Ecological Inference: Reconstructing Individual behavior from Aggregate Data; software available at `http://gking.harvard.edu/ei/`, free of charge and open-source (under the terms of the GNU GPL, v. 2).

Gary King [1]

April 22, 2009

[1]David Florence Professor of Government, Harvard University (Institute for Quantitative Social Sciences, 1737 Cambridge Street, Harvard University, Cambridge MA 02138; `http://GKing.Harvard.Edu`, `King@Harvard.Edu`, (617) 495-2027).

Contents

0.1 Acknowlegments

The code, which was originally written in Gauss, has been translated into R by Elena Villalon who also generated the plots that are presented in this document.

0.2 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 (EIBAD), A Solution to the Ecological Inference Problem: Reconstructing Individual Behavior from Aggregate Data (Princeton: Princeton University Press, 1997), by Gary King. Please read the book prior to trying this program (a sample chapter and other related information is available at King's website). Except where indicated, all references to page, section, chapter, table, and figure numbers in this document refer to the book.

Ecological inference, as traditionally defined, 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. As existing methods usually lead to inaccurate conclusions about the empirical world, the ecological inference problem had been to develop a method that gives accurate answers. 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 numerous areas of major significance 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 introduced in this book and estimated by this program, will produce precisely accurate results in every instance. However, potential difficulties are minimized here 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 all quantities of interest. I recommend reviewing Chapter 16 while using this program for actual research.

0.3 Overview: R-commands to run the application

As only four commands are required to use $\mathfrak{E}I$ the program can be easily run interactively, or in batch mode as a regular R program. Each command may also be used with many optional globals and subcommands. An example of these commands are as follows:

```
> library(ei)
```

```
> data(sample)
> t <- sample[[1]]
> x <- sample[[2]]
> n <- sample[[3]]
> dbuf <- ei(t,x,n,1,1)
> plot(dbuf)
> plot(dbuf, ``nonpar'')
> summary(dbuf)
> summary(dbuf, ``betab'')
```

In most applications, plot (which can also be invoked with the name eigraph) and summary would likely be run multiple times with different options chosen, and other commands would be included with these four to read in the data (t, x, and n).

These four commands described in this section through a very simple use of $\mathfrak{E}I$. (Refer to the reference section below for further details and more sophisticated uses. That section also includes sample data you can run with simple examples.) For this purpose, and without loss of generality, I 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 ($i = 1, \ldots, p$) 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.

- ei: To run the main procedure

  Use the format, dbuf = ei(t,x,n,1,1);, 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 remaining two inputs are for optional covariates; for the basic model, set them each to 1 for no covariates.) The output of this procedure is the list dbuf (i.e., data buffer). After running ei, it is a good idea to save dbuf on disk for futher analysis. The output data buffer from ei 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, two procedures are available:

- eigraph or plot: To graph relevant information

  For graphics, use plot(dbuf,"name") or eigraph(dbuf,"name");, where dbuf is the list that is the output of ei, and name can be any of a long list of ready-made graphs. For example, use plot(dbuf,"fit"); to assess the fit of the model, plot(dbuf,"tomog

"); to print a tomography graph, or eigraph(dbuf,"xgraph"); to display a scattercross graph.

- analysis: To obtain relevant information and numerical results

  For numerical information, use v <- summary(dbuf,"name"), where v is the item extracted, dbuf is the list that is the output of ei, and name can be any of a long list of output possibilities. For example, use betab for a vector of point estimates of β_i^b, ci80w for 80% confidence intervals for β_i^w, or sum to print a summary of district-level estimates and information.

0.4 `ei parametric`: Estimation for 2x2 tables

It runs ei setting the argument EnonPar=0, which is the default value, and is based on the assumption of the truncated bivarate normal. Uses the routine nlminb to find the local maximum-liklehood for the parameters optimization with bounds. Then, it uses the routine optim with the parameter estimations obtained with nlminb to calculate the numerically differentiated hessian. For a detailed description go to `http://gking.harvard.edu/ei`.

0.4.1 `ei parametric`: Demos

There are several examples in the demo directory, some of them have true values for β_B and β_W (i.e., cens1910, kyck88,matproII,and scsp). The following commands allow the user to run the demo data sets:

```
> library(ei)
> demo(eip.default)
> demo(eip.fultongen)
> demo(eip.in90)
> demo(eip.kyck88)
> demo(eip.cens1910)
> demo(eip.lavoteall)
> demo(eip.matproII)
> demo(eip.nj)
> demo(eip.pa90)
> demo(eip.scp)
```

Some of the data sets have many precincts (> 1000) and, even when the performance of "ei" is very efficient, the graphics may take sometime to display all the precincts.

0.4.2 `ei parametric`: Example

We are presenting an example with 75 precincts

```
> data(sample, package = "ei")
> dbuf <- ei(sample[[1]], sample[[2]], sample[[3]], 1, 1)

> message("Obtaining overall beta's and std errors")
> berr <- summary(dbuf, "paggs")
> berr

> message("Running graphics:")
```

```
> eigraph(dbuf, "tomoge")
```

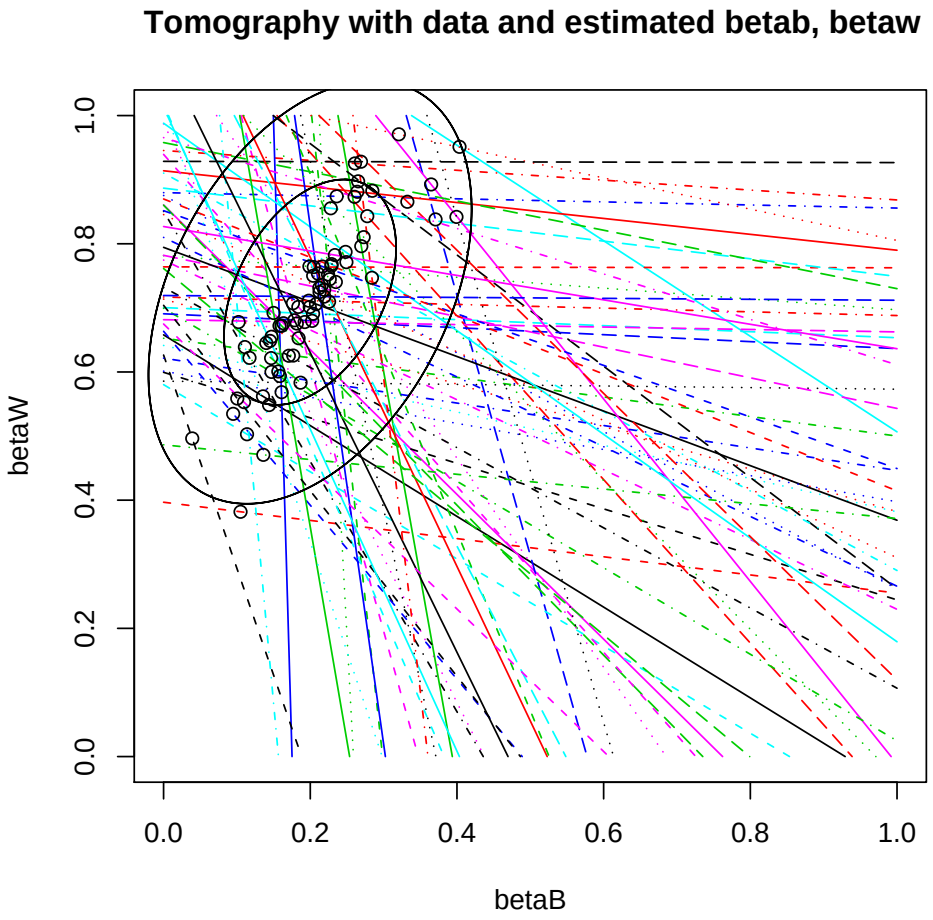

Figure 1: Tomography with estimated betaB and betaW

```
> eigraph(dbuf, "tomogci")
```

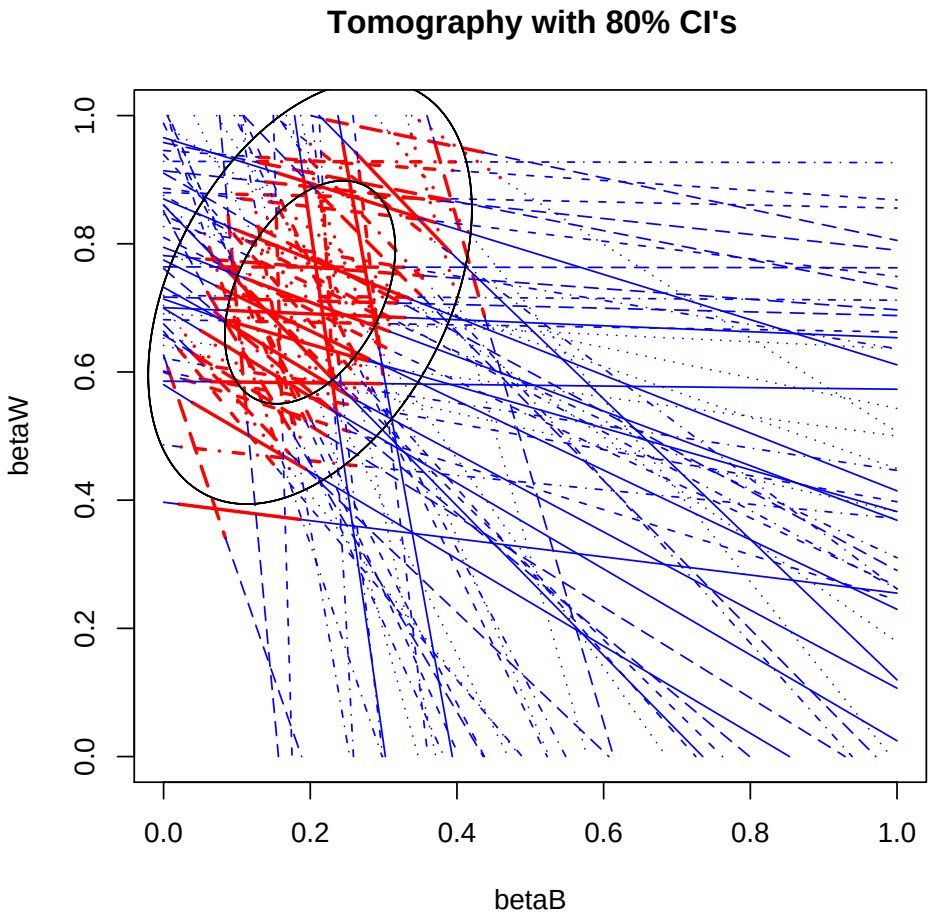

Figure 2: Tomography with 80% confidence intervals

```
> eigraph(dbuf, "tomogci95")
```

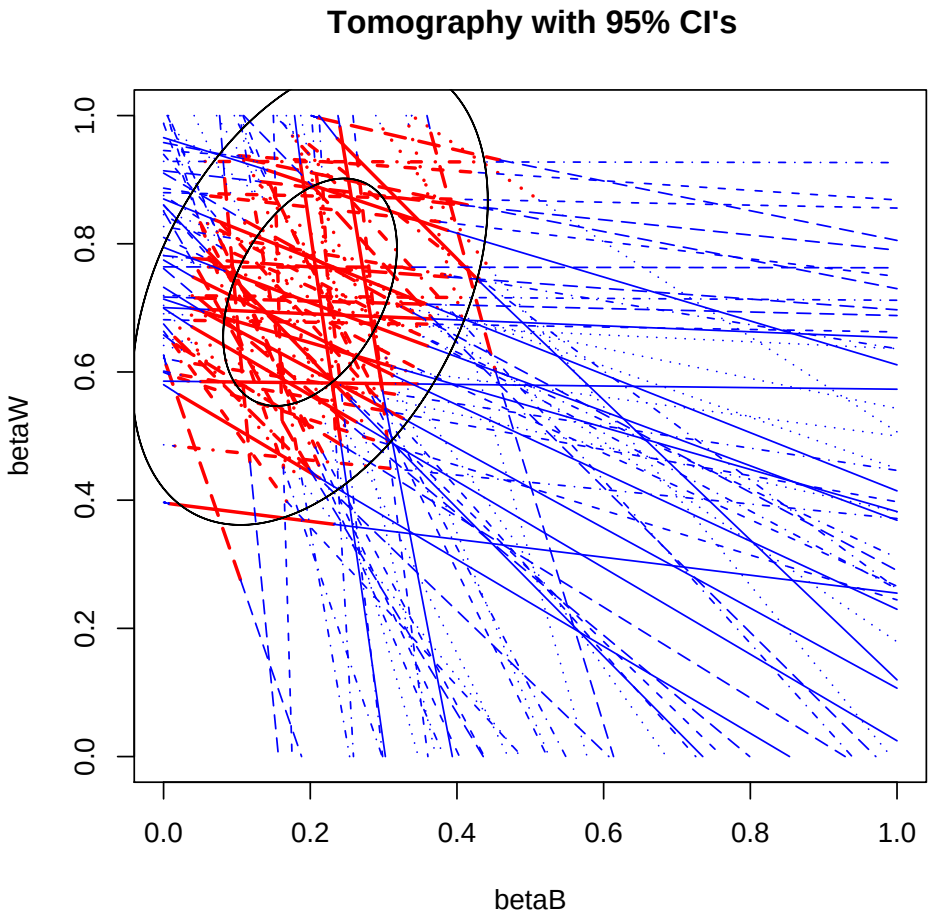

Figure 3: Tomography with 95% confidence intervals

```
> eigraph(dbuf, "estsims")
```

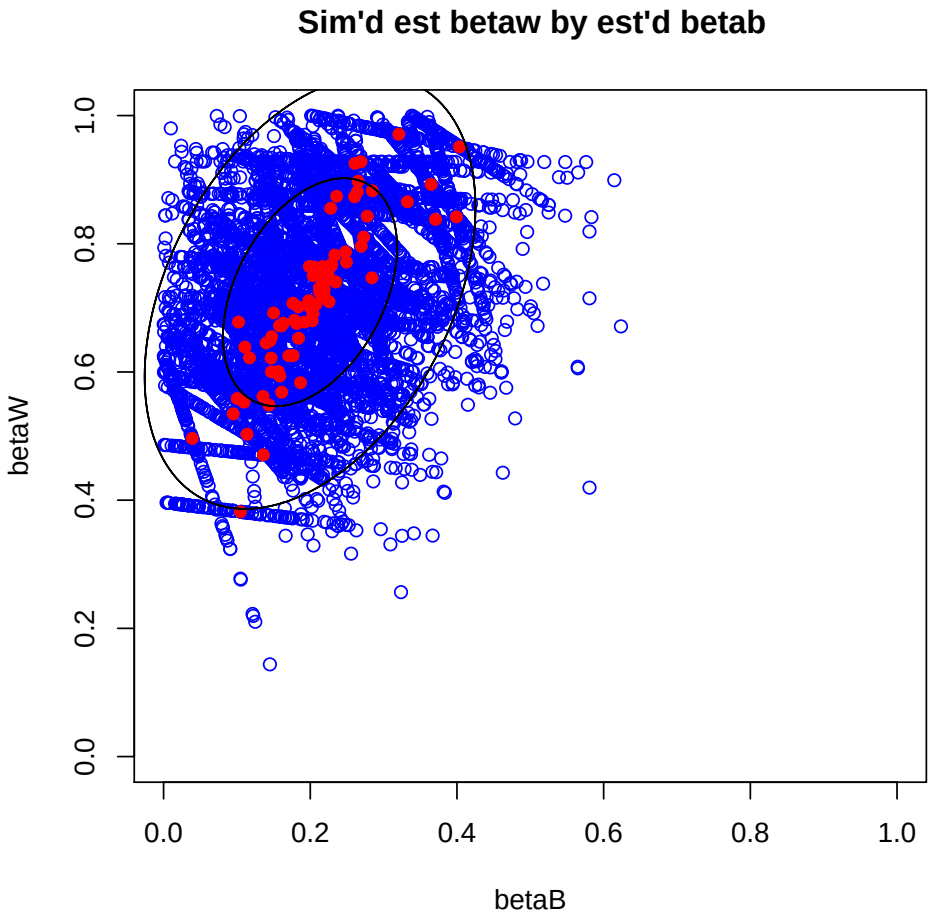

Figure 4: Simulated betaB's by simulated betaW's

```
> plot(dbuf)
```

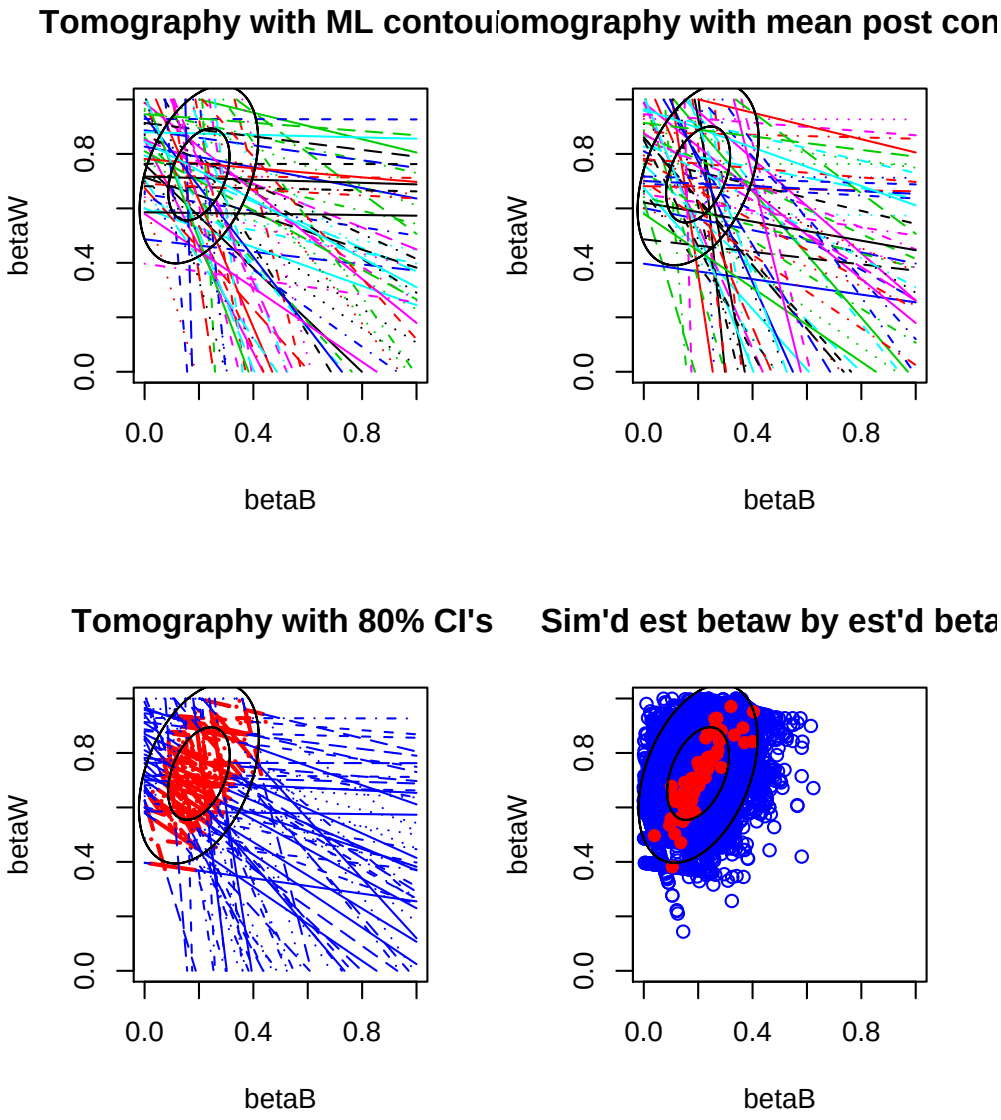

Figure 5: Tomographic plots

```
> eigraph(dbuf, "xtc")
```

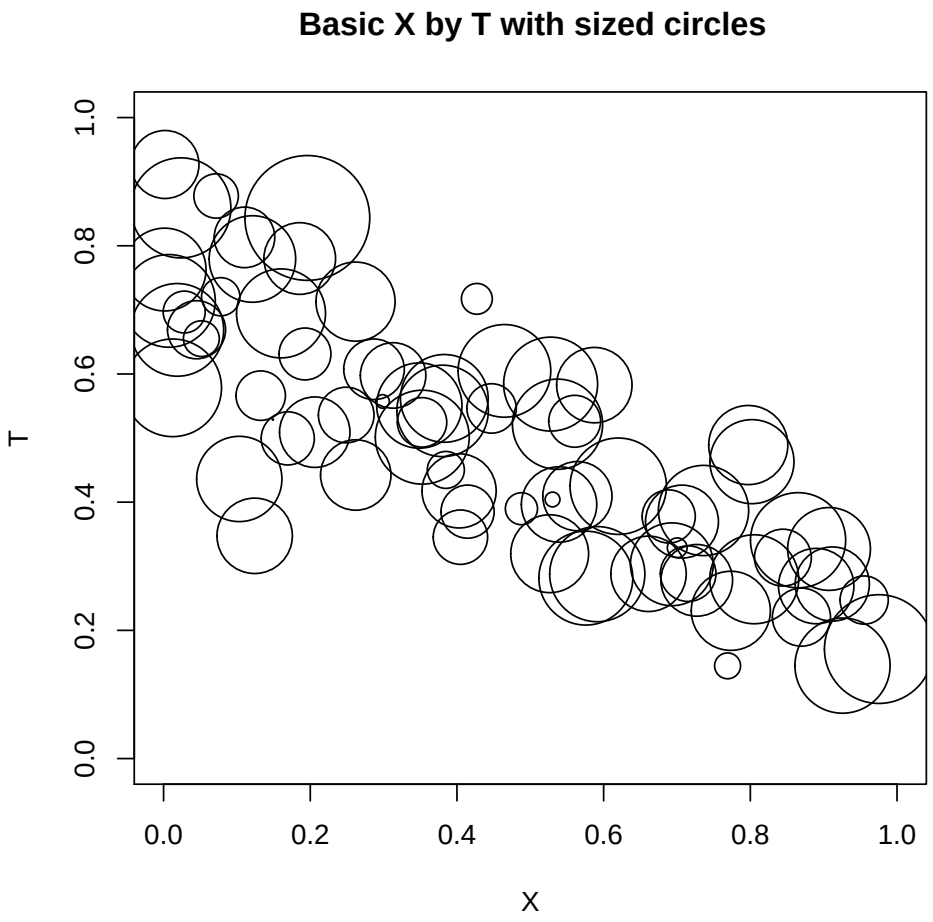

Figure 6: X by T with circles proportional to N

```
> eigraph(dbuf, "xgraphc")
```

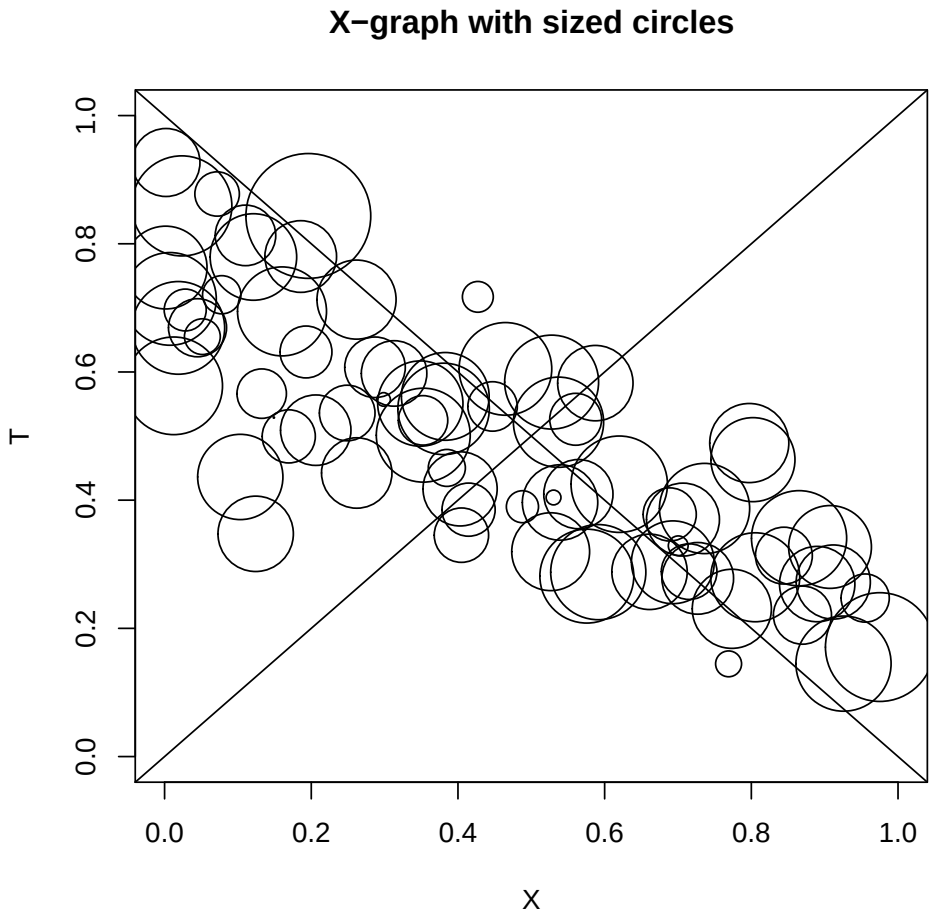

Figure 7: X-graph with data size proportional to N

```
> eigraph(dbuf, "goodman")
```

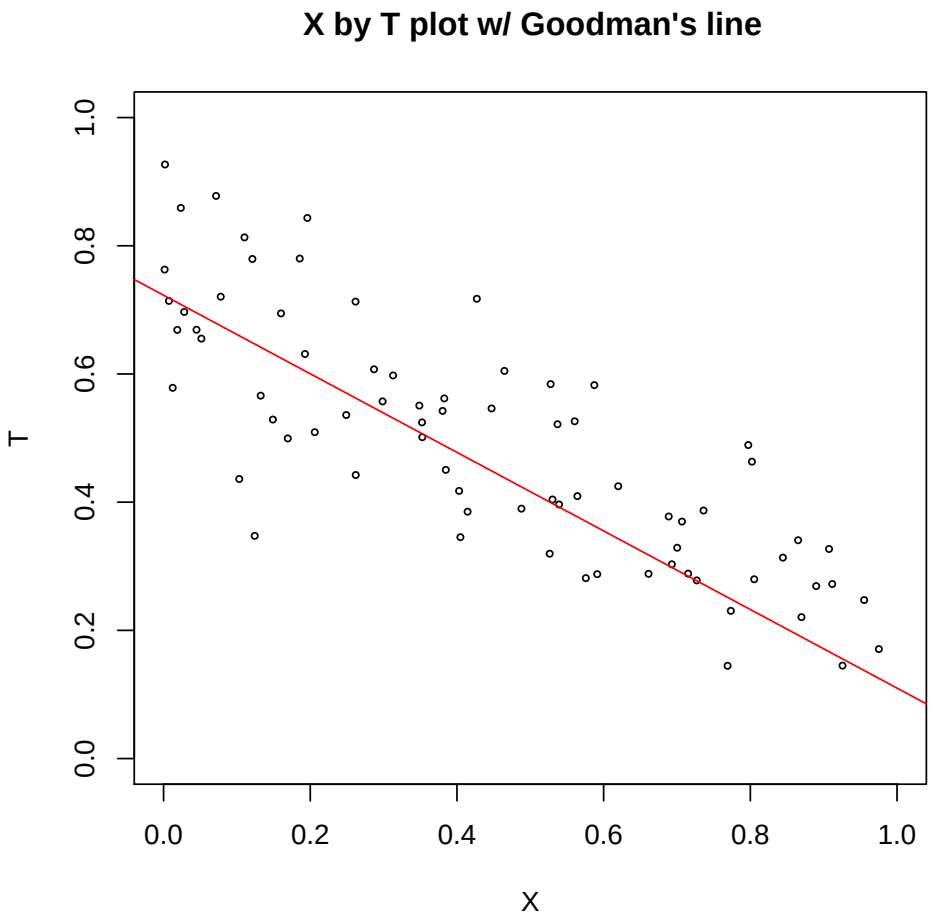

Figure 8: X by T with goodman's regression line plotted

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

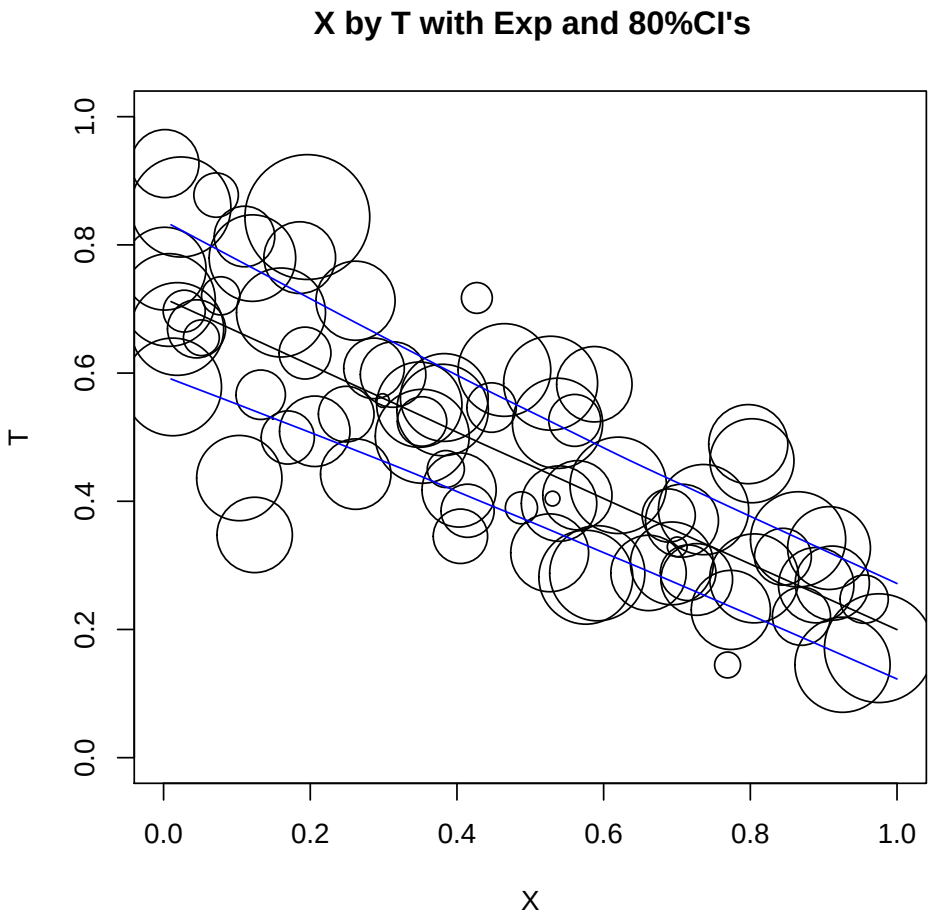

Figure 9: X by T with E(T|X) and cond'l 80% CI

```
> eigraph(dbuf, "xtfitg")
```

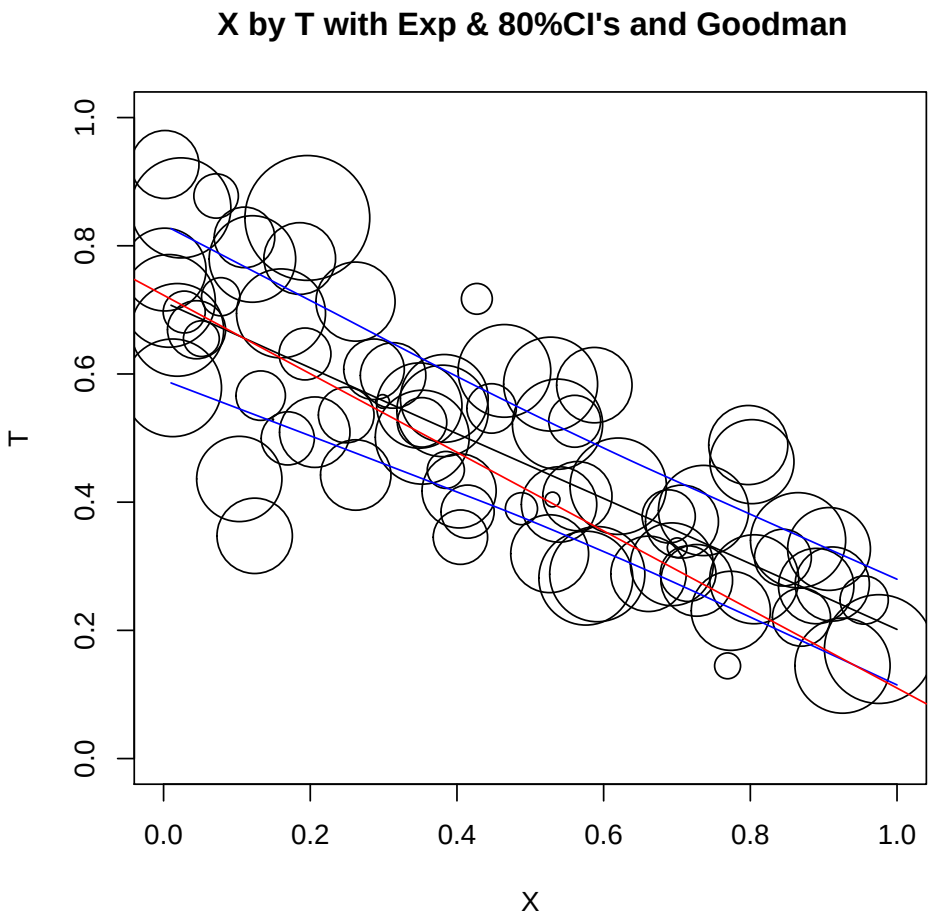

Figure 10: xtfit with goodman's regression

```
> eigraph(dbuf, "profile")
```

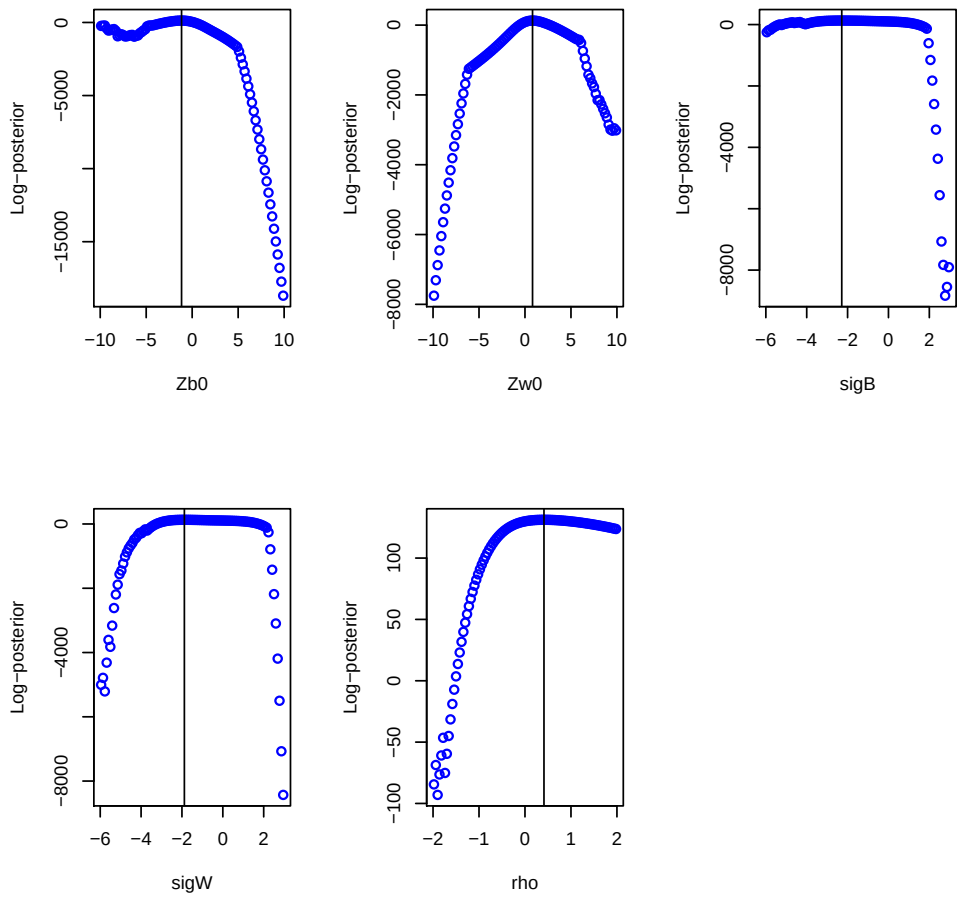

Figure 11: Profile posterior of elements of ϕ

```
> eigraph(dbuf, "profileR")
```

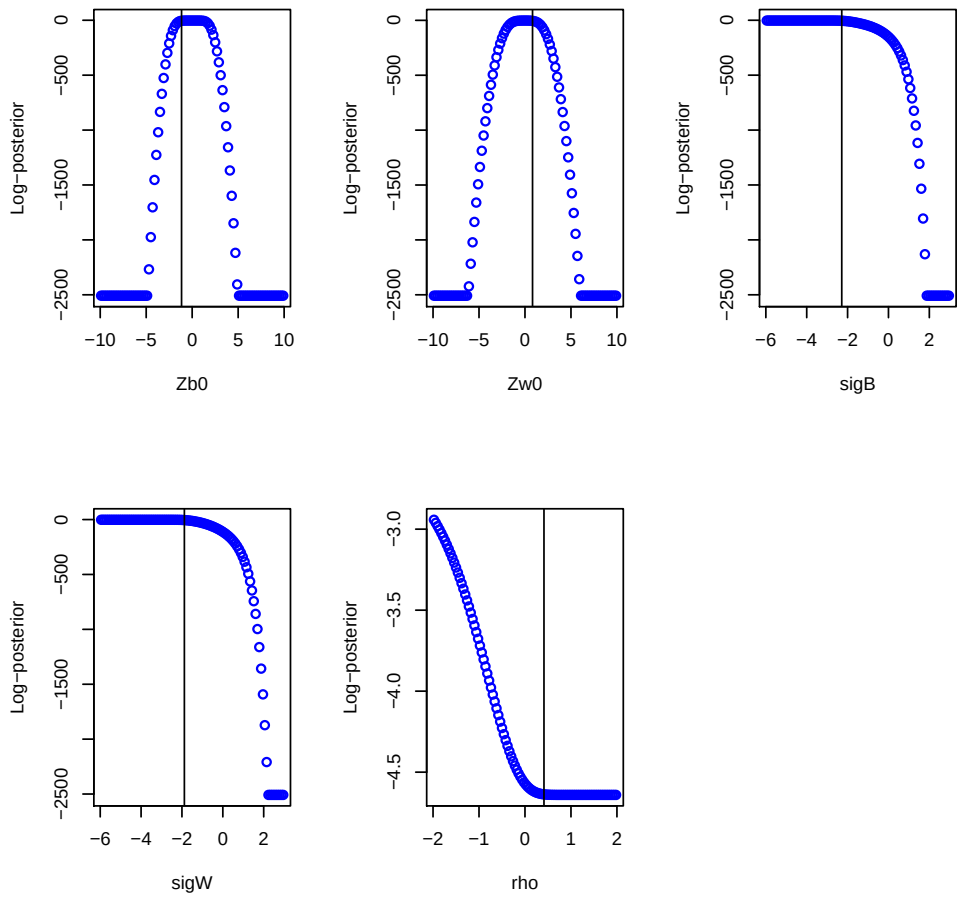

Figure 12: Profile of elements of ϕ

```
> eigraph(dbuf, "post")
```

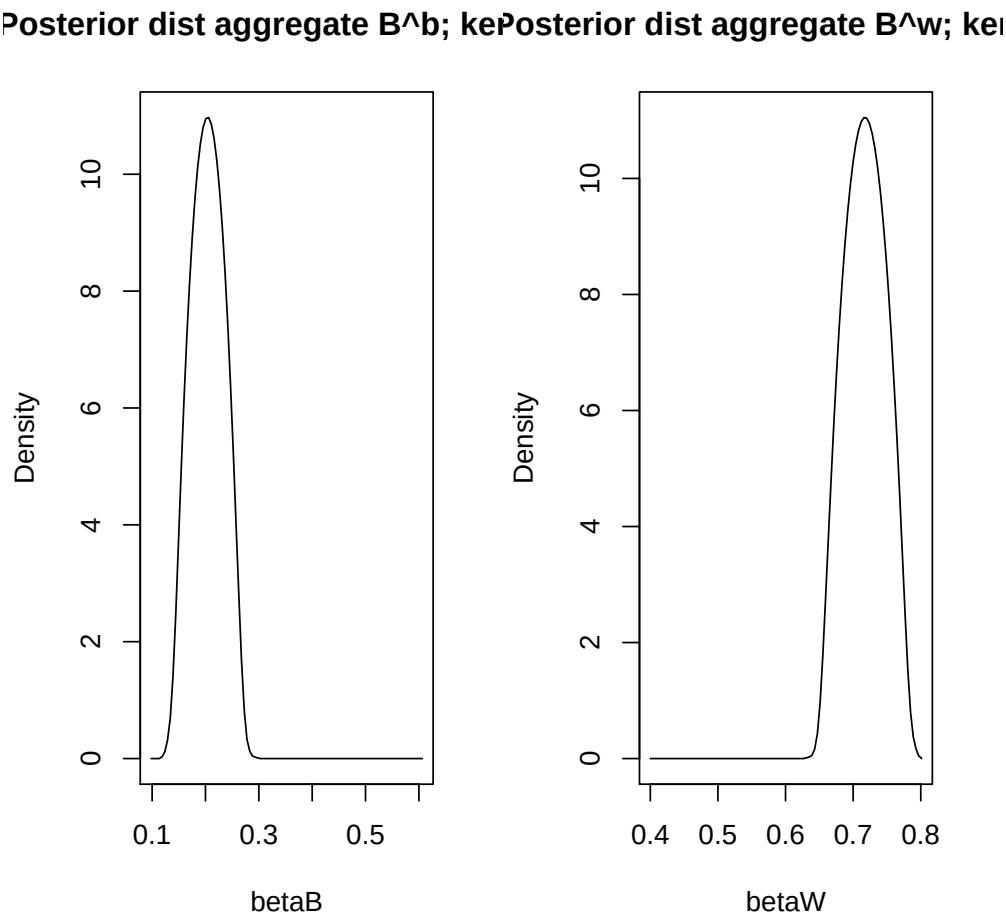

Figure 13: Density estimates of aggregates B^b and B^w

```
> message("Running beta with kern=E")
> eigraph(dbuf, "beta")
```

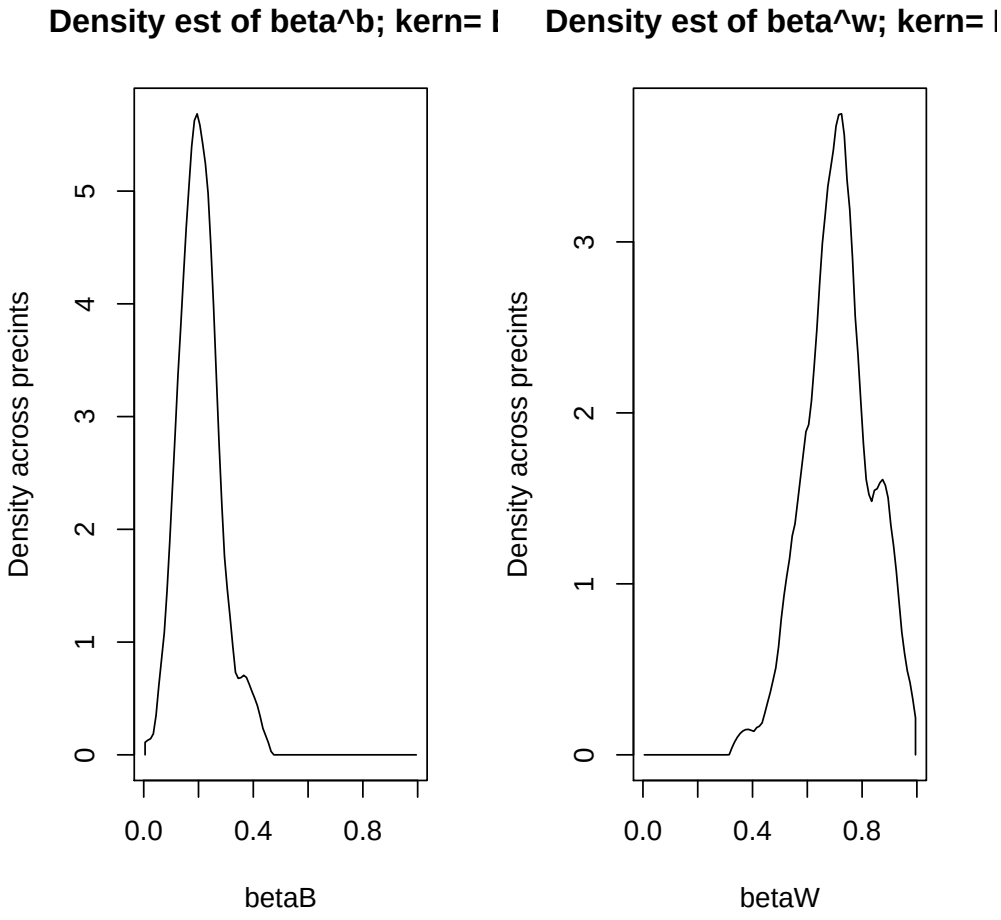

Figure 14: Density estimates of est'd betaB's and betaW's

```
> eigraph(dbuf, "results", kern = "E")
```

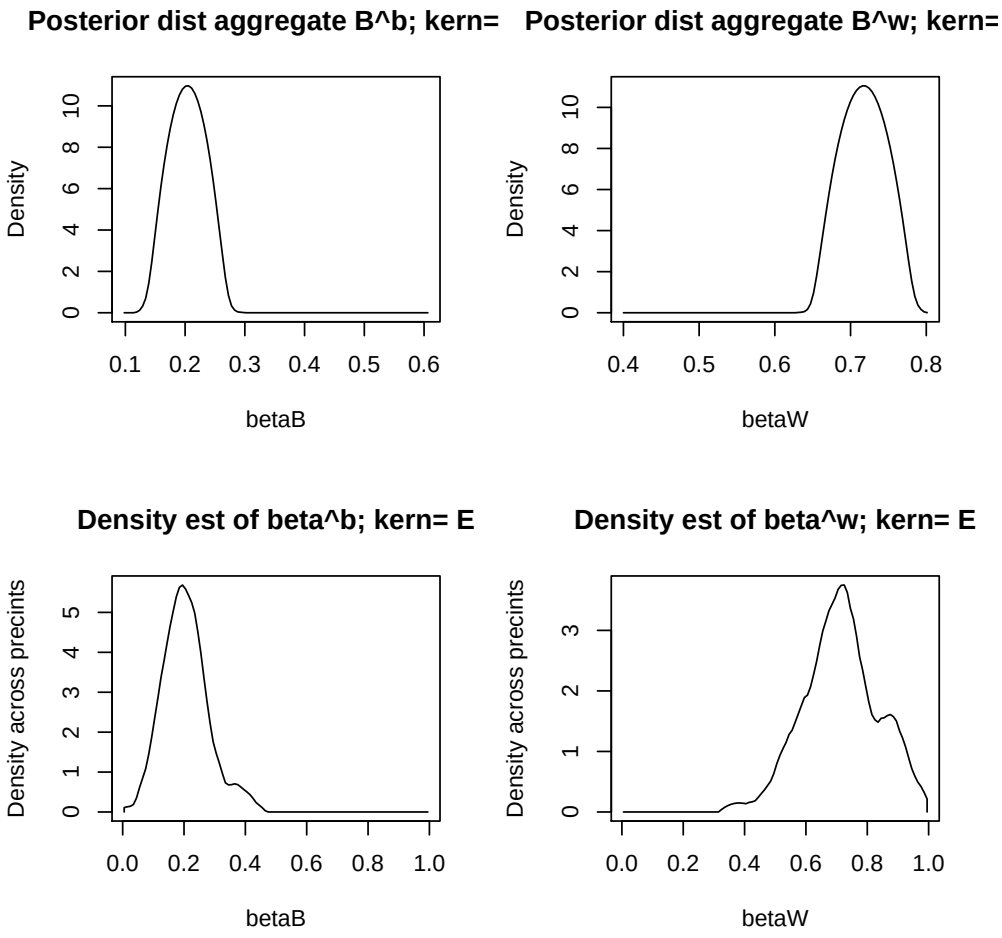

Figure 15: Combination of pos and beta plots

```
> eigraph(dbuf, "lines")
```

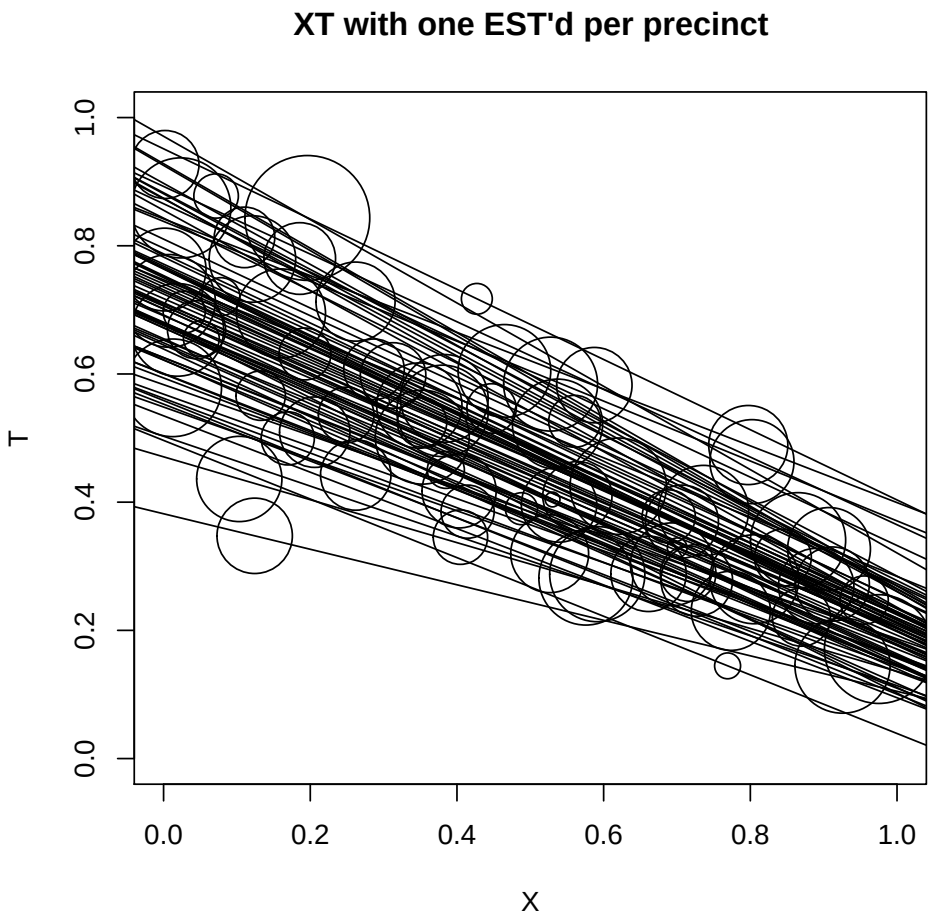

Figure 16: X by T with ESTIMATED line per precinct

```
> eigraph(dbuf, "bivar")
```

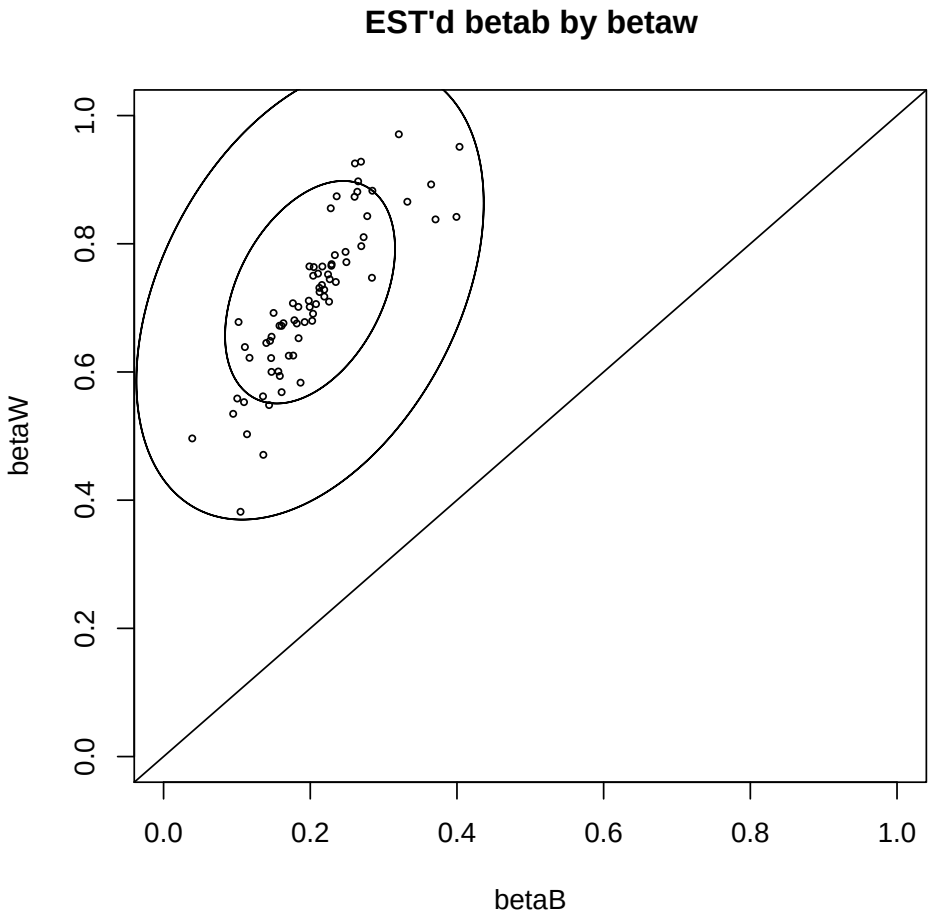

Figure 17: Estimated betaB by betaW

```
> eigraph(dbuf, "biasb")
```

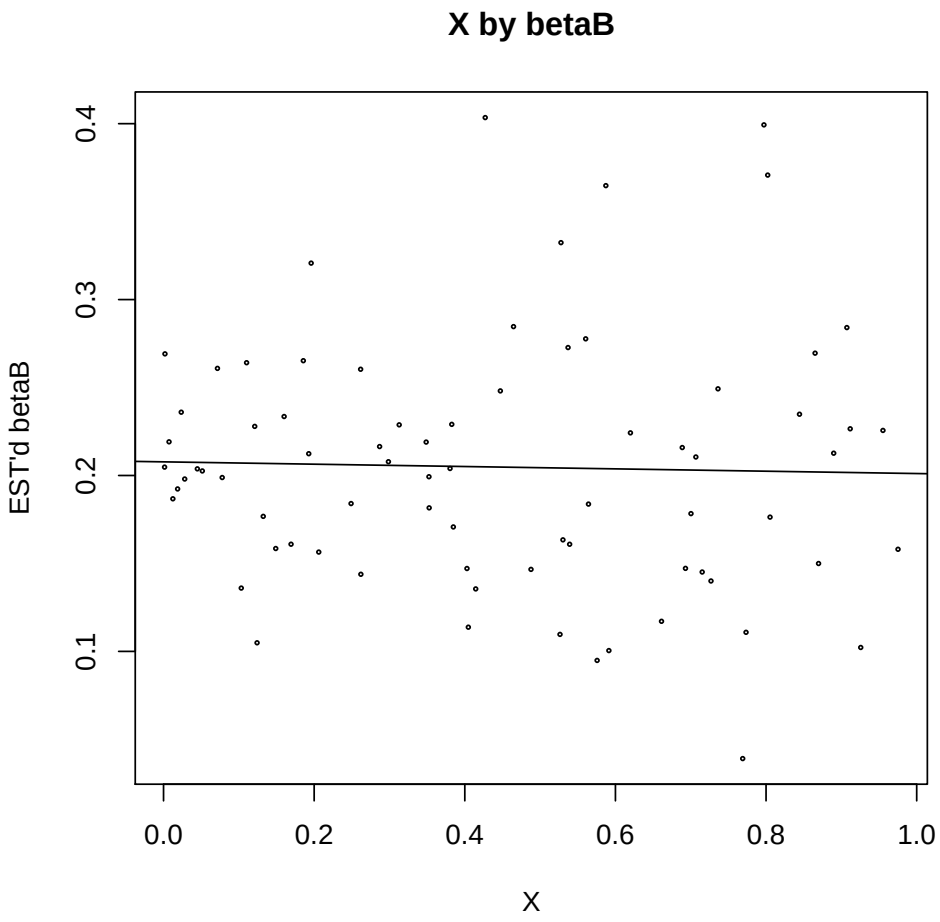

Figure 18: X by EST'd betaB

```
> eigraph(dbuf, "biasw")
```

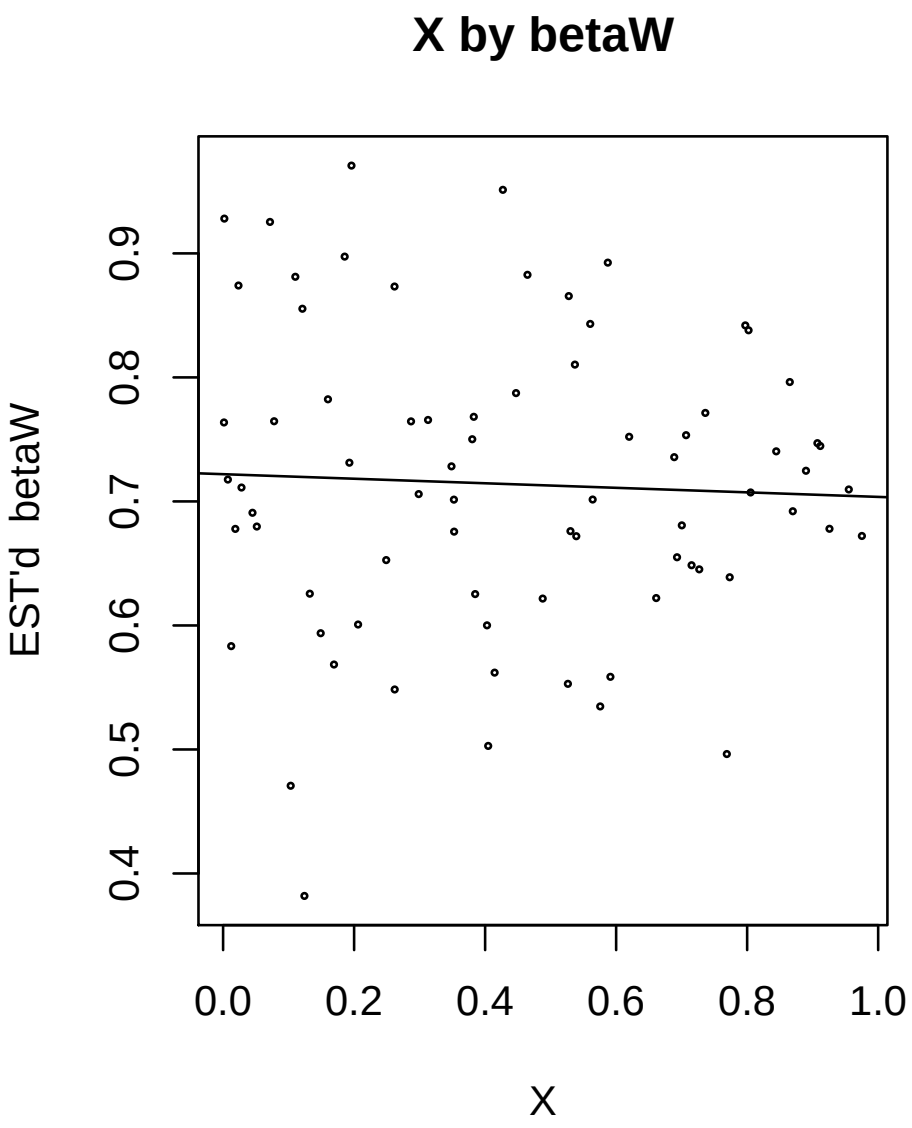

Figure 19: X by EST'd betaW

```
> eigraph(dbuf, "boundx")
```

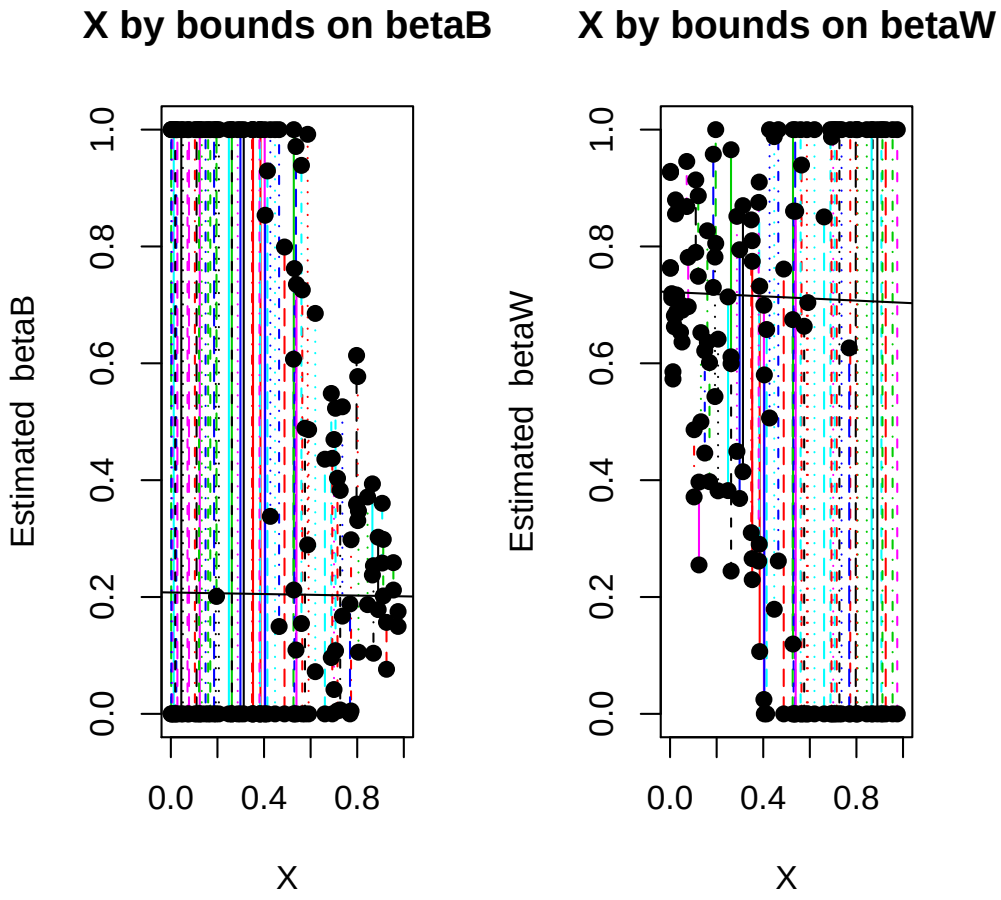

Figure 20: X by bounds on betaB and betaW

```
> eigraph(dbuf, "betast")
```

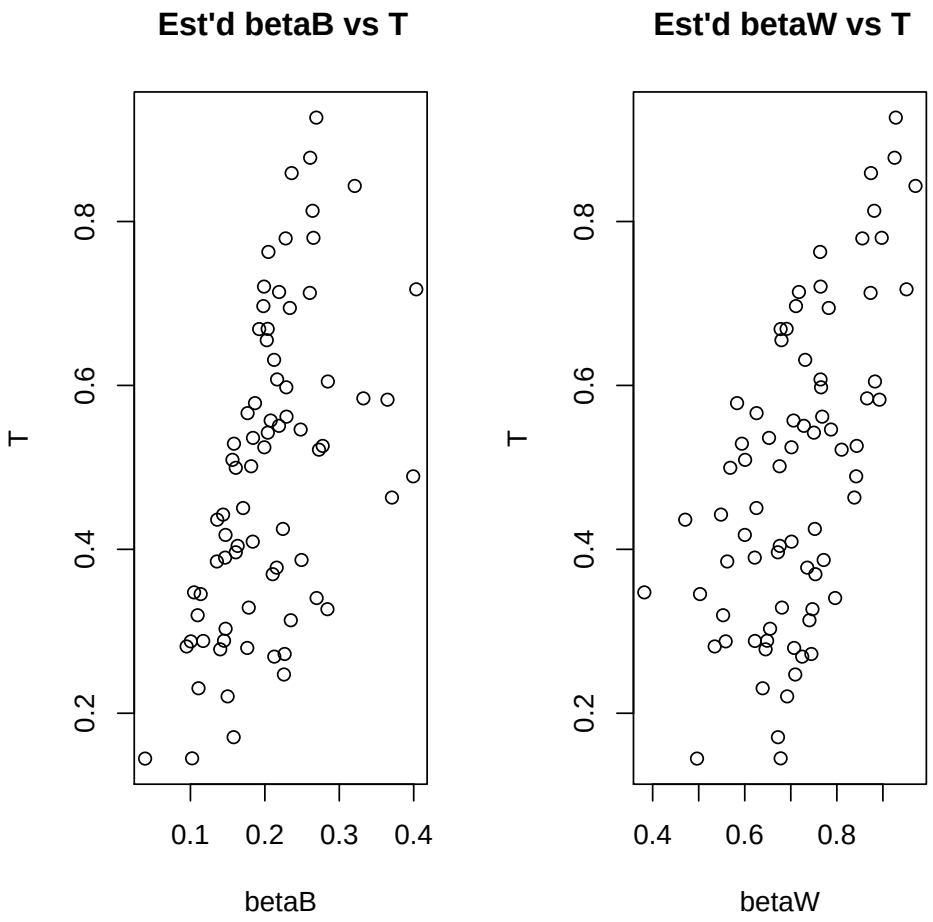

Figure 21: Estimated betaB and betaW vs turnout vote

0.5 ei non-parametric: Estimation for 2x2 tables

It does not require the assumption of truncated bivarate normal. It also provides a diagnostic for truncated bivariate normality, which can also be useful in deciding about the usefulness of the parametric version of the model. It can be run setting EnonPar=1, and it performance is less efficient thatn that of its parametric counterpart. As a matter of fact, we recommend in general to run the parametcric model as for long data sets the non-parametric models can take a long time and the results are not guarantee to be improved. For a detailed description go to `http://gking.harvard.edu/ei`.

0.5.1 ei non-parametric: Demos

There are several examples in the demo directory, they can be run with the commands:

```
> library(ei)
> demo(einp.default)
> demo(einp.fultongen)
> demo(einp.in90)
> demo(einp.kyck88)
> demo(einp.cens1910)
> demo(eip.lavoteall)
> demo(einp.matproII)
> demo(einp.nj)
> demo(einp.pa90)
> demo(einp.scp)
```

The non-parametric model is not efficient with large data sets that have many precincts (> 1000), and we recomend to run the parametric examples instead.

0.5.2 ei non-parametric: Example

```
> if (exists("dbuf")) rm(dbuf)
> if (!exists("sample")) data(sample, package = "ei")
> dbuf <- ei(sample[[1]], sample[[2]], sample[[3]], 1, 1, EnonPar = 1)

> message("Obtaining overall beta's and std errors")
> berr <- summary(dbuf, "paggs")
> berr

> message("Running graphics:")
```

```
> eigraph(dbuf, "tomoge")
```

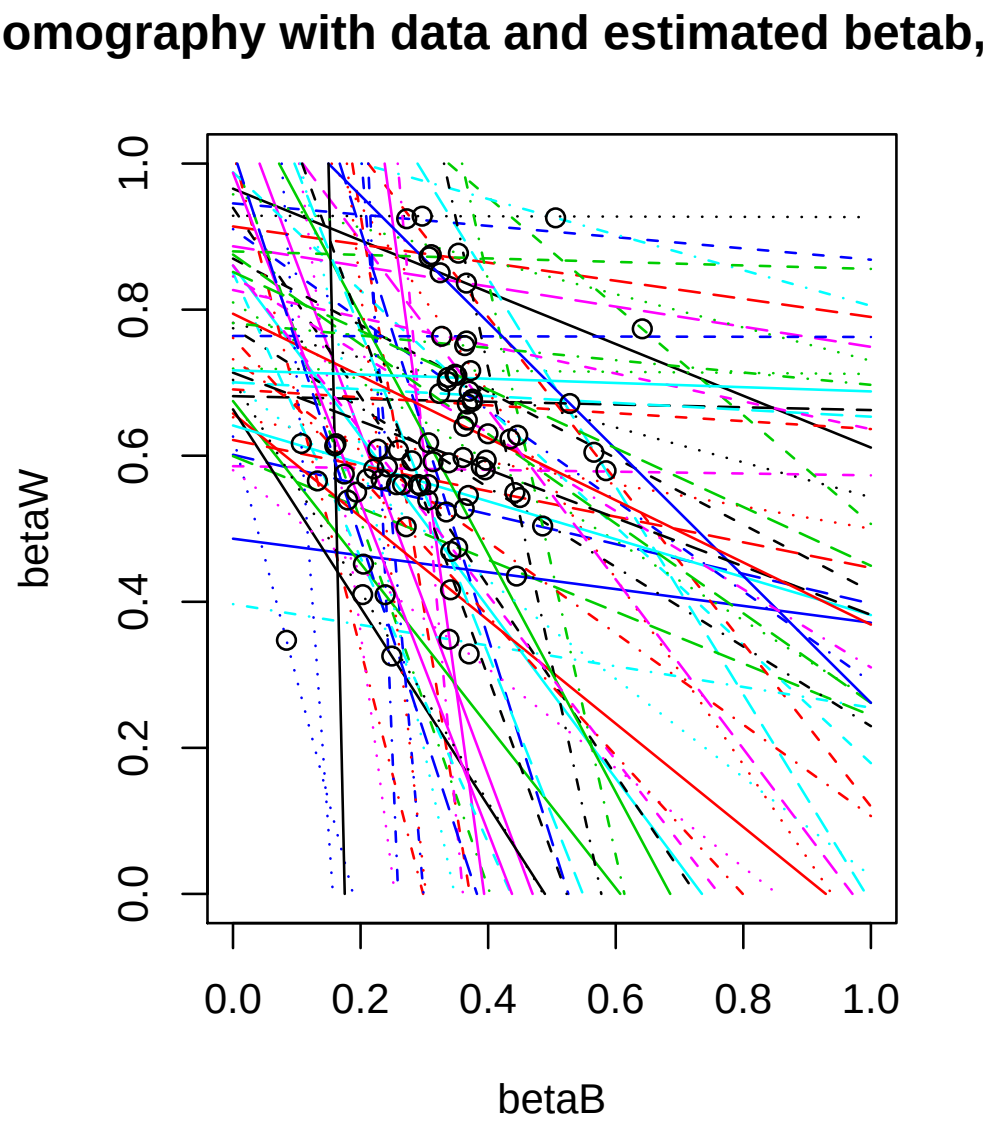

Figure 22: Non-parametric tomography with estimated beta's

```
> eigraph(dbuf, "tomogci")
```

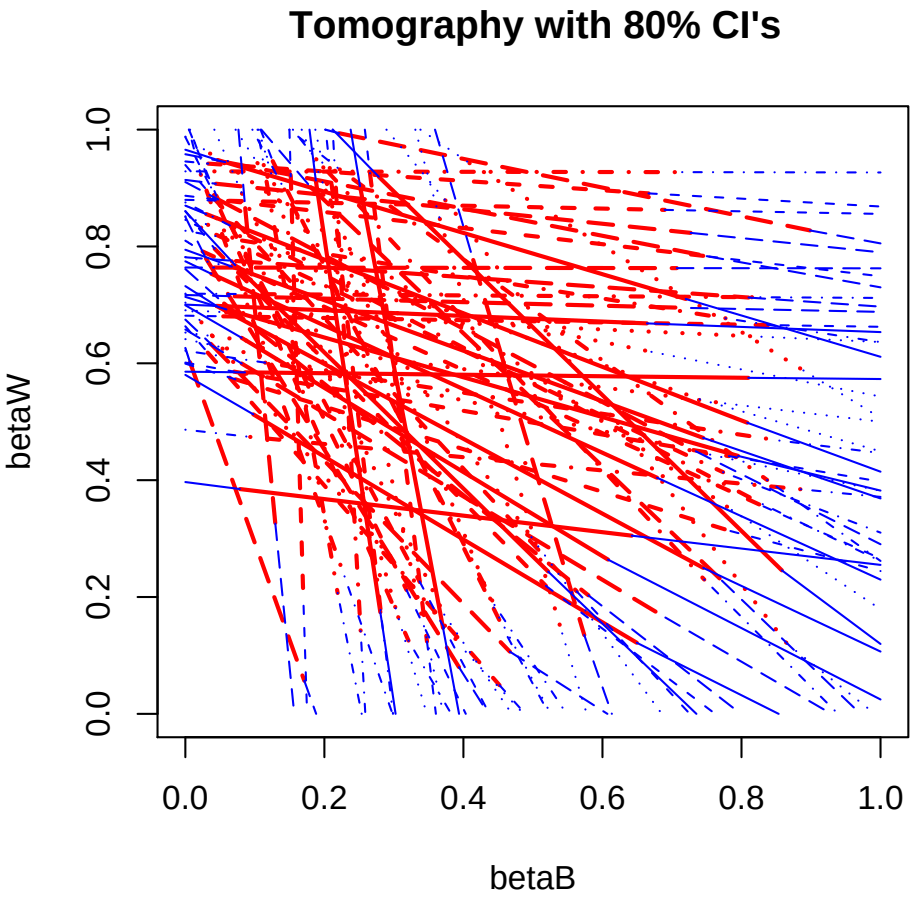

Figure 23: Non-parametric tomography with 80% CI

```
> eigraph(dbuf, "tomogci95")
```

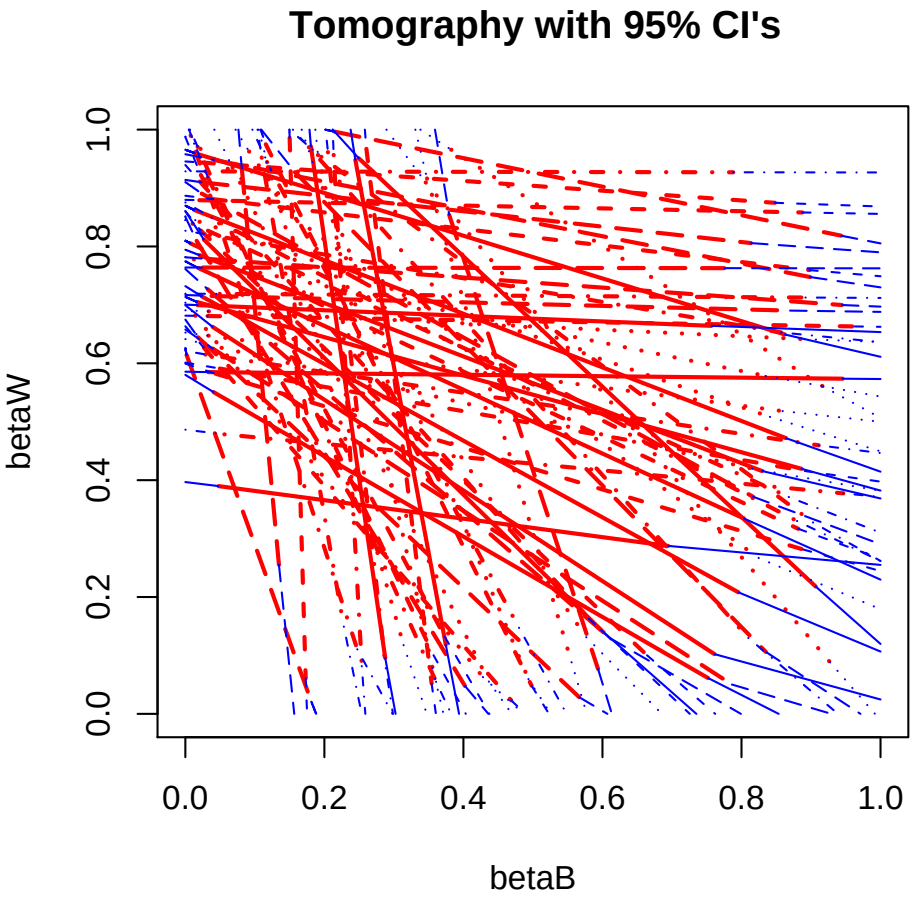

Figure 24: Non-parametric tomography with 95% CI

```
> eigraph(dbuf, "estsims")
```

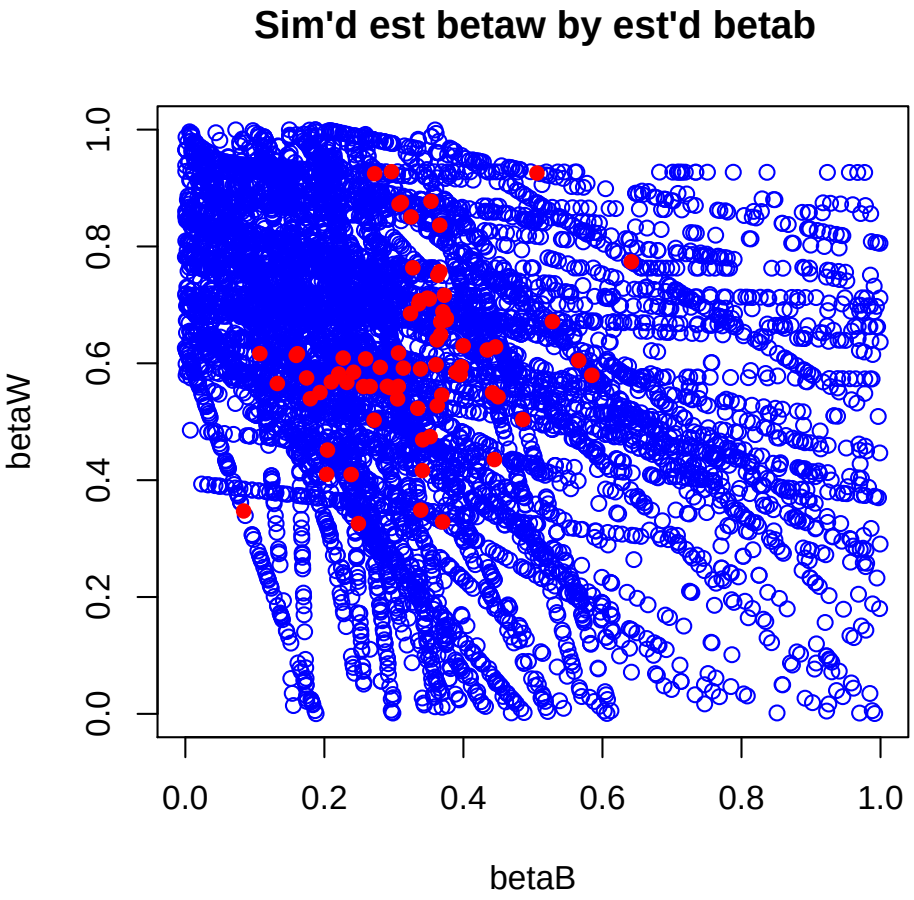

Figure 25: Non-parametric simulated betaB's by betaW's

```
> plot(dbuf)
```

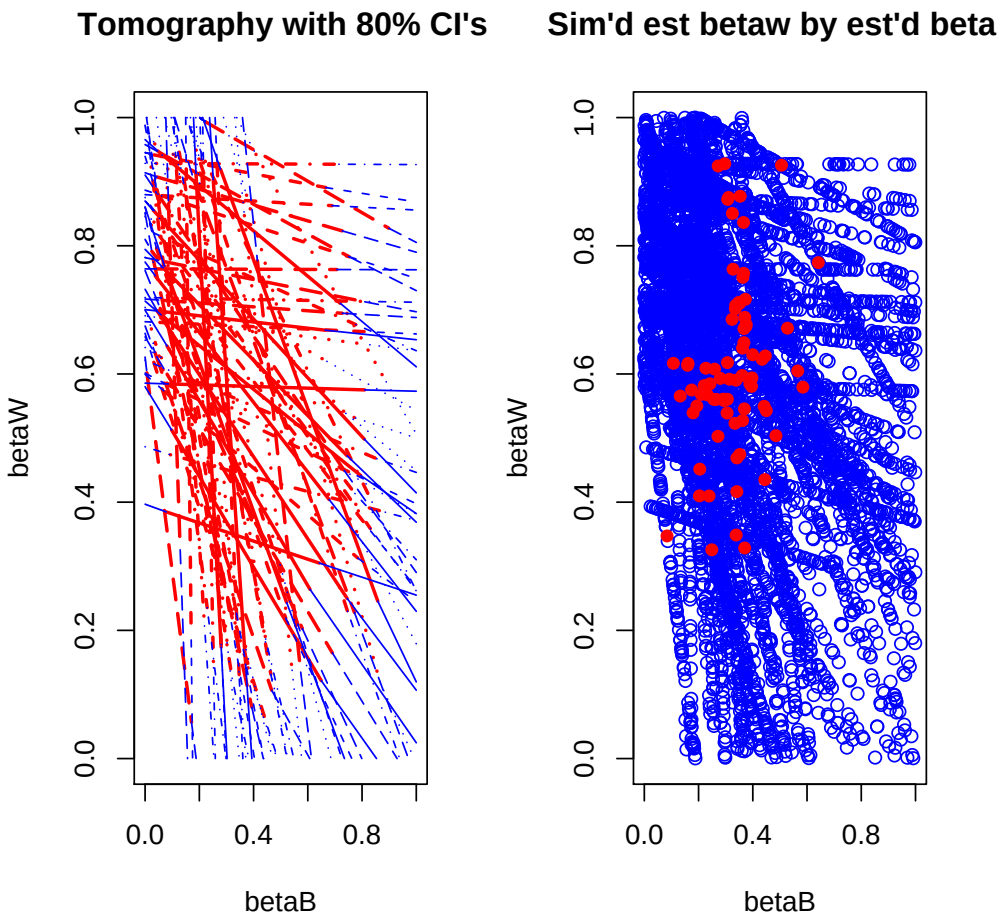

Figure 26: Non-parametric tomography plots

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

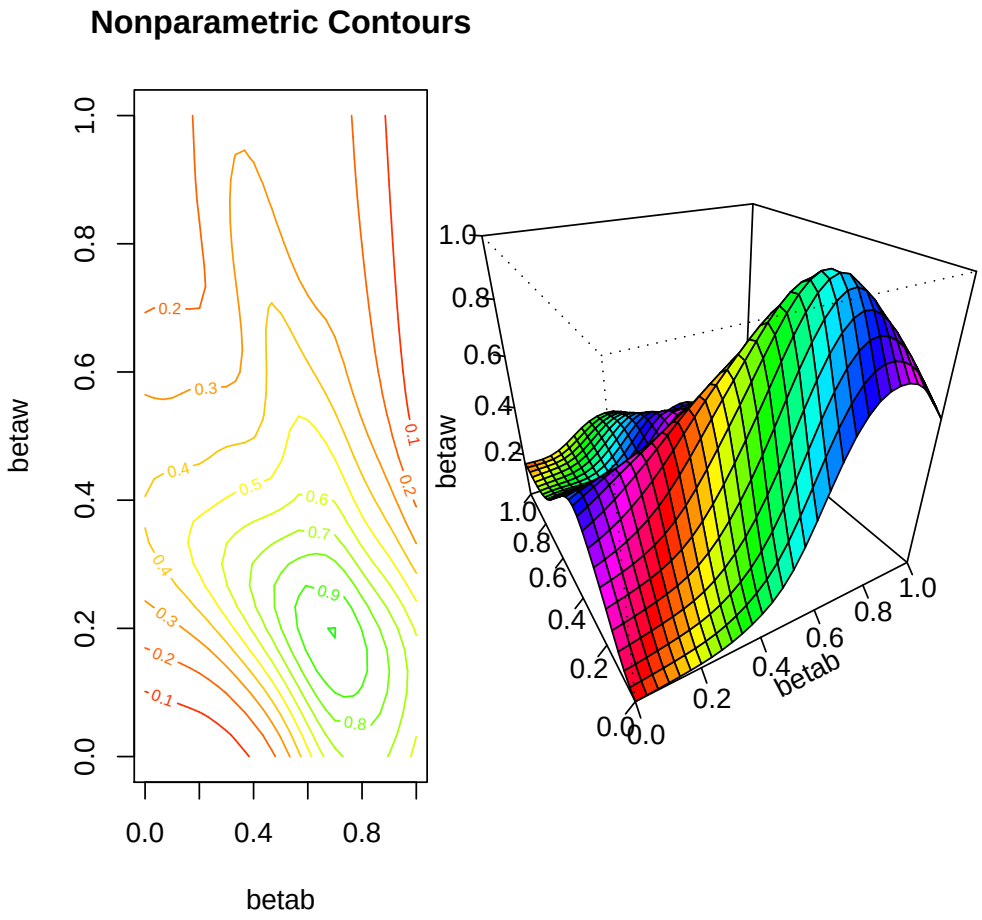

Figure 27: Nonparametric surface plot

```
> eigraph(dbuf, "post")
```

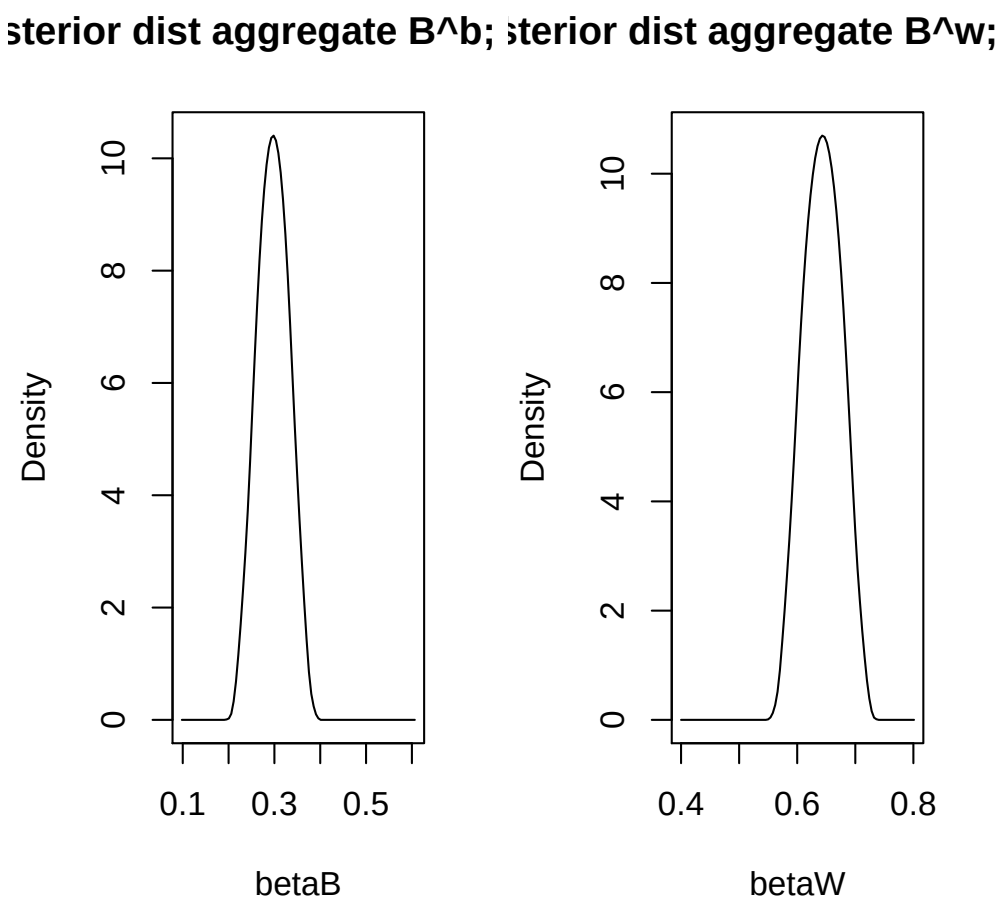

Figure 28: Non-parametric density estimates B^b and B^w

```
> message("Running beta with kern=E")
> eigraph(dbuf, "beta", kern = "E")
```

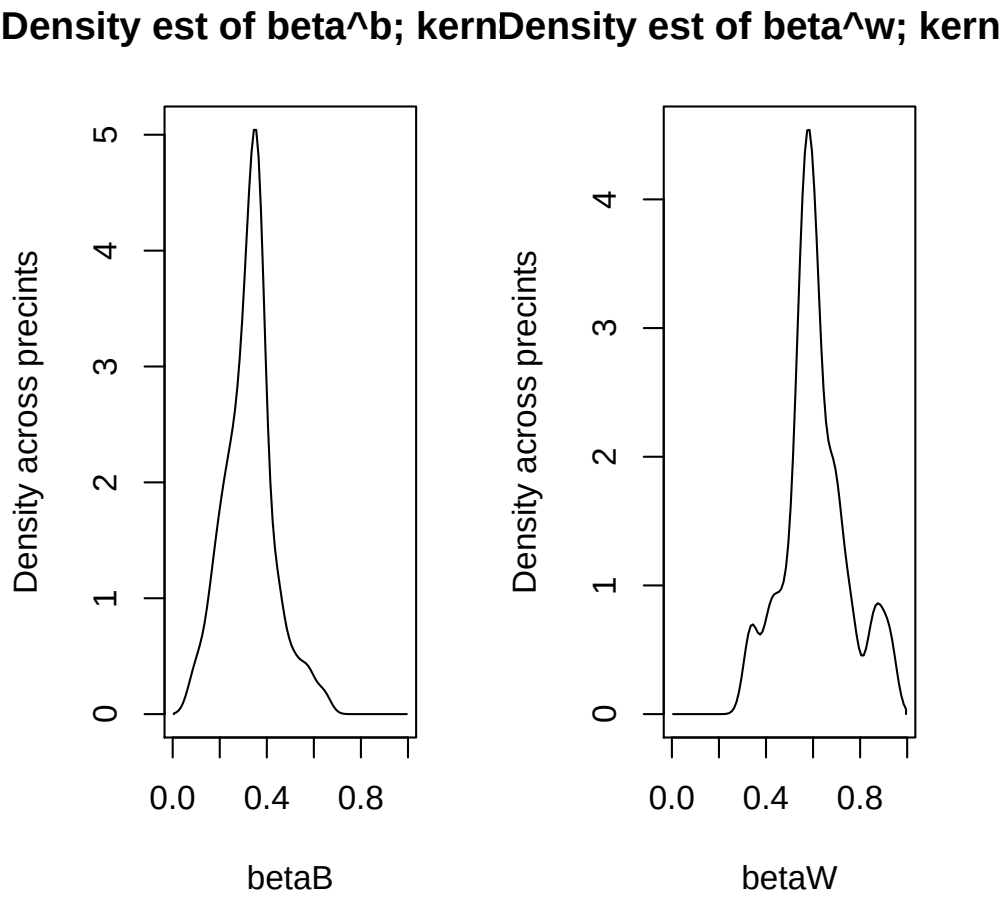

Figure 29: Non-parametric density estimates of est'd betaB's and betaW's

```
> message("Running beta with kern=TN")
> eigraph(dbuf, "beta", kern = "TN")
```

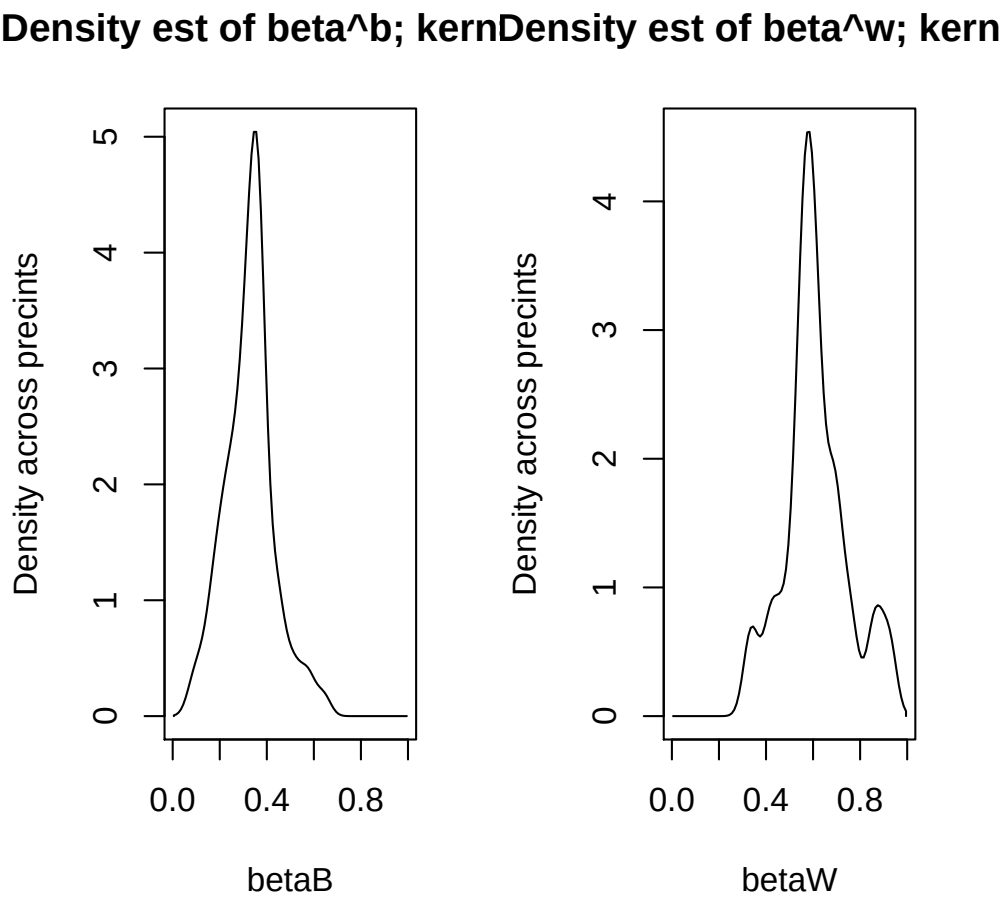

Figure 30: Non-parametric density of est'd betaB's and betaW's

```
> eigraph(dbuf, "results", kern = "E")
```

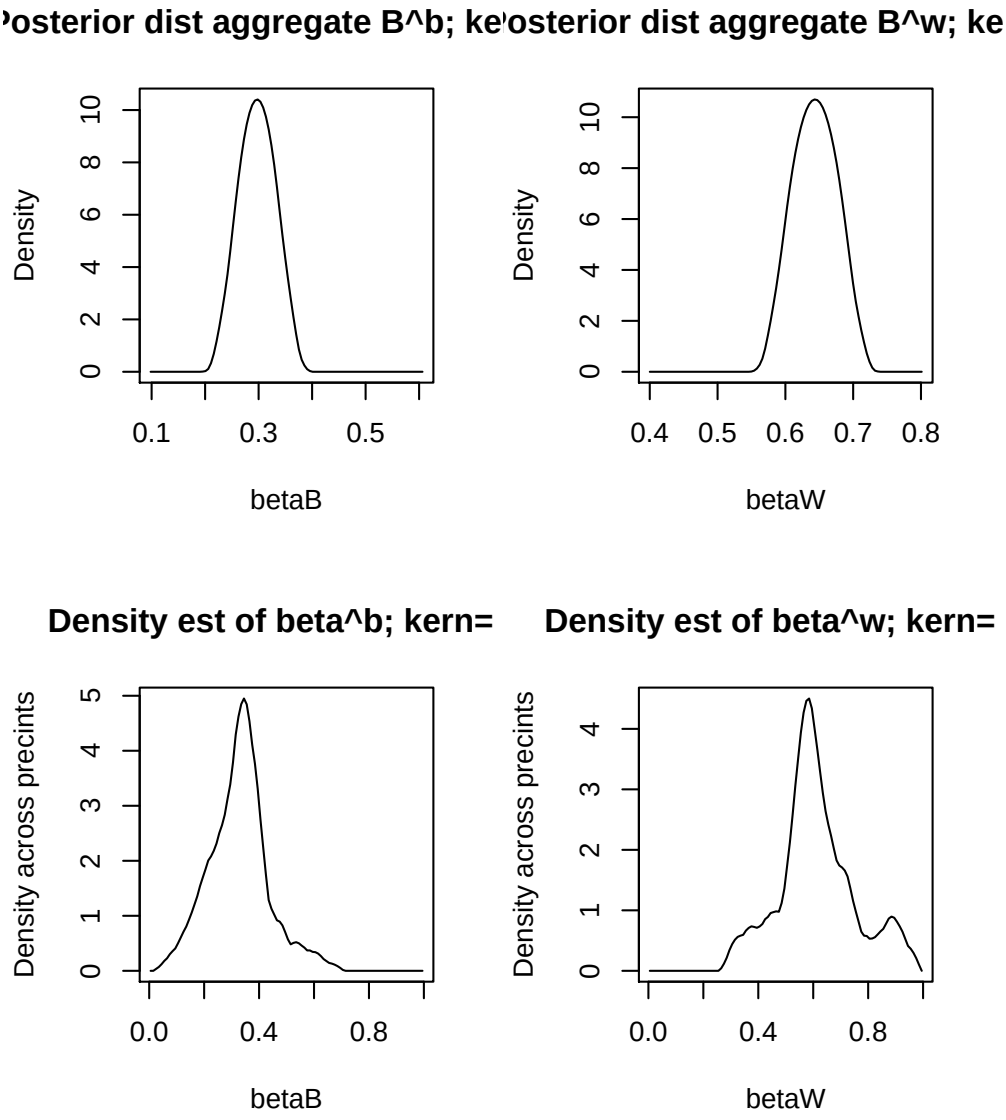

Figure 31: Non-parametric combination of pos and beta plots

```
> eigraph(dbuf, "bivar")
```

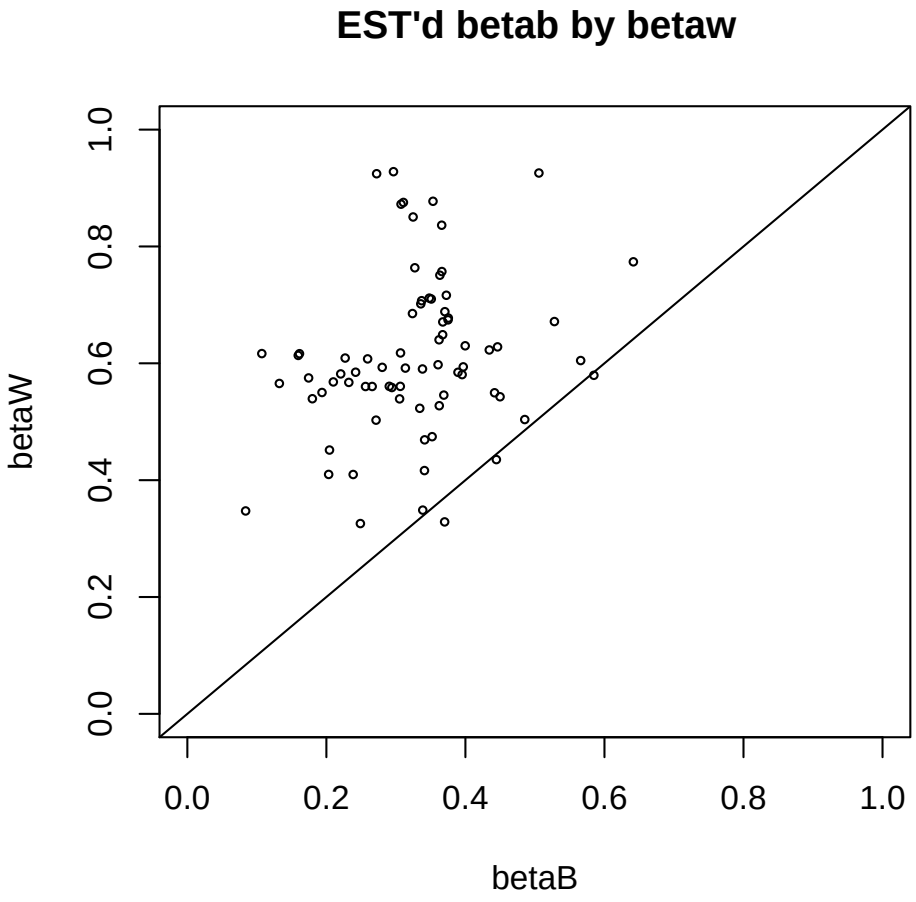

Figure 32: Non-parametric estimated betab by betaw

```
> eigraph(dbuf, "biasb")
```

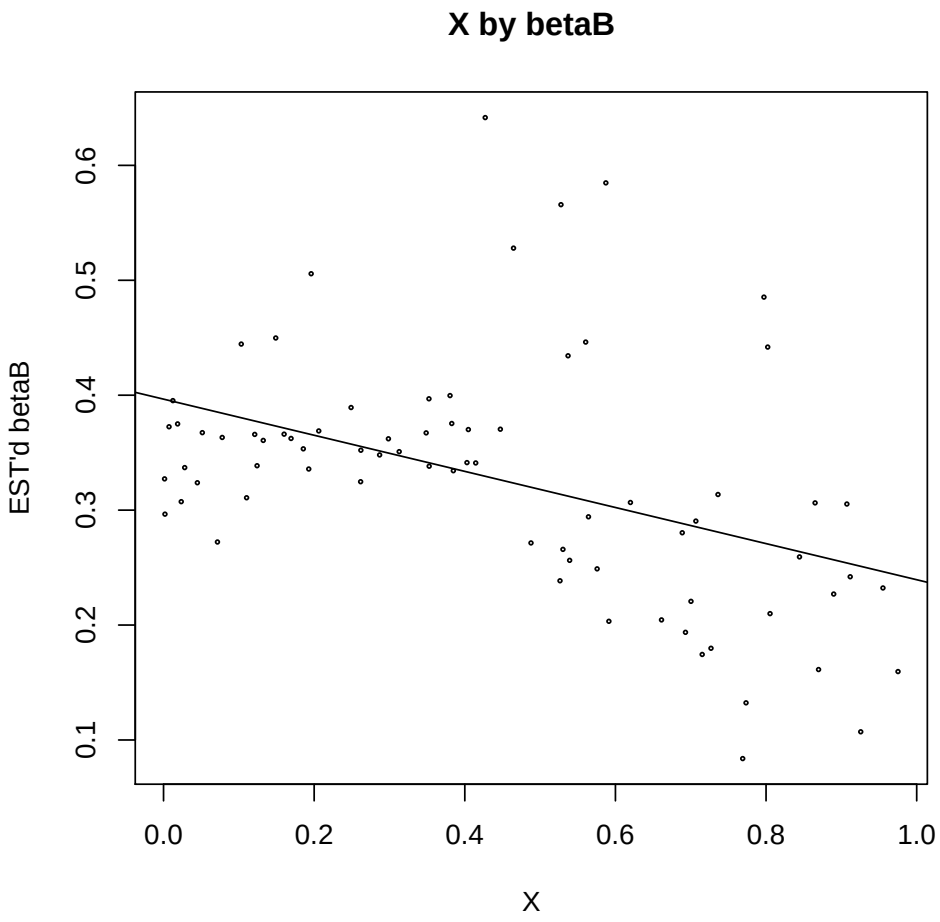

Figure 33: X by EST'd betaB

```
> eigraph(dbuf, "biasw")
```

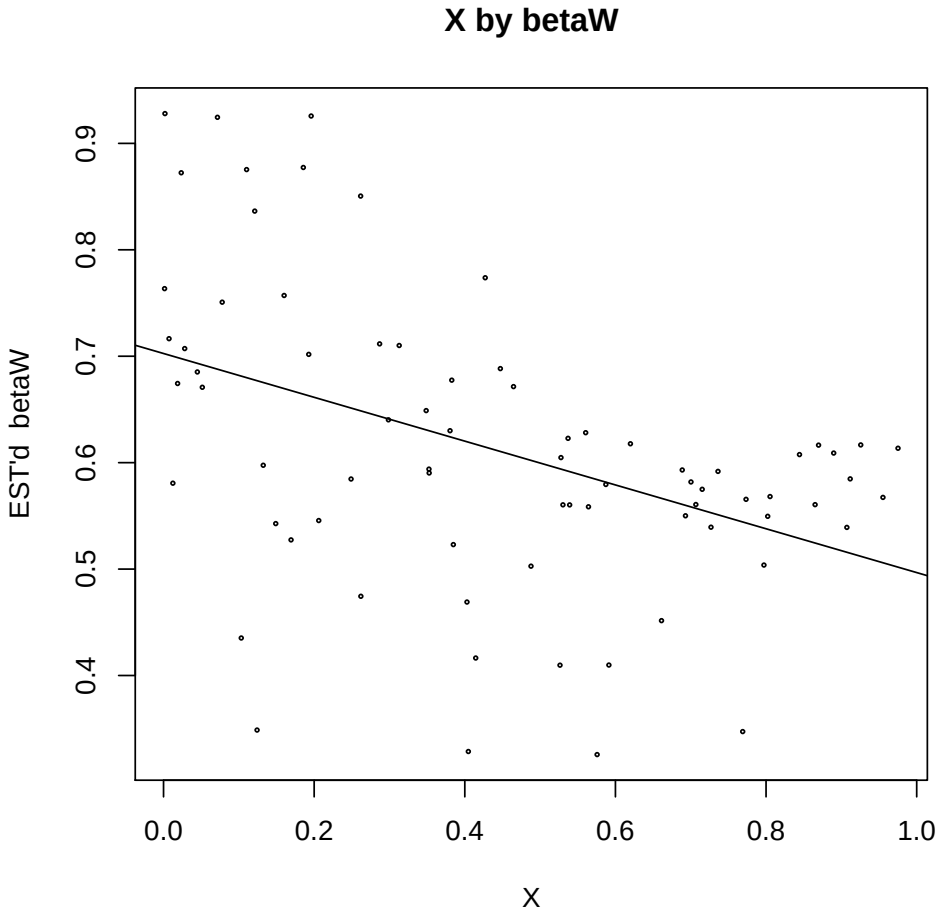

Figure 34: X by EST'd betaW

```
> eigraph(dbuf, "boundx")
```

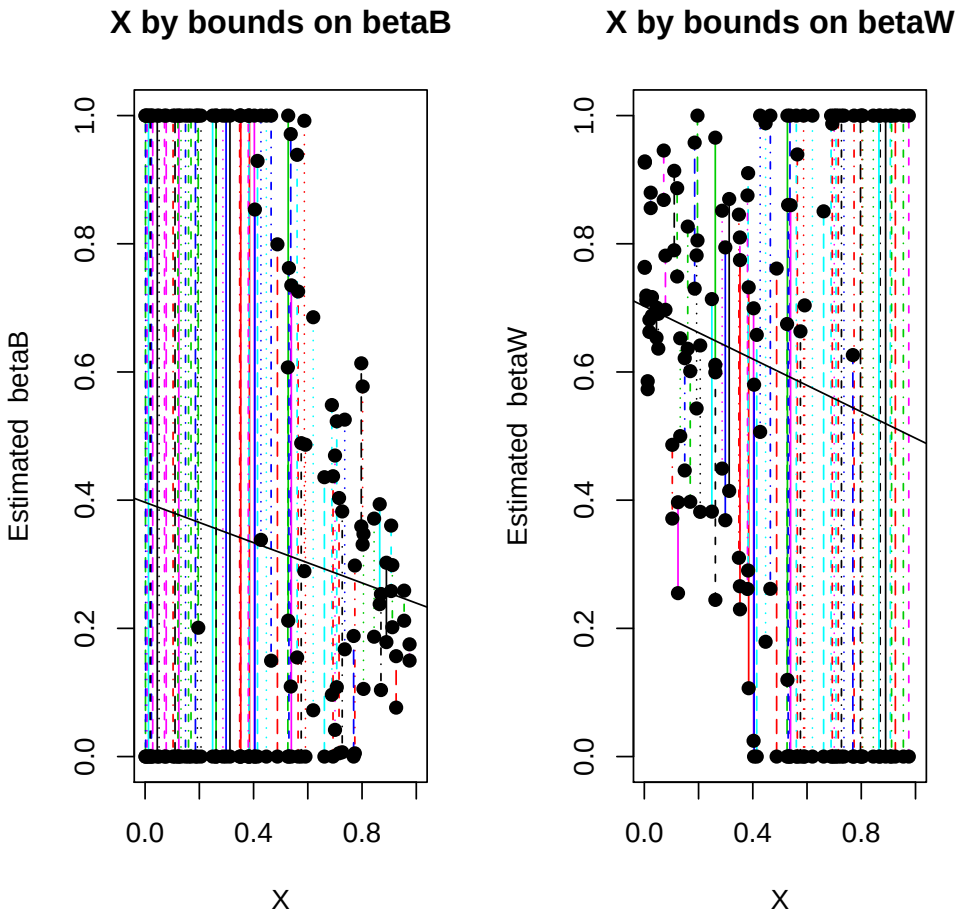

Figure 35: X by bounds on betaB and betaW

```
> eigraph(dbuf, "betast")
```

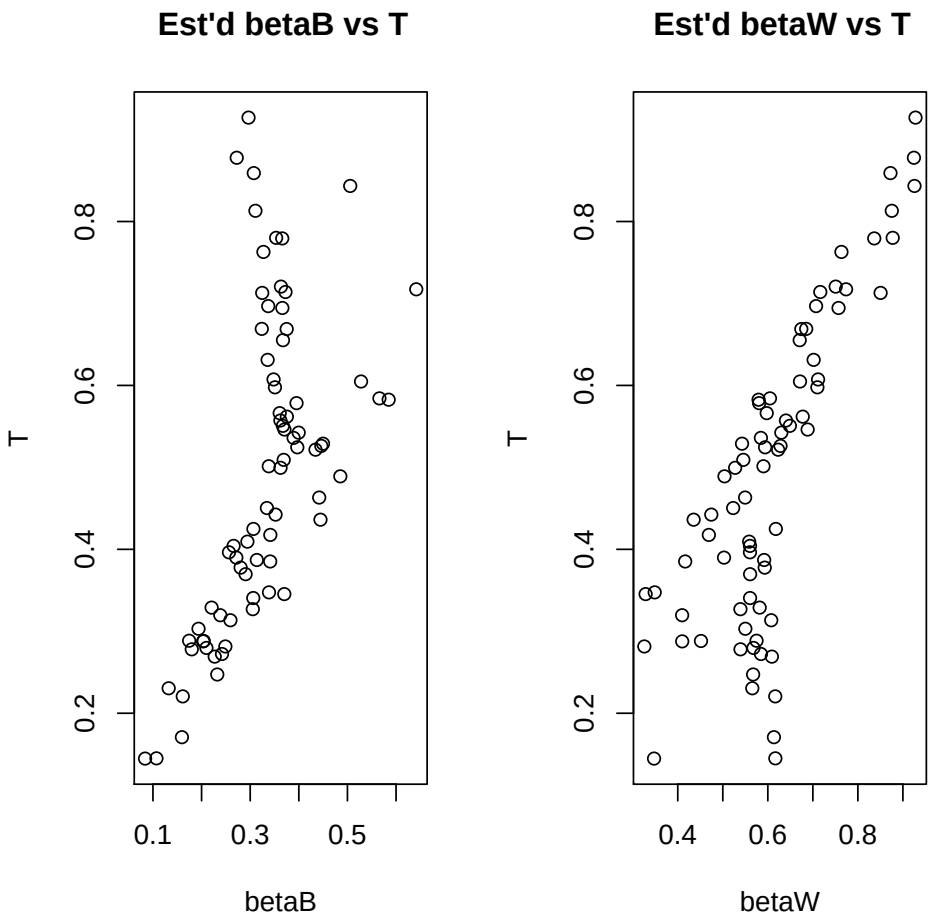

Figure 36: Non-parametric est'd betaB and betaW vs turnout vote

Bibliography

[1] Gary King, *A solution to the Echological Inference Problem: Reconstructing Individual Behaviour from Aggregate Data*, Princeton University Press (1997).

[2] Useful documentation also avalaible 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).