Package ‘gssquery’

March 13, 2020

Title Final Project for Stats 290

Version 1.0

Maintainer Jiao Lai <emmalai@stanford.edu>

Description This is a package for project of stat290. This package can be used to test the effect of US presidential elections on household happiness. Event study is exexuted for lead and lag effects. Placebo test is provided.

Depends R (>= 3.1.0)

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

Imports dplyr, tidyr, ggplot2, magrittr, rlang

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Jiao Lai [aut, cre]

R topics documented:

`Analysis.lm` *Linear regression of fixed effect model.*

Description

This function runs fixed effect model to analyze if a given independent variable affects the chosen outcome variable. Fixed effect(s) is included in the regression.

Usage

```
Analysis.lm(data, y, FE)
```

Arguments

`data`	data containing variables to be analyzed.
`y`	outcome variable.
`FE`	fixed effects to to included.

Value

the fit of lm

Examples

```
data(GSS)
LmData <- GetCleanLmData (GSS,'happy.score', 'election', 'year', 'age', 'income')
Analysis.lm(LmData,'happy.score','year')
```

`GetAttributes` *Get all attributes in given data.*

Description

This function returns all attributes in data

Usage

```
GetAttributes(data)
```

Arguments

`data`	given data

Value

The names of all attributes in data.

Examples

```
data(GSS)
GetAttributes(GSS)
```

`GetCleanLmData` *Get subset data containing certain attributes, which can be used for analyses such as linear regressions.*

Description

This function selects subset of data including outcome variable of interest, treatment of interest, fixed effects. Users can also select other control variables affecting the selected outcome variable.

Usage

```
GetCleanLmData(data, y, x, FE, common.na.strings, ...)
```

Arguments

`data` data containing variables to be analyzed, default is GSS.

`y` variable containing outcome variable in data, has to be numeric.

`x` variable containing treatment status in data, has to be indicator having two levels.

`FE` fixed effects to be included.

`common.na.strings`
values where to be replaced by NA

`...` control variables to be selected from data.

Value

Clean data with selected outcome, treatment, fixed effects, and control variables. The data can be used for analyses like linear regressions.

Examples

```
data(GSS)
GetCleanLmData(GSS,'happy.score','election','year',
c("DK", "IAP", "UNCODEABLE & IAP", ".a", ".b","DK NA"),'age','income')
```

`GSS` *Data on household happiness, income and other characteristics from ‘gss.norc.org‘.*

Description

This data is processed based on raw data downloaded from ‘gss.norc.org“, which can be found in folder ’data-raw’. Details of data preprocessing is included in ‘/data-raw/data_preprocessing.R‘. The dta consists of a named list of variables including ‘happy‘ and ‘year‘, which are the variables of interests. Other attributes such as ‘age‘, ‘income‘ are related control variables.

References

[gss.norc.org Data](http://gss.norc.org/get-the-data/stata) `gssquery` *A package which can perform fixed effect model and robustness test.*

Description

‘gssquery‘ is a package for final project of stats 290. This package can be used to perform fixed effect model and robustness tests. GSS data is included to run example of the effect of US presidential elections on household happiness. Robustness tests such as event study, leave one out falsification exercise, treatment re-shuffling are also provided.

Details

The package provides 9 functions: * [‘GetCleanLmData‘][GetCleanLmData()] which selects variables from data ‘GSS‘ and clean the data. * [‘Analysis.lm‘][Analysis.lm()] which runs linear regression of fixed effect model for given data. * [‘LagEffect‘][LagEffect()] which examines if there is any lag effect of the main results. * [‘LeadEffect‘][LeadEffect()] which examines if there is any lag effect of the main results. * [‘PlaceboLOT‘][PlaceboLOT()] which examines falsification exercise of leave one out to check the subset sensitivity of main results. * [‘GetAttributes‘][GetAttributes()] which returns all attributes in given data. * [‘gssquery‘][gssquery()] which returns attributes of user interest in a given data.. * [‘PlotAttributes‘][PlotAttributes()] which returns the bar plot or mean value by group of an attribute of user interest. * [‘PlaceboTSReshuffle‘][PlaceboTSReshuffle()] which tests if the main effect is sensitive to treatment re-shuffling. The source code of the package further demonstrates:

* [R markdown](https://rmarkdown.rstudio.com/) syntax in documenting functions, including references to functions and data within the same package and from other packages using the [‘roxygen2‘ package](https://cran.r-project.org/package=roxygen2/vignettes/rd-formatting.html) * Including a data set [‘GSS‘] as part of the package * Importing functions from other packages using ‘importFrom‘ directives * Use of [utils::globalVariables()] to get around warnings produced when checking the package, something that happens a lot when using [ggplot2::ggplot()], for instance.

See Also

[‘GetCleanLmData‘][GetCleanLmData()]

[‘Analysis.lm‘][Analysis.lm()]

[‘LagEffect‘][LagEffect()]

[‘LeadEffect‘][LeadEffect()]

[‘PlaceboLOT‘][PlaceboLOT()]

[‘GetAttributes‘][GetAttributes()]

[‘gssquery‘][gssquery()]

[‘PlotAttributes‘][PlotAttributes()]

[‘PlaceboTSReshuffle‘][PlaceboTSReshuffle()]

`LagEffect` *Event study: lag effect.*

Description

This functions examines if there is a lag effect on the relationship between x and y.

Usage

```
LagEffect(data, y, x, time, FE)
```

Arguments

`data`	data containing variables to be analyzed.
`y`	outcome variable.
`x`	treatment variable.
`time`	time fixed effect.
`FE`	fixed effects to to included.

Value

summary of lm fit on lag treatment.

Examples

```
data(GSS)
LmData <- GetCleanLmData (GSS,'happy.score', 'election', 'year', 'age', 'income')
LagEffect(LmData,'happy.score','election','year', 'year')
```

`LeadEffect` *Event study: lead effect.*

Description

This function examines if there is lead effect on the relationship between x and y.

Usage

```
LeadEffect(data, y, x, time, FE)
```

Arguments

`data`	data containing variables to be analyzed.
`y`	outcome variable.
`x`	treatment variable.
`time`	time fixed effect.
`FE`	fixed effects to to included.

Value

summary of regression on lead treatment.

Examples

```
data(GSS)
LmData <- GetCleanLmData (GSS,'happy.score', 'election', 'year', 'age', 'income')
LeadEffect(LmData,'happy.score','election','year', 'year')
```

`PlaceboLOU` *Placebo test: leave one out.*

Description

This function executes a leave one out falsification exercise on the effect of x on y. Specifically, observations can be seen as groups by ‘out‘. The total sample will be removed one group at a time, and then the main analysis is re-examined. This exercise tests if the main results are sensitive to certain subset of data.

Usage

```
PlaceboLOU(data, y, x, FE, out)
```

Arguments

`data`	data containing variables to be analyzed, default is GSS.
`y`	variable containing outcome variable in data.
`x`	variable containing treatment status in data, has to be indicator having two levels.
`FE`	fixed effects.
`out`	variable specifing groups.

Value

A data frame of coefficients from re-examined linear regression, and corresponding p value.

Examples

```
data(GSS)
LmData <- GetCleanLmData (GSS,'happy.score', 'election', 'year', 'age', 'income')
PlaceboLOU(LmData,'happy.score','election','year','year')
```

`PlaceboTSReshuffle` *Placebo test on effect of x on y.*

Description

This function executes a pacebo test on effect of x on y. Specifically, treatment status is randomly re-shuffled for 100 times, and the effect is re-examined on each re-shuffling. The coefficients from re-shuffled and original treatment status are ploted. If the effect is robust, then the point from original treatment status should be an outlier.

Usage

```
PlaceboTSReshuffle(data, y, x, FE)
```

Arguments

`data`	data containing variables to be analyzed, default is GSS.
`y`	variable containing outcome variable in data.
`x`	variable containing treatment status in data, has to be indicator having two levels.
`FE`	fixed effects.

Value

A plot of coefficients from re-shuffled and original treatment status.

Examples

```
data(GSS)
LmData <- GetCleanLmData (GSS,'happy.score', 'election', 'year', 'age', 'income')
PlaceboTSReshuffle(LmData,'happy.score','election','year')
```

`PlotAttribute` *Plot of specific attribute of data.*

Description

This function generate a bar-plot of an attribute with level <= 10, or a plot for mean value by group.

Usage

```
PlotAttribute(data, attr, group.by)
```

Arguments

`data`	data file
`attr`	an attribute from data
`group.by`	group by variable

Value

A plot of attribute of interest.

Examples

```
data(GSS)
PlotAttribute(GSS,'age','year')
```

QueryAttributes *Query specific attributes from data.*

Description

This function returns attributes of interest in a given data file.

Usage

```
QueryAttributes(data, ...)
```

Arguments

data	data file provided
...	attributes of interest

Value

attributes of interest in data.

Examples

```
data(GSS)
QueryAttributes(GSS,"age","income")
```

Index