

EI: A(n R) Program for Ecological Inference

Gary King¹
Margaret Roberts²

December 13, 2011

¹Albert J. Weatherhead III University Professor, Harvard University (Institute for Quantitative Social Sciences, 1737 Cambridge Street, Harvard University, Cambridge MA 02138; <http://GKing.Harvard.Edu>, King@Harvard.Edu, 617-500-7570.

²Ph.D. Candidate, Department of Government, 1737 Cambridge Street, Harvard University, Cambridge MA 02138

Contents

1	Introduction: Ecological Inference	2
2	Overview: R-commands	2
3	User's Guide: An Example	4
3.1	The Basic EI Algorithm	4
3.2	Extracting Quantities of Interest	6
3.3	Plotting in EI	6
4	Plotting RxC Information: eiRxCplot	11
5	Reference to EI Functions	13
5.1	ei: Ecological Inference Estimation	14
5.2	eiread: Quantities of Interest from Ecological Inference Estimation	16
5.3	plot.ei: Plotting Ecological Inference Estimates	18
5.4	eiRxCplot: Plotting Ecological Inference Estimates with eiRxC information	20

1 Introduction: Ecological Inference

This program provides a method of inferring individual behavior from aggregate data. It implements the statistical procedures, diagnostics, and graphics from the book, **A Solution to the Ecological Inference Problem: Reconstructing Individual Behavior from Aggregate Data** (Princeton: Princeton University Press, 1997), by Gary King. See <http://gking.harvard.edu/eicamera/kinroot.html>. Except where indicated, all references to page, section, chapter, table, and figure numbers in this document refer to the book.

Ecological inference is the process of using aggregate (i.e., “ecological”) data to infer discrete individual-level relationships of interest when individual-level data are not available. Ecological inferences are required in political science research when individual-level surveys are unavailable (e.g., local or comparative electoral politics), unreliable (racial politics), insufficient (political geography), or infeasible (political history). They are also required in public policy (e.g., for applying the Voting Rights Act) and other academic disciplines ranging from epidemiology and marketing to sociology and quantitative history. Most researchers using aggregate data have encountered some form of the ecological inference problem.

Because the ecological inference problem is caused by the lack of individual-level information, no method of ecological inference, including that estimated by this program, will always produce accurate results. However, potential difficulties are reduced by models that include more available information, diagnostics to evaluate when assumptions need to be modified, easy methods of modifying the assumptions, and uncertainty estimates for quantities of interest. We recommend reviewing Chapter 16 while using this program for actual research. Model dependence defines ecological inference when going anywhere beyond the bounds and tomography plots.

2 Overview: R-commands

In this section we describe the basic commands in EI. For this purpose, and without loss of generality, we use the running example from the book portrayed in Table 2.3 (page 31). This example uses the fraction of the voting age population who are black (X_i), the fraction turning out to vote (T_i), and the number of voting age people (N_i) in each precinct ($i = 1, \dots, p$) to infer the fraction of blacks who vote (β_i^b) and the fraction of whites who vote (β_i^w), also in each precinct. (For an extended example of EI, refer to the user’s guide below.) As only five commands are required to use EI, the program can be easily run interactively, or in batch mode as a regular R program. An example of a sequence of these commands are as follows:

```
> formula = t ~ x
> dbuf <- ei(formula=formula,total="n",data=data)
> summary(dbuf)
> eiread(dbuf, "betab","betaw")
> plot(dbuf, "tomog","betab","betaw","xtfit")
```

In most applications, `plot` and `eiread` would likely be run multiple times with different options chosen, and other commands would be included with these five to read in the data (`t`, `x`, and `n`).

- **ei**: To run the main procedure

Use the format, `dbuf = ei(formula,total="n",data=data)`, which takes three $p \times 1$ vectors as inputs: `t` (e.g. the fraction of the voting age population turning out to vote); `x` (e.g. the fraction of the voting age population who are black); and `n` (e.g. the total number of people in the voting age population). The output of this procedure is the list `dbuf` (i.e. an output object of class list called a “data buffer”). The data buffer includes the maximum likelihood estimates from the `ei` procedure. To simulate quantities of interest, then run `dbufsim = ei.sim(dbuf)`. After running `ei` and `ei.sim`, it is a good idea to save `dbuf` and `dbufsim` on disk for further analysis. The output data buffer from `ei` and `ei.sim` includes a large variety of different results useful for understanding the results of the analysis. A minimal set of nonrepetitive information is stored in this list (or data buffer), and a large variety of other information can be easily computed from it. Fortunately, you do not need to know whether the information you request is stored or computed as both are treated the same.

To extract information from the data buffer, three procedures are available:

- **summary**: To obtain a general summary of district-level information

`summary(dbuf)` will produce aggregate information about the `ei` results. If you have only run `ei`, `summary` will summarize the maximum likelihood estimates. If you have run `ei.sim`, `summary` will produce a summary of both the maximum likelihood estimates and simulated quantities of interest. It also includes information on the values specified for the prior.

- **plot**: To graph relevant information

For graphics, use `plot(dbuf, "name");`, where `dbuf` is the `ei`.object that is the output of `ei` or `ei.sim`, and `name` can be any number of a long list of ready-made graphs. For example, `plot(dbuf,"tomog")` will plot a tomography graph, and `plot(dbuf,"xt")` will display a scattercross graph. Any number of graphs can be selected in plot for output. For example, `plot(dbuf,"tomog","xt")` will produce both a tomography graph and a scattercross graph.

Some of the plots can be produced after estimating the maximum likelihood estimates using `ei`, but before simulating quantities of interest with `ei.sim`. These initial plots can be useful in determining which priors to include. However, for some plots, `plot` will need the simulated quantities of interest. These plots only run on data buffers from `ei.sim`.

- **eiread**: To obtain relevant information and numerical results

For numerical information, use `v <- eiread(dbuf, "name")`, where `v` is the item extracted, `dbuf` is the data buffer output from `ei` or `ei.sim`, and `name` can be any of a list of output possibilities. For example, `eiread(dbufsim, "betab")` will give a vector of point estimates of β_i^b , `eiread(dbufsim, "ci80w")` will give 80% confidence intervals for β_i^w . Any number of items can be selected for output. For example, `eiread(dbufsim, "betab", "ci80b")` will produce a list of the point estimates and 80% confidence intervals for β_i^b .

3 User's Guide: An Example

We now show how to use EI through a simple example. We use data on voter registration and racial background of people from 268 counties in four Southern U.S. States: Florida, Louisiana, and South Carolina. These are the same data used in Chapter 10. The data include the total voting age population in each county (**n**), the proportion of the population in each county who are black (**x**), and the proportion of the population in each county who are registered to vote (**t**). The proportion of whites registered can be computed by $(1-x)$. You can load the data into R using the command

```
> library(ei)
> data(matproii)
```

The statistical goal is to estimate the fraction of blacks registered (β_i^b) and the fraction of whites registered (β_i^w) in each county. These quantities of interest are generally unknown. However, EI also includes an option that allows the user to assess the reliability of the EI algorithm when the true quantities of interest are known. To illustrate the use of this “truth” option, the data also include the true fractions of registered blacks (**tb**) and the true fraction of registered whites (**tw**).

3.1 The Basic EI Algorithm

To begin, we perform ecological inference by calling the function **ei**. **ei** first computes and maximizes a likelihood distribution based on the inputted data. Then we use the function **ei.sim** to estimate county-level quantities of interest based on the possible values, or “bounds”, for each county and the overall likelihood distribution.

To run this algorithm, at a minimum we need to have three vectors. Two vectors, **t** and **x**, contain the aggregate known information about the counties of interest. In this example, **t** is the proportion of voters registered in each county, and **x** is the proportion of blacks in each county. These two vectors contain aggregate information, since we are interested in the proportion of voters in each country who are black. The last vector we need is **n**, the number of people of interest in each county. In this example, **n** is the number of people of voting age.

We proceed by performing ecological inference without covariates on the data:

```
> formula = t ~ x
> dbuf = ei(formula=formula,total="n",data=matproii)
```

When the data argument is included, the arguments “**t**”, “**x**”, “**n**” refer to names of the variables in the dataset and therefore must be strings. If the data argument is not specified, **ei(t,x,n)** can be used where **t**, **x** and **n** are the vectors containing the data. To include a covariate on β_i^b simply specify a covariate vector for the option **Zb**, or a string that refers to the name of the covariate in the dataset. Similarly, to include a covariate on β_i^w , specify a covariate vector for for the option **Zw**, or a string that refers to the name of the covariate in the dataset.

Next, we use **summary(dbuf)** and **summary(dbufsim)** to obtain general information about the estimation.

```

> summary(dbuf)

Erho = 0.5 , Esigma = 0.5 , Ebeta = 0.5 , N = 268 , Resamp = 4

Maximum likelihood results in scale of estimation (and se's)
      Bb0      Bw0      sigB      sigW      rho Zb Zw
1.2670 1.9348 -1.1151 -1.3272 1.6051  0  0
0.2786 0.2764  0.2096  0.1616 0.3096  0  0

Untruncated psi's
      BB      BW      SB      SW      RHO
0.9604 1.1264 0.327 0.2672 0.9202

Truncated psi's (ultimate scale)
      BB      BW      SB      SW      RHO
0.6199 0.8261 0.1972 0.1418 0.7867

Aggregate Bounds
      betab  betaw
lower 0.2125 0.7025
upper 0.9754 0.9200

Estimates of Aggregate Quantities of Interest
      mean      sd
Bb 0.5691 0.0137
Bw 0.8183 0.0039

```

The `summary` function provides basic information about the ecological inference estimation, the maximum likelihood estimates on three scales, and an overall summary of the quantities of interest. First, it reports the values of the priors used in ecological inference (`Erho`, `Esigma`, `Ebeta`). It also reports the number of counties in the dataset (`N`), as well as the number of importance sampling iterations required to produce estimates of the quantities of interest (`resamp`).

When `summary` is used on a `ei.sim` object, `summary` produces information both about the maximum likelihood estimates and the quantities of interest. First, it provides the maximum likelihood estimates in the scale of estimation. Next, it provides the values of the MLEs when they are transformed into an untruncated bivariate normal distribution. These estimates provide information about the location of the highest density of estimates for the proportion of blacks who are registered to vote and the proportion of whites who are registered to vote. Last, it provides the values of the MLEs on the ultimate truncated bivariate normal scale. These estimates are an unweighted average of the fraction of blacks and whites registered to vote over all the counties in the sample. In this example, the EI algorithm predicts that the unweighted average of the proportion of blacks registered to vote across counties is 0.57 and the unweighted average of the proportion of whites registered to vote is 0.82.

Finally, `summary` produces information on aggregate quantities of interest. The aggregate bounds

are the mean of the bounds on the proportions of black and white voters, weighted by the population in each county. The aggregate quantities of interest are the weighted mean of the proportion of registered blacks and the proportion of registered whites in each county. In this example the weighted average proportion of blacks who are registered is 0.57, and the weighted average proportion of whites who are registered is 0.82.

3.2 Extracting Quantities of Interest

`eiread` extracts quantities of interest in a list format from the object `dbuf`. For example,

```
> bb.out <- eiread(dbuf, "betab", "sbetab")
```

extracts the point estimates and estimated standard deviations for β_i^b , the estimates of the proportion of registered blacks in each county. The user can then use `bb.out$betab` to obtain a vector of the point estimates for β_i^b , and `bb.out$sbetab` to obtain a vector of the standard deviations for β_i^b .

`eiread()` takes any number of arguments to extract any number of quantities of interest from the data buffer. `?eiread` can be used to find a list of quantities of interest that are available in EI. Among the most useful arguments are "betab" and "betaw", which report the point estimates; "sbetab" and "sbetaw", which report the standard deviations of the point estimates; and "CI80b" and "CI80w", which report the confidence intervals of the point estimates.

3.3 Plotting in EI

Plotting EI output is extremely useful for understanding the results of the ecological inference algorithm and diagnosing problems with estimation. First, we graph a tomography plot of the data (Figure 1).

```
> plot(dbuf, "tomog")
```

Each line on the map represents the possible values for β_i^b and β_i^w for one county. The contour lines identify the portion of the lines that have the highest probability of containing the true estimates of β_i^b and β_i^w . These contour lines provide information about the overall pattern of registration. Note that the area with highest probability is in the upper right-hand corner, where the proportion of whites registered is between 0.75 and 1 and the proportion of blacks registered is between 0.5 and 1. Further, we see that the lines are clustered in the upper half of the figure, indicating that the possible values of β_i^w will have a lower variance than the possible values of β_i^b .

Figure 2 is a double plot of the point estimates and their confidence intervals. To compute this, we call `plot` to generate two graphs: a tomography plot with the point estimates generated from the algorithm, and a tomography plot with 80% confidence intervals on the point estimate.

```
> plot(dbuf, "tomogE", "tomogCI")
```

This plot is useful to visualize the actual estimates and confidence intervals for each county. We can see that the point estimates and confidence intervals are clustered in the same area as the contours from the previous plot. Further, the point estimates and confidence intervals only fall on the lines

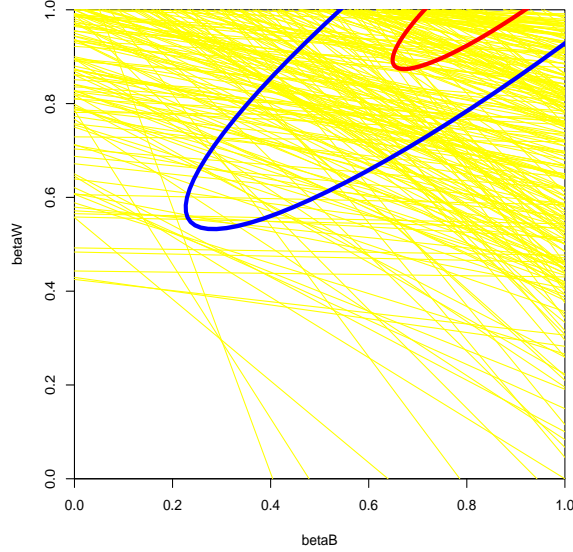


Figure 1: Tomography Plot with ML Contours

that indicate the possible values of β_i^b and β_i^w .

Figure 3 shows plots that indicate the distribution of β_i^b and β_i^w . To produce these plots, run:

```
> plot(dbuf, "betab", "betaw")
```

Density plots are useful to visualize the location and uncertainty of the point estimates. The green line represents the density of the simulated point estimates, and the black tick marks are a rug plot of the point estimates, β_i^b and β_i^w . You can see that the variance of the point estimates from β_i^b is much higher than the variance of point estimates from β_i^w .

Figure 4 portrays the results of the EI algorithm by plotting the proportion of blacks in each country by the proportion of registered voters in each county. To produce this plot, use:

```
> plot(dbuf, "xtfit")
```

The circles around each of the points in this plot are proportional to the population of each county. The graph represents the likelihood estimates by plotting the expected number of registered voters given the proportion of blacks in each county, represented by the yellow line. The red lines in the plot are the 80% confidence interval around the regression line. The higher uncertainty in the estimates of black registration can be seen by the absence of points on the right hand side of the graph and the larger confidence interval on the right hand side of the graph, where the proportion of blacks in the county is relatively large.

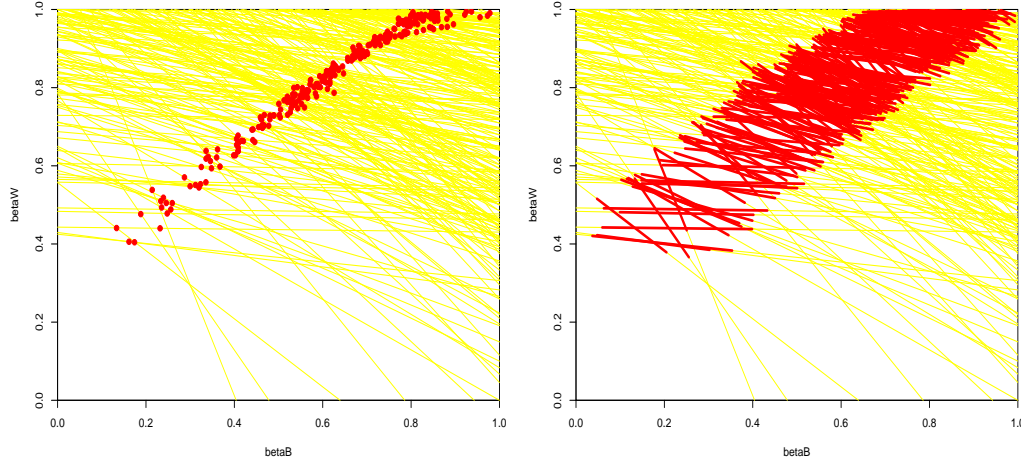


Figure 2: Tomography Plots: Point Estimates (left) and Confidence Intervals (right)

Finally, if we have data on the true proportions of registered blacks and registered whites in each county, as we do in this dataset, we can use plots in `EI` to assess how well the algorithm works on the given data. To do this, rerun the `ei` algorithm, adding the truth vectors, `tb` and `tw` as the `truth` argument.

```
> truth = cbind(matproii$tb,matproii$tw)
> formula = t ~ x
> dbuf = ei(formula=formula,total="n",data=matproii,truth=truth)
```

Then use `plot` to compare the estimates of the `EI` algorithm to the true proportions of white and black registered voters in each county.

```
> plot(dbuf, "truth")
```

The “truth” plot (Figure 5) has four components. The top two figures have the posterior densities of the aggregate quantities of interest B^b and B^w . These densities tell us in what range the algorithm predicts that the point estimates lie. For example, the density of B^b is wider than the density of B^w , indicating that our estimates of B^w lie in a smaller range than than our estimates of B^b . The true aggregate values, computed from the `truth` data we inputed, are indicated by the vertical bars. The fact that the true B^b and B^w are within the densities we computed confirms that the model is did a good job at predicting the true proportions of registered white and black voters.

The bottom two figures in Figure 5 plot the estimated values of β_i^b and β_i^w against the true values. The size of each of the circles plotted is proportional to the number of blacks in the county in the first graph and the number of whites in each county in the second graph. If the estimated values were exactly equal to the true values, all of the points would be on the 45° line. Because the points fall quite close to the 45° and do not deviate from the line in a systematic way, we can see that the `EI` algorithm predicts the point estimates quite well.

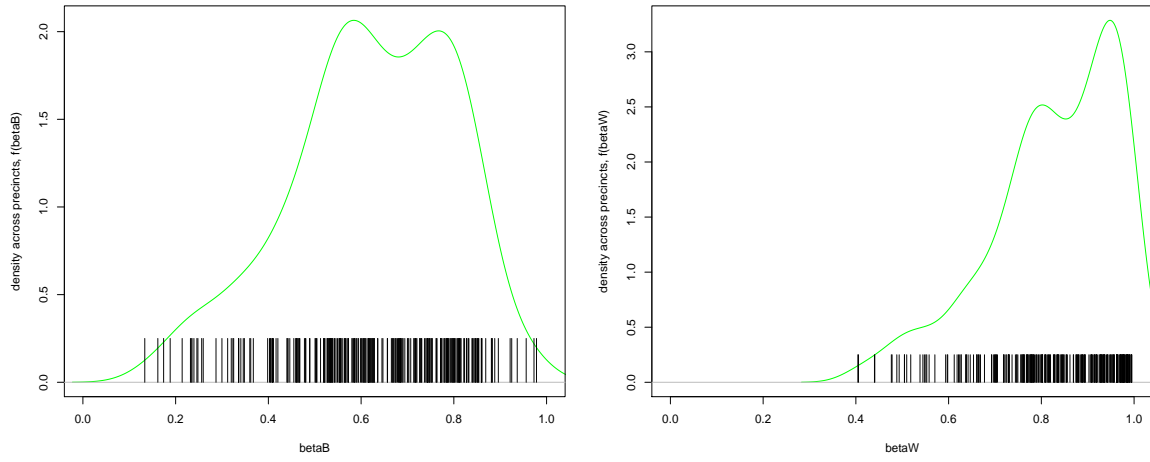


Figure 3: Plots with Distribution of Point Estimates for β_i^b (left) and β_i^w (right), for all i

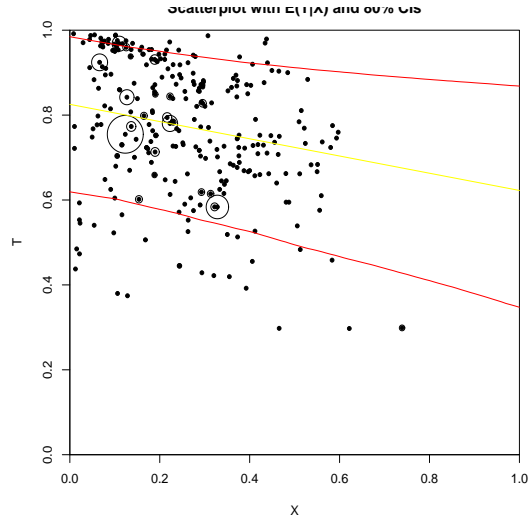


Figure 4: X and T Scatterplot with $E(T|X)$ and 80% CIs

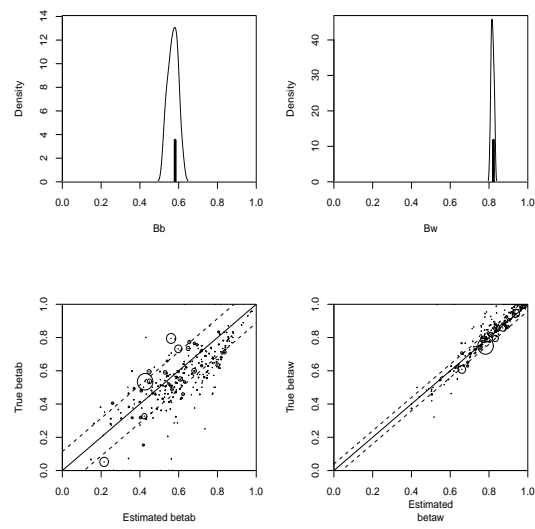


Figure 5: Comparing Estimates to the Truth at the County Level

4 Plotting RxC Information: eiRxCplot

`eiRxCplot` allows the user to include RxC information into the tomography plot in order to visualize the RxC dynamic in a 2x2 analysis. We illustrate this by using the data `eiRxCsample` within the `ei` package. `eiRxCsample` contains both demographic and turnout information for a district. The variable `x` represents the proportion of non-whites in each precinct, the variable `t` represents the proportion of people in each precinct that turned out to vote, and `n` represents the total number of people in each precinct. The data also contain the variables `whit.ei`, `black.ei`, and `hisp.ei`, which have the proportion of whites, blacks, and hispanics in each precinct.

We might be interested in doing a 2x2 analysis to understand non-white vs. white turnout in each precinct. We could use `ei` to run this analysis, as we did above:

```
> data(eiRxCsample)
> formula = t ~ x
> dbuf = ei(formula=formula, total="n", data=eiRxCsample)
```

However, we might be interested in whether black turnout is different from hispanic turnout, even though we have grouped them together in our 2x2 analysis as “non-whites”. To visualize information in our analysis that comes from each different demographic group, we can use `eiRxCplot`.

```
> eiRxCplot(dbuf, groups=c("black.ei", "whit.ei", "hisp.ei"), data=eiRxCsample, groupnames=c("Black", "White", "Hispanic"))
```

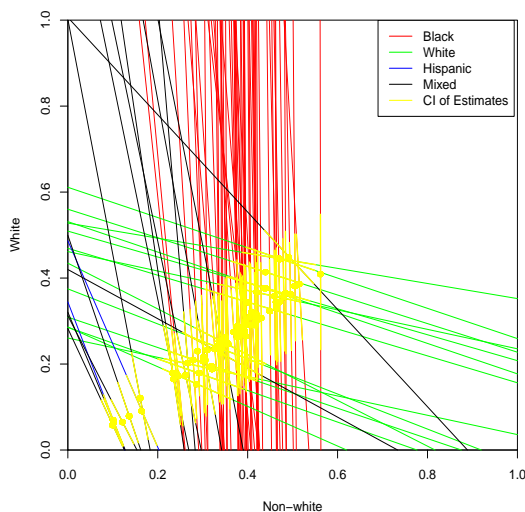


Figure 6: Visualizing RxC information in a 2x2 Analysis

`eiRxCplot` plots the same tomography plot that can be plotted using the command `plot(dbuf)`, but also includes RxC information by coloring precincts that have over a certain threshold of one demographic group. For example, in Figure 6, we see that precincts that are more than 65% black (colored in red) are very informative and have a non-white turnout rate of between 20% and 60%. There are fewer precincts that are more than 65% hispanic (colored in blue), and these precincts

have a non-white turnout rate between 0% and 20%. As a result, the `eiRxCplot` tells us two things. First, our results for non-white turnout are mainly driven by majority black precincts, rather than majority hispanic precincts. Second, black turnout in this district is generally higher than hispanic turnout. Even though our original analysis does not distinguish between black and hispanic groups, `eiRxCplot` gives us a lot of information about the eiRxC dynamic within the district.

5 Reference to EI Functions

5.1 ei: Ecological Inference Estimation

Description

`ei` is the main command in the package `EI`. It gives observation-level estimates (and various related statistics) of β_i^b and β_i^w given variables T_i and X_i ($i = 1, \dots, n$) in this accounting identity: $T_i = \beta_i^b * X_i + \beta_i^w * (1 - X_i)$. Results are stored in an `ei` object, that can be read with `summary()` or `eiread()` and graphed in `plot()`.

Usage

```
ei(t, x, n, Zb=1, Zw=1, data=NA, erho=.5, esigma=.5, ebeta=.5, ealphab=NA,
  ealphaw=NA, truth=NA, precision=4)
ei.sim(dbuf)
```

Arguments

<code>t</code>	$p \times 1$ vector, or name of vector if data argument is included
<code>x</code>	$p \times 1$ vector, or name of vector if data argument is included
<code>n</code>	$p \times 1$ vector, or name of vector if data argument is included
<code>Zb</code>	$p \times k^b$ matrix of covariates
<code>Zw</code>	$p \times k^w$ matrix of covariates
<code>data</code>	optional data frame that if specified should contain <code>t,x,ns</code> . If using covariates and data is specified, data should also contain <code>Zb</code> and <code>Zw</code> .
<code>erho</code>	The standard deviation of the normal prior on ϕ_5 for the correlation. Default = 0.5.
<code>esigma</code>	The standard deviation of an underlying normal distribution, from which a half normal is constructed as a prior for both $\check{\sigma}_b$ and $\check{\sigma}_w$. Default = 0.5
<code>ebeta</code>	Standard deviation of the "flat normal" prior on \check{B}^b and \check{B}^w . The flat normal prior is uniform within the unit square and dropping outside the square according to the normal distribution. Set to zero for no prior. Setting to positive values probabilistically keeps the estimated mode within the unit square. Default = 0.5
<code>ealphab</code>	$\text{cols}(Zb) \times 2$ matrix of means (in the first column) and standard deviations (in the second) of an independent normal prior distribution on elements of α^b . If you specify <code>Zb</code> , you should probably specify a prior, at least with mean zero and some variance (default is no prior). (See Equation 9.2, page 170, to interpret α^b).
<code>ealphaw</code>	$\text{cols}(Zw) \times 2$ matrix of means (in the first column) and standard deviations (in the second) of an independent normal prior distribution on elements of α^w . If you specify <code>Zw</code> , you should probably specify a prior, at least with mean zero and some variance (default is no prior). (See Equation 9.2, page 170, to interpret α^w).

truth	A length(t) x 2 matrix of the true values of the quantities of interest.
precision	The number of decimal places that are printed in summary and eiread functions.

Details

The EI algorithm is run using the **ei** command. A summary of the results can be seen graphically using **plot(ei.object)** or numerically using **summary(ei.object)**. Quantities of interest can be calculated using **eiread(ei.object)**.

References

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

Examples

```
data(sample)
dbuf <- ei("t","x","n",data=sample)
summary(dbuf)
dbufsim <- ei.sim(dbuf)
summary(dbufsim)
```


5.2 eiread: Quantities of Interest from Ecological Inference Estimation

Description

`eiread` is the command that pulls quantities of interest from the `ei` object. The command returns a list of quantities of interest requested by the user.

Usage

```
eiread(ei.object, ...)
```

Arguments

<code>ei.object</code>	An <code>ei</code> object from the function <code>ei</code> .
<code>...</code>	A list of quantities of interest for <code>eiread()</code> to return. See values below.

Value

<code>betab</code>	$p \times 1$ point estimate of β_i^b based on its mean posterior. See section 8.2
<code>betaw</code>	$p \times 1$ point estimate of β_i^w based on its mean posterior. See section 8.2
<code>sbetab</code>	$p \times 1$ standard error for the estimate of β_i^b , based on the standard deviation of its posterior. See section 8.2
<code>sbetaw</code>	$p \times 1$ standard error for the estimate of β_i^w , based on the standard deviation of its posterior. See section 8.2
<code>phi</code>	Maximum posterior estimates of the CML
<code>psisims</code>	Matrix of random simulations of ψ . See section 8.2
<code>bounds</code>	$p \times 4$: bounds on β_i^b and β_i^w , lowerB ~ upperB ~ lowerW ~ upperW. See Chapter 5.
<code>abounds</code>	2×2 : aggregate bounds rows:lower, upper; columns: betab, betaw. See Chapter 5.
<code>aggs</code>	Simulations of district-level quantities of interest \hat{B}^b and \hat{B}^w . See Section 8.3.
<code>maggs</code>	Point estimate of 2 district-level parameters, \hat{B}^b and \hat{B}^w based on the mean of aggs. See Section 8.3.
<code>VCaggs</code>	Variance matrix of 2 district-level parameters, \hat{B}^b and \hat{B}^w . See Section 8.3.
<code>CI80b</code>	$p \times 2$: lower~upper 80% confidence intervals for β_i^b . See section 8.2.
<code>CI80w</code>	$p \times 2$: lower~upper 80% confidence intervals for β_i^w . See section 8.2.
<code>eaggbias</code>	Regressions of estimated β_i^b and β_i^w on a constant term and X_i .
<code>goodman</code>	Goodman's Regression. See Section 3.1

References

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

Examples

```
data(sample)
dbuf <- ei("t","x","n",data=sample)
eiread(dbuf, "phi")
dbufsim <- ei.sim(dbuf)
eiread(dbufsim, "betab", "betaw")
```

5.3 plot.ei: Plotting Ecological Inference Estimates

Description

‘plot’ method for the class ‘ei’.

Usage

```
plot.ei(ei.object, ...)
```

Arguments

<code>ei.object</code>	An ei object from the function <code>ei</code> .
<code>...</code>	A list of options to return in graphs. See values below.

Value

<code>tomogD</code>	Tomography plot with the data only. See Figure 5.1, page 81.
<code>tomog</code>	Tomography plot with ML contours. See Figure 10.2, page 204.
<code>tomogCI</code>	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.
<code>tomogCI95</code>	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.
<code>tomogE</code>	Tomography plot with estimated mean posterior β_i^b and β_i^w points.
<code>tomogP</code>	Tomography plot with mean posterior contours.
<code>betab</code>	Density estimate (i.e., a smooth version of a histogram) of point estimates of β_i^b 's with whiskers.
<code>betaw</code>	Density estimate (i.e., a smooth version of a histogram) of point estimates of β_i^w 's with whiskers.
<code>xt</code>	Basic X_i by T_i scatterplot.
<code>xtc</code>	Basic X_i by T_i scatterplot with circles sized proportional to N_i .
<code>xtfit</code>	X_i by T_i plot with estimated $E(T_i X_i)$ and conditional 80% confidence intervals. See Figure 10.3, page 206.
<code>xtfitg</code>	<code>xtfit</code> with Goodman's regression line superimposed.
<code>estsims</code>	All the simulated β_i^b 's by all the simulated β_i^w '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 β_i^b (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 β_i^w (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).

References

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

Examples

```
data(sample)
dbuf <- ei("t","x","n",data=sample)
plot(dbuf, "tomog")
dbufsim <- ei.sim(dbuf)
plot(dbufsim, "tomog", "betab", "betaw", "xtfit")
```

5.4 eiRxCplot: Plotting Ecological Inference Estimates with eiRxC information

Usage

```
eiRxCplot(ei.object, groups, data, groupnames=groups,  
threshold=.65, random=FALSE,prop=.15,title="EI RxC Plot",  
xaxis="betab",yaxis="betaw", estimates=TRUE, legendpos=c(.69,1))
```

Arguments

<code>ei.object</code>	An <code>ei</code> object from the function <code>ei</code> or <code>ei.sim</code> .
<code>groups</code>	A list of groups to be represented in the eiRxC plot
<code>data</code>	data where the groups are contained and from which the <code>ei</code> object was produced
<code>groupnames</code>	A list of group names corresponding to the groups to be printed in the legend
<code>threshold</code>	The threshold where precincts in which a group has more than this proportion will have tomography lines colored to represent that group.
<code>random</code>	TRUE to print only a random selection of tomography lines
<code>prop</code>	Specifies the proportion of tomography lines to be printed if <code>random</code> is TRUE
<code>title</code>	Title for the plot
<code>xaxis</code>	X-axis label for the plot
<code>yaxis</code>	Y-axis label for the plot
<code>estimates</code>	Specifies whether or not the quantities of interest should be included in the plot. If <code>estimates=FALSE</code> , <code>ei.sim</code> is not needed to use <code>eiRxCplot</code> .
<code>legendpos</code>	Specifies coordinate position for the legend.

References

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

Examples

```
data(eiRxCsample)  
dbuf <- ei("t","x","n",data=eiRxCsample)  
dbufsim <- ei.sim(dbuf)  
eiRxCplot(dbufsim, groups=c("black.ei", "whit.ei", "hisp.ei"), data=eiRxCsample,  
groupnames=c("Black", "White", "Hispanic"), title="Non-white vs. White  
Turnout", xaxis="Non-white", yaxis="White")
```

References

- [1] Gary King, *A Solution to the Ecological Inference Problem: Reconstructing Individual Behaviour from Aggregate Data*, Princeton University Press (1997).
- [2] Useful documentation also available at <http://GKing.Harvard.Edu>.
- [3] For R language see <http://www.r-project.org>.
- [4] Venables, W.N., and Ripley, B.D., *Statistics and Computing*, Springer (2002).