Package 'popstudy'

October 18, 2023

Type Package

Title Applied Techniques to Demographic and Time Series Analysis

Version 1.0.1

Depends R (>= 3.5.0)

Maintainer Cesar Gamboa-Sanabria <info@cesargamboasanabria.com>

Description The use of overparameterization is proposed with combinatorial analysis to test a broader spectrum of possible ARIMA models.
In the selection of ARIMA models, the most traditional methods such as correlograms or others, do not usually cover many alternatives to define the number of coefficients to be estimated in the model, which represents an estimation method that is not the best.
The popstudy package contains several tools for statistical analysis in demography and time series based in Shryock research (Shryock et. al. (1980) <https://books.google.co.cr/books?id=8Oo6AQAAMAAJ>).

Imports ggplot2, magrittr, lubridate, dplyr, tidyr, stats, demography, forecast, ggpubr, moments, grid, DescTools, rcompanion, utils, corrr, Hmisc, corrplot, correlation, parallel, here, stringr, scales, rainbow, Rdpack

License GPL-3

Encoding UTF-8

RdMacros Rdpack

LazyData true

URL https://www.cesargamboasanabria.com

RoxygenNote 7.2.3

Suggests testthat, snow

NeedsCompilation no

Author Cesar Gamboa-Sanabria [aut, mdc, cph, cre] (<https://orcid.org/0000-0001-6733-4759>)

Repository CRAN

Date/Publication 2023-10-17 23:50:02 UTC

R topics documented:

anonymous *anonymous*

Description

Anonymizing a data frame by avoiding vulnerability to a rainbow table attack.

Usage

```
anonymous(data, ID, string_length = 15, SEED = NULL)
```

Arguments

data	data.frame. A dataset with the a variable to change its values.
ID	character. A string with the variable name to change its values.
string_length	numeric. It defines the string length of the new identification variable.
SEED	to be passed to set.seed to keep the the same new id's.

Value

anonymous function returns a list with two data frames:

data	original data with the new variable
dictionary	data frame with the original variable and the new one

Author(s)

Cesar Gamboa-Sanabria

References

Oechslin P (2003). "Making a Faster Cryptanalytic Time-Memory Trade-Off." In Boneh D (ed.), *Advances in Cryptology - CRYPTO 2003*, 617–630. ISBN 978-3-540-45146-4.

Examples

```
library(dplyr)
df <- select(mutate(mtcars, id=rownames(mtcars)), id, !contains("id"))
anonymous(df, ID="id", string_length = 5, SEED=160589)
```

Beers *Beers multipliers*

Description

Method to open five-year grouped ages into specific ages.

Usage

```
Beers(data, ...)
```

Arguments

data	data.drame. It contains at least two variables: five-year grouped ages and population.
...	Arguments to be passed to dplyr::select, i.e., age and population, respectively.

Value

Beers returns a data.frame with specific ages and populations.

Author(s)

Cesar Gamboa-Sanabria

References

Shryock HS, Siegel JS, Larmon EA, of the Census USB (1980). *The Methods and Materials of Demography*, number v. 1 in The methods and materials of demography. Department of Commerce, Bureau of the Census. https://books.google.co.cr/books?id=8Oo6AQAAMAAJ.

See Also

Sprague

Examples

```
Beers(Ecuador1990, age, population)
```

births_deaths *Births and deaths data*

Description

Simulated data for Lexis Diagram examples.

Usage

```
data("births_deaths")
```

Format

The format is: List of 2 $ births: tibble [32 x 3] (S3: tbl_df/tbl/data.frame) ..$ sex : chr [1:32] "male" "male" "male" "male" ... ..$ date_reg: Date[1:32], format: ... ..$ births : num [1:32] 121558 126446 130839 130911 127524 ... $ deaths: tibble [112 x 4] (S3: tbl_df/tbl/data.frame) ..$ sex : chr [1:112] "male" "male" "male" "male" ... ..$ date_reg: Date[1:112], format: ... ..$ age : num [1:112] 0 0 0 0 0 0 0 0 0 0 ... ..$ deaths : num [1:112] 11411 10494 10814 9872 9457 ...

Examples

```
data(births_deaths)
summary(births_deaths)
```

CEB *Children Ever Born Data*

Description

Children Ever Born Data from Bolivia's 2001 Census data.

Usage

```
data("CEB")
```

Format

A data frame with 27 observations on 8 variables for each five-year grouped age.

Source

https://www.ine.gob.bo/

Examples

```
data(CEB)
summary(CEB)
```

correlate_df *correlate_df*

Description

Compute correlations in a data frames.

Usage

```
correlate_df(data, keep_class = NULL)
```

Arguments

data	data.frame. A dataset with the variables to correlate.
keep_class	list. A list that contains desire classes for specyfic variables.

Details

correlate_df takes data.frame class objects and works only with numeric, factor, and ordered class variables, so a previous data cleaning is needed for optimal results. A variable is considered nominal when it is a factor variable with more than two levels, and it is no ordered. When a numeric variable has only two different values, it is considered a binary variable. Also, when a factor variable has only two levels, it is regarded as a binary variable. The computed correlation will depend on the paired-variables class: Pearson method when both variables are numeric, Kendall correlation with a numeric and an ordinal variable, point-biserial with a numeric and a binary variable, Polychoric correlation with two ordinal variables, Tetrachoric correlation when both are binary, Rank-Biserial when one is ordinal, and the other is binary; and Kruskal's Lambda with one binary and one nominal, or both nominal variables. A Gaussian linear model is fitted to estimate the multiple correlation coefficient in the specific cases of one nominal variable and another numerical or ordered, so the user should take it carefully.

Value

correlate_df function returns a list with three objects: A data-frame with the correlation matrix and two correlation plots.

Author(s)

Cesar Gamboa-Sanabria

References

Khamis H (2008). "Measures of Association: How to Choose?" *Journal of Diagnostic Medical Sonography*, **24**(3), 155-162. doi:10.1177/8756479308317006.

Examples

```
df <- data.frame(cont1=rnorm(100),
cont2=rnorm(100),
ordi1=factor(sample(1:5, 100, replace = TRUE), ordered = TRUE),
ordi2=factor(sample(1:7, 100, replace = TRUE), ordered = TRUE),
bin1=rbinom(100, 1, .4),
bin2=rbinom(100, 1, .6),
nomi1=factor(sample(letters[1:8], 100, replace = TRUE)),
nomi2=factor(sample(LETTERS[1:8], 100, replace = TRUE)))

correlate_df(df)
```

`CR_births` *CR_births*

Description

Births registers in Costa Rica.

Usage

```
data("CR_births")
```

Format

A data frame with 8434 observations on the following 2 variables.

`date_reg` a Date

`births` a numeric vector

Source

https://inec.cr/

Examples

```
data(CR_births)
summary(CR_births)
```

CR_deaths *CR_deaths*

Description

Deaths registers in Costa Rica.

Usage

```
data("CR_deaths")
```

Format

A data frame with 229462 observations on the following 3 variables.

`date_reg` a Date

`age` a numeric vector

`deaths` a numeric vector

Source

https://inec.cr/

Examples

```
data(CR_deaths)
summary(CR_deaths)
```

CR_fertility_rates_1950_2011 *Costa Rica fertility rates*

Description

Fertility rates for Costa Rica 1950-2011.

Usage

```
data("CR_fertility_rates_1950_2011")
```

Format

A data frame with 2170 observations on the following 3 variables.

`Year` a numeric vector

`Age` a numeric vector

`Female` a numeric vector with fertility rates

Source

https://inec.cr/

Examples

```
data(CR_fertility_rates_1950_2011)
summary(CR_fertility_rates_1950_2011)
```

`CR_mortality_rates_1950_2011`
Costa Rica mortality rates

Description

Mortality rates for Costa Rica 1950-2011.

Usage

```
data("CR_mortality_rates_1950_2011")
```

Format

A data frame with 2170 observations on the following 4 variables.

`Year` a numeric vector

`Age` a numeric vector

`Female` a numeric vector with female mortality rates

`Male` a numeric vector with male mortality rates

`Total` a numeric vector with total mortality rates

Source

https://inec.cr/

Examples

```
data(CR_mortality_rates_1950_2011)
summary(CR_mortality_rates_1950_2011)
```

CR_mortality_rates_2010_2015

Costa Rica Mortality Rates

Description

Mortality rates for Costa Rica in 2010-2015

Usage

```
data("CR_mortality_rates_2010_2015")
```

Format

A data frame with 7656 observations on the following 4 variables.

`Year` a numeric vector

`Age` a numeric vector

`Female` a numeric vector with female mortality rates

`Male` a numeric vector with male mortality rates

Source

https://inec.cr/

Examples

```
data(CR_mortality_rates_2010_2015)
summary(CR_mortality_rates_2010_2015)
```

CR_populations_1950_2011

Costa Rica population

Description

Estimated y projected populations for Costa Rica 1950-2011.

Usage

```
data("CR_populations_1950_2011")
```

Format

A data frame with 7656 observations on the following 4 variables.

`Year` a numeric vector

`Age` a numeric vector

`Female` a numeric vector with female population

`Male` a numeric vector with male population

`Total` a numeric vector with total population

Source

https://inec.cr/

Examples

```
data(CR_populations_1950_2011)
summary(CR_populations_1950_2011)
```

CR_populations_1950_2015

Costa Rica population

Description

Estimated y projected populations for Costa Rica 1950-2015.

Usage

```
data("CR_populations_1950_2015")
```

Format

A data frame with 7656 observations on the following 4 variables.

`Year` a numeric vector

`Age` a numeric vector

`Female` a numeric vector with female population

`Male` a numeric vector with male population

Source

https://inec.cr/

Examples

```
data(CR_populations_1950_2015)
summary(CR_populations_1950_2015)
```

CR_women_childbearing_age_1950_2011 *Costa Rica population*

Description

Estimated y projected populations for Costa Rica 1950-2011.

Usage

```
data("CR_women_childbearing_age_1950_2011")
```

Format

A data frame with 7656 observations on the following 4 variables.

`Year` a numeric vector

`Age` a numeric vector

`Female` a numeric vector with women of reproductive age population

Source

https://inec.cr/

Examples

```
data(CR_women_childbearing_age_1950_2011)
summary(CR_women_childbearing_age_1950_2011)
```

descriptive_plot *descriptive_plot*

Description

Plot density with descriptive statistics for numerical values.

Usage

```
descriptive_plot(data, ..., labels = NULL, ylab = "Density")
```

Arguments

`data`	data.frame.
`...`	additional arguments to be passed to `dplyr::select()`.
`labels`	A vector with x-axis labels.
`ylab`	y-axis label.

Value

`descriptive_plot` function returns a plot with density and descriptive statistics.

Author(s)

Cesar Gamboa-Sanabria

Examples

```
df <- data.frame(var1=rpois(50, 6), var2=rgamma(50, shape=5,rate=.4), var3=rnorm(50, 10))
descriptive_plot(df, var1, var3)
```

Ecuador1990 *Ecuador1990*

Description

Ecuador census data in 1990 by grouped ages.

Usage

```
data("Ecuador1990")
```

Format

A data frame with 21 observations on the following 4 variables.

`age` a factor with levels `0-4` `5-9` `10-14` `15-19` `20-24` `25-29` `30-34` `35-39` `40-44` `45-49` `50-54` `55-59` `60-64` `65-69` `70-74` `75-79` `80-84` `85-89` `90-94` `95-99` `100+`

`male` a numeric vector with males population

`female` a numeric vector with female population

`population` a numeric vector Ecuador population

Source

https://microdata.worldbank.org/index.php/catalog/499

Examples

```
data(Ecuador1990)
summary(Ecuador1990)
```

`El_Badry` *El-Badry method*

Description

The method corrects the zero parity omission error.

Usage

```
El_Badry(data, age, CEB, childs, req_ages = NULL)
```

Arguments

`data`	data.drame. It contains at least three variables: five-year grouped ages, number of childs and Children Ever Born (CEB).
`age`	variable name in `data` of the five-year grouped age.
`CEB`	variable name in `data` with number of Children Ever Born .
`childs`	variable name in `data` with the number of childs for each five-year grouped age and number of Children Ever Born.
`req_ages`	optional character string that specifies the five-year grouped age to estimates the intercept.

Value

`Moultrie` returns a list with two elements: a data.frame with corrected children for each number of Children Ever Born and five-year grouped ages and a data.frame with combinations of five-year grouped age to estimate intercept, slope, and R-squared. By default, the method uses the best value of R-squared to apply the El Badry correction.

Author(s)

Cesar Gamboa-Sanabria

References

Moultrie TA, Dorrington RE, Hill AG, Hill K, Timæus IM, Zaba B (2013). *Tools for demographic estimation*. International Union for the Scientific Study of Population.

See Also

`CEB` `Moultrie`

Examples

```
CEB_data <- tidyr::gather(CEB, ages, childs, -Children_Ever_Born)
results <- Moultrie(CEB_data, ages, childs, Children_Ever_Born)
CEB_data <- tidyr::pivot_wider(results, names_from=age, values_from=childs)
CEB_data <- tidyr::gather(CEB_data, ages, children, -CEB)
El_Badry(CEB_data,ages, CEB, children)
```

---

grouped_age_CR_pop *grouped_age_CR_pop*

Description

Costa Rica population by 5-year-group ages in 2011.

Usage

```
data("grouped_age_CR_pop")
```

Format

A data frame with 16 observations on the following 2 variables.

age an ordered factor with levels 0 - 4 < 5 - 9 < 10 - 14 < 15 - 19 < 20 - 24 < 25 - 29 < 30 - 34 < 35 - 39 < 40 - 44 < 45 - 49 < 50 - 54 < 55 - 59 < 60 - 64 < 65 - 69 < 70 - 74 < 75 and more

pop a numeric vector with the populaion

Source

https://inec.cr/

Examples

```
data(grouped_age_CR_pop)
str(grouped_age_CR_pop)
```

`growth_exp` *Exponential growth*

Description

Assuming an exponential behavior estimates the population size at time t, the growth rate, or population at time 0.

Usage

```
growth_exp(Nt = NULL, N0 = NULL, r = NULL, t0, t, time_interval, date = FALSE)
```

Arguments

`Nt`	numeric. The population at time t. If null and date = FALSE, then estimate the population at time t.
`N0`	numeric. The population at time 0. If null and date = FALSE, then estimate the population at time 0.
`r`	numeric. The growth rate. If null and date = FALSE, then estimate the growth rate for the time period [t0,t].
`t0`	numeric. An object of class character with the date for the first population.
`t`	numeric. An object of class character with the date for the second population.
`time_interval`	character. A string with the time interval to calculate Delta_t.
`date`	logical. If TRUE, then estimates the moment t when Nt reaches a specific value.

Value

`growth_exp` returns a data frame with N0, Ntr, t0, t, delta, and time_interval for desire parameters.

Author(s)

Cesar Gamboa-Sanabria

References

Shryock HS, Siegel JS (2013). *The Methods and Materials of Demography*, Studies in Population. Elsevier Science. ISBN 9781483289106, https://books.google.co.cr/books?id=HVW0BQAAQBAJ.

See Also

`growth_linear`, `growth_logistic`

Examples

```
# According to the Panama census in 2000-05-14,
# the population was 2,839,177. In 2010-05-16, the census
# calculates 3,405,813 population.
# To get r:

growth_exp(N0=2839177, Nt=3405813, t0="2000-05-14", t="2010-05-16", time_interval = "years")

# To get Nt at 2000-06-30:

growth_exp(N0=2839177, r=0.0182, t0="2000-05-14", t="2000-06-30", time_interval = "years")

# The time when the population will be 5,000,000.

growth_exp(N0=2839177, Nt=5000000, r=0.0182, t0="2000-05-14", date=TRUE)
```

---

growth_linear *Linear growth*

Description

Assuming an linear behavior, estimates the population size at time t, the growth rate, or population at time 0.

Usage

```
growth_linear(
  Nt = NULL,
  N0 = NULL,
  r = NULL,
  t0,
  t,
  time_interval,
  date = FALSE
)
```

Arguments

Nt	numeric. The population at time t. If null and date = FALSE, then estimate the population at time t.
N0	numeric. The population at time 0. If null and date = FALSE, then estimate the population at time 0.
r	numeric. The growth rate. If null and date = FALSE, then estimate the growth rate for the time period [t0,t].
t0	numeric. An object of class character with the date for the first population.

t	numeric. An object of class character with the date for the second population.
time_interval	character. A string with the time interval to calculate Delta_t.
date	logical. If TRUE, then estimates the moment t when Nt reaches a specific value.

Value

growth_linear returns a data frame with N0, Ntr, t0, t, delta, and time_interval for desire parameters.

Author(s)

Cesar Gamboa-Sanabria

References

Shryock HS, Siegel JS (2013). *The Methods and Materials of Demography*, Studies in Population. Elsevier Science. ISBN 9781483289106, https://books.google.co.cr/books?id=HVW0BQAAQBAJ.

See Also

growth_exp,growth_logistic

Examples

```
# According to the Panama census at 2000-05-14,
# the population was 2,839,177. In 2010-05-16, the census
# calculates 3,405,813 population.
# To get r:

growth_linear(N0=2839177, Nt=3405813, t0="2000-05-14", t="2010-05-16", time_interval = "years")

# To get Nt at 2000-06-30:

growth_linear(N0=2839177, r=0.0182, t0="2000-05-14", t="2000-06-30", time_interval = "years")

# The time when the population will be 5,000,000.

growth_linear(N0=2839177, Nt=5000000, r=0.0182, t0="2000-05-14", date=TRUE)
```

`growth_logistic` *Logistic growth*

Description

Given two pivots and limits, estimates the growth assuming a logistic behavior.

Usage

```
growth_logistic(pivot_values, pivot_years, upper, lower, t)
```

Arguments

`pivot_values`	numeric. Reference values to estimate, like TFR for two specific years.
`pivot_years`	numeric. Reference years to estimate for both values in `pivot_values`.
`upper`	numeric. Upper asymptotic value.
`lower`	numeric. Lower asymptotic value.
`t`	numeric. Year to get logistic value.

Value

`growth_logistic` returns the logistic estimation for specified year.

Author(s)

Cesar Gamboa-Sanabria

References

Shryock HS, Siegel JS (2013). *The Methods and Materials of Demography*, Studies in Population. Elsevier Science. ISBN 9781483289106, https://books.google.co.cr/books?id=HVW0BQAAQBAJ.

See Also

`growth_exp`, `growth_linear`

Examples

```
# Given TFR values 3.32 and 2.85 for the years 1986 and 1991, respectively,
# estimate the TFR in 1987 assuming 1.5 as lower limit and 8 as upper limit.

growth_logistic(pivot_values = c(3.32, 2.85), pivot_years = c(1986, 1991),
upper = 8, lower=1.5, t=1987)
```

karup_king *karup_king*

Description

Separate grouped-age data to simple ages data using Karup-King separation factors.

Usage

```
karup_king(data)
```

Arguments

data	data.frame. A dataset with two variables: age, the group age each 5 years; and pop, the population for that age.

Value

karup_king function returns a a data frame with separated simple ages.

Author(s)

Cesar Gamboa-Sanabria

References

Shryock HS, Siegel JS, Larmon EA, of the Census USB (1980). *The Methods and Materials of Demography*, number v. 2 in The Methods and Materials of Demography. U.S. Department of Commerce, Bureau of the Census. https://books.google.co.cr/books?id=SuXrAAAAMAAJ.

See Also

grouped_age_CR_pop

Examples

```
karup_king(grouped_age_CR_pop)
```

karup_king_factors *karup_king_factors*

Description

Karup-King separation factors.

Usage

```
data("karup_king_factors")
```

Format

A data frame with 76 observations on the following 7 variables.

`age` a character vector with simple ages

`f1` a numeric vector, Karup-King factor

`f2` a numeric vector, Karup-King factor

`f3` a numeric vector, Karup-King factor

`d1` a numeric vector, used in `karup_king` function, do not edit by hand

`d2` a numeric vector, used in `karup_king` function, do not edit by hand

`d3` a numeric vector, used in `karup_king` function, do not edit by hand

References

Shryock HS, Siegel JS, Larmon EA, of the Census USB (1980). *The Methods and Materials of Demography*, number v. 2 in The Methods and Materials of Demography. U.S. Department of Commerce, Bureau of the Census. https://books.google.co.cr/books?id=SuXrAAAAMAAJ.

Examples

```
data(karup_king_factors)
str(karup_king_factors)
```

Lexis *Lexis diagram*

Description

Plot a Lexis Diagram from births and deaths data for a given year, month, and day with specific simple ages.

Usage

```
Lexis(
  deaths_data,
  births_data,
  first.date = NULL,
  choose_year,
  choose_month,
  choose_day,
  ages,
  factors = NULL
)
```

Arguments

deaths_data	data.frame. A dataset with three variables: date_reg, the registered death date, age, the age of decease; and deaths, the deaths number for that date. See CR_deaths.
births_data	data data.frame. A dataset with two variables: date_reg, the registered birth date; and births, the births number for that date. See CR_births.
first.date	character. Optional argument that specifies the first date of interest.
choose_year	numeric. The year from which the countdown begins until the desired minimum age is reached.
choose_month	numeric. The month from which the countdown begins until the desired minimum age is reached.
choose_day	numeric. The day from which the countdown begins until the desired minimum age is reached.
ages	numeric. An ages vector to plot the diagram.
factors	numeric. Optional argument to set specific factors to set alpha and delta sections in Lexis Diagram.

Value

Lexis function returns a list with two objects: diagram, the Lexis diagram; and deaths, the estimated deaths number.

Author(s)

Cesar Gamboa-Sanabria

References

Rau R, Bohk-Ewald C, Muszynska MM, Vaupel JW (2017). *Visualizing Mortality Dynamics in the Lexis Diagram*, The Springer Series on Demographic Methods and Population Analysis. Springer International Publishing. ISBN 9783319648200, https://books.google.co.cr/books?id=ttpCDwAAQBAJ.

Examples

```
Lexis(CR_deaths, CR_births, choose_year=2011, choose_month=1, choose_day=1, ages=0:9)$diagram

##Lexis diagram with specific factors
data("births_deaths")
Births <- dplyr::filter(births_deaths$births, sex=="male")
Deaths <- dplyr::filter(births_deaths$deaths, sex=="male")
Lexis(deaths_data=Deaths, births_data=Births, first.date = "1999-01-01",
choose_year=2007, choose_month=1, choose_day=1, ages=0:4,
factors = c(.2,.41,.47,.48,.48))$diagram
```

Lifetable *Life Table*

Description

Estimates a lifetable from mortality rates and population data.

Usage

```
Lifetable(
  rates,
  pops,
  sex,
  max_age = NULL,
  first_year,
  threshold,
  jump,
  element = c("mx", "qx", "lx", "dx", "Lx", "Tx", "ex", "rx"),
  ...
)
```

Arguments

rates	character. A character string that specifies mortality data path. The dataset is a .txt file like CR_mortality_rates_2010_2015.
pops	character. A character string that specifies population data path. The dataset is a .txt file like CR_populations_1950_2015.
sex	character. "female" or "male".
max_age	numeric. Desire omega age. If NULL, Lifetable function takes the dataset's maximum age.
first_year	numeric. First year to start estimation.
threshold	numeric. Maximum forecast year.
jump	character. Same purpose to jumpchoice argument in forecast function.

element	character. Wanted estimation element, one of "mx", "qx", "lx", "dx", "Lx", "Tx", "ex" or "rx".
...	additional arguments to be passed to read.demogdata, such as label.

Value

Lifetable function returns a list with both data frames, wide and long format, for specified element in argument element for desire years.

Author(s)

Cesar Gamboa-Sanabria

References

Wunsch G, Mouchart M, Duchêne J (2002). *The Life Table: Modelling Survival and Death*, European Studies of Population. Springer Netherlands. ISBN 9781402006388, https://books.google.co.cr/books?id=ySex55d4nlsC.

Examples

```
## Not run:
 write.table(CR_mortality_rates_2010_2015,
 file = "CR_mortality_rates_2010_2015.txt",
 sep = "\t", row.names = FALSE, quote = FALSE)

 write.table(CR_populations_1950_2015,
 file = "CR_populations_1950_2015.txt",
 sep = "\t", row.names = FALSE, quote = FALSE)

 Lifetable("CR_mortality_rates_2010_2015.txt", "CR_populations_1950_2015.txt",
 sex="female", first_year=2011, threshold=2150, jump="actual", max_age = 100,
 element="ex", label="CR")

## End(Not run)
```

---

mortality_projection *mortality_projection*

Description

Forecasting mortality rates.

Usage

```
mortality_projection(
  mortality_rates_path,
  total_population_path,
  omega_age,
  horizon,
  first_year_projection,
  ...
)
```

Arguments

mortality_rates_path	character. Path to Mortality rates in a .txt file.
total_population_path	character. Path to Populations in a .txt file.
omega_age	numeric. Maximum age.
horizon	numeric. The forecast horizon.
first_year_projection	numeric. Year for the base population.
...	additional arguments to be passed to forecast::Arima().

Value

mortality_projection returns an object of class fmforecast with with both female and male mortality projections and the components of demography::forecast.lca().

Author(s)

Cesar Gamboa-Sanabria

Examples

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

data(CR_mortality_rates_1950_2011)

#CR_mortality_rates_1950_2011 %>%
#write.table(.,
#file = "CR_mortality_rates_1950_2011.txt",
#sep = "\t",
#row.names = FALSE,
#col.names = TRUE,
#quote = FALSE)
```

```
data(CR_populations_1950_2011)

#CR_populations_1950_2011 %>%
#write.table(.,
#file = "CR_populations_1950_2011.txt",
#sep = "\t",
#row.names = FALSE,
#col.names = TRUE,
#quote = FALSE)

#result <- mortality_projection(mortality_rates_path = "CR_mortality_rates_1950_2011.txt",
#total_population_path = "CR_populations_1950_2011.txt",
#omega_age = 115, first_year_projection = 2011, horizon = 2150)


## End(Not run)
```

`Moultrie` *Moultrie rule for Children Ever Born*

Description

Moultrie's proposal for correction of Children Ever Born in five-year grouped ages.

Usage

```
Moultrie(data, ...)
```

Arguments

`data`	data.drame. It contains at least three variables: five-year grouped ages, number of childs and Children Ever Born (CEB).
`...`	Arguments to be passed to `dplyr::select`, i.e., five-year grouped ages, number of childs and Children Ever Born.

Value

`Moultrie` returns a data.frame with corrected childs for each number of Children Ever Born and five-year grouped ages.

Author(s)

Cesar Gamboa-Sanabria

References

Moultrie TA, Dorrington RE, Hill AG, Hill K, Timæus IM, Zaba B (2013). *Tools for demographic estimation*. International Union for the Scientific Study of Population.

See Also

CEB El_Badry

Examples

```
CEB_data <- tidyr::gather(CEB, ages, childs, -Children_Ever_Born)
results <- Moultrie(CEB_data, ages, childs, Children_Ever_Born)
tidyr::pivot_wider(results, names_from=age, values_from=childs)
```

Myers *Myer's Blended Index*

Description

An upgrade over the Whipple index allows analyzing digit's attraction (or repulsion) from 0 to 9.

Usage

```
Myers(data, ...)
```

Arguments

data	data.drame. It contains at least two variables: specific ages and population.
...	Arguments to be passed to dplyr::select, i.e., age and population, respectively.

Value

Myers returns a list with two objects:

Mmat	a data.frame with specific digits index
MI	the Myer's Blend Index.

Author(s)

Cesar Gamboa-Sanabria

References

Shryock HS, Siegel JS, Larmon EA, of the Census USB (1980). *The Methods and Materials of Demography*, number v. 1 in The methods and materials of demography. Department of Commerce, Bureau of the Census. https://books.google.co.cr/books?id=8Oo6AQAAMAAJ.

Examples

```
results <- Myers(Panama1990, age, pop)
results$Mmat
results$MI
```

`netmigration_projection` *netmigration_projection*

Description

Forecasting net migration.

Usage

```
netmigration_projection(
  mortality_rates_path,
  TFR_path,
  total_population_path,
  WRA_path,
  omega_age,
  horizon,
  first_year_projection
)
```

Arguments

`mortality_rates_path`	character. Path to Mortality rates in a .txt file.
`TFR_path`	character. Path to Fertility rates in a .txt file.
`total_population_path`	character. Path to Populations in a .txt file.
`WRA_path`	character. Path to Women of Reproductive Age in a .txt file.
`omega_age`	numeric. Maximum age.
`horizon`	numeric. The forecast horizon.
`first_year_projection`	numeric. Year for the base population.

Value

`netmigration_projection` returns an object of class `fmforecast` with the forecast netmigration models and the components of `demography::forecast.fdmpr()`.

Author(s)

Cesar Gamboa-Sanabria

Examples

```
## Not run:

library(dplyr)

data(CR_mortality_rates_1950_2011)

#CR_mortality_rates_1950_2011 %>%
#write.table(.,
#file = "CR_mortality_rates_1950_2011.txt",
#sep = "\t",
#row.names = FALSE,
#col.names = TRUE,
#quote = FALSE)


data(CR_populations_1950_2011)

#CR_populations_1950_2011 %>%
#write.table(.,
#file = "CR_populations_1950_2011.txt",
#sep = "\t",
#row.names = FALSE,
#col.names = TRUE,
#quote = FALSE)

data(CR_fertility_rates_1950_2011)

#CR_fertility_rates_1950_2011 %>%
#write.table(.,
#file = "CR_fertility_rates_1950_2011.txt",
#sep = "\t",
#row.names = FALSE,
#col.names = TRUE,
#quote = FALSE)


data(CR_women_childbearing_age_1950_2011)

#CR_women_childbearing_age_1950_2011 %>%
#write.table(.,
#file = "CR_women_childbearing_age_1950_2011.txt",
#sep = "\t",
#row.names = FALSE,
#col.names = TRUE,
#quote = FALSE)
```

```
#result <- netmigration_projection(mortality_rates_path = "CR_mortality_rates_1950_2011.txt",
#total_population_path = "CR_populations_1950_2011.txt",
#TFR_path = "CR_fertility_rates_1950_2011.txt",
#WRA_path = "CR_women_childbearing_age_1950_2011.txt",
#omega_age = 115, first_year_projection = 2011, horizon = 2150)


## End(Not run)
```

`op.arima` *op.arima*

Description

Estimates the best predictive ARIMA model using overparameterization.

Usage

```
op.arima(
  arima_process = c(p = 1, d = 1, q = 1, P = 1, D = 1, Q = 1),
  seasonal_periodicity,
  time_serie,
  reg = NULL,
  horiz = 12,
  prop = 0.8,
  training_weight = 0.2,
  testing_weight = 0.8,
  parallelize = FALSE,
  clusters = detectCores(logical = FALSE),
  LAMBDA = NULL,
  ISP = 100,
  ...
)
```

Arguments

`arima_process`	numeric. The ARIMA(p,d,q)(P,D,Q) process.
`seasonal_periodicity`	numeric. The seasonal periodicity, 12 for monthly data.
`time_serie`	ts. The univariate time series object to estimate the models.
`reg`	Optionally, a vector or matrix of external regressors, which must have the same number of rows as time_serie.
`horiz`	numeric. The forecast horizon.
`prop`	numeric. Data proportion for training dataset.

`training_weight`
: numeric. Importance weight for the goodness of fit and precision measures in the training dataset.

`testing_weight`
: numeric. Importance weight for the goodness of fit and precision measures in the testing dataset.

`parallelize`
: logical. If TRUE, then use parallel processing.

`clusters`
: numeric. The number of clusters for the parallel process.

`LAMBDA`
: Optionally. See `forecast::Arima()` for details.

`ISP`
: numeric. Overparameterization indicator to filter the estimated models in the (0,100] interval.

`...`
: additional arguments to be passed to `forecast::Arima()`.

Value

`op.arima` returns an object of class `list` with the following components:

`arima_models`
: all models defined by the `arima_process` argument.

`final_measures`
: goodness of fit and precision measures for each model.

`bests`
: a sorted list with the best ARIMA models.

`best_model`
: a list of "Arima", see `forecast::Arima()`

Author(s)

Cesar Gamboa-Sanabria

References

Gamboa-Sanabria C (2022). *La Sobreparametrizacion en el ARIMA: una aplicacion a datos costarricenses*. Master's thesis, Universidad de Costa Rica.

Examples

```
op.arima(arima_process = c(2,1,2,2,1,2),
time_serie = AirPassengers,
seasonal_periodicity = 12, parallelize=FALSE)
```

Panama1990 *Panama1990*

Description

Panama census data in 1990 by specific ages.

Usage

```
data("Panama1990")
```

Format

A data frame with 100 observations on the following 2 variables.

`age` a character vector with specific ages

`pop` a numeric vector with population for each age

Source

https://ccp.ucr.ac.cr/

Examples

```
data(Panama1990)
summary(Panama1990)
```

popstudy *popstudy Package*

Description

Applied techniques to demographic and time series analysis.

Author(s)

Cesar Gamboa-Sanabria <info@cesargamboasanabria.com>

population_projection *population_projection*

Description

Forecasting population using the components method.

Usage

```
population_projection(...)
```

Arguments

...	required arguments for mortality_projection, TFR_projection and netmigration_projection.

Value

population_projection returns an object of class list with the following components:

mort	mortality projections from mortality_projection.
fert	fertility projections from TFR_projection.
mig	netmigration projections from netmigration_projection.
pop	the national projections by sex and year.

Author(s)

Cesar Gamboa-Sanabria

See Also

mortality_projection TFR_projection netmigration_projection

Examples

```
## Not run:

library(dplyr)

data(CR_mortality_rates_1950_2011)

#CR_mortality_rates_1950_2011 %>%
#write.table(.,
#file = "CR_mortality_rates_1950_2011.txt",
#sep = "\t",
#row.names = FALSE,
#col.names = TRUE,
```

```
#quote = FALSE)


data(CR_populations_1950_2011)

#CR_populations_1950_2011 %>%
#write.table(.,
#file = "CR_populations_1950_2011.txt",
#sep = "\t",
#row.names = FALSE,
#col.names = TRUE,
#quote = FALSE)

data(CR_fertility_rates_1950_2011)

#CR_fertility_rates_1950_2011 %>%
#write.table(.,
#file = "CR_fertility_rates_1950_2011.txt",
#sep = "\t",
#row.names = FALSE,
#col.names = TRUE,
#quote = FALSE)


data(CR_women_childbearing_age_1950_2011)

#CR_women_childbearing_age_1950_2011 %>%
#write.table(.,
#file = "CR_women_childbearing_age_1950_2011.txt",
#sep = "\t",
#row.names = FALSE,
#col.names = TRUE,
#quote = FALSE)

#result <- population_projection(mortality_rates_path = "CR_mortality_rates_1950_2011.txt",
#total_population_path = "CR_populations_1950_2011.txt",
#TFR_path = "CR_fertility_rates_1950_2011.txt",
#WRA_path = "CR_women_childbearing_age_1950_2011.txt",
#omega_age = 115, first_year_projection = 2011, horizon = 2020)


## End(Not run)
```

`project_structure` *project_structure*

Description

Create a basic structure for a project repo.

Usage

```
project_structure()
```

Value

`project_structure` does not return a value, it only creates basic diretories and files in the current working direcotory/repository.

Author(s)

Cesar Gamboa-Sanabria

Examples

```
## Not run:
project_structure()

## End(Not run)
```

---

read_from_dir *read_from_dir*

Description

Get full path from a file.

Usage

```
read_from_dir(file, path = NULL)
```

Arguments

`file`	The file name.
`path`	The file location.

Value

`read_from_dir` returns an object of class character with the normalizaed path for a file.

Author(s)

Cesar Gamboa-Sanabria

Examples

```
## Not run:
file.create("test_file.txt")
read_from_dir("test_file.txt")

## End(Not run)
```

`required_packages` *required_packages*

Description

Install/load the required packages from CRAN.

Usage

```
required_packages(...)
```

Arguments

`...`	packages names.

Value

`required_packages` does not return a value, it only install and load the desired packages.

Author(s)

Cesar Gamboa-Sanabria

Examples

```
## Not run:
#If you need to install and load the tidyr, dplyr and ggplot2 packages, run the following line:
#required_packages(tidyr, dplyr, ggplot2)

## End(Not run)
```

`Sprague` *Sprague multipliers*

Description

Method to open five-year grouped ages into specific ages.

Usage

```
Sprague(data, ...)
```

Arguments

`data`	data.drame. It contains at least two variables: five-year grouped ages and population.
`...`	Arguments to be passed to `dplyr::select`, i.e., age and population, respectively.

Value

`Sprague` returns an object of class data.frame with population for specific ages.

Author(s)

Cesar Gamboa-Sanabria

References

Shryock HS, Siegel JS, Larmon EA, of the Census USB (1980). *The Methods and Materials of Demography*, number v. 1 in The methods and materials of demography. Department of Commerce, Bureau of the Census. https://books.google.co.cr/books?id=8Oo6AQAAMAAJ.

See Also

`Beers`

Examples

```
Sprague(Ecuador1990, age, population)
```

TFR_projection *TFR_projection*

Description

Forecasting total fertility rates.

Usage

```
TFR_projection(TFR_path, WRA_path, horizon, first_year_projection, ...)
```

Arguments

TFR_path	character. Path to Fertility rates in a .txt file.
WRA_path	character. Path to Women of Reproductive Age in a .txt file.
horizon	numeric. The forecast horizon.
first_year_projection	numeric. Year for the base population.
...	additional arguments to be passed to forecast::Arima().

Value

TFR_projection returns an object of class fmforecast with the forecast fertility rates and the components of demography::forecast.fdm().

Author(s)

Cesar Gamboa-Sanabria

Examples

```
library(dplyr)

data(CR_fertility_rates_1950_2011)

#CR_fertility_rates_1950_2011 %>%
#write.table(.,
#file = "CR_fertility_rates_1950_2011.txt",
#sep = "\t",
#row.names = FALSE,
#col.names = TRUE,
#quote = FALSE)


data(CR_women_childbearing_age_1950_2011)
```

```
#CR_women_childbearing_age_1950_2011 %>%
#write.table(.,
#file = "CR_women_childbearing_age_1950_2011.txt",
#sep = "\t",
#row.names = FALSE,
#col.names = TRUE,
#quote = FALSE)

#result <- TFR_projection(TFR_path = "CR_fertility_rates_1950_2011.txt",
#WRA_path = "CR_women_childbearing_age_1950_2011.txt",
#omega_age = 115, first_year_projection = 2011, horizon = 2150)
```

Index