Package ‘eiCompare’

June 16, 2017

Type Package
Title Compares EI, Goodman, RxC Estimates
Version 2.1
Date 2017-06-13
Author Loren Collingwood
Maintainer Loren Collingwood <loren.collingwood@ucr.edu>
Description Compares estimates from three ecological inference rou-
License GPL (>= 2)
Depends eiPack, methods, tmvtnorm, foreach, ggplot2, ei
Imports data.table, ucminf, R.utils, mvtnorm, msm, mnormt, cubature,
elipse, plotrix, plyr, tidyr, magrittr, stringr
NeedsCompilation no
Repository CRAN
Date/Publication 2017-06-16 17:59:02 UTC

R topics documented:

eiCompare-package .................................................. 2
bayes_table_make .................................................. 2
corona .............................................................. 4
cor_06 .............................................................. 5
ei.reg.bayes.conf.int .............................................. 6
ei_compare-class ................................................... 7
ei_est_gen .......................................................... 9
ei_rc_congruence .................................................. 11
ei_rc_good_table .................................................. 12
goodman_generalize ................................................. 14
lac_10 .............................................................. 15
plot.ei ............................................................. 16
plot.ei_compare ................................................... 18
summary.ei ........................................................ 20
**eiCompare-package**  
*Compares EI, Goodman, RxC Estimates*

**Description**  
Compares estimates from three ecological inferences routines, based on King et al.’s approach.

**Details**  
See demo(demo, "eiCompare") for examples on how to use code

**Author(s)**  
Loren Collingwood  
Maintainer: Loren Collingwood <loren.collingwood@ucr.edu>

**References**  

---

**bayes_table_make**  
*EI:RxC Bayes Table Make*

**Description**  
Creates data.frame() table from eiPack RxC output, in the same format as ei_est_gen

**Usage**  
```r
bayes_table_make(ei_bayes_object, cand_vector, table_names)
```

**Arguments**  
- `ei_bayes_object`  
  Output from eiPack ei.reg.bayes() function  
- `cand_vector`  
  Character vector of candidate name variables, usually "pct_johns" or something  
- `table_names`  
  Character vector of column names, e.g., c("RxC: Pct Hisp", "RxC: Pct Asian")

**Value**  
Data frame object in similar vein to ei_est_gen
bayes_table_make

Author(s)

Loren Collingwood <loren.collingwood@ucr.edu>

References


Examples

```r
# TOY DATA EXAMPLE
canda <- runif(5)
candb <- 1-canda
white <- runif(5)
black <- 1 - white
total <- round( runif(5, min=20, max=40), 0)

toy <- data.frame(canda, candb, white, black, total)

cands <- c("canda", "candb")
table_names <- c("RxC: PCT Black", "RxC PCT White")

# Generate formula for passage to ei.reg.bayes() function
form <- formula(cbind(canda,candb) ~ cbind(black, white))
# Run Bayesian model
suppressWarnings {
  ei_bayes <- ei.reg.bayes(form, data=toy, sample=100, truncate=TRUE)
}
# Table Creation, using function bayes_table_make
ei_bayes_res <- bayes_table_make(ei_bayes, cand_vector= cands, table_names = table_names)
print(ei_bayes_res)

# Load Package Data
data(corona)

# Create Character Vectors

cands <- c("pct_husted","pct_spiegel","pct_ruth","pct_button","pct_montanez","pct_fox")
table_names <- c("RxC: Pct Hisp", "RxC: Pct Asian", "RxC: Pct White")

# Generate formula for passage to ei.reg.bayes() function
form <- formula(cbind(pct_husted,pct_spiegel,pct_ruth,pct_button,pct_montanez,pct_fox) ~ cbind(pct_hisp, pct_asian, pct_white))
# Run Bayesian model
suppressWarnings {
  ei_bayes <- ei.reg.bayes(form, data=corona, sample=10000, truncate=TRUE)
}
# Table Creation, using function bayes_table_make
ei_bayes_res <- bayes_table_make(ei_bayes, cand_vector= cands, table_names = table_names)
print(ei_bayes_res)
```
Description

Data taken from a 2014 California election, with precinct results and racial demographics for Corona, CA precincts.

Usage

data("corona")

Format

A data frame with 46 observations on the following 12 variables.

precinct  a numeric vector
totvote  a numeric vector
pct_husted  a numeric vector
pct_spiegel  a numeric vector
pct_ruth  a numeric vector
pct_button  a numeric vector
pct_montanez  a numeric vector
pct_fox  a numeric vector
pct_hisp  a numeric vector
pct_asian  a numeric vector
pct_white  a numeric vector
pct_non_lat  a numeric vector

References

Riverside County, CA board of elections

Examples

data(corona)
head(corona)
str(corona)
Description

Precinct vote data from a Corona, CA 2006 election

Usage

data("cor_06")

Format

A data frame with 47 observations on the following 8 variables.

- precinct a numeric vector
- totvote a numeric vector
- pct_latino a numeric vector
- pct_other a numeric vector
- pct_breitenbucher a numeric vector
- pct_montanez a numeric vector
- pct_spiegel a numeric vector
- pct_skipworth a numeric vector

References

Riverside County, CA board of elections

Examples

data(cor_06)
# Look at data
head(cor_06)
str(cor_06)
ei.reg.bayes.conf.int  Creates EI Reg Bayes Tables

Description

Creates EI reg bayes tables with confidence bands

Usage

    ei.reg.bayes.conf.int(ei_bayes)

Arguments

    ei_bayes  Object result of call to ei.reg.bayes() function.

Value

    Matrix object, table of results

Author(s)

    Loren Collingwood <loren.collingwood@ucr.edu>

References

    eiPack, King et. al. (http://gking.harvard.edu/eiR)

Examples

```r
# TOY DATA EXAMPLE
canda <- c(.1, .09, .85, .9, .92)
candb <- 1 - canda
white <- c(.8, .9, .10, .08, .11)
black <- 1 - white
total <- c(30, 80, 70, 20, 29)
toy <- data.frame(canda, candb, white, black, total)

# CREATE VECTORS
cands <- c("canda")
race_group <- c("- black")  # only use one group for example
table_names <- c("EI: PCT Black", "EI: PCT White")

# RUN ei_est_gen()
# KEEP DATA TO JUST ONE ROW FOR EXAMPLE (time) ONLY!
results <- ei_est_gen(cands, race_group, "total",
                      data = toy[c(1,3,5),], table_names = table_names, sample=100)

# Generate formula for passage to ei.reg.bayes() function
```
form <- formula(cbind(canda, candb) ~ cbind(black, white))
# Run Bayesian model
suppressWarnings {
  ei_bayes <- ei.reg.bayes(form, data=toy, sample=100, truncate=TRUE)
}

# Produce Table
ei.reg.bayes.conf.int(ei_bayes)

# Warning: Takes a while to run
# Load corona data
data(corona)
# Generate character vectors
cands <- c("pct_husted", "pct_spiegel", "pct_ruth", "pct_button", "pct_montanez", "pct_fox")
race_group3 <- c("- pct_hisp", "- pct_asian", "- pct_white")
table_names <- c("EI: Pct Lat", "EI: Pct Asian", "EI: Pct White")
# Run EI iterative Fitting
results <- ei.est.gen(cand_vector=cands, race_group = race_group3,
total = "totvote", data = corona, table_names = table_names)

# EI: RxC model
# Generate formula
form <- formula(cbind(pct_husted, pct_spiegel, pct_ruth, pct_button, pct_montanez, pct_fox) ~ cbind(pct_hisp, pct_asian, pct_white))
suppressWarnings {
  ei_bayes <- ei.reg.bayes(form, data=corona, sample=10000, truncate=TRUE)
}
# Produce Table
ei.reg.bayes.conf.int(ei_bayes)

---

### ei_compare-class

**Class** "ei_compare"

**Description**

An S4 class object stemming from ei_rc_good_table(), used for plotting, and examining comparison results.

**Objects from the Class**

Objects can, in principle, be created by calls of the form `new("ei_compare", ...`). However, the preferred form is to have them called ei_rc_good_table()

**Slots**

- data: Object of class "data.frame"
- groups: Object of class "character"
Author(s)

Loren Collingwood <loren.collingwood@ucr.edu>

Examples

# TOY DATA EXAMPLE
canda <- c(0.1, 0.09, 0.85, 0.9, 0.92)
candb <- 1-canda
white <- c(0.8, 0.9, 0.1, 0.08, 0.11)
black <- 1 - white
total <- c(30, 80, 70, 20, 29)
toy <- data.frame(canda, candb, white, black, total)

# CREATE VECTORS
cands <- c("canda")
race_group <- c("- black") # only use one group for example
table_names <- c("EI: PCT Black", "EI: PCT White")

# RUN ei_est_gen()
# KEEP DATA TO JUST ONE ROW FOR EXAMPLE (time) ONLY!
results <- ei_est_gen(cands, race_group, "total",
data = toy[c(1,3,5),], table_names = table_names, sample=100)

# Generate formula for passage to ei.reg.bayes() function
form <- formula(cbind(canda,candb) ~ cbind(black, white))
# Run Bayesian model
suppressWarnings (  
ei_bayes <- ei.reg.bayes(form, data=toy, sample=100, truncate=TRUE)
)

table_names <- c("RxC: PCT Black", "RxC: PCT White")
cands <- c("canda", "candb")
ei_bayes_res <- bayes_table_make(ei_bayes, cand_vector=cands, table_names = table_names)
ei_bayes_res <- ei_bayes_res[c(1,2,5),]
# Combine Results, results in object of class ei_compare
ei_rc_combine <- ei_rc_good_table(results, ei_bayes_res,
groups= c("Black", "White")
)
# Produces data and character vector, which can be sent to plot()
ei_rc_combine

# Warning: Takes a while to run
# Load corona data
data(corona)
# Generate character vectors
cands <- c("pct_husted","pct_spiegel","pct_ruth","pct_button","pct_montanez","pct_fox")
race_group3 <- c("- pct_hisp", "- pct_asian", "- pct_white")
table_names <- c("EI: Pct Lat", "EI: Pct Asian", "EI: Pct White")
# Run EI iterative Fitting
results <- ei_est_gen(cand_vector=cands, race_group = race_group3,
total = "totvote", data = corona, table_names = table_names)
# EI: RxC model
# Generate formula
form <- formula(cbind(pct_husted, pct_spiegel, pct_ruth, pct_button, pct_montanez, pct_fox) ~ cbind(pct_hisp, pct_asian, pct_white))
ei_bayes <- ei.reg.bayes(form, data=corona, sample=10000, truncate=TRUE)

# RxC table names
table_names <- c("RxC: Pct Hisp", "RxC: Pct Asian", "RxC: Pct White")

# Table Creation, using function bayes_table_make in ei_est_generalize.R file
ei_bayes_res <- bayes_table_make(ei_bayes, cand_vector= cands, table_names = table_names)

# Combine Results, results in object of class ei_compare
ei_rc_combine <- ei_rc_good_table(results, ei_bayes_res, groups = c("Latino", "Asian", "White"))

# Produces data and character vector, which can be sent to plot()
ei_rc_combine

---

**ei_est_gen**

**Iterative EI Estimation**

**Description**

Iteratively fits EI models for candidates and racial/ethnic groups

**Usage**

\[
ei_est_gen(cand_vector, race_group, total, rho = 10, data, table_names, sample = 1000, tomog = F, density_plot = F, beta_yes=F,...)
\]

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cand_vector</td>
<td>Character vector of candidate names, taken from the dataset</td>
</tr>
<tr>
<td>race_group</td>
<td>Character vector of formula, e.g., &quot;~ pct_latino&quot;</td>
</tr>
<tr>
<td>total</td>
<td>Character vector (e.g., &quot;totvote&quot;) of total variable name from data, variable in data is numeric</td>
</tr>
<tr>
<td>rho</td>
<td>Rho parameter for ei() estimate, defaults to 10, numeric</td>
</tr>
<tr>
<td>data</td>
<td>data.frame() object containing the data</td>
</tr>
<tr>
<td>table_names</td>
<td>Character vector of table names with same length as race_group. Used for formatting output. If only one racial group, must provide &quot;Pct. Other&quot; as second element of vector</td>
</tr>
<tr>
<td>sample</td>
<td>Number of samples used for EI calculation, default = 1000</td>
</tr>
<tr>
<td>tomog</td>
<td>Logical to display tomography plot. If true will will save pdf plot to working directory. Default is FALSE</td>
</tr>
</tbody>
</table>
density_plot  Logical to display density plot of betab and betaw. If true will save pdf plot to working directory. Default is FALSE

beta_yes  Logical to export betas (b, w) in list object in addition to table of results. Default is FALSE

...  Arguments passed onto ei() function

Value

Data frame/table object containing EI individually estimated results. If beta_yes=T, two list items, first the data frame table of results, second dataframe of betas themselves.

Note

If this results in an error, "Error in .subset2(x, i, exact = exact) : invalid subscript type ‘list’", just rerun the algorithm again.

Author(s)

Loren Collingwood <loren.collingwood@ucr.edu>

References


Examples

# TOY DATA EXAMPLE
canda <- c(.1, .09, .85, .9, .92)
candb <- 1-canda
white <- c(.8, .9, .10, .08, .11)
black <- 1 - white
total <- c(30,80, 70, 20, 29)
toy <- data.frame(canda, candb, white, black, total)

# CREATE VECTORS
cands <- c("canda")
race_group <- c("~ black") # only use one group for example
table_names <- c("EI: PCT Black", "EI: PCT White")

# RUN ei_est_gen()
# KEEP DATA TO JUST ONE ROW FOR EXAMPLE (time) ONLY!
ei_est_gen(cands, race_group, "total",
         data = toy[c(1,3,5),], table_names = table_names, sample=100)

# WARNING -- May take a little while to execute
# Load Package Data
data(corona)
# Create Character Vectors
cands <- c("pct_husted","pct_spiegel","pct_ruth","pct_button","pct_montanez","pct_fox")
race_group3 <- c("- pct_hisp", "- pct_asian", "- pct_white")
table_names <- c("EI: Pct Hisp", "EI: Pct Asian", "EI: Pct White")

# Run ei_est_gen function
results <- ei_est_gen(cand_vector=cands, race_group = race_group3, total = "totvote", data = corona, table_names = table_names)

results
# Run ei_est_gen function; Exporting betas into data frame
results_w_betas <- ei_est_gen(cand_vector=cands, race_group = race_group3, total = "totvote", data = corona, table_names = table_names, beta_yes=T)

res1 <- results_w_betas[[1]]# table of mean estimates
res1
res2 <- results_w_betas[[2]]# betas of estimates for each precinct

---

**ei_rc_congruence**

**Congruence for 2x2**

**Description**

Calculates congruence scores between EI and RxC for the 2x2 Scenario

**Usage**

```r
ei_rc_congruence(ei_rc_table, cand_race, group_race)
```

**Arguments**

- **ei_rc_table** Object produced from `ei_rc_good_table()`, where `include_good=F`, of class `ei_compare`
- **cand_race** Numeric vector indicating race of the candidates in order they show up in table rownames, where 1=Latino; 2=Black; 3=Asian; 4=White/Non
- **group_race** Numeric vector, taking similar values as `cand_race` where 1=Latino; 2=Black; 3=Asian; 4=White/Non

**Value**

Table of congruence scores

**Author(s)**

Loren Collingwood <loren.collingwood@ucr.edu>, Matt Barreto <barretom@ucla.edu>
Examples

# LA County 2010 Insurance Commissioner Race
#ei_rc_combine <- ei_rc_good_table(results, ei_bayes_res,
#   groups = c("Latino", "Non Latino"))

load('system.file("extdata/congruence_table.RData", package=eiCompare')

ei_rc_congruence(ei_rc_combine2_2, c(1,4), c(1,4))

---

**ei_rc_good_table**  
*Create EI Comparison Table*

**Description**

Takes output from EI model, EI RxC model, Goodman regression, and puts them into a data frame table for useful analysis and comparison.

**Usage**

`ei_rc_good_table(ei, rc, good, groups, include_good = F)`

**Arguments**

- `ei`  
  Table/data frame object result from `ei_est_gen`. This assumes beta_yes=F in `ei_est_gen()`. See example below for beta_yes=T in `ei_est_gen()`.

- `rc`  
  Table/data frame from EI:RxC process from `bayes_table_make()`

- `good`  
  Table/data frame from Goodman regression, from `goodman_generalize()`. Default is nothing

- `groups`  
  Character vector of voting blocks (e.g., c("Latino", "White"))

- `include_good`  
  Logical, default is FALSE (F), Set to TRUE (T) if including a Goodman table/data object

**Value**

Object of class `ei_compare` containing a 1. `data.frame()` slot of comparisons across the three models; 2. Character vector of group names used for later plotting

**Note**

Most of the time the user will not include the Goodman table, as they are interested in the EI vs. EI:RxC comparison

**Author(s)**

Loren Collingwood <loren.collingwood@ucr.edu>
eiPack, King et. al. (http://gking.harvard.edu/eiR)

Examples

# TOY DATA EXAMPLE
canda <- c(.1, .09, .85, .9, .92)
candb <- 1-canda
white <- c(.8, .9, .10, .08, .11)
black <- 1 - white
total <- c(30,80, 70, 20, 29)
toy <- data.frame(canda, candb, white, black, total)

# CREATE VECTORS
cands <- c("canda")
race_group <- c("- black") # only use one group for example
table_names <- c("EI: PCT Black", "EI: PCT White")

# RUN ei_est_gen()
# KEEP DATA TO JUST ONE ROW FOR EXAMPLE (time) ONLY!
results <- ei_est_gen(cands, race_group, "total",
data = toy[c(1,3,5),], table_names = table_names, sample=100)

# Generate formula for passage to ei.reg.bayes() function
form <- formula(cbind(canda,candb) ~ cbind(black, white))
# Run Bayesian model
suppressWarnings {
  ei_bayes <- ei.reg.bayes(form, data=toy, sample=100, truncate=TRUE)
}

table_names <- c("RxC: PCT Black", "RxC: PCT White")
cands <- c("canda", "candb")
ei_bayes_res <- bayes_table_make(ei_bayes, cand_vector= cands, table_names = table_names)
ei_bayes_res <- ei_bayes_res[c(1,2,5),]
# Combine Results, results in object of class ei_compare
ei_rc_combine <- ei_rc_good_table(results, ei_bayes_res,
groups= c("Black", "White")
)
# Produces data and character vector, which can be sent to plot()
ei_rc_combine

# Warning: Takes a while to run
# Load corona data
data(corona)
# Generate character vectors
cands <- c("pct_husted","pct_spiegel","pct_ruth","pct_button","pct_montanez","pct_fox")
race_group3 <- c("~ pct_hisp", "~ pct_asian", "~ pct_white")
table_names <- c("EI: Pct Lat", "EI: Pct Asian", "EI: Pct White")
# Run EI iterative Fitting
goodman_generalize

Goodman Regression Generalization

Description

Makes summary table out of multiple heckman regression results, for multiple candidates and groups

Usage

goodman_generalize(cand_vector, race_group, total, data, table_names, ...)

Arguments

cand_vector  Character vector of candidate names, taken from the dataset
race_group   Character vector of formula, e.g., "~ pct_latino"
total       Character vector (e.g., "totvote") of total variable name from data, variable in data is numeric
data         data.frame() object containing the data
table_names  Character vector of table names with same length as race_group. Used for formatting output
...          Arguments passed onto lm() function
Value
 Object of class data.frame() returned containing table summary of all the Goodman regressions

Author(s)
Loren Collingwood <loren.collingwood@ucr.edu>

References

See Also
ei_rc_good_table

Examples
# Load corona data
data(corona)
# Generate character vectors
cands <- c("pct_husted","pct_spiegel","pct_ruth","pct_button","pct_montanez","pct_fox")
race_group3 <- c("~ pctl_hisp", "~ pctl_asian", "~ pctl_white")

# Goodman Regression
good_corona <- goodman_generalize(cands, race_group3, "totvote", corona, table_names)

lac_10  Los Angeles County Data

Description
Los Angeles County precinct dataset from 2010 election

Usage
data("lac_10")

Format
A data frame with 4980 observations on the following 10 variables.

precinct a character vector
tot_reg a numeric vector
i_jones a numeric vector
i_delatore a numeric vector
votescast a numeric vector
lat_voters a numeric vector
pct_latino a numeric vector
pct_delatorre a numeric vector
pct_jones a numeric vector
pct_other a numeric vector

Source
Los Angeles county

Examples

data(lac_10)
head(lac_10)
str(lac_10)

Description
'plot' method for the class 'ei'.

Usage

## S3 method for class 'ei'
## S3 method for class 'ei'
plot(x, ...)

Arguments

x An ei object from the function ei.
...
A list of options to return in graphs. See values below.

Details
Returns any of a set of possible graphical objects, mirroring those in the examples in King (1997). Graphical option lci is a logical value specifying the use of the Law of Conservation of Ink, where the implicit information in the data is represented through color gradients, i.e. the color of the line is a function of the length of the tomography line. This can be passed as an argument and is used for "tomogD" and "tomog" plots.
**Value**

- **tomogD**: Tomography plot with the data only. See Figure 5.1, page 81.
- **tomog**: Tomography plot with ML contours. See Figure 10.2, page 204.
- **tomogCI**: Tomography plot with 80% confidence intervals. Confidence intervals appear on the screen in red with the remainder of the tomography line in yellow. The confidence interval portion is also printed thicker than the rest of the line. See Figure 9.5, page 179.
- **tomogCI95**: Tomography plot with 95% confidence intervals. Confidence intervals appear on the screen in red with the remainder of the tomography line in yellow. The confidence interval portion is also printed thicker than the rest of the line. See Figure 9.5, page 179.
- **tomogE**: Tomography plot with estimated mean posterior $B_{ib}$ and $B_{iw}$ points.
- **tomogP**: Tomography plot with mean posterior contours.
- **betab**: Density estimate (i.e., a smooth version of a histogram) of point estimates of $B_{ib}$'s with whiskers.
- **betaw**: Density estimate (i.e., a smooth version of a histogram) of point estimates of $B_{iw}$'s with whiskers.
- **xt**: Basic $X_i$ by $T_i$ scatterplot.
- **xtc**: Basic $X_i$ by $T_i$ scatterplot with circles sized proportional to $N_i$.
- **xtfit**: $X_i$ by $T_i$ plot with estimated $E(T_i|X_i)$ and conditional 80% confidence intervals. See Figure 10.3, page 206.
- **xtfitg**: xfit with Goodman’s regression line superimposed.
- **estsim**: All the simulated $B_{ib}$’s by all the simulated $B_{iw}$’s. The simulations should take roughly the same shape of the mean posterior contours, except for those sampled from outlier tomography lines.
- **boundXb**: $X_i$ by the bounds on $B_{ib}$ (each precinct appears as one vertical line), see the lines in the left graph in Figure 13.2, page 238.
- **boundXw**: $X_i$ by the bounds on $B_{iw}$ (each precinct appears as one vertical line), see the lines in the right graph in Figure 13.2, page 238.
- **truth**: Compares truth to estimates at the district and precinct-level. Requires truth in the ei object. See Figures 10.4 (page 208) and 10.5 (page 210).
- **movieD**: For each observation, one tomography plot appears with the line for the particular observation darkened. After the graph for each observation appears, the user can choose to view the next observation (hit return), jump to a specific observation number (type in the number and hit return), or stop (hit “s” and return).
- **movie**: For each observation, one page of graphics appears with the posterior distribution of $B_{ib}$ and $B_{iw}$ and a plot of the simulated values of $B_{ib}$ and $B_{iw}$ from the tomography line. The user can choose to view the next observation (hit return), jump to a specific observation number (type in the number and hit return), or stop (hit “s” and return).

**Note**

Used within the ei_est_gen()
Author(s)

Gary King «email: king@harvard.edu» and Molly Roberts «email: roberts8@fas.harvard.edu»

References


Examples

```r
# TOY DATA EXAMPLE
canda <- c(.1, .09, .85, .9, .92)
candb <- 1-canda
white <- c(.8, .9, .10, .08, .11)
black <- 1-white
total <- c(30,80, 70, 20, 29)
toy <- data.frame(canda, candb, white, black, total)

# CREATE VECTORS
cands <- c("canda")
race_group <- c("- black") # only use one group for example
table_names <- c("EI: PCT Black", "EI: PCT White")

# RUN ei_est_gen()
# KEEP DATA TO JUST ONE ROW FOR EXAMPLE (time) ONLY!
results <- ei_est_gen(cands, race_group, "total",
                      data = toy[1,5,], table_names = table_names,
                      sample=100, tomog=TRUE)

# Delete Pdf files produced
getwd() # Show working directory
list.files() # show all files in directory
# To remove pdf files, make sure no other pdf files in directory
# system ("rm *.pdf ")
```

plot.ei_compare  

Plot Method for class ei_compare

Description

Allows quick plotting, using plot() of EI vs EI:RxC differences. Produces ggplot2 output, amazing.

Usage

```r
## S3 method for class 'ei_compare'
plot(x, ...)
```
Arguments

x Object of class ei_compare, from the ei_rc_good_table() function

... Arguments passed onto plot() and par()

Details

Limited amount of plotting flexibility. If user wants more flexibility extract relevant objects from ei_rc_good_table() output and do your own plotting!

Value

ggplot2 graph output of EI and RxC differences

Author(s)

Loren Collingwood <loren.collingwood@ucr.edu>

References

eiPack, King et. al. (http://gking.harvard.edu/eiR)

Examples

# TOY DATA EXAMPLE
canda <- c(.1, .09, .85,.9, .92)
candb <- 1-canda
white <- c(.8, .9, .10, .08, .11)
black <- 1 - white
total <- c(30,80, 70, 20, 29)
toy <- data.frame(canda, candb, white, black, total)

# CREATE VECTORS
cands <- c("canda")
race_group <- c("~ black") # only use one group for example
table_names <- c("EI: PCT Black", "EI: PCT White")

# RUN ei_est_gen()
# KEEP DATA TO JUST ONE ROW FOR EXAMPLE (time) ONLY!
results <- ei_est_gen(cands, race_group, "total",
data = toy[1,], table_names = table_names, sample=100)

# Generate formula for passage to ei.reg.bayes() function
form <- formula(cbind(canda,candb) ~ cbind(black, white))
# Run Bayesian model
suppressWarnings (ei_bayes <- ei.reg.bayes(form, data=toy, sample=100, truncate=TRUE))

table_names <- c("RxC: PCT Black", "RxC: PCT White")
cands <- c("canda", "candb")
ei_bayes_res <- bayes_table_make(ei_bayes, cand_vector = cands, table_names = table_names)
ei_bayes_res <- ei_bayes_res[c(1,2,5),]

# Combine Results, results in object of class ei_compare
ei_rc_combine <- ei_rc_good_table(results, ei_bayes_res,
                                   groups = c("Black", "White")
)

# Produces data and character vector, which can be sent to plot()
ei_rc_combine

# PLOT EI DIFFERENCES
plot(ei_rc_combine)

# Warning: Takes a while to run
# Load corona data
data(corona)
# Generate character vectors
cands <- c("pct_husted", "pct_spiegel", "pct_ruth", "pct_button", "pct_montanez", "pct_fox")
race_group3 <- c("- pct_hisp", "- pct_asian", "- pct_white")
table_names <- c("EI: Pct Lat", "EI: Pct Asian", "EI: Pct White")

# Run EI iterative Fitting
results <- ei_est_gen(cand_vector = cands, race_group = race_group3,
                      total = "totvote", data = corona, table_names = table_names)

# EI: RxC model
# Generate formula
form <- formula(cbind(pct_husted,pct_spiegel,pct_ruth,pct_button,pct_montanez,pct_fox) ~ cbind(pct_hisp, pct_asian, pct_white))
suppressWarnings(
    ei_bayes <- ei.reg.bayes(form, data=corona, sample=10000, truncate=TRUE)
)

# RxC table names
table_names <- c("RxC: Pct Hisp", "RxC: Pct Asian", "RxC: Pct White")

# Table Creation, using function bayes_table_make in ei_est_generalize.R file
ei_bayes_res <- bayes_table_make(ei_bayes, cand_vector = cands, table_names = table_names)

# Goodman Regression
good_corona <- goodman_generalize(cands, race_group3, "totvote", corona, table_names)

# Combine Results, results in object of class ei_compare
ei_rc_g_combine <- ei_rc_good_table(results, ei_bayes_res, good_corona,
                                     groups = c("Latino", "Asian", "White")
)

# Plot the Results
plot(ei_rc_g_combine)
**summary.ei**

**Description**

‘summary’ method for the class ‘ei’.

**Usage**

```r
## S3 method for class 'ei'
## S3 method for class 'ei'
summary(object, ...)  
```

**Arguments**

- `object`  
  An ei object from the function `ei`.

- `...`  
  A list of options to return in graphs. See values below.

**Note**

Used within `ei_est_gen()` function

**Author(s)**

Gary King «email: king@harvard.edu» and Molly Roberts «email: roberts8@fas.harvard.edu»

**References**


**Examples**

```r
# TOY DATA EXAMPLE  
canda <- c(.1, .09, .85, .9, .92)
candb <- 1-canda
white <- c(.8, .9, .10, .08, .11)
black <- 1 - white
total <- c(30,80, 70, 20, 29)
toy <- data.frame(canda, candb, white, black, total)

# CREATE VECTORS  
cands <- c("canda")
race_group <- c("black")  # only use one group for example  
table_names <- c("EI: PCT Black", "EI: PCT White")

# RUN ei_est_gen()  
# KEEP DATA TO JUST ONE ROW FOR EXAMPLE (time) ONLY!  
results <- ei_est_gen(cands, race_group, "total",  
                      data = toy[c(1,3,5),], table_names = table_names, sample=100)
results
```
Index

*Topic classes
  ei_compare-class, 7
*Topic datasets
  cor_06, 5
  corona, 4
  lac_10, 15
*Topic package
  eiCompare-package, 2

bayes_table_make, 2

cor_06, 5
corona, 4

ei.reg.bayes.conf.int, 6
ei_compare-class, 7
ei_est_gen, 9
ei_rc_congruence, 11
ei_rc_good_table, 12, 15
eiCompare(eiCompare-package), 2
eiCompare-package, 2

goodman_generalize, 14

lac_10, 15

plot.ei, 16
plot.ei_compare, 18

summary.ei, 20