

Package: **ffishltm** (via r-universe)

September 7, 2024

Type Package

Title A package for analyzing FL FWC Freshwater long-term monitoring (LTM) data

Description Provides functions for summarizing and analyzing FL FWC Freshwater Fish Long-term Monitoring Data.

Date 2021-07-06

Version 0.0.4

Encoding UTF-8

Imports tibble, tidyr, dplyr, tibble, ggplot2, viridis, stringr, magrittr

RoxygenNote 7.3.1

Roxygen list(markdown = TRUE)

Suggests rmarkdown, knitr, testthat (>= 3.0.0)

VignetteBuilder knitr

Depends R (>= 3.5.0)

License MIT + file LICENSE

Config/testthat/edition 3

Repository <https://jasono6359.r-universe.dev>

RemoteUrl <https://github.com/jasonO6359/ffishltm>

RemoteRef main

RemoteSha 462fbc036fbe807687bed45cb63a41614f1706f1

Contents

add_zero_count	2
age.key	3
check_expected_columns	4
cpue	4
cpue.plot	5
cpue_plot	6

create_ltm_analysis_template	8
create_ltm_report	9
deprecate	9
fish_table	10
ffishltm	10
get_wpt	11
guild.comp	11
helper_addgapyears	12
is.outlier	13
is.StdFishQry	13
len.dist	14
len_dist	15
loch_blcr_2020_age	16
ltm.data.summary	16
ltm.import	17
newnans	18
predict_wt	19
sfr_fish	19
species.history	20
species_history	21
theme_ltm	22
Wr	23

Index 24

add_zero_count	<i>Add zero count data for species missing from sample</i>
----------------	--

Description

Add zero count data for species missing from sample

Usage

```
add_zero_count(data, sample_ids, species_id, count, format = "long")
```

Arguments

data	dataset as data.frame or tibble
sample_ids	sample id variables
species_id	species ID field
count	field containing species counts
format	defaults to "long" - data returned with 1 row per sample_id x species_id, else if not "long" returns data in "wide" format, with one row per sample_id and 1 column per unique species in the species_id.

Value

tibble

Examples

```
fish_data <- tibble(
  site = c(1:5,3:7),
  species_name = c(rep("BLUE", 5), rep("RESU", 5)),
  ct = c(5, 2, 6, 4, 7,
        8, 6, 7, 2, 9),
  minutes = rep(10, 10))

fish_data %>%
  add_zero_count(sample_ids = c(site,minutes), species_id = species_name, count = ct)
```

age.key *Create age-length key*

Description

Generate age-length key from a subsample of aged fish

Usage

```
age.key(dataset, age_column, length_column, length_unit = NA)
```

Arguments

dataset	dataset containing at least 2 columns, one containing an age field, and one containing length field
age_column	string specifying the name of the column containing ages (in years)
length_column	string specifying the name of the column containing lengths
length_unit	string specifying the length units options are: "mm" = millimeters; "cm" = centimeters, if none specified defaults to "mm"

Value

returns age-length key

Examples

```
data(loch_blcr_2020_age)
loch_blcr_alk <- age.key(dataset=loch_blcr_2020_age, age_column="age", length_column="tl", length_unit = 'mm')
```

```
check_expected_columns
```

HELPER: check for expected column names

Description

helper function triggers error if expected column names are missing

Usage

```
check_expected_columns(input_table, expected_names)
```

Arguments

```
input_table
expected_names
```

Examples

```
test_data <- data.frame(col1 = 1, col2 = 2, col3 = "A")
check_expected_columns(test_data, c("col1", "col2"))
```

```
cpue
```

Calculate Mean Catch-per-unit Effort (CPUE)

Description

Calculate Mean Catch-per-unit Effort (CPUE)

Usage

```
cpue(data, count, effort)
```

Arguments

data	data.frame or tibble containing samples as rows, and species counts as columns along with an effort column
effort	name of column (unquoted) specifying the sample effort value (typically minutes)
species	name of column (unquoted) for species for which CPUE is desired

Value

named vector containing "Mean CPUE" and "SE" estimates

Examples

```

fish_data <- tibble(
  site = c(1:5,3:7),
  species_name = c(rep("BLUE", 5), rep("RESU", 5)),
  ct = c(5, 2, 6, 4, 7,
        8, 6, 7, 2, 9),
  minutes = rep(10, 10))

fish_data %>%
  cpue(ct, "minutes") # can specify column name as either a symbol or character

fish_data %>%
  group_by(species_name) %>% # use dplyr::group_by to specify grouping variables.
  cpue(ct, "minutes")

# fish_data %>%
#   add_zero_count(c(site,minutes), species_name, ct) %>% #use [add_zero_count()]to account for missing absence data
#   group_by(species_name) %>%
#   cpue(ct, "minutes")

```

cpue.plot

*Catch-Per-Unit-Effort Timeseries Plot – DEPRECATED May 2024***Description**

DEPRECATED as of 5/6/2024 use [cpue_plot\(\)](#) instead.

Usage

```

cpue.plot(
  datafile,
  speciesList = list(),
  species_size_strata = list(),
  years = list(),
  seasons = list(),
  print = FALSE,
  figure_filename = NA,
  fig_scale = 1
)

```

Arguments

datafile	data, should be output from ltm.data.summary function
speciesList	list of selected species, can specify by common name, scientific name or species code
species_size_strata	optional argument, specifies size strata groups for which CPUEs should be calculated, see example below for proper convention.

years	list of years to include in figure
seasons	—currently functionless, will update in future version, to avoid errors make sure that all seasons in input dataset are the same—
print	boolean, if TRUE figure will be saved to file
figure_filename	if print=TRUE, figure will be saved to this filename
fig_scale	adjust to scale output figure size

Details

Create CPUE timeseries for species and years selected

Value

Returns the summarized data used to construct the plot

Examples

```
# import and format data
data(newnans)
newn_sum <- ltm.data.summary("Newnans Lake", newnans)
# CPUE plots for bluegill, largemouth bass, and brown bullheads
newnans_cpue <- cpue.plot(newn_sum,
  speciesList=c("BLUE", "LMB", "BRBU"),
  years = c(2016:2020))
# CPUE plots by size class
newnans_cpue2 <- cpue.plot(newn_sum,
  speciesList=c("BLUE", "LMB", "BRBU"),
  species_size_strata = list(
    BLUE = list(
      YOY = c(0,8),
      Quality = c(18,50)),
    LMB = list(
      YOY = c(0,20),
      Quality = c(30,50),
      Trophy = c(51,100))
  ),
  years = c(2016:2020)
)
```

cpue_plot

Catch-Per-Unit-Effort Timeseries Plot

Description

Create CPUE timeseries for species and years selected

Usage

```
cpue_plot(
  datafile,
  speciesList = list(),
  species_size_strata = list(),
  years = list(),
  seasons = list(),
  print = FALSE,
  figure_filename = NA,
  fig_scale = 1,
  return_object = "data"
)
```

Arguments

datafile	data, should be output from ltm.data.summary function
speciesList	list of selected species, can specify by common name, scientific name or species code
species_size_strata	optional argument, specifies size strata groups for which CPUEs should be calculated, see example below for proper convention.
years	list of years to include in figure
seasons	—currently functionless, will update in future version, to avoid errors make sure that all seasons in input dataset are the same—
print	boolean, if TRUE figure will be saved to file
figure_filename	if print=TRUE, figure will be saved to this filename
fig_scale	adjust to scale output figure size
return_object	<i>string</i> if "data" then function returns the dataset that is used to generate the figure, else if "ggplot" then function returns ggplot object

Value

either summarised data or a ggplot object. Specify the return object type with return_object

Examples

```
# import and format data
data(newnans)
newn_sum <- ltm.data.summary("Newnans Lake", newnans)
# CPUE plots for bluegill, largemouth bass, and brown bullheads
newnans_cpue <- cpue_plot(newn_sum,
  speciesList=c("BLUE", "LMB", "BRBU"),
  years = c(2016:2020))
# CPUE plots by size class
newnans_cpue2 <- cpue_plot(newn_sum,
  speciesList=c("BLUE", "LMB", "BRBU"),
```

```

species_size_strata = list(
  BLUE = list(
    YOY = c(0,8),
    Quality = c(18,50)),
  LMB = list(
    YOY = c(0,20),
    Quality = c(30,50),
    Trophy = c(51,100))
),
years = c(2016:2020)
)

```

```
create_ltm_analysis_template
```

HELPER: Create analysis R script for LTM report template

Description

A helper function, intended to be called from within `create_ltm_report()`

Usage

```

create_ltm_analysis_template(
  catch_data = "qryFish_standard.csv",
  age_data = NULL,
  habitat_data = "qrySite_comprehensive.csv"
)

```

Arguments

<code>catch_data</code>	<i>string</i> name of a csv file containing catch data. Data should be in format generated by 'Standard Fish Query' and stored in the /data directory of the ltm report template.
<code>age_data</code>	FUTURE FEATURE <i>string</i> name of a csv file containing age data
<code>habitat_data</code>	FUTURE FEATURE <i>string</i> name of a csv file containing habitat data. Data should be in format generated by 'Comprehensive Site Data' query and stored in the /data directory of the ltm report template.

Value

string containing r script text

Examples

```

## Not run:
create_ltm_analysis_template()

## End(Not run)

```

create_ltm_report *Create a new LTM report template*

Description

Create a new LTM report template

Usage

```
create_ltm_report(dir_name, report_name = dir_name, dir_root = getwd())
```

Arguments

dir_name *string* specifying the name of the report directory to create

report_name *string* specifying the report name, used in file names so avoid spaces defaults to name specified in directory

dir_root *string* specifying the path to the root directory in which to create the report project. Defaults to the current working directory.

Value

boolean - returns TRUE if the function executes without error

Examples

```
## Not run:  
create_ltm_report(dir_name = "test_report",  
                 report_name = "LakeDoe_LMB_2020")  
  
## End(Not run)
```

deprecate *INTERNAL: Create a deprecation warning*

Description

INTERNAL: Create a deprecation warning

Usage

```
deprecate(new_function)
```

Value

warning message

Examples

```

deprecated_function <- function() {
  deprecate("new_function")
}
deprecated_function()

```

fish_table	<i>LTM Fish Lookup Table</i>
------------	------------------------------

Description

Species lookup table for freshwater fish.

Usage

```
fish_table
```

Format

A dataframe with 321 obs and 7 variables

F_code Species field code

common_name common name

scientific_name binomial scientific name

eco_type categorical variable grouping species as "Exotic", "Fresh", or "Salt"

family_scientific family name, scientific (latin)

family_common family name, common

fwc_id unique identification number for species, can be used as primary key

flfishlrm	<i>ffishlrm: A package for summarizing and analyzing FL FWC LTM Data</i>
-----------	--

Description

ADD PACKAGE DETAILS See

Details

```
vignette("flfishlrm", package="flfishlrm")
```

intro to flfishlrm for more details

ffishlrm functions

[ltm.data.summary](#)

[species.history](#)

[len.dist](#)

get_wpt	<i>Extract waypoint data from .gpx files</i>
---------	--

Description

Extract waypoint data from .gpx files

Usage

```
get_wpt(gpx)
```

Arguments

gpx	A .gpx file generated using a Garmin GPSmap 78s
-----	---

Value

data frame containing point ID, lat, long and elevation

Examples

```
# df <- get_wpt('Waypoints_01-DEC-22.gpx')
```

guild.comp	<i>Guild Composition Plot</i>
------------	-------------------------------

Description

Create Guild Composition plot

Usage

```
guild.comp(
  ltm_dataset,
  waterbody_list = list(),
  years = list(),
  color_scale = "default",
  save = FALSE
)
```

Arguments

ltm_dataset	data, should be output from ltm.data.summary function
waterbody_list	—currently functionless, will update in future version—
years	list of years to include in figure
color_scale	color scale for output figure defaults to default r color scale, other options include "grayscale" for black and white, and "viridis" for viridis magma scale
save	boolean, if TRUE figure will be saved to file

Value

returns summarize data used to construct figures

Examples

```
data(newnans)
newn_sum <- ltm.data.summary(file=newnans)
newnans_guild <- guild.comp(newn_sum)
```

helper_addgapyears	<i>Helper function to add blank rows to account for missing years in a yearly summary table</i>
--------------------	---

Description

currently used within `ltm.data.summary()`

Usage

```
helper_addgapyears(sumTable)
```

Arguments

sumTable data.frame or tibble

Value

table

Examples

```
test_data <- structure(list(
  Year = c("2006 - 2007", "2007 - 2008", "2008 - 2009", "2009 - 2010",
    "2010 - 2011", "2012 - 2013", "2013 - 2014", "2014 - 2015",
    "2015 - 2016", "2019 - 2020", "2020 - 2021"),
  yr = c(2006, 2007, 2008, 2009, 2010, 2012, 2013, 2014, 2015, 2019, 2020),
  BOW = c(0.226666666666667, 0.24, 0.188, 0.252, 0.132, 0.02, 0.116, 0.112,
    0.064, 0.26, 0.128), LMB = c(0.006666666666667, 0.0733333333333333,
    0.064, 0.084, 0.04, 0.02, 0.048, 0.148, 0.212, 0.136, 0.12)),
  row.names = c(NA, -11L), class = c("tbl_df", "tbl", "data.frame"))
```

```
helper_addgapyears(test_data)
```

is.outlier	<i>Check if Outlier</i>
------------	-------------------------

Description

Checks whether value is an outlier

Usage

```
is.outlier(x)
```

Arguments

x vector of numerical values

Value

boolean

Examples

```
numbers = c(1:10,1:10,1:10,1000)
numbers_outlier <- is.outlier(numbers)
numbers[numbers_outlier]#'
```

is.StdFishQry	<i>Standard Fish Query Test</i>
---------------	---------------------------------

Description

Check if data is from LTM Standard Fish Query

Usage

```
is.StdFishQry(x)
```

Arguments

x dataset

Value

true/false

Examples

```

data(newnans)
is.StdFishQry(newnans)
####
is.StdFishQry(newnans[, -1])
###
new2 <- newnans
new2$newcolumn <- NA
is.StdFishQry(new2)

```

len.dist

*Length Distribution Plot – DEPRECATED May 2024***Description**

Create Length Distribution histograms for species and years selected

Usage

```

len.dist(
  datafile,
  speciesList = list(),
  years = list(),
  seasons = list(),
  print = FALSE,
  figure_filename = NA,
  fig_scale = 1
)

```

Arguments

datafile	data, should be output from ltm.data.summary function
speciesList	list of selected species, can specify by common name, scientific name or species code
years	list of years to include in figure
seasons	—currently functionless, will update in future version, to avoid errors make sure that all seasons in input dataset are the same—
print	boolean, if TRUE figure will be saved to file
figure_filename	if print=TRUE, figure will be saved to this filename
fig_scale	adjust to scale output figure size

Value

Returns the summarized data used to construct the histogram plot

Examples

```
data(newnans)
newn_sum <- ltm.data.summary("Newnans",newnans)
newnans_LD <- len.dist(newn_sum, speciesList=c("BLUE", "LMB", "BRBU"), years = c(2016:2020))
```

len_dist	<i>Length Distribution Plot</i>
----------	---------------------------------

Description

Create Length Distribution histograms for species and years selected

Usage

```
len_dist(
  datafile,
  speciesList = list(),
  years = list(),
  seasons = list(),
  print = FALSE,
  figure_filename = NA,
  fig_scale = 1,
  return_object = "data"
)
```

Arguments

datafile	data, should be output from ltm.data.summary function
speciesList	list of selected species, can specify by common name, scientific name or species code
years	list of years to include in figure
seasons	—currently functionless, will update in future version, to avoid errors make sure that all seasons in input dataset are the same—
print	boolean, if TRUE figure will be saved to file
figure_filename	if print=TRUE, figure will be saved to this filename
fig_scale	adjust to scale output figure size
return_object	<i>string</i> if "data" then function returns the dataset that is used to generate the figure, else if "ggplot" then function returns ggplot object

Value

either summarised data or a ggplot object. Specify the return object type with return_object

Examples

```
data(newnans)
newn_sum <- itm.data.summary("Newnans",newnans)
newnans_LD <- len_dist(newn_sum, speciesList=c("BLUE", "LMB", "BRBU"), years = c(2016:2020))
```

loch_blc_r_2020_age *Lochloosa Black Crappie Fall 2020 Aged Subsample*

Description

Contains total length and ages for the aged subsample of Black Crappie collected on Lochloosa Lake, FL during fall 2020 LTM trawl surveys

Usage

```
loch_blc_r_2020_age
```

Format

A dataframe with 103 obs and 5 variables

WaterBody Waterbody where collection was made

collected collection date

Species Species code; BLCR = Black Crappie *Pomoxis nigromaculatus*

tl total length in mm

age age determined from otolith reads assuming Jan 1 birthdate, in yrs

itm.data.summary *LTM Data Summary*

Description

Summarize raw LTM data from "Standard Fish Query"

Usage

```
itm.data.summary(
  waterbodyname = "No Waterbody Specified",
  file,
  outtables = 0,
  printfigs = 0,
  print_directory = getwd()
)
```


Arguments

waterbodyname	The name of the selected Waterbody as you would like it to appear in filenames and figure titles
file	path to the raw query file, should be a .csv file
outtables	option to specify which summary tables to save to working directory, default 0 exports no summary tables
printfigs	option to specify which figures to save to print directory, default 0 prints no figures, 1 prints all figures
print_directory	specify directory to save exported figures

Value

list of summarized datasets

Examples

```
#Import demo data for Newnans Lake
data(newnans)
#dplyr::summarise Newnans Lake data
newn_sum <- ltm.data.summary("Newnans Lake", file = newnans)
# Access summary tables within the newn_sum object
newn_sum$SpeciesList
newn_sum$CPUE_number
newn_sum$RawData
newn_sum$Comp_num
```

ltm.import

Import data from FWC LTM Query

Description

Imports LTM Query data and performs a set of checks that will output warnings if file does not contain expected input, or if file contains data that may not work with 'flfishltm' functions

Usage

```
ltm.import(datafile)
```

Arguments

datafile an ltm query dataset, can be either ".csv" or ".Rdata" format

Value

imported dataset

Examples

```
data(newnans)
newn <- ltm.import(newnans)
```

 newnans

LTM Fish Community Data for Newnans Lake, FL

Description

Dataset containing fish collection for Newnans Lake for the period 2006- 2020. Data is output of "Standard Fish Query"

Usage

```
newnans
```

Format

A data frame with 13,326 rows and 40 variables:

ID unique record ID #

WaterBody lake, river or other resource where data were collected, Factor

County FL County of Collection

Date collection date

Time collection time

SamplingType Sampling protocol: "Standard", "Haphazard", "NS-Random", etc TODO FILL IN ALL PROTOCOLS

Target Target species

Season Sampling season

Gear Sampling gear used

Effort Effort, expressed as time in seconds

DistanceM Distance covered during sample, in meters

Site site identification name or number

BeginLat Latitude of starting location, decimal degrees

BeginLong Longitude of starting location, decimal degrees

predict_wt	<i>Predict Weigth at Length</i>
------------	---------------------------------

Description

Predict Weigth at Length

Usage

```
predict_wt(species_code, tl, error = 0.5)
```

Arguments

species_code	FWC LTM species code
tl	Total length (mm)
error	numeric error scale expressed as a % of estimate, defaults to 0.5

Value

tibble containing tl, prediction, upper and lower bounds

Examples

```
predict_wt("BLUE", 150)  
predict_wt("BLUE", 150)
```

sfr_fish	<i>Santa Fe River Fish Community Electrofishing Data</i>
----------	--

Description

Dataset containing fish collection for Santa Fe River for the period 2006- 2021. Data is output of "Standard Fish Query"

Usage

```
sfr_fish
```

Format

A dataframe with 321 obs and 7 variables

ID unique record ID #

WaterBody lake, river or other resource where data were collected, Factor

County FL County of Collection

Date collection date

Time collection time

SamplingType Sampling protocol: "Standard", "Haphazard", "NS-Random", etc TODO FILL IN ALL PROTOCOLS

Target Target species

Season Sampling season

Gear Sampling gear used

Effort Effort, expressed as time in seconds

DistanceM Distance covered during sample, in meters

Site site identification name or number

BeginLat Latitude of starting location, decimal degrees

BeginLong Longitude of starting location, decimal degrees

Details

** NEED TO UPDATE METADATA **

species.history

Species detection history – DEPRECATED Aug 2023

Description

DEPRECATED as of 8/22/2023 use `species_history()` instead.

Usage

```
species.history(
  LTMdataset,
  exclude_species = c(),
  save = FALSE,
  filename = NA,
  fig_res = NA,
  fig_width = NA,
  fig_height = NA,
  return_object = "data"
)
```

Arguments

save boolean, if TRUE figure will be saved to file
 filename = filename for saved figure
 fig_res = specify figure resolution
 fig_width = specify figure width
 fig_height = specify figure height
 return_object *string* if "data" then function returns the dataset that is used to generate the figure, else if "ggplot" then function returns ggplot object
 ltm_dataset data, should be output from ltm.data.summary function

Details

Create Species detection history plot

Value

either summarised data or a ggplot object. Specify the return object type with return_object

Examples

```

data(newnans)
newn_sum <- ltm.data.summary(file=newnans)
newnans_dethist <- species.history(newn_sum,save=FALSE)

```

species_history	<i>Species detection history</i>
-----------------	----------------------------------

Description

Create Species detection history plot

Usage

```

species_history(
  LTMdataset,
  exclude_species = c(),
  save = FALSE,
  filename = NA,
  fig_res = NA,
  fig_width = NA,
  fig_height = NA,
  return_object = "data"
)

```

Arguments

<code>exclude_species</code>	vector of <i>string</i> identifying the species names to exclude from the output figure. Should match the spelling of the species names in the figure.
<code>save</code>	boolean, if TRUE figure will be saved to file
<code>filename</code>	= filename for saved figure
<code>fig_res</code>	= specify figure resolution
<code>fig_width</code>	= specify figure width
<code>fig_height</code>	= specify figure height
<code>return_object</code>	<i>string</i> if "data" then function returns the dataset that is used to generate the figure, else if "ggplot" then function returns ggplot object
<code>ltm_dataset</code>	data, should be output from <code>ltm.data.summary</code> function

Value

either summarised data or a ggplot object. Specify the return object type with `return_object`

Examples

```
data(newnans)
newn_sum <- ltm.data.summary(file=newnans)
newnans_dethist <- species.history(newn_sum, save=FALSE)
```

`theme_ltm`
Custom LTM theme for ggplot

Description

Custom LTM theme for ggplot

Usage

```
theme_ltm()
```

Value

theme for ggplot figures

Examples

```
data(newnans)
ggplot2::ggplot(data=newnans, ggplot2::aes(x=TotalLength,y=TotalWeight)) +
ggplot2::geom_point() +
theme_ltm()
```

Wr *Calculate Relative Weight*

Description

Calculates the Relative Weight for a sample of fish with both lengths and weights

Usage

```
Wr(dataset = NA, length_col = NA, weight_col = NA, a = NA, b = NA)
```

Arguments

dataset	dataset containing at least 2 columns, one containing a length field, and one containing weight field
length_col	string specifying the name of the column containing length
weight_col	string specifying the name of the column containing weights
a	value of the alpha parameter for relative weight calculation (specific to species of interest)
b	value of the beta parameter for relative weight calculation (specific to species of interest)

Value

returns vector of relative weights

Examples

```
library(dplyr)
data(newnans)
#Calculate relative weight for all LMB collected from Newnans Lake
newn_lmb <- newnans %>% dplyr::filter(SpeciesCode == "LMB" )
newnans_lmb_Wr <- Wr(dataset = newn_lmb,
                    length_col = "TotalLength",
                    weight_col = "TotalWeight",
                    a = .3245,
                    b = 3.21)
```

Index

* datasets

- fish_table, 10
- loch_blcrcr_2020_age, 16
- newnans, 18
- sfr_fish, 19

add_zero_count, 2

age.key, 3

check_expected_columns, 4

cpue, 4

cpue.plot, 5

cpue_plot, 6

cpue_plot(), 5

create_ltm_analysis_template, 8

create_ltm_report, 9

create_ltm_report(), 8

deprecate, 9

fish_table, 10

flfishltm, 10

FUTURE FEATURE, 8

get_wpt, 11

guild.comp, 11

helper_addgapyears, 12

is.outlier, 13

is.StdFishQry, 13

len.dist, 10, 14

len_dist, 15

loch_blcrcr_2020_age, 16

ltm.data.summary, 10, 16

ltm.data.summary(), 12

ltm.import, 17

newnans, 18

predict_wt, 19

sfr_fish, 19

species.history, 10, 20

species_history, 21

species_history(), 20

theme_ltm, 22

Wr, 23