Package: tbeptools (via r-universe)

August 27, 2024

Title Data and Indicators for the Tampa Bay Estuary Program

Version 3.0.0

Date 2024-08-07

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 (>= 3.6), methods

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

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

Remotes ropensci/rnoaa

License MIT + file LICENSE

Encoding UTF-8

VignetteBuilder knitr

LazyData true

LazyDataCompression xz

RoxygenNote 7.2.3

URL `https://github.com/tbep-tech/tbeptools`

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

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

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

RemoteRef HEAD

RemoteSha af03208727fb54b8d86dcfc0cac6c98be8c27c13

Contents

`acres` *Tampa Bay intertidal and supratidal land use and cover*

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)
```

`anlz_attain` *Get attainment categories*

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` *Get site attainments*

Description

Get site attainment categories for chlorophyll or light attenuation

Usage

```
anlz_attainsite(
  avedatsite,
  thr = c("chla", "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 simplication 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` *Estimate annual means*

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` *Estimate annual means by site*

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` *Assign threshold categories to Enterococcus data*

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_importrain`. 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

`fibdata`	input FIB `data.frame` as returned by `read_importfib`
`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
`assf`	logical indicating if the data are further processed as a simple features object with additional columns for `show_fibmap`

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 `areasel` argument can indicate valid entries in the `area` column of `fibdata`. For example, use either `"Alafia River"` or `"Hillsborough River"` for the corresponding river basins, where rows in `fibdata` are filtered based on the the selection. All stations are returned if this argument is set as `NULL` (default). The Alafia River basin includes values in the `area` column of `fibdata` as `"Alafia River"` and `"Alafia River Tributary"`. The Hillsborough River basin includes values in the `area` column of `fibdat` as `"Hillsborough River"`, `"Hillsborough River Tributary"`, `"Lake Thonotosassa"`, `"Lake Thonotosassa Tributary"`, and `"Lake Roberta"`. Not all areas may be present based on the selection. All valid options for `areasel` include `"Alafia River"`, `"Hillsborough River"`, `"Big Bend"`, `"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"`. One to any of the options can be used.

Value

A `data.frame` if 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_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` *Analyze Fecal Indicator Bacteria categories over time by station*

Description

Analyze Fecal Indicator Bacteria categories over time by station

Usage

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

Arguments

`fibdata`	input data frame as returned by `read_importfib` or `read_importentero`
`yrrng`	numeric vector indicating min, max years to include, defaults to range of years in data, see details
`stas`	optional vector of stations to include, see details
`indic`	character for choice of fecal indicator. Allowable options are `fcolif` for fecal coliform, or `entero` for Enterococcus. A numeric column in the data frame must have this name.
`threshold`	optional numeric for threshold against which to calculate exceedances for the indicator bacteria of choice. If not provided, defaults to 400 for `fcolif` and 130 for `entero`.
`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_importrain`. 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

Details

This function is used to create output for plotting a matrix stoplight graphic for FIB categories by station and year. Each station and year combination is categorized based on the likelihood of fecal indicator bacteria concentrations exceeding some threshold in a given year. For fecal coliform, the default threshold is 400 CFU / 100 mL in a given year (using Fecal Coliform, `fcolif` in `fibdata`). For Enterococcus, the default threshold is 130 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%, 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 using the monthly samples (`lagyr = 3`). See `show_fibmatrix` for additional details.

`yrrng` can be specified several ways. If `yrrng = 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., `yrrng = c(2000, 2010)` or `yrrng = c(2000, NA)`. In the latter case, the maximum year will be defined by the data.

Value

A `tibble` object with FIB summaries by year and station, including columns for the estimated geometric mean of fecal coliform or Enterococcus concentrations (`gmean`), the proportion of samples exceeding 400 CFU / 100 mL (fecal coliform) or 130 CFU / 100 mL (Enterococcus) (`exced`), the count of samples (`cnt`), and a category indicating a letter outcome based on the proportion of exceedences (`cat`).

See Also

`show_fibmatrix`

Examples

```
anlz_fibmatrix(fibdata, indic = 'fcolif')

# use different indicator:
anlz_fibmatrix(fibdata, indic = 'entero')

# use different dataset; does not contain an 'fcolif' column so we must specify indic:
anlz_fibmatrix(enterodata, indic = 'entero', lagyr = 1)
# same entero data; lower threshold - changes 'cat' scores
anlz_fibmatrix(enterodata, indic = 'entero', lagyr = 1, threshold = 30)

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

# subset to only dry samples
```

```
anlz_fibmatrix(enterodata, indic = 'entero', lagyr = 1, subset_wetdry = "dry",
               temporal_window = 2, wet_threshold = 0.5)
```

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

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_importrain`. 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` *Evaluate Habitat Master Plan progress for report card*

Description

Evaluate Habitat Master Plan progress for report card

Usage

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

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

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` *Estimate hydrological estimates and adjustment factors for bay segments*

Description

Estimate hydrological estimates and adjustment factors for bay segments

Usage

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

Arguments

`yrs`	numeric vector indicating years to return
`noaa_key`	user-supplied NOAA key, see details
`trace`	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 "Authentication" section under the help file for ncdc in the defunct rnoaa package. 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=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX\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 = 2023)
```

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 = 2023)
```

`anlz_refs` *Convert references csv to bib*

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 <- 'https://raw.githubusercontent.com/tbep-tech/tbep-refs/master/tbep-refs.csv'
bibs <- anlz_refs(path)

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

## End(Not run)
```

`anlz_sedimentaddtot` *Add contaminant totals to sediment data*

Description

Add contaminant totals to sediment data

Usage

```
anlz_sedimentaddtot(
  sedimentdata,
  yrrng = c(1993, 2022),
  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` *Get average concentrations for a sediment parameter by bay segment*

Description

Get average concentrations for a sediment parameter by bay segment

Usage

```
anlz_sedimentave(
  sedimentdata,
  param,
  yrrng = c(1993, 2022),
  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` *Get sediment PEL ratios at stations in Tampa Bay*

Description

Get sediment PEL ratios at stations in Tampa Bay

Usage

```
anlz_sedimentpel(
  sedimentdata,
  yrrng = c(1993, 2022),
  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` *Get average concentrations for a sediment parameter by bay segment*

Description

Get average concentrations for a sediment parameter by bay segment

Usage

```
anlz_sedimentpelave(
  sedimentdata,
  yrrng = c(1993, 2022),
  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_tbbimed` *Get annual medians of Tampa Bay Benthic Index scores by bay segment*

Description

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

Usage

```
anlz_tbbimed(
  tbbiscr,
 bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB", "All", "All (wt)"),
  rev = FALSE,
  yrrng = c(1993, 2022)
)
```

Arguments

`tbbiscr`	input data frame as returned by `anlz_tbbiscr`
`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

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.

Value

A data frame of annual medians by bay segment

Examples

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

`anlz_tbbiscr` *Get Tampa Bay Benthic Index scores*

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

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` *Get all raw metrics for Tampa Bay Nekton Index*

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 calcualted

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 fomat. If `all = TRUE`, all metrics are returned, otherwise only `NumTaxa`, `BenthicTaxa`, `TaxaSelect`, `NumGuilds`, and `Shannon` are returned.

Examples

```
anlz_tbnimet(fimdata)
```

`anlz_tbniscr` *Get Tampa Bay Nekton Index scores*

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 = 2023)
```

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 = 2023)
```

`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 = 2023, 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% of observations exceed DO saturation criteria are also estimated. The `do_bnml` and `do_prop` columns show a 1 or 0 for a given year to indicate if more than ten percent of observations were below DO percent saturation of 42. The first column is based on a binomial probability exceedance that takes into account the number of observations in the year and the second column is based on a simple proportional estimate from the raw data.

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 (`chla_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 = 2023)
head(dat)
```

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

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, 2023),
  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 indicating year ranges to evaluate
`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` *Get annual averages of seagrass frequency occurrence by bay segments, year, and species*

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, 2023),
 species = c("Halodule", "Syringodium", "Thalassia", "Ruppia", "Halophila", "Caulerpa",
    "Dapis"),
  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 indicating year ranges to evaluate
`species`	chr string of species to summarize, one to many of "Halodule", "Syringodium", "Thalassia", "Ruppia", "Halophila", "Caulerpa", "Dapis"
`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`. Also note that totals across species (`total = T`) are not returned 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` *Get seagrass average abundance and occurrence across transects*

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) and Dapis (cyanobacteria), whereas all attached and drift algae species are aggregated. Drift or attached macroalgae and cyanobacteria (Dapis) estimates may not be accurate prior to 2021.

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` *Get attainment categories for a chosen year*

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_importwq`
`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 chloropyll concentration, light attenuations, segment targets, and attainment categories.

Examples

```
# defaults to current year
anlz_yrattain(epcdata, yrsel = 2023)
```

`benthicdata` *Benthic data for the Tampa Bay Benthic Index current as of 03132024*

Description

Benthic data for the Tampa Bay Benthic Index current as of 03132024

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` *Terrain basemap*

Description

Terrain basemap

Usage

```
bsmap
```

Format

A `ggmap` object

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

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/tbeptools/blob/master/data-raw/catchpixels-raw.R`)

`catchprecip` *Daily precip by catchment for selected Enterococcus stations*

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 from 1995-2023, 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. Generated by `data-raw/catchprecip-raw.R` (view on github: `https://github.com/tbep-tech/tbeptools/blob/master/data-raw/catchprecip-raw.R`)

Source

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

`enterodata` *Enterococcus data from 53 key Enterococcus stations since 1995*

Description

Enterococcus data from 53 key Enterococcus stations since 1995

Usage

```
enterodata
```

Format

A data frame with 6266 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 sample dataset containing Enterococcus from 53 stations in the TBEP watershed from 1995-2023. Generated by `data-raw/enterodata-raw.R` (view on github: `https://github.com/tbep-tech/tbeptools/blob/master/data-raw/enterodata-raw.R`)

Source

Water Quality Portal, `https://waterqualitydata.us`

`epcdata` *All bay data as of 20240201*

Description

All bay data as of 20240201

Usage

```
epcdata
```

Format

A data frame with 27961 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` *All Fecal Indicator Bacteria (FIB) data as of 20240814*

Description

All Fecal Indicator Bacteria (FIB) data as of 20240814

Usage

```
fibdata
```

Format

A data frame with 77526 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

Examples

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

nrow(fibdata)
ncol(fibdata)
```

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

file.remove(xlsx)

## End(Not run)
```

`fimdata` *FIM data for Tampa Bay Nekton Index current as of 07092024*

Description

FIM data for Tampa Bay Nekton Index current as of 07092024

Usage

```
fimdata
```

Format

A data frame with 52042 rows 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 7771 features and 2 fields, +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs

Reference num

bay_segment chr

Examples

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

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

## 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 65*

Description

Florida Department of Environmental Protection, Impaired Waters Rule, Run 65

Usage

```
iwrraw
```

Format

A data frame 537946 rows and 11 variables

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(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)
```

`phytodata` *Phytoplankton data current as of 03202024*

Description

Phytoplankton data current as of 03202024

Usage

```
phytodata
```

Format

A nested `tibble` with 21143 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)
```

`read_dlcurrent` *Download latest file from epchc.org*

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 is not current or `download_latest = TRUE`. The function does nothing if `download_latest = FALSE`.

Examples

```
## Not run:
locin <- '~/Desktop/RWMDataSpreadsheet_ThroughCurrentReportMonth.xlsx'
urlin1 <- 'https://epcbocc.sharepoint.com/:x:/s/Share/'
urlin2 <- 'EYXZ5t16UlFGk1rzIU91VogBa8U37lh8z_Hftf2KJISSHg?e=8r1SUL'
urlin <- paste0(urlin1, urlin2, '&download=1')
read_dlcurrent(locin = locin, urlin = urlin)

## End(Not run)
```

`read_formbenthic` *Format benthic data for the Tampa Bay Benthic Index*

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/:x:/s/Share/'
urlin2 <- 'EQUCWBuwCNdGuMREYAyAD1gBKC98mYtCHMWX0FYLrbT4KA?e=nDfnnQ'
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_importepc`
`all`	logical indicating if all stations with FIB data are returned, default is `FALSE`

Details

Formats input data from `read_importepc` 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_importepc`

Examples

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

# load raw data and assign to object
epcall <- read_importepc(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
csv <- url('https://raw.githubusercontent.com/tbep-tech/tbni-proc/master/data/TampaBay_NektonIndexData.csv')

datin <- read.csv(csv)

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

`read_formphyto` *Format phytoplankton data*

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. Bacillariophyta and Cyanobacteria are retained at the phylum level. 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)

## End(Not run)
```

`read_formsediment` *Format sediment data*

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/:x:/s/Share/'
urlin2 <- 'Ef9utuKCHD9LliarsOPKCJwB5kxgCObf0tY5x5wX20JQUA?e=DuTseb'
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` *Format seagrass transect data from Water Atlas*

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 <- 'http://dev.seagrass.wateratlas.usf.edu/api/assessments/all__use-with-care'
jsn <- fromJSON(url)
trndat <- read_formtransect(jsn)

## End(Not run)

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

`read_formwq` *Format water quality data*

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 chloropyll and secchi observations

See Also

`read_importwq`, `read_importepc`

Examples

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

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` *Download and import benthic data for Tampa Bay*

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://epcbocc.sharepoint.com/:u:/s/Share/EQUCWBuwCNdGuMREYAyAD1gBKC98mYtCHMWX0FYLrbT4KA?e=nDfnnQ&download=1` (viewable at `https://epcbocc.sharepoint.com/:f:/s/Share/EtOJfziTTa9FliL1oROb9OsBRZU 0NRC162hHjQ?e=4gUXgJ`).

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` *Download Enterococcus data from the Water Quality Portal*

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

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 "RWMDataSpreadsheet" worksheet from the file located at `xlsx`. The file is downloaded from `https://epcbocc.sharepoint.com/:x:/s/Share/EYXZ5t16UlFGk1rzIU91VogBa8U37lh8z_Hftf2KJISSHg?e=8r1SUL&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_importwq` 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_importwq`, `read_importfib`

Examples

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

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

## End(Not run)
```

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

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_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`
`all`	logical indicating if all stations with FIB data are returned, default is `TRUE`, see details

Details

Loads the "RWMDataSpreadsheet" worksheet from the file located at `xlsx`. The file is downloaded from `https://epcbocc.sharepoint.com/:x:/s/Share/EYXZ5t16UlFGk1rzIU91VogBa8U37lh8z_Hftf2KJISSHg?e=8r1SUL&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, function is used internally within `read_importfib`

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/RWMDataSpreadsheet_ThroughCurrentReportMonth.xlsx'

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


## End(Not run)
```

`read_importfim` *Load local FIM data for the Tampa Bay Nekton Index*

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_dlcurrent` 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` *Load local phytoplankton cell count file*

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_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

Phytoplankton cell count data downloaded from https://epcbocc.sharepoint.com/:x:/s/Share/ETAfRQ5drmRHntDd1O8s3FQ
OIVDxrA?e=eSmtxD&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_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_importsediment` *Download and import sediment data for Tampa Bay*

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/Share/Ef9utuKCHD9LliarsOPKCJwB5kxgCObf0tY5x5wX20JQUA?e=DuTseb&download=1` (viewable at `https://epcbocc.sharepoint.com/:f:/s/Share/EtOJfziTTa9FliL1oROb9OsBRZU-nO60fu_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` *Load local water quality file*

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 `xlsx` 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 "RWMDataSpreadsheet" worksheet from the file located at `xlsx`. The file is downloaded from `https://epcbocc.sharepoint.com/:x:/s/Share/EYXZ5t16UlFGk1rzIU91VogBa8U37lh8z_Hftf2KJISSHg?e=8r1SUL&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 (`chla`), 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/RWMDataSpreadsheet_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` *Import data from the Water Quality Portal*

Description

Import data from the Water Quality Portal

Usage

```
read_importwqp(org, type, trace = F)
```

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 `FALSE`

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), or `"21FLTPA_WQX"` (Florida Department of Environmental Protection, Southwest District). 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.

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_transect` *Import JSON seagrass transect data from Water Atlas*

Description

Import JSON seagrass transect data from Water Atlas

Usage

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

Arguments

`training`	logical if training data are imported or the complete database
`raw`	logical indicating if raw, unformatted data are returned, see details

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 `http://dev.seagrass.wateratlas.usf.edu/api/assessments/training`. If `training` = `FALSE`, the entire transect survey database is imported from `http://dev.seagrass.wateratlas.usf.edu/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.

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` *Seagrass coverage by year*

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),
   Acres = c(40420, 21650, 23285, 25226, 25753, 26518, 26916,
        24841, 26078, 27021, 28299, 29647, 32897, 34642, 40294.71,
        41655.16, 40651.55, 34298, 30137),
   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)
 ), class = "data.frame", row.names = c(NA, -19L))

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


## End(Not run)
```

`sedimentdata` *Sediment data for the Tampa Bay current as of 12142023*

Description

Sediment data for the Tampa Bay current as of 12142023

Usage

```
sedimentdata
```

Format

A `data.frame` with 216627 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

BetweenTELPEL 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_annualassess` *Create a table for the annual management outcome assessments*

Description

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

Usage

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

Arguments

`epcdata`	data frame of epc data returned by `read_importwq`
`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
`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(epcdata, yrsel = 2023)
show_annualassess(epcdata, yrsel = 2023, caption = TRUE)
```

`show_boxplot` *Plot monthly chlorophyll or light attenuation values for a segment*

Description

Plot monthly chlorophyll or light attenuation values for a bay segment

Usage

```
show_boxplot(
  epcdata,
  param = c("chla", "la"),
  yrsel = NULL,
  yrrng = c(1975, 2023),
  ptsz = 0.5,
  bay_segment = c("OTB", "HB", "MTB", "LTB"),
  trgs = NULL,
  family = NA,
  labelexp = TRUE,
  txtlab = TRUE,
  partialyr = FALSE
)
```

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
`yrsel`	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 `yrsel`
`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

Details

Points not included in `yrsel` 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,
  xtxt = 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
`xtxt`	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 = 2023, site = '2', species = 'Halodule', varplo = 'Abundance')
```

`show_enteromap` *Map Enterococcus results by month, year, and location*

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
)
```

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_importrain`. 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

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

Description

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

Usage

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

Arguments

`fibdata`	input FIB `data.frame` as returned by `read_importfib`
`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

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_fibmap` for details on the categories

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')
```

`show_fibmatrix` *Plot a matrix of Fecal Indicator Bacteria categories over time by station*

Description

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

Usage

```
show_fibmatrix(
  fibdata,
  yrrng = NULL,
  stas = NULL,
  indic = NULL,
  threshold = NULL,
  lagyr = 3,
  subset_wetdry = c("all", "wet", "dry"),
  precipdata = NULL,
  temporal_window = NULL,
  wet_threshold = NULL,
  txtsz = 3,
  asreact = FALSE,
  nrows = 10,
  family = NA,
  angle = 90,
  size = 10,
  hjust = 0,
  plotly = FALSE,
  width = NULL,
  height = NULL
)
```

Arguments

`fibdata`	input data frame as returned by `read_importfib` or `read_importentero`
`yrrng`	numeric vector indicating min, max years to include, defaults to range of years in data, see details
`stas`	optional vector of stations to include, see details

`indic`	character for choice of fecal indicator. Allowable options are `fcolif` for fecal coliform, or `entero` for Enterococcus. A numeric column in the data frame must have this name.
`threshold`	optional numeric for threshold against which to calculate exceedances for the indicator bacteria of choice. If not provided, defaults to 400 for `fcolif` and 130 for `entero`.
`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_importrain`. 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
`txtsz`	numeric for size of text in the plot, applies only if `tab = FALSE`. Use `txtsz = NULL` to suppress.
`asreact`	logical indicating if a `reactable` object is returned
`nrows`	if `asreact = TRUE`, a numeric specifying number of rows in the table
`family`	optional chr string indicating font family for text labels
`angle`	numeric for angle of x-axis text labels
`size`	numeric for size of the x-axis text labels
`hjust`	numeric for horizontal justification of x-axis text labels
`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

The matrix color codes years and stations based on the likelihood of fecal indicator bacteria concentrations exceeding 400 CFU / 100 mL for Fecal Coliform or 130 CFU / 100 mL for Enterococcus. 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.

The default stations for fecal coliform data 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 in `fibdata` can be plotted using the `stas` argument.

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`.

Examples

```
show_fibmatrix(fibdata)

# change the threshold
show_fibmatrix(fibdata, threshold = 200)

# change the indicator
show_fibmatrix(fibdata, indic = 'entero')

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

`show_hmpreport` *Show Habitat Master Plan progress report card*

Description

Show Habitat Master Plan progress report card

Usage

```
show_hmpreport(
  acres,
  subtacres,
  hmptrgs,
  typ,
  twocol = FALSE,
  strata = c("Subtidal", "Intertidal", "Supratidal"),
  ycollapse = FALSE,
  text = 2.5,
  xang = 25,
```

```
  family = NA,
  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"`
`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 symbols if `twocol = T`, use `NULL` to suppress
`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_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 = NA,
  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` *Creates a plotly matrix from any matrix function input*

Description

Creates a plotly matrix from any matrix function input

Usage

```
show_matrixplotly(
  mat,
  family = NA,
  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_importwq`
`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, yrsel = 2023, 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 = 2022,
  family = NA,
  lastlab = "acres",
  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
`lastlab`	chr string of text to append to label in last bar of the barplot
`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)
```

`show_sedimentalratio` *Plot metal concentrations against aluminum*

Description

Plot metal concentrations against aluminum

Usage

```
show_sedimentalratio(
  sedimentdata,
  param,
  yrrng = c(1993, 2022),
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"),
  funding_proj = "TBEP",
  lnsz = 1,
  base_size = 12,
  plotly = FALSE,
  family = NA,
  width = NULL,
  height = NULL
)
```

Arguments

`sedimentdata`	input sediment `data.frame` as returned by `read_importsediment`
`param`	chr string for which parameter to plot, must be a metal
`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

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

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` *Plot sediment concentration averages by bay segment*

Description

Plot sediment concentration averages by bay segment

Usage

```
show_sedimentave(
  sedimentdata,
  param,
  yrrng = c(1993, 2022),
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"),
  funding_proj = "TBEP",
  lnsz = 1,
  base_size = 12,
  plotly = FALSE,
  family = NA,
  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

Examples

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

`show_sedimentmap` *Make a map for sediment contaminants at stations in Tampa Bay*

Description

Make a map for sediment contaminants at stations in Tampa Bay

Usage

```
show_sedimentmap(
  sedimentdata,
  param,
  yrrng = c(1993, 2022),
  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` *Plot summary of PEL averages by bay segment*

Description

Plot summary of PEL averages by bay segment

Usage

```
show_sedimentpelave(
  sedimentdata,
  yrrng = c(1993, 2022),
  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 = NA,
  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

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

`p`	`ggplot` object as output from `show_sedimentpelave`
`width`	numeric for width of the plot in pixels
`height`	numeric for height of the plot in pixels

Value

A `plotly` data object

Examples

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

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

Description

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

Usage

```
show_sedimentpelmap(
  sedimentdata,
  yrrng = c(1993, 2022),
  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` *Create a colorized table for water quality outcomes and exceedances by segment*

Description

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

Usage

```
show_segmatrix(
  epcdata,
  txtsz = 3,
  trgs = NULL,
  yrrng = c(1975, 2023),
  bay_segment = c("OTB", "HB", "MTB", "LTB"),
  abbrev = FALSE,
  family = NA,
  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
`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 of `plotly = TRUE`
`height`	numeric for height of the plot in pixels, only applies of `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` *Plot chlorophyll and secchi data together with matrix outcomes*

Description

Plot chlorophyll and secchi data together with matrix outcomes

Usage

```
show_segplotly(
  epcdata,
  bay_segment = c("OTB", "HB", "MTB", "LTB"),
  yrrng = c(1975, 2023),
  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` *Map site attainment categories for a selected year*

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,
  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)`
`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 `mosel = c(1, 12)`

Value

A static `ggplot` object is returned

Examples

```
show_sitemap(epcdata, yrsel = 2023)
```

`show_sitesegmap` *Map site and bay segment attainment categories for a selected year*

Description

Map site and bay segment attainment categories for a selected year

Usage

```
show_sitesegmap(
  epcdata,
  yrsel,
  param = c("chla", "la"),
  trgs = NULL,
  thrs = FALSE,
  partialyr = FALSE,
  base_size = 12,
  family = NA
)
```

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
`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, yrsel = 2023)
```

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

Description

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

Usage

```
show_tbbimatrix(
  tbbiscr,
 bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB", "All", "All (wt)"),
  yrrng = c(1993, 2022),
  alph = 1,
  txtsz = 3,
  family = NA,
  rev = FALSE,
  position = "top",
  plotly = FALSE,
  width = NULL,
  height = NULL
)
```

Arguments

`tbbiscr`	input data frame as returned by `anlz_tbbiscr`
`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
`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)").

Value

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

Examples

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

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

Description

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

Usage

```
show_tbnimatrix(
  tbniscr,
  bay_segment = c("OTB", "HB", "MTB", "LTB"),
  perc = c(32, 46),
  alph = 1,
  txtsz = 3,
  family = NA,
  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 = NA,
  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` *Plot Tampa Bay Nekton Index scores over time as average across bay segments*

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` *Creates a plotly object for TBNI score plots*

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` or `show_tbniscrall`
`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_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 = 2023)
show_tdlcrk(dat)
```

`show_tdlcrkindic` *Plotly barplots of tidal creek context indicators*

Description

Plotly barplots of tidal creek context indicators

Usage

```
show_tdlcrkindic(
  id,
  cntdat,
  yr = 2023,
  thrsel = 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
`thrsel`	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 = 2023)

set.seed(123)
id <- sample(unique(cntdat$id), 1)
show_tdlcrkindic(id, cntdat, thrsel = 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 = 2023,
  thrsel = 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
`thrsel`	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 = 2023)

set.seed(123)
id <- sample(unique(cntdat$id), 1)
show_tdlcrkindiccdf(id, cntdat, thrsel = TRUE)
```

`show_tdlcrkline` *Add a line or annotation to a plotly graph*

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"),
  thrsel = FALSE,
  horiz = TRUE,
  annotate = FALSE
)
```

Arguments

`varin`	chr string for the indicator
`thrsel`	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, chloropyll
show_tdlcrkline('CHLAC', thrsel = TRUE)
```

`show_tdlcrkmatrix` *Plot the tidal creek report card matrix*

Description

Plot the tidal creek report card matrix

Usage

```
show_tdlcrkmatrix(
  dat,
  class = c("3M", "2"),
  score = c("Prioritize", "Investigate", "Caution", "Monitor"),
  family = NA,
  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 = 2023)
show_tdlcrkmatrix(dat)
```

`show_tdlcrkradar` *Radar plots for tidal creek indicators*

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 = 2023, radar = TRUE)

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

`show_thrplot` *Plot annual water quality values, targets, and thresholds for a segment*

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("chla", "la"),
  trgs = NULL,
  yrrng = c(1975, 2023),
  family = NA,
  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"),
  yrrng = c(1998, 2023),
  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 indicating year ranges to evaluate
`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 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

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), and/or Dapis (cyanobacteria). 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'),
  varplo = 'Abundance')

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

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

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, 2023),
  species = c("Halodule", "Syringodium", "Thalassia", "Halophila", "Ruppia", "Caulerpa",
    "Dapis"),
  total = TRUE,
  alph = 1,
  family = NA,
  plotly = FALSE,
  asreact = FALSE,
  width = NULL,
  height = NULL,
  sppcol = NULL
)
```

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 indicating year ranges to evaluate
`species`	chr string of species to summarize, one to many of "Halodule", "Syringodium", "Thalassia", "Ruppia", "Halophila", "Caulerpa", "Dapis"
`total`	logical indicating if total frequency occurrence for all 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 species by date and transect in each bay segment. Drift or attached macroalgae (e.g., Caulerpa) and cyanobacteria (Dapis) estimates may not be accurate prior to 2021.

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 Yarbro, 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` *Show matrix of seagrass frequency occurrence by bay segments and year*

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, 2023),
  alph = 1,
  txtsz = 3,
  family = NA,
  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
`yrrng`	numeric indicating year ranges to evaluate
`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 species 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 *Plot frequency occurrence for a seagrass transect by time for all species*

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"),
  yrrng = c(1998, 2023),
  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 indicating year ranges to evaluate
`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) 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 colorized table for chlorophyll or light attenuation exceedances*

Description

Create a colorized table for chlorophyll or light attenuation exceedances

Usage

```
show_wqmatrix(
  epcdata,
  param = c("chla", "la"),
  txtsz = 3,
  trgs = NULL,
  yrrng = c(1975, 2023),
  bay_segment = c("OTB", "HB", "MTB", "LTB"),
  asreact = FALSE,
  nrows = 10,
  abbrev = FALSE,
  family = NA,
  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` *Bay stations by segment*

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

`subtacres` *Tampa Bay subtidal cover*

Description

Tampa Bay subtidal cover for Habitat Master Plan reporting

Usage

```
subtacres
```

Format

A data frame

Examples

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

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

## End(Not run)
```

`swfwmdtbseg` *Spatial data object of SWFWMD Tampa Bay segments*

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)
```

`targets` *Bay segment targets*

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

Examples

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

# import and clean
tbnispp <- read.csv('../tbni-proc/data/TBIndex_spp_codes.csv',
    header = TRUE, stringsAsFactors = FALSE) %>%
  mutate(
    NODCCODE = as.character(NODCCODE),
    NODCCODE = case_when(NODCCODE == "9.998e+09" ~ "9998000000",
                         TRUE ~ NODCCODE)
  )

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

## End(Not run)
```

`tbseg` *Spatial data object of Tampa Bay segments*

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))
```

`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` *Spatial data object of Tampa Bay segments plus watersheds*

Description

Spatial data object of Tampa Bay segments plus waterhseds

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` *Spatial data object of Tampa Bay watershed*

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 65*

Description

Spatial data object of tidal creeks in Impaired Waters Rule, Run 65

Usage

```
tidalcreeks
```

Format

A simple features sf object (MULTILINESTRING) with 615 features and 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 04122024*

Description

Seagrass transect data for Tampa Bay current as of 04122024

Usage

```
transect
```

Format

A data frame with 152580 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` *Seagrass transect locations*

Description

Seagrass transect locations

Usage

```
trnlns
```

Format

A `sf` LINESTRING object

Examples

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

trnlns <- st_read('T:/05_GIS/SEAGRASS_TRANSECTS/transect_routes.shp') %>%
   st_transform(crs = 4326) %>%
   dplyr::filter(!as.character(Site) %in% c('S8T1', 'S8T2', 'S8T3', 'S3T2')) %>%
   dplyr::mutate_if(is.factor, as.character) %>%
   dplyr::filter(Site %in% trnpts$TRAN_ID)

# add bearing, positive counter-clockwise from east
bearing <- lapply(trnlns$geometry, function(x) geosphere::bearing(x[, c(1:2)])[[1]]) %>%
  unlist()

trnlns$bearing <- bearing

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

## End(Not run)
```

`trnpts` *Seagrass transect starting locations*

Description

Seagrass transect starting locations

Usage

```
trnpts
```

Format

A `sf` POINT object

Examples

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

trnpts <- st_read('T:/05_GIS/SEAGRASS_TRANSECTS/TransectBasics2019.shp') %>%
   st_transform(crs = 4326) %>%
   dplyr::rename(MonAgency = 'MON_AGENCY') %>%
   dplyr::filter(!as.character(TRAN_ID) %in% c('S8T1', 'S8T2', 'S8T3', 'S3T2'))

# s4t10 is slightly outside boundary for lower tampa bay
s4t10 <- trnpts %>%
   dplyr::filter(TRAN_ID %in% 'S4T10') %>%
   dplyr::mutate(bay_segment = 'LTB')

trnpts <- trnpts %>%
   sf::st_intersection(sf::st_make_valid(tbsegshed)) %>%
   dplyr::select(-long_name) %>%
   dplyr::mutate_if(is.factor, as.character) %>%
   dplyr::bind_rows(s4t10)

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

## End(Not run)
```

`util_fibicons` *Return leaflet icon set for FIB maps*

Description

Return leaflet icon set for FIB maps

Usage

```
util_fibicons(indic)
```

Arguments

`indic`	character indicating `"entero"` or `"fcolif"` for *Enterococcus* or Fecal Coliform

Details

Used internally with `show_enteromap` and `show_fibmap`, former uses wet/dry icons for *Enterococcus* and latter uses *E. Coli*/*Enterococcus* icons

Value

A leaflet icon set as returned by `iconList`.

Examples

```
util_fibicons(indic = 'entero')
util_fibicons(indic = 'fcolif')
```

`util_fiblevs` *A list of Fecal Indicator Bacteria (FIB) category breakpoints and labels*

Description

A list of Fecal Indicator Bacteria (FIB) category breakpoints and labels

Usage

```
util_fiblevs()
```

Value

A `list` showing cut points and labels for FIB categories

Examples

```
util_fiblevs()
```

`util_html` *Convert character string to html class*

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_map` *Create an empty leaflet map from sf input*

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_orgin` *Get organization name from organization identifier in USEPA Water Quality Portal*

Description

Get organization name from organization identifier in USEPA Water Quality Portal

Usage

```
util_orgin(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_orgin('21FLHILL_WQX')
util_orgin('21FLPASC_WQX', stanm = TRUE)
```

Index