Package ‘eiCompare’

November 14, 2018

Type Package

Title Compares EI, Goodman, RxC Estimates

Version 2.3

Date 2018-11-13

Author Loren Collingwood

Maintainer Loren Collingwood <loren.collingwood@ucr.edu>

Description Compares estimates from three ecological inference routines, based on King (1997) <ISBN: 0691012407>, <http://gking.harvard.edu/eicamera/kinroot.html>; King et. al. (2004
abs.shtml>.

License GPL (>= 2)

Depends eiPack, methods, tmvtnorm, foreach, ggplot2, ei, RCurl, RJSONIO, sjmisc, wru

Imports data.table, ucminf, R.utils, mvtnorm, msm, mnormt, cubature, ellipse, plotrix, plyr, tidyr, magrittr, stringr

NeedsCompilation no

R topics documented:

`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

Gary King (1997). A Solution to the Ecological Inference Problem. Princeton: Princeton University Press. Lau, Olivia, Ryan Moore, and Michael Kellerman. eiPack: Ecological Inference and Higher-Dimension Data Management

`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

```
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

Author(s)

Loren Collingwood <loren.collingwood@ucr.edu>

References

O. Lau, R. T. Moore, and M. Kellermann. eipack: RxC ecological inference and higher-dimension data management. New Functions for Multivariate Analysis, 18(1):43, 2006.

Examples

```
# 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)
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)
ei_bayes_res
```

bisg_prep_race_predict

Bayesian Improved Surname Geocoding Data Preparation and Race Prediction

Description

Preps data for BISG estimation via WRU package

Usage

```
bisg_prep_race_predict(df, voterid=NULL, precinct=NULL, surname_char, state,
                                   census.geo="block", census.key, census.data,
                                   census.surname = TRUE, surname.only = FALSE,
                                  surname.year = 2010, age = FALSE, sex = FALSE,
                                   party, retry = 0)
```

Arguments

`df` data.frame() object, containing voter file information with FIPS code information extracted using unit_comb_extract() function, which brings in relevant columns

`voterid` String of voterid column name. Default = NULL. Probably want this.

`precinct` String of precinct column name. Default = NULL. Probably want this.

`surname_char` String of surname_char column name. This column is passed to wru package.

`state` String of two digit state abbreviation, e.g., "NY", or "CA".

`census.geo` String of geographic level used for BISG. Default is Census "block". An optional character vector specifying what level of geography to use to merge in U.S. Census 2010 geographic data. Currently "county", "tract", "block", and "place" are supported. Note: sufficient information must be in user-defined voter.file object. If census.geo = "county", then voter.file must have column named county. If census.geo = "tract", then voter.file must have columns named county and tract. And if census.geo = "block", then voter.file must have columns named county, tract, and block. If census.geo = "place", then voter.file must have column named place. Specifying census.geo will call census_helper function to merge Census geographic data at specified level of geography.

`census.key` A character object specifying user's Census API key. Required if census.geo is specified, because a valid Census API key is required to download Census geographic data.

`census.data` A list indexed by two-letter state abbreviations, which contains pre-saved Census geographic data. Can be generated using get_census_data function from wru package.

`census.surname` A TRUE/FALSE object. If TRUE, function will call merge_surnames to merge in Pr(Race | Surname) from U.S. Census Surname List (2000 or 2010) and Spanish Surname List. If FALSE, voter.file object must contain additional fields specifying Pr(Race | Surname), named as follows: p_whi for Whites, p_bla for Blacks, p_his for Hispanics/Latinos, p_asi for Asians, and/or p_oth for Other. Default is TRUE.

`surname.only` A TRUE/FALSE object. If TRUE, race predictions will only use surname data and calculate Pr(Race | Surnname). Default is FALSE.

`surname.year` A number to specify the year of the census surname statistics. These surname statistics is stored in the data, and will be automatically loaded. The default value is 2010, which means the surname statistics from the 2010 census will be used. Currently, the other available choice is 2000.

`age` An optional TRUE/FALSE object specifying whether to condition race predictions on age (in addition to surname and geolocation). Default is FALSE. Must be same as age in census.data object. May only be set to TRUE if census.geo option is specified. If TRUE, voter.file should include a numerical variable age.

`sex` optional TRUE/FALSE object specifying whether to condition race predictions on sex (in addition to surname and geolocation). Default is FALSE. Must be same as sex in census.data object. May only be set to TRUE if census.geo option is specified. If TRUE, voter.file should include a numerical variable sex, where sex is coded as 0 for males and 1 for females.

`party` An optional character object specifying party registration field in voter.file, e.g., party = "PartyReg". If specified, race/ethnicity predictions will be conditioned on individual's party registration (in addition to geolocation). Whatever the name of the party registration field in voter.file, it should be coded as 1 for Democrat, 2 for Republican, and 0 for Other.

`retry` The number of retries at the census website if network interruption occurs. Default = 0.

Value

List object of of two data.frames. List item 1 is initial voter file data.frame. List object 2 (named bisg) includes race predictions.

Author(s)

Loren Collingwood <loren.collingwood@ucr.edu>

References

wru R package. Kabir Khanna, Kosuke Imai, Hubert Jin. Imai and Khanna (2015) "Improving Ecological Inference by Predicting Individual Ethnicity from Voter Registration Records" <DOI:10.1093/pan/mpw001>

Examples

```
# EXAMPLE: NOT RUN #
# NOTE: You need to have a census key to run this #

data(ny_voter)
head(ny_voter)

# Load ny_census object #
# load ( system.file("extdata/ny_census.RData",package="eiCompare") )

# Now load your Census key #
# [add in here where you would do that]
# key_census = ""
```

```
####################################################
# Prep Data & Perform Bayesian Surname Geolocation #
####################################################
# Extract second list object (called bisg, note $bisg)
#bisg <- bisg_prep_race_predict(df=ny_voter,
#                     voterid = "Voter.ID",
#                     precinct = "SD..Poll",
#                     surname_char = "Last.Name",
#                     state = "NY",
#                     census.key = key_census,
#                     census.data = ny_census)$bisg
```

corona *Corona Precinct Racial Bloc Voting Data*

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)
```

cor_06 *Corona 2006*

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

```
# 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(.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
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

`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
`rho`	Rho parameter for ei() estimate, defaults to 10, 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. If only one racial group, must provide "Pct. Other" as second element of vector
`sample`	Number of samples used for EI calculation, default = 1000
`tomog`	Logical to display tomography plot. If true will will save pdf plot to working directory. Default is FALSE
`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

eiPack. Gary King (1997). A Solution to the Ecological Inference Problem. Princeton: Princeton University Press.

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

```
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>

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[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))
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)


# 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")
)
ei_rc_combine

# If set beta_yes=T in ei_est_gen():
#ei_rc_combine2 <- ei_rc_good_table(results[[1]], ei_bayes_res, groups= c("Black", "White"))
```

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

eiPack King et. al. (http://gking.harvard.edu/eiR) L. A. Goodman. Ecological regressions and behavior of individuals. American Sociological Review, 1953.

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("~ pct_hisp", "~ pct_asian", "~ pct_white")

# Goodman Regression
table_names <- c("Good: Pct Lat", "Good: Pct Asian", "Good: Pct Wht")
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)
```

---

latlong2fips *Latitude-Longitude Coordinates to FIPS Geocode*

Description

Converts latitude/longitude coordinates to 15-digit FIPS code. Communicates with FCC API.

Usage

```
latlong2fips(latitude, longitude, number)
```

Arguments

latitude	Numeric. Latitude coordinate.
longitude	Numeric. Longitude coordinate.
number	Numeric. Usually part of a loop index counter

Value

Character string 15-digit FIPS code corresponding to Lat/Long entry

Author(s)

Loren Collingwood <loren.collingwood@ucr.edu>

References

https://geo.fcc.gov/api/census/block/

Examples

```
# EXAMPLE: NOT RUN #
#census_block <- list()
#num_catch <- rep(NA, nrow(nom_geo))

#for (i in 1:nrow(nom_geo)) {

#  census_block[[i]] <- latlong2fips(nom_geo$lat[i], nom_geo$lon[i], i)
#}

# Row Bind the list into a data.frame object #
#fips_df <- rbindlist(census_block)
```

`mbd_two` *Multinomial Dirichlet Bayes Draw Two Candidates*

Description

Extract posterior means and credible intervals. Need to label candidate vote variables: V1, V2, when two=F, add V3; Hispanic = VtdHVap_cor, White = VtdAVap_cor, Black = VtdBVap_cor

Usage

```
mbd_two(md, colnames, two=TRUE)
```

Arguments

`md`	object from ei.MD.bayes() return
`colnames`	Vector of candidate names. Stick to c(V1,V2) or c(V1,V2,V3)
`two`	Logical. Two candidates (TRUE), or three (FALSE)

Value

List with two data frames

Author(s)

Loren Collingwood <loren.collingwood@ucr.edu>

Examples

```
# EXAMPLE: NOT RUN #
```

`mbd_two_minority` *Multinomial Dirichlet Bayes Draw Two Candidates, and White/Minority voters*

Description

Extract posterior means and credible intervals. Need to call variables V1, V2. When two=F, add in V3; Race names = VtdAVap_cor, VtdMVap_cor

Usage

```
mbd_two_minority(md, colnames, two=TRUE)
```

Arguments

`md` object from ei.MD.bayes() return

`colnames` Vector of candidate names. Stick to c(V1,V2) or c(V1,V2,V3)

`two` Logical. Two candidates (TRUE), or three (FALSE)

Value

List with two data frames

Author(s)

Loren Collingwood <loren.collingwood@ucr.edu>

Examples

```
# EXAMPLE: NOT RUN #
```

`md_bayes_draw` *MD Bayes Draw*

Description

Tunes and estimates MD Bayes algorithm (ei.MD.bayes)

Usage

```
md_bayes_draw(dat, race_vote_split, form,
              ntunes=10, totaldraws = 100000, seed=12345,
              sample = 100000, thin = 100, burnin = 100000,
              ret.mcmc=TRUE)
```

Arguments

`dat` data.frame() object of just raw candidate vote and raw population counts. Put vote results in first set of columns, put population counts next

`race_vote_split` Numeric vector of length 2 indicating where vote column ends (e.g., 3), and population counts begin (e.g., 4): c(3,4)

`form` Formula object, e.g.: cbind(V1, V2, novote) ~ cbind(VtdAVap_cor, VtdBVap_cor, VtdHVap_cor, VtdOVap_cor)

`ntunes` Numeric; how much to tune tuneMD. Default = 10

`totaldraws` Numeric; How many total draws from MD. Default = 100000

`seed` Numeric. Default = 12345

`sample` Numeric. Default = 100000

`thin` Numeric. Default = 10

`burnin` Numeric. Default = 100000

`ret.mcmc` Logical. Default = TRUE

Value

Matrix object, of simulation reults

Author(s)

Loren Collingwood <loren.collingwood@ucr.edu>

References

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

Examples

```
# TOY DATA EXAMPLE
canda <- c(10,8, 10, 4, 8)
candb <- 20-canda
white <- c(15, 12, 18, 6, 10)
black <- 20 - white
toy <- data.frame(canda, candb, white, black)

# Generate formula for passage to ei.reg.bayes() function
form <- formula(cbind(canda,candb) ~ cbind(black, white))
# Then excute md_bayes_draw(); not run here due to time
# md_bayes_draw(toy, c(2,3), form )
```

`mean_and_ci` *mean_and_ci*

Description

Internal

Usage

```
mean_and_ci(cbind_dat, ci=c(.025, .975))
```

Arguments

`cbind_dat`	cbind object
`ci`	Credible intervals. Default: c(.025, .975)

Value

Mean and credible interval

Author(s)

Loren Collingwood <loren.collingwood@ucr.edu>

Examples

```
    # EXAMPLE: NOT RUN #
```

---

ny_fips *New York state FIPS codes*

Description

New York state FIPS codes example of 500 voters

Usage

```
data("ny_fips")
```

Format

A data frame with 500 observations on the following 2 variables.

`row_id` unique id

`FIP` 15 digit FIPS code including state, county, tract, block

Source

FCC

Examples

```
data(ny_fips)
str(ny_fips)
```

---

ny_voter *New York voter file sample*

Description

New York voter file sample; example of 500 voters

Usage

```
data("ny_voter")
```

Format

A data frame with 500 observations on the following 10 variables.

`Voter.ID` Unique voter file id number, jumbled

`SD..Poll` Precinct id

`fips` 15-digit fips code

`st` state fips code

`county` county fips code

`tract` tract fips code

`block` block fips code

`st_cty` state county fips code

`st_cty_tract` state county tract fips code

`Last.Name` Voters' surname

Source

ERCSD board of election

Examples

```
data(ny_voter)
str(ny_voter)
```

---

plot.ei *Plotting Ecological Inference Estimates*

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` xtfit with Goodman's regression line superimposed.

`estsims` 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

Gary King (1997). A Solution to the Ecological Inference Problem. Princeton: Princeton University Press.

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, 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 ouput, amazing.

Usage

```
## 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[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

# 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
table_names <- c("Good: Pct Lat", "Good: Pct Asian", "Good: Pct Wht")
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)
```

precinct_agg_combine *precinct_agg_combine*

Description

Combines voter file surname geocoded data into aggregated precinct counts. To be performed on data object resulting from successful completion of bisg_prep_race_predict() function.

Usage

```
precinct_agg_combine(dat, precinct = "precinct")
```

Arguments

`dat`	data.frame() object
`precinct`	Character of precinct column name. Default = "precinct", which is what is output from bisg_prep_race_predict()

Value

Aggregated dataset of nrow() precinct size, including racial size precinct estimates. Dataset suitable for EI/RxC.

Author(s)

Loren Collingwood <loren.collingwood@ucr.edu>

`race_check_2_3` *race_check_2_3*

Description

Checks that both sides of the RxC equation for White/Minority and White, Black, Hispanic, Other, respectively, add up to the same values. If small rounding issues, adjusts the "other" race category.

Usage

```
race_check_2_3(dat, split=c(3,4), catch=FALSE, catch_col=NULL,
                      print_sides=TRUE)
```

Arguments

`dat`	data.frame() object. One no vote/third party vote column, with candidate votes (for either 2 or 3 candidates), then up to four demographics with last as other
`split`	Numeric vector of length 2. Default is c(3,4), for two candidates and one catch-all. c(4,5) for three candidates and one catch all.
`catch`	Logical (TRUE/FALSE). Catch negative values. Default is FALSE
`catch_col`	Column names to be catched.
`print_sides`	Logical (TRUE/FALSE). Print out evaluations. Default is TRUE

Value

Dataset of Left side (Votes) vs. Right side (Demographics). diff column can be tagged on to exiting 'other' category to expedite data preparation process.

Author(s)

Loren Collingwood <loren.collingwood@ucr.edu>

Examples

```
  # EXAMPLE: NOT RUN #
```

summary.ei *Summarize Ecological Inference Estimates*

Description

‘summary’ method for the class ‘ei’.

Usage

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

Arguments

object	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

Gary King (1997). A Solution to the Ecological Inference Problem. Princeton: Princeton University Press.

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)
results
```

`unit_comb_extract` *Extract relevant geographic units from 15-digit FIPS code*

Description

Extract relevant geographic units from 15-digit FIPS code

Usage

```
unit_comb_extract(fips)
```

Arguments

`fips`	Character vector. 15-digit FIPS code

Value

List with separate datasets of length fips, including values for fips, state, county, tract, block, state-county, and state-county-tract.

Author(s)

Loren Collingwood <loren.collingwood@ucr.edu>

Index