Package ‘vietnamdata’

May 26, 2017

Type Package

Title Data and Empirical Tools for Quantitative Social Science Research on Vietnam

Version 0.1.0

Author Minh D. Trinh [aut, cre]

Maintainer Minh D. Trinh <mdtrinh@mit.edu>

Description Provides several datasets that are useful for social researchers interested in Vietnam, and convenient functions for empirical analysis using these datasets. It includes data on Vietnam's provincial macroeconomics, budget cycles, demographics, and aggregate results from the PCI and PAPI surveys. To facilitate empirical analyis at the provincial level, the package provides tools to conduct randomization inference implementations of common statistical methods.

Depends R (>= 2.10)

Imports ggplot2, permute, Synth, wfe, vietnamcode, snowfall, stringdist, zoo, stringr

License CC0

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Suggests dplyr

URL https://github.com/mdtrinh/vietnamdata

BugReports https://github.com/mdtrinh/vietnamdata/issues

R topics documented:

`bestmatch` *Best string match using string distance*

Description

Find the best match for one character string in a vector of character strings using Levenshtein distance.

Usage

```
bestmatch(string, stringVector)
```

Arguments

`string`	elements to be approximately matched: will be coerced to `character` unless it is a list consisting of `integer` vectors. Identical to the `x` parameter in `stringdist::amatch()`
`stringVector`	a vector of strings to be used as lookup table for matching. Will be coerced to `character` unless it is a list consisting of `integer` vectors. Identical to the `table` parameter in `stringdist::amatch()`

Details

The function `bestmatch` is a quick wrapper around the `stringdist::amatch()` function using `method = "lv"` and `maxDist = 1`

Value

a vector of the same length as `string` containing the position of the closest match of each element of `string` in `stringVector`. Identical to the result returned by `stringdist::amatch()`.

`fill.blanks` *"Fill down the blank"*

Description

For a pre-sorted vector, fill each missing value with that of the preceding element.

Usage

```
fill.blanks(x)
```

Arguments

`x`	A pre-sorted vector that may contain `NA` values or empty strings.

Details

`fill.blanks()` is a wrapper around the `zoo::na.locf()` function in the package `zoo`. It can handle both missing values from all data types including character vectors where missing values are often encoded as empty strings.

Value

A vector of the same length and type as `x`, with all `NA` values or empty strings replaced by the value of the preceding element in the vector.

`find.numeric` *Identify numeric values from a character vector*

Description

Identify all numbers from a character vector (e.g. ID variables). Can return indices of these numbers, their values, or boolean indicators for whether each value is numeric.

Usage

```
find.numeric(x, response = c("index", "value", "boolean"))
```

Arguments

`x`	A character vector containing possible numeric values that need identified.
`response`	The type of values to be returned. One of "index" (default), "value", or "boolean".

Value

If `response == "index"`, the indices of all numeric values. If `response == "value"`, the values of these numeric values, with non-numeric values replaced by `NA`. If `response == "boolean"`, boolean indicators for whether each value in `x` is numeric or string.

genperms *Generate flexible permutation matrix for blocked, clustered or simpler designs*

Description

An alternative version of `ri::genperms`. Given user-input vectors of clusters or blocks, generate an exact permutation matrix, or a randomly sampled permutation matrix if the number of actual permutations is too high. Improves upon `ri::genperms` by allowing permutation of non-binary vectors.

Usage

```
genperms(x, block = NULL, clus = NULL, maxiter = 10000)
```

Arguments

`x`	a vector to be permuted. Can be continuous
`block`	a vector of equal length as `x`, with unique values indicating different blocks
`clus`	a vector of equal length as `x`, with unique values indicating different blocks
`maxiter`	maximum number of permutations to be included in the permutation matrix. If it is possible to perform exact permutation with fewer permutations, then the exact permutation matrix is produced.

Details

Unlike its counterpart in the `ri` package, this function can perform permutation of input vectors that are not binary. It also accepts as arguments for `blockvar` and `clustvar` other data types than `integer`. `genperms` is primarily based on the `permute` package.

Value

A permutation matrix where each row is a permutation of the input vector `x`

References

Gerber, Alan S. and Donald P. Green. 2012. Field Experiments: Design, Analysis, and Interpretation. New York: W.W. Norton.

`is.nan.data.frame` *Identify* `NaN` *in a dataframe.*

Description

Identify cells with `NaN` in a data frame. Improve on the defeault `is.nan()` function, which only works on vectors, by allowing data frames as input.

Usage

```
## S3 method for class 'data.frame'
is.nan(x)
```

Arguments

`x` A data frame to be tested.

Value

A matrix of the same dimension as `x`, with TRUE/FALSE values for whether each cell in the original data frame is a number or not. `NaN` means 'Not a Number'.

`neglog` *Negative-log transformation*

Description

An analogue to the log transformation for negative values.

Usage

```
neglog(x, base = exp(1), offset = 0.5)
```

Arguments

`x` a numeric or complex vector

`base` a positive or complex number: the base with respect to which logarithms are computed. Defaults to e =`exp(1)`

`offset` a number specifying an offset to avoid `neglog(x)` returning `-Inf`. Defaults to 0.5

Details

`neglog(x)` is calculated such that $neglog(x) = sign(x) \times \log|x + offset|$. The offset ensures that no `-Inf` value is returned. The default offset is 0.5.

Value

A vector of the same length as `x` containing the transformed values.

`papi` *etnam PAPI Data*

Description

A data frame containing results from the Public Administration Performance Index (PAPI) survey, aggregated at the province level for all Vietnamese provinces from 2005 to 2016. It contains 15 variables, which include province and year IDs, weighted and unweighted aggregate PAPI scores, and component PAPI scores.

Usage

```
papi
```

Format

An object of class `data.frame` with 378 rows and 32 columns.

Details

Raw data for `pci` is scraped from the website of the Vietnam Provincial Governance and Public Administrataion Performance Index Project at `papi.org.vn`

`pci` *Vietnam PCI Data*

Description

A data frame containing results from the Provincial Competitiveness Index (PCI) survey, aggregated at the province level for all Vietnamese provinces from 2005 to 2016. It contains 15 variables, which include province and year IDs, weighted and unweighted aggregate PCI scores, and component PCI scores.

Usage

```
pci
```

Format

An object of class `data.frame` with 738 rows and 15 columns.

Details

Raw data for `pci` is scraped from the website of the Provincial Competitiveness Index Project at `www.pcivietnam.org`

plot.riFit — *Plot Point Estimate and Randomization Distribution for Randomization Inference Result*

Description

A plot showing the point estimate for the treatment effect as a red line, and the randomization distribution of the treatment effect as gray bars.

Usage

```
## S3 method for class 'riFit'
plot(x, title = NULL, xlab = NULL, ylab = NULL,
  scale = F, xmin, xmax, axe.y = F, ...)
```

Arguments

`x` An object of the class `riFit` generated by `rireg` or `riwfe`.

`title` The main title (on top)

`xlab` X axis label

`ylab` Y axis label

`scale` logical; if TRUE the point estimates and the randomization distribution is demeaned and normalized before plotting (default is FALSE)

`xmin` a numeric indicating the minimum value of the plot's main axix

`xmax` a numeric indicating the maximum value of the plot's main axix

`axe.y` logical; if TRUE the y-axis is shown (default is FALSE)

`...` Other arguments passed on to `ggplot2::ggplot`. Currently unused

plot.riSynth — *Plot Point Estimate and Randomization Distribution for Randomization Inference Result from riSynth*

Description

A plot showing the point estimate for the treatment effect as a red line, and the randomization distribution of the treatment effect as gray bars.

Usage

```
## S3 method for class 'riSynth'
plot(x, title = NULL, xlab = NULL, ylab = NULL,
  scale = F, xmin, xmax, att = T, axe.y = F, ...)
```

Arguments

`x`	an `riSynth` object
`title`	The main title (on top)
`xlab`	X axis label
`ylab`	Y axis label
`scale`	logical; if TRUE the point estimates and the randomization distribution is demeaned and normalized before plotting (default is FALSE)
`xmin`	a numeric indicating the minimum value of the plot's main axix
`xmax`	a numeric indicating the maximum value of the plot's main axix
`att`	logical; currently unused
`axe.y`	logical; if TRUE the y-axis is shown (default is FALSE)
`...`	Other arguments passed on to `ggplot2::ggplot`. Currently unused

Details

`plot.riSynth` converts a `riSynth` object into a `riFit` object and then make a call to `plot.riFit`.

`rireg` *Randomization Inference on Treatment Effect using Linear Regression*

Description

Estimates of the treatment effect using linear models through `lm`, then conducts inference using randomization inference by permuting the treatment vector to obtain the sharp null distribution

Usage

```
rireg(data, outcome, treatment, covs, blockvar = NULL, clustvar = NULL,
  maxiter = 10000)
```

Arguments

`data`	a data frame containing the variables in the model
`outcome`	a character. Name of the outcome variable.
`treatment`	a character. Name of the treatment variable.
`covs`	a character vector. Names of the covariates to be used in the model.
`blockvar`	an optional character vector. Name of the block variable if the randomization inference procedure requires block randomization. The variable named by `blockvar` will be used as input for the `genperms` function.
`clustvar`	an optional character vector. Name of the cluster variable if the randomization inference procedure requires clustered randomization. The variable named by `clustvar` will be used as input for the `genperms` function.
`maxiter`	a positive integer. The maximum number of permutations to be included in the permutation matrix for the randomization distribution. Used as input for the `genperms` function.

Details

Estimates of the treatment effects are obtained by OLS regression. When covariates are included, the randomization distribution is obtained by permuting the "partialled-out" treatment vector i.e. the vector of residuals from a regression of treatment on covariates. Internally `rireg` makes call to `genperms`. The variable whose names are given by `blockvar` and `clustvar` will be coerced into input vectors for the `block` and `clus` arguments of the `genperms` function.

Value

An object of class `riFit`

`riSynth` *Randomization Inference on ATT using Synthetic Control*

Description

Estimates average treatment effect on the treated using the Synthetic Control Method, then conducts inference using randomization inference by permuting the treatment vector to obtain the sharp null distribution

Usage

```
riSynth(data, outcome, treatment, covs, treatment.year, pretreatment.year,
  posttreatment.year = NULL, unit.variable, unit.names.variable,
  time.variable, include.past.Y = TRUE, snowfall = FALSE, maxiter = 1000)
```

Arguments

`data` a data frame containing the variables in the model

`outcome` a character. Name of the outcome variable.

`treatment` a character. Name of the treatment variable.

`covs` a character vector. Names of the covariates to be used in the model.

`treatment.year` the year or time period at which treatment occurs

`pretreatment.year`
the years or time periods before treatment occurs. Observations from these years or time periods will be used to create the synthetic contorl

`posttreatment.year`
the years or time periods after treatment occurs. Observations form these years or time periods will be used to calculate treatment effects.

`unit.variable` A scalar identifying the column number or column-name character string associated unit numbers. The unit.varibale has to be numeric.

`unit.names.variable`
A scalar or column-name character string identifying the column with the names of the units. This variable has to be of mode character.

`time.variable` A scalar identifying column number or column-name character string associated with period (time) data. The time variable has to be numeric.

`include.past.Y` a logical; if TRUE past values of the outcomes are included as a covariate to produce the synthetic control

`snowfall` a logical; if TRUE the function performs parallel computing using the snowfall package

`maxiter` maximum number of permutations to be included in the permutation matrix. If it is possible to perform exact permutation with fewer permutations, then the exact permutation matrix is produced.

Details

Estimates of treatment effects are obtained by the Synthetic Control Method through the `Synthatt` function. The randomization distribution is obtained by permuting the treatment vector. Internally, `riSynth` makes call to `genperms`. It is not yet possible to perform block or clustered randomization. The arguments `treatment.year`, `pretreatment.year`, `posttreatment.year`, `unit.variable`, `unit.names.variable`, `time.variable`, `include.past.Y` are input directly into the call for `Synth::Synth`.

Internally, `riSynth` makes use of the `Synth` function in the `Synth` package to calculate estimates of the ATT and the `permute` function in the `permute` package to create the randomization distribution. For computers with multiple cores, `riSynth` can perform parallel computation to improve computation speed.

Value

An object of class "riSynth"

`riSynthToriFit` *Convert* `riSynth` *object to* `riFit`

Description

Convert a `riSynth` to a `riFit` object

Usage

```
riSynthToriFit(riSynth.obj)
```

Arguments

`riSynth.obj` An object of the class `riSynth` generated by `riSynth`.

Details

Useful for plotting. Converts all `tau.att` elements in the `riSynth` object to corresponding `beta` elements in `riFit` and drops all `tau.i` elements.

Value

a riFit object

riwfe *Randomization Inference on Treatment Effect using Weighted Fixed Effects Regression*

Description

Estimates of the treatment effect using Weighted Fixed Effects Regression through the wfe function in the wfe package, then conducts inference using randomization inference by permuting the treatment vector to obtain the sharp null distribution

Usage

```
riwfe(data, outcome, treatment, covs, blockvar = NULL, clustvar = NULL,
  maxiter = 1000, unit.index, time.index, method, qoi = "ate",
  estimator = NULL, unbiased.se = TRUE)
```

Arguments

data	a data frame containing the variables in the model
outcome	a character. Name of the outcome variable.
treatment	a character. Name of the treatment variable.
covs	a character vector. Names of the covariates to be used in the model.
blockvar	an optional character vector. Name of the block variable if the randomization inference procedure requires block randomization. The variable named by blockvar will be used as input for the genperms function.
clustvar	an optional character vector. Name of the cluster variable if the randomization inference procedure requires clustered randomization. The variable named by clustvar will be used as input for the genperms function.
maxiter	a positive integer. The maximum number of permutations to be included in the permutation matrix for the randomization distribution. Used as input for the genperms function.
unit.index	a character string indicating the name of unit variable used in the models. The index of unit should be factor.
time.index	a character string indicating the name of time variable used in the models. The index of time should be factor.
method	method for weighted fixed effects regression, either unit for unit fixed effects; time for time fixed effects. The default is unit.
qoi	one of "ate" or "att". The default is "ate".
estimator	an optional character string indicating the estimating method. One of "fd" or "did". The default is NULL.
unbiased.se	logical. If TRUE, bias-asjusted heteroskedasticity-robust standard errors are used. See Stock and Watson (2008). Should be used only for balanced panel. The default is FALSE.

Details

Estimates of the treatment effects are obtained by OLS regression. When covariates are included, the randomization distribution is obtained by permuting the outcome vector. This is equivalent to permuting the treatment vector and all associated covariates. Unlike `rireg`, the `riwfe` function does not make use of the partialling-out method because the function `wfe` from the `wfe` package that `riwfe` calls does not allow the treatment variable to be omitted. Internally, `riwfe` makes call to `genperms`. The variable whose names are given by `blockvar` and `clustvar` will be coerced into input vectors for the `block` and `clus` arguments of the `genperms` function. The arguments `unit.index`, `time.index`, `method`, `qoi`, `estimator`, `unbiased.se` are input directly into the call for `wfe`.

Value

An object of class `riFit`

`SynthATT` *Synthetic Control ATT*

Description

Estimates the average treatment effect on the treated using the Synthetic Control Method, then conducts inference using randomization inference by permuting the treatment vector to obtain the sharp null distribution

Usage

```
SynthATT(data, outcome, treatment, covs, treatment.year, pretreatment.year,
  posttreatment.year = NULL, unit.variable, unit.names.variable,
  time.variable, include.past.Y = TRUE, snowfall = FALSE)
```

Arguments

`data` a data frame containing the variables in the model

`outcome` a character. Name of the outcome variable.

`treatment` a character. Name of the treatment variable.

`covs` a character vector. Names of the covariates to be used in the model.

`treatment.year` the year or time period at which treatment occurs

`pretreatment.year` the years or time periods before treatment occurs. Observations from these years or time periods will be used to create the synthetic contorl

`posttreatment.year` the years or time periods after treatment occurs. Observations form these years or time periods will be used to calculate treatment effects.

`unit.variable` A scalar identifying the column number or column-name character string associated unit numbers. The unit.varibale has to be numeric.

`unit.names.variable`
A scalar or column-name character string identifying the column with the names of the units. This variable has to be of mode character.

`time.variable` A scalar identifying column number or column-name character string associated with period (time) data. The time variable has to be numeric.

`include.past.Y` a logical; if TRUE past values of the outcomes are included as a covariate to produce the synthetic control

`snowfall` a logical; if TRUE the function performs parallel computing using the snowfall package

Details

For each unit in the treatment group, `Synthatt` creates a synthetic control using weighted averages of the units in the control group, then estimates an unit-specific treatment effect. The Average Treatment Effect on the Treated is calculated by taking the average of all unit-specific treatment effects.

Internally, `Synthatt` makes use of the `synth` and `dataprep` functions in the `Synth` package. For computers with multiple cores, `Synthatt` can perform parallel computation to improve computation speed.

Value

An list with two objects: `tau.i`, a vector of unit-specific treatment effects, and `tau.att` the Average Treatment Effect on the Treated

`vietnamdata` *vietnamdata: Data and Empirical Tools for Quantitative Political Science Research on Vietnam*

Description

The `vietnamdata` package provides several datasets that are useful for political scientists interested in Vietnam, and convenient functions for empirical analysis using these datasets

Details

The package contains four key datasets: planned national budget breakdowns by provinces, realized national budget breakdowns by provinces, Provincial Competitiveness Index (PCI) aggregate and component scores by provinces, and Public Administration Performance Index (PAPI) aggregate and component scores by provinces.

It also provides convenient functions to perform Randomization Inference implementation for some common empirical methods such as Linear Regression, Weighted Fixed Effects Regression, and Synthetic Control. These functions are often helpful for finite-sample problems that often characterize provincial-level quantitative analyses of Vietnam.

Datasets

- VNbudget_plan is a data frame containing planned national budget broken down by provinces for all Vietnamese provinces from 2006 to 2016. Planned budgets are issued at the beginning of every year. It contains 46 variables, which include province and year IDs, total revenues, total expenditures, central transfers broken down by categories, as well as log transformations, lags and change values for the latter.
- VNbudget_final is a data frame containing realized national budget broken down by provinces for all Vietnamese provinces from 2006 to 2014. Realized budgets are calculated at the end of every year. It contains 38 variables, which include province and year IDs, total revenues, total expenditures, central transfers broken down by categories, as well as log transformations, lags and change values for the latter.
- pci is a data frame containing results from the Provincial Competitiveness Index (PCI) survey, aggregated at the province level for all Vietnamese provinces from 2005 to 2016. It contains 15 variables, which include province and year IDs, weighted and unweighted aggregate PCI scores, and component PCI scores.
- papi is a data frame containing results from the Public Administration Performance Index (PAPI) survey, aggregated at the province level for all Vietnamese provinces from 2005 to 2016. It contains 15 variables, which include province and year IDs, weighted and unweighted aggregate PAPI scores, and component PAPI scores.

Empirical tools

- rireg performs Randomization Inference for Ordinary Least Squares regressions
- riwfe performs Randomization Inference for Weighted Fixed Effects regressions
- riSynth performs Randomization Inference for the Synthetic Control method, with option for parallel computation
- plot.riFit and plot.riSynth produces simple visualization for results from rireg, riwfe, and riSynth
- genperms produces permutation matrices for an arbitrary vector with options for block and clustered randomization. Can permute non-binary vectors

Note

The functions in this package makes heavy use of existing functions from the wfe, Synth, and permute packages. All errors in implementation, howerver, are my own.

References

Abadie, Alberto, Alexis Diamond, and Jens Hainmueller. "Synth: An R package for synthetic control methods in comparative case studies." (2011).

Kim, In Song, Kosuke Imai, and Maintainer In Song Kim. "Package ‘wfe’." (2014).

Knaus, Jochen. "snowfall: Easier cluster computing (based on snow)." (2010).

Simpson, Gavin L. "Restricted permutations; using the permute package." (2012).

Wickham, Hadley. ggplot2: elegant graphics for data analysis. Springer, 2016.

`VNbudget_final` *Vietnam Provincial Budget Finals*

Description

A data frame containing realized national budget broken down by provinces for all Vietnamese provinces from 2006 to 2014. Realized budgets are calculated at the end of every year. It contains 38 variables, which include province and year IDs, total revenues, total expenditures, central transfers broken down by categories, as well as log transformations, lags and change values for the latter.

Usage

```
VNbudget_final
```

Format

An object of class `data.frame` with 569 rows and 37 columns.

Details

Raw data for `VNbudget_final` is scraped from the website of the Vietnamese Ministry of Finance at `www.mof.gov.vn`

`VNbudget_plan` *Vietnam Provincial Budget Plans*

Description

A data frame containing planned national budget broken down by provinces for all Vietnamese provinces from 2006 to 2016. Planned budgets are issued at the beginning of every year. It contains 46 variables, which include province and year IDs, total revenues, total expenditures, central transfers broken down by categories, as well as log transformations, lags and change values for the latter.

Usage

```
VNbudget_plan
```

Format

An object of class `data.frame` with 759 rows and 46 columns.

Details

Raw data for `VNbudget_plan` is scraped from the website of the Vietnamese Ministry of Finance at `www.mof.gov.vn`

`year` *Extract year from date*

Description

Extract the year from a date variable in multiple formats

Usage

```
year(date)
```

Arguments

`date`	a `character` object or vector indicating dates in one of three formats: exact date in d/m/Y, exact year without date, and date as number of days since 01/01/1900.

Details

`year` can handle dates in the following formats: date in d/m/Y, year without date, and date as number of days since 01/01/1900.

Value

The year of the date in `date`

Index