Package: FishEconProdOutput (via r-universe)

October 2, 2024

Type Package

Title Price Method Fisheries Economics Total Factor Productivity Outputs

Version 0.1.1

URL https://EmilyMarkowitz-NOAA.github.io/FishEconProdOutput/, https://github.com/EmilyMarkowitz-NOAA/FishEconProdOutput

BugReports https://github.com/EmilyMarkowitz-NOAA/FishEconProdOutput/issues/

License GPL-3

Depends R (>= 3.6.0)

Description Here we provide methodology guidelines on how to calculate fishery productivity measurement at the individual fishery and aggregate sector levels. Attention is given to the constructions of output and total factor productivity based on available data and a bottom-up approach. Given that there is no nation-wide standard cost survey, we recommend starting with measuring TFP at the fishery level based on a translog gross output production possibility frontier using index number techniques. Special attention is given to measuring quality-adjusted physical capital inputs in the bottom-up approach.

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports data.table, grDevices, tidyverse, tidyr, ggplot2, dplyr, plyr, rlist, taxize, filesstrings, ggpubr, xlsx, readxl, knitr

VignetteBuilder knitr

Suggests rmarkdown

Repository https://noaa-fisheries-integrated-toolbox.r-universe.dev

RemoteUrl https://github.com/EmilyMarkowitz-NOAA/FishEconProdOutput

RemoteRef HEAD

RemoteSha 8bb9430698e631d29ce1e0427aa0068d8b7cb021

Contents

`counter00X` *Counter*

Description

This funciton advances a value of 'counter0' +1 each time it is used.

Usage

```
counter00X(counter0)
```

Arguments

`counter0`	value to be advanced by 1.

Value

counter

Examples

```
counter00X(c(1, 2))
```

`itis_reclassify` *Reclassify ITIS species based off a list of higher taxonomic groupings*

Description

Reclassify ITIS species based off a list of higher taxonomic groupings

Usage

```
itis_reclassify(tsn, categories, uncategorized_name = "Uncategorized")
```

Arguments

`tsn`	A vector of Taxonomic Serial Numbers to be evaluated.
`categories`	A list of the categories and associated TSN values. within a list of a category, a minus (-) in front of a number is short hand to remove organisms within that tsn's taxonomy from being listed in a category. See the example for an instance where that makes sense.
`uncategorized_name`	A string of what to call the missing value.

Value

df_out, tsn_indata

Examples

```
itis_reclassify(tsn = c(83677, # subphylum Crustacea; shellfish
                        172746, # Scophthalmus aquosus; finfish
                        173747, # class Reptilia; uncategorized as part of tetrapoda
                        98678), # Cancer borealis; shellfish
                categories = list('Finfish' = c(914179, #  Infraphylum Gnathostomata
                                                -914181), # Tetrapoda; - = do NOT include
                                  "Shellfish" = c(82696, # Phylum Arthropoda
                                                  69458)), # Phylum Mollusca
                uncategorized_name = "uncategorized")
```

`land` *Modified Landings Data*

Description

Modified and cleaned data from NOAA Fisheries Office of Science and Technology's Fisheries Statistics Division's Commercial Landings Query, Available at: https://foss.nmfs.noaa.gov/apexfoss/f?p=215:200:::::: for all coastal states combined with state and regional data.

Usage

```
data(land)
```

Format

A data frame with 53940 rows and 10 variables:

Year four-digit year

Pounds weight of fish caught, in pounds

Dollars value of fish caught, in USD

category category of organism. For our analysis, we aggregated landings and revenue data into two different fisheries: finfish (defined by all organisms in the infraphylum Gnathostomata) and shellfish (defined by all organisms in the phyla Arthropoda and Mollusca)

Tsn Taxonomic Serial Number (TSN) as defined by the Integrated Taxonomic Information System Distinguishing species fishery categories was done easily with the R package 'taxize'

State The state the fish was caught in, in full name

Region The region the fish was caught in, in full name

abbvreg The region the fish was caught in, abbrevated

Source

NOAA Fisheries FOSS

Examples

```
data(land)
```

`numbers0` *How Many Speices are in a Dataset numeric identifier*

Description

This funciton standardizes the length of the category or species numbers e.g.,(numbers of 33, 440, and 1 are converted to 033, 440, and 001)

Usage

```
numbers0(x)
```

Arguments

`x`	x is a string of all the numbers you are interested in 'standardizing'.

Examples

```
numbers0(x = c(1,14,302))
```

OutputAnalysis *Run Analysis for the US and several regions.*

Description

Run Analysis for the US and several regions.

Usage

```
OutputAnalysis(
  landings_data,
  category0,
  baseyr,
  titleadd,
  dir_analyses,
  reg_order = c("National", "North Pacific", "Pacific", "Western Pacific (Hawai`i)",
    "New England", "Mid-Atlantic", "Northeast", "South Atlantic", "Gulf of Mexico"),
  reg_order_abbrv = c("US", "NP", "Pac", "WP", "NE", "MA", "NorE", "SA", "GOM"),
  skipplots = FALSE,
  save_outputs_to_file = TRUE
)
```

Arguments

`landings_data` Landings data with the following columns: "Year", "Pounds", "Dollars", category0, "Tsn", "State"

`category0` A character string. The column where the category is defined.

`baseyr` Numeric year (YYYY). The base year you are assessing the anaylsis with. Typically this is the earliest year in the data set, but it can be any year you choose.

`titleadd` A string to add to the file with the outputs to remind you why this particular analysis was interesting.

`dir_analyses` A directory that your analyses will be saved to (e.g., "./output/").

`reg_order` The US and each region that you would like to assess. Default = c("National", "North Pacific", "Pacific", "Western Pacific (Hawai‘i)", "New England", "Mid-Atlantic", "Northeast", "South Atlantic", "Gulf of Mexico").

`reg_order_abbrv` Acronym of the US and each region listed in reg_order. Default = c("US", "NP", "Pac", "WP", "NE", "MA", "NorE", "SA", "GOM").

`skipplots` TRUE (create and save plots) or don’t FALSE.

`save_outputs_to_file` TRUE (save outputs from analysis within function) or don’t FALSE.

Value

warnings_list, editeddata_list, index_list, spp_list, figures_list, gridfigures_list

Examples

```
browseVignettes("FishEconProdOutput")
```

---

plotnlines *Plot n lines in ggplot*

Description

This funciton plots n lines in a ggplot.

Usage

```
plotnlines(dat, titleyaxis = "", title0 = "")
```

Arguments

`dat`	Default data.
`titleyaxis`	y-axis title.
`title0`	Title of plot.

Examples

```
dat<-data.frame(Year = c(2016:2020, 2016:2020),
                val = rnorm(n = 10, mean = 500, sd = 100),
                cat = c(rep_len("A", 5), rep_len("B", 5)))
plotnlines(dat = dat,
           titleyaxis = "Normal Distribution of 10 Numbers",
           title0 = "Anywhere")
```

PriceMethodOutput *Price Method*

Description

This function calculates the Implicit Quanity Output at Fishery Level by systematically runing the Price Method Productivity Output analysis for all species of each cateorgy.

Usage

```
PriceMethodOutput(dat00, baseyr, title0 = "", place = "", category0)
```

Arguments

`dat00` Dataset.

`baseyr` Numeric year (YYYY). The base year you are assessing the anaylsis with. Typically this is the earliest year in the data set, but it can be any year you choose.

`title0` Title of analysis

`place` Area you are assessing the analysis for. This can also be used as a title.

`category0` A character string. The column where the category is defined. A character string.

`PriceMethodOutput_Category` *Price Methods - Category Level*

Description

This function systematically runs the Price Method Productivity Output analysis for all species of a cateorgy.

Usage

```
PriceMethodOutput_Category(
  dat00,
  ii,
  category,
  category0,
  baseyr,
  maxyr,
  minyr,
  warnings_list = ls()
)
```

Arguments

`dat00` Default dataset.

`ii` Category number.

`category` A character string. A unique string from the 'category0' column of the group being evaluated.

`category0` A character string. The column where the category is defined.

`baseyr` Numeric year (YYYY). The base year you are assessing the anaylsis with. Typically this is the earliest year in the data set, but it can be any year you choose.

`maxyr` The maxium year to assess in the dataset.

`minyr` The minium year to assess in the dataset.

`warnings_list` A list where warnings are stored. If using this function in the PriceMethodOutput it will be inherited. If using outside of that function, put ls().

tornb *Tornqvist Price Index Base Year Function*

Description

Tornqvist Price Index Base Year Function

Usage

```
tornb(dat, Year = "Year", pvar = "p", vvar = "v", prodID = "prod", baseyr)
```

Arguments

`dat`	The dataset you would like to use.
`Year`	Name of the column holding year data.
`pvar`	Name of the column holding price data.
`vvar`	Name of the column holding value data.
`prodID`	Name of the column holding prodID data.
`baseyr`	The year dollar values need to be in.

Examples

```
tornb(dat = data.frame("Year" = c(2001:2020, 2001:2020, 2001:2020, 2001:2020),
                       "p" = rnorm(n = 80, mean = 1, sd = .1),
                       "v" = rnorm(n = 80, mean = 500, sd = 300),
                       "prod" = c(rep_len("A", 20), rep_len("B", 20),
                                  rep_len("C", 20), rep_len("D", 20))),
     Year = "Year",
     pvar = "p",
     vvar = "v",
     prodID = "prod",
     baseyr = 2015)
```

tornc *Tornqvist Price Index Base Year chain Function*

Description

Tornqvist Price Index Base Year chain Function

Usage

```
tornc(dat, Year = "Year", pvar = "p", vvar = "v", prodID = "prod", baseyr)
```

Arguments

`dat` The dataset you would like to use.

`Year` Name of the column holding year data.

`pvar` Name of the column holding price data.

`vvar` Name of the column holding value data.

`prodID` Name of the column holding prodID data.

`baseyr` The year dollar values need to be in.

Examples

```
tornc(dat = data.frame("Year" = c(2001:2020, 2001:2020, 2001:2020, 2001:2020),
                       "p" = rnorm(n = 80, mean = 1, sd = .1),
                       "v" = rnorm(n = 80, mean = 500, sd = 300),
                       "prod" = c(rep_len("A", 20), rep_len("B", 20),
                                  rep_len("C", 20), rep_len("D", 20))),
     Year = "Year",
     pvar = "p",
     vvar = "v",
     prodID = "prod",
     baseyr = 2015)
```

`xunits` *Standardize Units*

Description

This funciton standardizes units of a value. For example, 1,000,000 would become "1 Million."

Usage

```
xunits(val, combine = T)
```

Arguments

`val` Value to be evaluated.

`combine` TRUE/FALSE (Default = TRUE). Asks if you want two strings (FALSE) or 1 concatenated string (TRUE).

Examples

```
xunits(1234567890)
```

Index