HB-Reg v4

Software for Hierarchical Bayes Regression

(Updated March 26, 2013)

Sawtooth Software, Inc.
Orem, UT
http://www.sawtoothsoftware.com

Bryan Orme, Editor

About Technical Support

We've designed this manual to teach you how to use our software and to serve as a reference to answer your questions. If you still have questions after consulting the manual, we offer telephone support.

When you call us, please be at your computer and have at hand any instructions or files associated with your problem, or a description of the sequence of keystrokes or events that led to your problem. This way, we can attempt to duplicate your problem and quickly arrive at a solution.

For customer support, contact our Orem, Utah office at 801/477-4700, email: support@sawtoothsoftware.com.

Table of Contents

Foreword

Quick Start Instructions

Some Details Regarding HB Estimation

Preparing Data Files for HB-Reg

Coding Independent Variables

Building Models (Specify Models Tab)

Analyzing Data Using HB-Reg

How Good Are the Results?

Appendix: Technical Details

References

1 Foreword

1.1 Acknowledgements

In producing this software we have been helped by several sources listed in the References. We have benefited particularly from the materials provided by Professor Greg Allenby in connection with his tutorials at the American Marketing Association's Advanced Research Techniques Forum. For many of the technical improvements starting in version 3, we have benefited from direction provided by Peter Lenk, of the University of Michigan.

1.2 What's New in HB-Reg v4

1. Version 4 features a new user interface, borrowed extensively from the Sawtooth Software's MBC (Menu-Based Choice) software product.

2. HB-Reg 4 can automatically code Categorical Independent variables as dummy-coded, linear, or log-linear. This removes extra data processing requirements, making it easy for you to switch back and forth between different independent variable coding schemes.

3. HB-Reg 4 automatically builds a "proper prior covariance matrix" when you specify a categorical independent variable (when dummy-coding is activated). This is the same procedure as recommended by leading Bayesian academic Peter Lenk that is also used in our CBC/HB software.

4. Covariates may now be included in HB runs. For more information on covariates in HB, please see the white paper entitled: Application of Covariates within Sawtooth Software's CBC/HB Program: Theory and Practical Example (2009). This paper may be downloaded from our Technical Papers library on our website: www.sawtoothsoftware.com.

5. Ability to run multiple HB runs simultaneously, utilizing multiple cores of your machine's processor.

6. Big gains in speed for exceptionally large datasets (>128MB) due to an improved memory caching technique. We used an enormous HB-Reg dataset for testing (250MB) that experienced a 20x increase in speed under HB-Reg v4 compared to v3.

1.3 Introduction

In the analysis of marketing research data, there are many occasions when the researcher has a sample of respondents, stores, or other experimental units and wishes to estimate separate regression coefficients for each unit.

Consider three examples:

> 1. In full-profile conjoint analysis, survey respondents give preference ratings for hypothetical product concepts. Regression analysis is often used, where the independent variables are columns of a "design matrix" describing the concepts and the dependent variable consists of preference ratings. The researcher often wants to estimate regression coefficients that will be interpreted as "part-worths." Because respondents are expected to differ in their values, the researcher wants to produce estimates for each respondent individually.
>
> 2. Survey respondents in a customer satisfaction study provide ratings of several companies. Some ratings are on "explanatory" variables, such as customer service, product durability, convenience of use, etc. Other ratings are more general, such as overall satisfaction with the companies' products. One goal of the study is to infer the relative importance of each explanatory factor in determining overall satisfaction. Because respondents may differ, the researcher wants to produce estimates for each respondent individually.
>
> 3. During a pricing experiment in grocery stores, the prices of several products are varied systematically in different time periods and sales of each product are measured with scanner data. The independent variables are product prices and other factors such as the presence of displays, coupons, and other promotions. The dependent variables are product sales. Because it is believed that customers of different stores may behave differently, the researcher wants to estimate price effects and cross-effects for each store individually.

In each situation, separate regression estimates are desired for each individual (respondent or store). However, in each case there is likely to be a degrees-of-freedom problem, with many parameters to be estimated for each individual, but relatively few observations per individual.

In the past, researchers have often tried to handle this problem by ignoring heterogeneity among individuals, pooling all the data, and estimating a single set of regression coefficients that describe the "average" individual. However, an alternative solution has recently become available to marketing researchers with the introduction of "hierarchical Bayes" (HB) methods. Several articles (see for example Lenk, *et al*. 1996 and Allenby, *et al*. 1998) have shown that hierarchical Bayes estimation can do a creditable job of estimating individual parameters even when there are *more* parameters than observations per individual. This is done by considering each individual to be a sample from a population of similar individuals and "borrowing" information from other individuals in the estimation for each one.

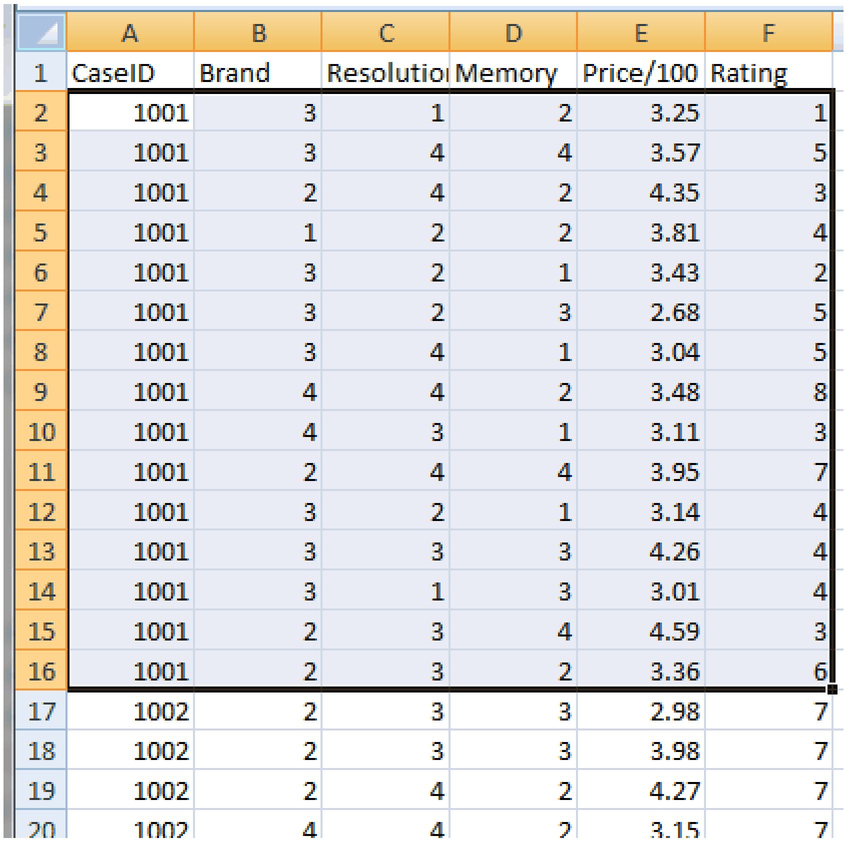
HB-Reg is appropriate for the situations described above. It estimates a hierarchical random coefficients model using a Monte Carlo Markov Chain algorithm. In the material that follows we describe the hierarchical model and the Bayesian estimation process.

Although HB methods are often computationally arduous, our software has the advantage of being written in compiled code (C#). Compiled code programs are usually considerably faster than those written in higher-level languages such as Gauss.

This is one of many HB products that Sawtooth Software has provided. As examples, CBC/HB, ACA/HB, and CVA/HB are specialized applications for use with data generated by Sawtooth Software's CBC, ACA, and CVA conjoint analysis software. Because they are used in narrowly defined contexts, they require relatively little data processing on the part of the user. HB-Reg is more general, being applicable to data from a variety of sources. To permit this generality we must assume that the user is capable of arranging input data in a format acceptable to HB-Reg.

2 Quick Start Instructions

1. Prepare a datafile in .csv (comma-separated values) format using a program like Excel. Each respondent should have multiple observations (rows) in the data file when using HB-Reg. The first row should contain labels.

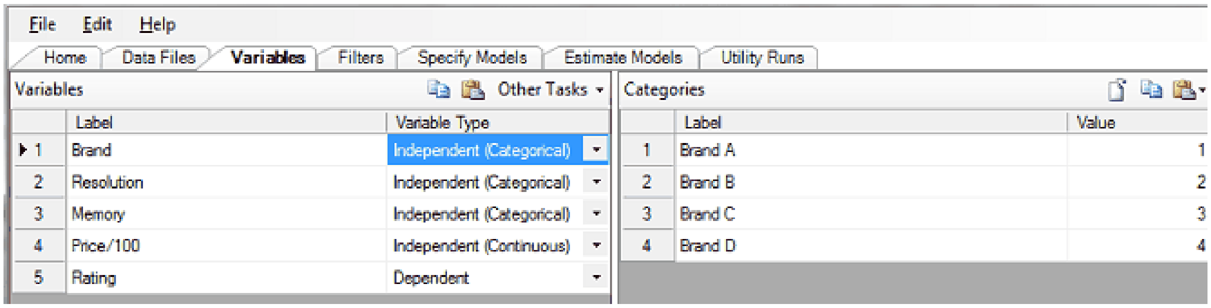


	A	B	C	D	E	F
1	CaseID	Brand	Resolutio	Memory	Price/100	Rating
2	1001	3	1	2	3.25	1
3	1001	3	4	4	3.57	5
4	1001	2	4	2	4.35	3
5	1001	1	2	2	3.81	4
6	1001	3	2	1	3.43	2
7	1001	3	2	3	2.68	5
8	1001	3	4	1	3.04	5
9	1001	4	4	2	3.48	8
10	1001	4	3	1	3.11	3
11	1001	2	4	4	3.95	7
12	1001	3	2	1	3.14	4
13	1001	3	3	3	4.26	4
14	1001	3	1	3	3.01	4
15	1001	2	3	4	4.59	3
16	1001	2	3	2	3.36	6
17	1002	2	3	3	2.98	7
18	1002	2	3	3	3.98	7
19	1002	2	4	2	4.27	7
20	1002	4	4	2	3.15	7

Nominal variables (such as Brand, Color, etc.) should be coded as integers starting with 1. HB-Reg can automatically perform the coding to convert them to dummy-coded independent variables. You may code continuous variables as you like, with decimal places of precision and negative or positive values. HB-Reg will use these columns "as-is" during parameter estimation.

2. When you open the HB-Reg software, you probably will see the *Project Wizard* that allows you to select a data file (if not, click ***File | Open...***). Use the ***Browse...*** button to browse to the .csv data file you prepared in step 1.

3. Click the *Variables* tab, and specify which variable is the *Dependent variable* and which are the *Independent Variables*. You should also specify for each independent variable if it is *Categorical* (nominal) or *Continuous*.

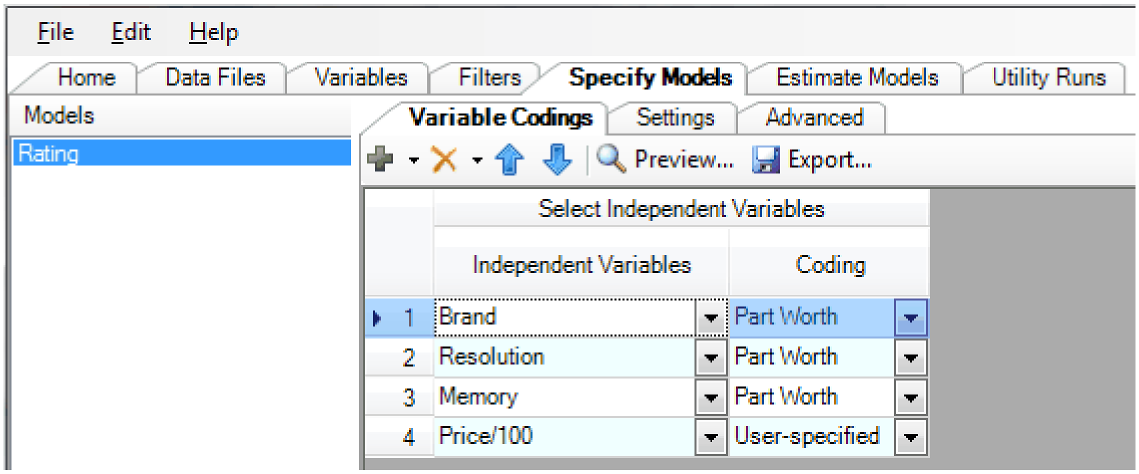


	Label	Variable Type
1	Brand	Independent (Categorical)
2	Resolution	Independent (Categorical)
3	Memory	Independent (Categorical)
4	Price/100	Independent (Continuous)
5	Rating	Dependent

	Label	Value
1	Brand A	1
2	Brand B	2
3	Brand C	3
4	Brand D	4

Note: some variables are read into HB-Reg as Categorical (such as discrete levels of price, where a code of 1 means $10, 2 means $20, and 3 means $30) even though these seem like continuous variables. You may later (in Step 4) request that HB-Reg software code these as linear or log-linear for parameter estimation. By reading these into HB-Reg as Categorical, you give yourself flexibility to later model these as either part-worth (dummy-coded) or linear/log-linear.

4. Click the *Specify Models* tab. On the *Variable Coding* tab under the *Models* column, click the label of the *Dependent Variable* for which you wish to build a model. Use the *Add a Row* ✚ icon to add independent (predictor) variables to the model. Use the drop-down controls to specify how you wish HB-Reg to code these independent variables (Part-Worth, Linear, or Loglinear).



	Select Independent Variables	
	Independent Variables	Coding
1	Brand	Part Worth
2	Resolution	Part Worth
3	Memory	Part Worth
4	Price/100	User-specified

You may preview how HB-Reg is constructing the design matrix by clicking the *Preview...* button.

You may change HB settings (number of iterations, etc.) under the *Settings* tab.

5. When you are ready for parameter estimation, click the *Estimate Models* tab. Click the blue *Estimate* link.

6. When the run is complete, you may review your results and export individual-level parameters from the *Utility Runs* tab. Along the left side are listed any runs you've completed. Highlight a run and click the *Export...* button on the toolbar to export the results to an Excel file.

3 Some Details Regarding HB Estimation

HB-Reg uses Bayes methods to estimate the parameters of a randomized coefficients regression model. In this section we provide a non-technical description of the underlying model and the algorithm used for estimation. Further details are provided in the Appendix. To make matters clearer, we focus on one of the earlier examples, which we repeat here:

> Survey respondents in a customer satisfaction study provide ratings of several companies. Some ratings are on relatively specific "explanatory" variables, such as customer service, product durability, convenience of use, etc. Other ratings are more general, such as overall satisfaction with the companies' products. One goal of the study is to infer the relative importance of each explanatory factor in determining overall satisfaction. Because respondents may differ in what is important, the researcher wants to produce unique estimates for each respondent.

Regression analysis is a statistical technique that seeks a set of "weights" which, when applied to explanatory variables, predict or explain another variable of interest. The explanatory variables are often called "independent variables" and the variable we want to explain or predict is often called the "dependent variable."

The weights are generally called "regression coefficients," though in this context we might call them "part-worths" or "importance weights." The regression equation is usually of the form

$$y = x_1 b_1 + x_2 b_2 + \ldots + x_n b_n + e$$

Here the variable y is a respondent's rating of a company (or product or service) on "overall satisfaction." The x variables are that same respondent's ratings of the same company on other variables that we believe may be important in determining how satisfied that customer is with that company. (In many contexts, we include an "intercept," which is an estimate of y if all the independent variables had values of zero. That can be accommodated in this formula by letting one of the x's have constant values of unity (1), though the HB-Reg software provides a simple checkbox that includes an intercept in the computation.) The symbol "e" on the right side of the equation stands for "error," and represents our inability to predict y with complete accuracy by adding up weighted sums of the x's. If the respondent has rated several companies, we have several values of y, several corresponding sets of x's, and several corresponding values of e.

Under certain conditions, regression analysis can provide estimates of the b's for this respondent.

- Usually we assume the errors are random, have mean of zero, and are independent of the x's. If we are using "least squares" regression, we also assume the sum of squared errors is as small as possible.

- The respondent must have rated at least as many companies as the number of variables for which we seek importance weights. Another way of saying this is that the number of unknowns (the b's) must be no larger than the number of data points (the y's).

- The respondent's ratings on different variables (the x's) must have some degree of independence from one another. For example, if there were two variables for which each company got equal ratings, we would have no way of deciding how importance should be allocated among those two variables.

Unfortunately, researchers doing customer satisfaction studies usually find that none of these conditions is satisfied.

The first condition is seldom satisfied because survey respondents tend to bunch their ratings at the top ends of the scale, so random variability tends to be smaller for highly rated products. Researchers have tried for many years to overcome this problem, one attempt being Rossi *et al.* (1999). Although this is an important problem, its complete solution will probably require new methods of data collection that encourage respondents to discriminate more finely among companies they like.

The second condition is an even more serious impediment to the estimation of individual importance weights. The persons funding the research are often interested in a large number of possible explanatory variables, but it is usually not possible for each respondent to provide knowledgeable ratings of a large number of companies. Respondents get bored when asked to rate many companies on many attributes and the quality of their output suffers. Also, many respondents are familiar with only a few companies, so their ratings of other companies contain little real information. Fortunately, HB methods can provide significant help in overcoming this problem. Unlike conventional regression analysis, HB-Reg can provide reasonable estimates for each respondent's importance weights, even when each respondent rates fewer companies than the number of variables for which weights are to be estimated. The ability of HB-Reg to provide reasonable individual-level estimates in this case may be enhanced by constraining the signs of the coefficients to be positive or negative.

Failure to satisfy the third condition is also a serious problem. Often respondents fail to distinguish among variables as precisely as researchers would like. The researcher may want to learn whether "reliability" is more or less important than "durability;" but if those words mean nearly the same thing to a respondent, he or she is likely to produce identical ratings on each variable. This failure is known as "colinearity," and describes the condition in which ratings on one variable are predictable from ratings on others. For most efficient estimation, we would like a respondent's ratings on several variables to be completely independent of one another, but that is almost never true. Fortunately, HB methods can also provide significant help in overcoming colinearity.

The model underlying HB-Reg is called "hierarchical" because it has two levels. At the upper level, respondents are considered as members of a population of similar individuals. Their importance weights are assumed to have a multivariate normal distribution described by a vector of means and a matrix of variances and covariances.

At the lower level, each individual's importance weights are assumed to be related to his ratings by the simple equation above. That is to say, when deciding on his level of overall satisfaction with a company, he is assumed to consider several explanatory variables, multiplying his rating of that company on each variable by an importance weight and adding up those products.

Suppose there are N individuals, each of whom has rated products on n explanatory variables. If we were to do ordinary regression analysis separately for each respondent, we would be estimating N*n importance weights. With the hierarchical model we also estimate N*n importance weights and we further estimate n mean importance weights for the population as well as an n x n matrix of variances and covariances for the *distribution* of individuals' importance weights. Because the hierarchical model requires that we estimate a larger number of parameters, one might expect it would work less well than ordinary regression analysis. However, because each individual is assumed to be drawn from a population of similar individuals, information can be "borrowed" from other individuals in estimating parameters for each one, with the result that estimation is usually enhanced.

In particular, it becomes possible to estimate individual parameters even though each respondent has rated only a small number of products and even though there may be considerable colinearity in a respondent's ratings on explanatory variables. For example, suppose an individual has given similar ratings to two variables, such as reliability and durability, so that an ordinary regression analysis might be unable to allocate importance between them. But since we assume this respondent is drawn from a distribution with known characteristics, we can use information about that distribution to resolve ambiguities for each individual.

3.1 The Hierarchical Model

To recapitulate, the model used by HB-Reg is called “hierarchical” because it has two levels, ***higher*** and ***lower***.

At the ***higher level***, we assume that individuals’ regression weights are described by a multivariate normal distribution. Such a distribution is characterized by a vector of means and a matrix of covariances. To make this explicit, we assume individual regression weights have the multivariate normal distribution,

$$\boldsymbol{\beta}_i \sim \mathbf{Normal}(\boldsymbol{\alpha}, \mathbf{D})$$

where:

$\boldsymbol{\beta}_i$ = a vector of regression (or importance) weights for the ith individual,

$\boldsymbol{\alpha}$ = a vector of means of the distribution of individuals’ regression weights,

$\mathbf{D}$ = a matrix of variances and covariances of the distribution of regression weights across individuals.

At the ***lower level*** we assume that, given an individual’s regression weights, values of the dependent variable are described by the model:

$$\mathbf{y}_{ij} = \mathbf{x}_{ij}' \boldsymbol{\beta}_i + \mathbf{e}_{ij}$$

where:

$\mathbf{y}_{ij}$ = the dependent variable for observation j by respondent i,

$\mathbf{x}_{ij}'$ = a row vector of values of independent variables for the jth observation for respondent i,

$\mathbf{e}_{ij}$ = random error term, distributed normally with mean of zero and variance $\boldsymbol{\sigma}^2$.

Continuing the customer satisfaction example, this model says that individuals have vectors of importance weights $\boldsymbol{\beta}_i$ drawn from a multivariate normal distribution with mean vector $\boldsymbol{\alpha}$ and covariance matrix $\mathbf{D}$. Individual i’s rating of overall satisfaction with the jth company $\mathbf{y}_{ij}$ is normally distributed, with mean equal to the sum of that respondent’s ratings on the independent variables, each weighted by the corresponding importance coefficient (which is equal to the vector product $\mathbf{x}_{ij}' \boldsymbol{\beta}_i$) and variance equal to some value $\boldsymbol{\sigma}^2$.

The parameters to be estimated are the vectors $\boldsymbol{\beta}_i$ of part-worths for each individual, the vector $\boldsymbol{\alpha}$ of means of the distribution of regression weights, the matrix $\mathbf{D}$ of the variances and covariances of that distribution, and the scalar $\boldsymbol{\sigma}^2$.

3.2 Iterative Estimation of the Parameters

The parameters are estimated by an iterative process which is quite robust and for which final results do not depend on starting values. As initial estimates of each parameter we use values of zero or unity. We use zeros as initial estimates of the betas, alpha, and the covariances, and we use unity as initial estimates of the variances and of sigma. Given those initial values, each iteration consists of these steps (further details are provided in the Appendix):

> Using present estimates of the betas and $\mathbf{D}$, generate a new estimate of $\boldsymbol{\alpha}$. We assume $\boldsymbol{\alpha}$ is distributed normally with mean equal to the average of the betas and covariance matrix equal to $\mathbf{D}$ divided by the number of respondents. A new estimate of $\boldsymbol{\alpha}$ is drawn randomly from that distribution.
>
> Using present estimates of the betas and $\boldsymbol{\alpha}$, draw a new estimate of $\mathbf{D}$ from the inverse Wishart distribution.
>
> Using present estimates of $\boldsymbol{\alpha}$, $\mathbf{D}$, and $\boldsymbol{\sigma}$, generate new estimates of the betas. We use different methods for doing this, depending on the format of the input data. If every respondent has the same values for his explanatory variables (as is frequently the case in full-profile conjoint analysis) we use a “normal draw” procedure to get a new estimate of beta for each individual. That is to say, we draw a random vector from the distribution characterizing his regression weights. If every respondent can have unique values for his explanatory variables (as is usually the case in customer satisfaction research) we obtain a new estimate of beta for each individual using a Metropolis Hastings algorithm.
>
> Using present estimates of $\boldsymbol{\alpha}$, $\mathbf{D}$, and the betas, generate a new estimate of $\boldsymbol{\sigma}$. For this purpose we again use the inverse Wishart distribution.

In each of these steps we re-estimate one set of parameters conditionally, given current values for the other three. This technique is known as “Gibbs sampling,” and eventually converges to the correct distributions for each set of parameters. Another name for this procedure is a “Monte Carlo Markov Chain,” deriving from the fact that the estimates in each iteration are determined from those of the previous iteration by a constant set of probabilistic transition rules. This Markov property assures that the iterative process converges.

This process is continued for a large number of iterations, typically 10,000 or more. After we are confident of convergence, the process is continued for many further iterations and the actual draws of beta for each individual as well as estimates of $\boldsymbol{\alpha}$, $\mathbf{D}$, and $\boldsymbol{\sigma}$ are saved to the hard disk. The final estimates of regression coefficients for each individual and also of $\boldsymbol{\alpha}$, $\mathbf{D}$, and $\boldsymbol{\sigma}$, are obtained by averaging those values that have been saved.

3.3 Parameter Constraints

There are modeling situations in which the researcher knows ahead of time that certain parameters must not be less in magnitude than others. As one example, conjoint studies frequently include product attributes for which almost everyone would be expected to prefer one level to another. However, estimated part worths sometimes turn out not to have those expected orders. This can be a problem, since part worths with the wrong slopes are likely to yield nonsense results and can undermine users' confidence. As another example, a model may include company ratings on different aspects of service or product quality, where higher ratings imply greater satisfaction. If those variables are used to predict some overall outcome such as likelihood of purchase or overall rating for the company, one should expect all betas to be positive. Due to the sparse nature of the data and random noise, many of the individual-level betas may be negative, but the researcher may want to constrain them to be positive.

HB-Reg provides the capability of enforcing constraints between two parameters, or sign constraints for individual parameters. The same constraints are applied for all respondents, so constraints should only be used for variables that have unambiguous a-priori orders or signs.

Evidence to date suggests that constraints can be useful when the researcher is primarily interested in the accuracy of individual models (such as for classification or estimating "hit rates"). However, constraints appear to be less useful, and indeed can be harmful, if the researcher is primarily interested in making aggregate predictions, such as predictions of shares of choices within choice simulators. The use of constraints can also get in the way of hypothesis testing, where the researcher may require the unconstrained distribution of parameters.

In a paper available on the Sawtooth Software Web site (Johnson, 2000) we explored several ways of enforcing constraints with HB among part-worths in the conjoint analysis context. Realizing that most conjoint analysis users are probably interested in predicting individual choices as well as aggregate shares, we examined the success of each method with respect to both hit rates and share predictions. One of the methods seemed consistently successful was referred to in that paper as "Simultaneous Tying." We have implemented that method in HB-Reg. We call it "Simultaneous" because it applies constraints during estimation, so the presence of the constraints affects the estimated values.

Simultaneous Tying

This method features a change of variables between the "upper" and "lower" parts of the HB model. For the upper model, we assume that each individual has a vector of (unconstrained) betas, with distribution:

$$\boldsymbol{\beta}_i \sim \textbf{Normal}(\boldsymbol{\alpha}, \textbf{D})$$

where:

$\boldsymbol{\beta}_i$ = unconstrained betas for the ith individual,
$\boldsymbol{\alpha}$ = means of the distribution of unconstrained betas,
$\textbf{D}$ = variances and covariances of the distribution of unconstrained betas.

With this model, we consider two sets of betas for each respondent: unconstrained and constrained. The unconstrained betas are assumed to be distributed normally in the population and are used in the upper model. However, the constrained betas are used in the lower model to evaluate likelihoods.

We speak of "recursively tying" because, if there are several variables involved in constraints, tying two values to satisfy one constraint may lead to the violation of another. The algorithm cycles through the constraints repeatedly until they are all satisfied.

When constraints are in force, the estimates of population means and covariances are based on the unconstrained betas. However, since the constrained betas are of primary interest, we plot the constrained betas to the screen. Only the constrained betas are saved to the output draws and point estimates files.

When constraints are in place, measures of fit (average r-squared) are *decreased*. Constraints always decrease the goodness-of-fit for the sample in which estimation is done. This is accepted in the hope that the constrained solution will work better for predictions in out-of-sample situations.

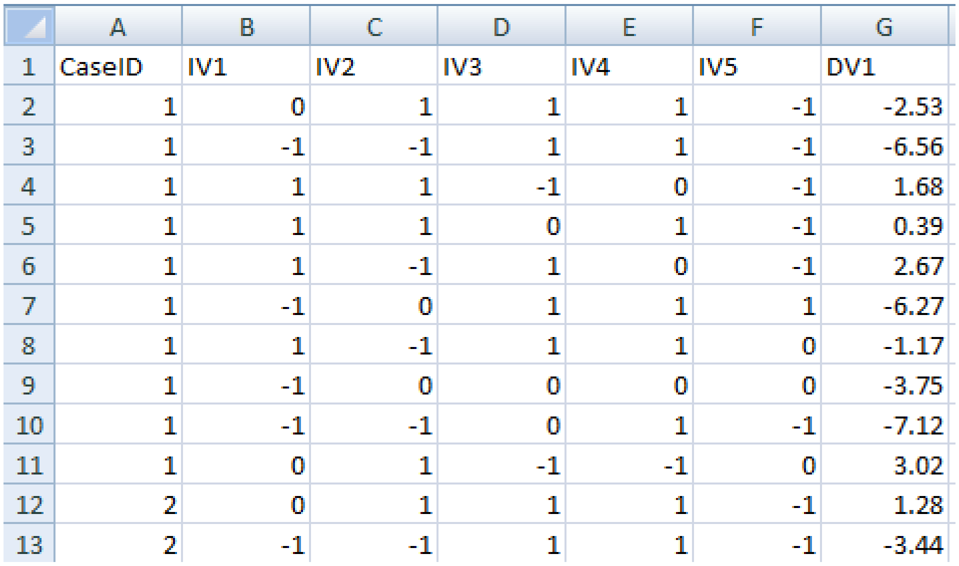
4 Preparing Data Files for HB-Reg

4.1 Preparing Data Files for HB-Reg

Independent and dependent variables must be scaled with appropriate ranges (variance) so that HB converges quickly and properly. In our experience, variables should be coded in the magnitude of about single digits. This means that prices of $100,000 to $500,000 should be divided by 100,000 so that they range from 1 to 5, etc. Also take care that the range should not be too small. For example, a variable that runs from -0.00001 to 0.00001 would not lead to quick and proper convergence.

File Format

We have made it very easy to prepare data files for HB-Reg. We use the .csv (comma-separate values) text format, which may be saved from Excel and by most software that is used for collecting and processing data. The first row contains labels, but all data in the remaining rows must be numeric only. Missing data are not supported. Here are the first few rows from our Sample1 data set as displayed by Excel:



	A	B	C	D	E	F	G
1	CaseID	IV1	IV2	IV3	IV4	IV5	DV1
2	1	0	1	1	1	-1	-2.53
3	1	-1	-1	1	1	-1	-6.56
4	1	1	1	-1	0	-1	1.68
5	1	1	1	0	1	-1	0.39
6	1	1	-1	1	0	-1	2.67
7	1	-1	0	1	1	1	-6.27
8	1	1	-1	1	1	0	-1.17
9	1	-1	0	0	0	0	-3.75
10	1	-1	-1	0	1	-1	-7.12
11	1	0	1	-1	-1	0	3.02
12	2	0	1	1	1	-1	1.28
13	2	-1	-1	1	1	-1	-3.44

Rows

Each row represents an observation and each respondent (or other unit of analysis) should have multiple observations (in the example above, each respondent has 10 observations). Each respondent ID must be unique. The number of observations may differ per respondent, but the observations for each case must be available in consecutive rows (each respondent's data must be grouped together).

Columns

Regression analysis runs usually have multiple Independent Variables and a single Dependent Variable. Your data file can have as many Independent Variables and Dependent

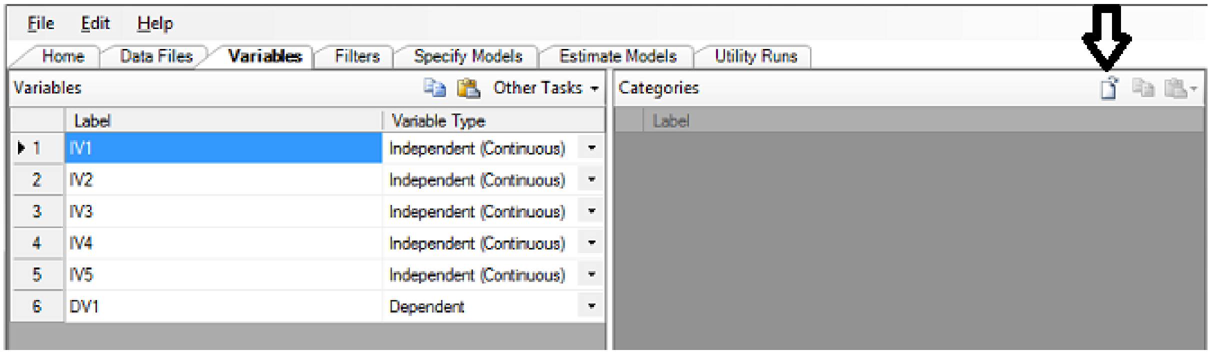
Variables as you wish (though only one Dependent Variable may be specified in any one model). When you build models, you'll be asked to indicate which variables are independent and dependent, as well as which ones to include within your estimation run.

Reading the File into HB-Reg

Once you have prepared your .csv data file, you simply browse to it from the *Data Files* tab, or when prompted to do so in the project wizard dialog. You may also supply an optional (separate) demographics file that is in .csv format, with Respondent ID followed by segmentation variables (to use as filters or covariates).

Variable Type

After HB-Reg reads your data, you can open the *Variables* tab to see how it has interpreted the file. A row is shown in the data table corresponding to each column in your data file. The labels that you had specified in the first row of your data file are displayed. For example, if you open the Sample1.csv file available in the Sample1 project (you can open this project by clicking ***Help | Sample Data Sets***), you see:



Variables

	Label	Variable Type
1	IV1	Independent (Continuous)
2	IV2	Independent (Continuous)
3	IV3	Independent (Continuous)
4	IV4	Independent (Continuous)
5	IV5	Independent (Continuous)
6	DV1	Dependent

Categories

Label

For each variable in your data file, you need to specify whether it is:

- Independent (Continuous)
- Independent (Categorical)
- Dependent

Continuous (User-Specified)

Independent variables are *Continuous* when they take on many unique metric values, such as the weight of an individual in kilos, ratings on a 10-point scale, or factor scores with decimal places of precision. Continuous independent variables are used as-is in the design matrix for parameter estimation. In the popular CBC/HB software, such variables are called "User-Specified," meaning you as the user have specified exactly how each independent variable should be coded for parameter estimation.

Independent (Categorical)

Independent variables are Categorical if they take on a limited number of discrete values, such as colors of product concepts (red=1, blue=2, green=3), ratings (1=definitely would not buy, 2=probably would not buy, 3=might/might not buy, 4=probably would buy, 5=definitely would buy), or even a limited number of discrete prices (1=$100, 2=$150, 3=$200, 4=$250). For an attribute like color, it isn't appropriate to treat the values in a metric fashion, where a code of 3 (green) is three times as large as a code of 1 (red). Categorical independent variables are typically dummy-coded and HB-Reg will automatically dummy-code these variables for you if you request "part-worth" functional form on the *Specify Models* tab.

If your categorical independent variable is something like a 5-point rating scale or four discrete levels of price, then the Categorical coding type gives you flexibility to fit a part-worth functional form, a linear function, or a loglinear function (you select these options on the Specify Models tab). In all cases, HB-Reg does the coding automatically for you.

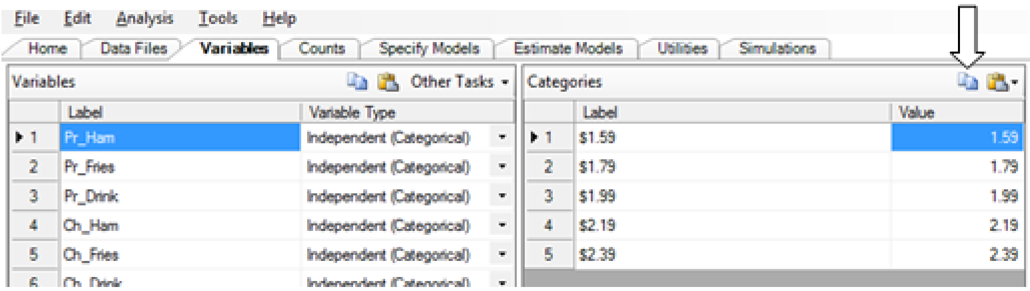
Dependent

This is the variable that is to be predicted by the independent variables. Dependent variables may be integers or continuous, with additional decimal places of precision if desired.

4.2 Labels File

If you have specified any Categorical Independent Variables, then you can specify labels and values for the categories using the *Variables* tab. You can do this by typing directly into the interface, or you can import labels from a .csv that you can prepare with a program such as Excel.

This dialog supports cutting and pasting from programs such as Word or Excel, so you don't have to re-type labels and values that you have written elsewhere:



	Label	Variable Type
1	Pr_Ham	Independent (Categorical)
2	Pr_Fries	Independent (Categorical)
3	Pr_Drink	Independent (Categorical)
4	Ch_Ham	Independent (Categorical)
5	Ch_Fries	Independent (Categorical)
6	Ch_Drink	Independent (Categorical)

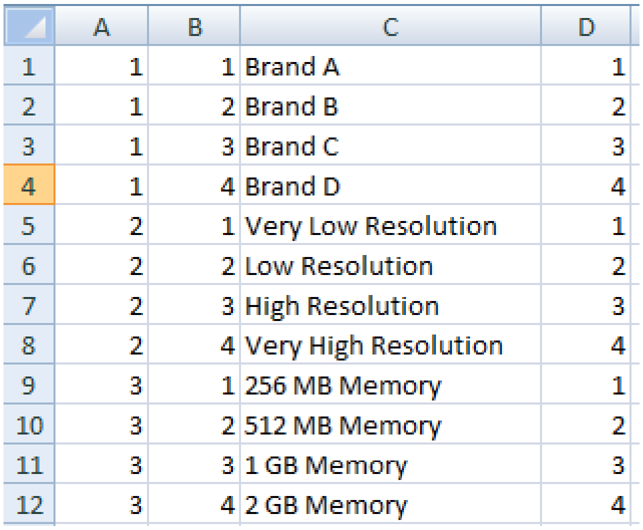
	Label	Value
1	$1.59	1.59
2	$1.79	1.79
3	$1.99	1.99
4	$2.19	2.19
5	$2.39	2.39

Figure 3.3 (Establishing Variable Labels and Values on Variables Tab)

On the *Variables* tab, you specify the labels used for the categories of each of the independent and dependent variables in your study. In Figure 3.3, we show how the labels have been specified for each of the five levels of hamburger price (**Pr_Ham**), as well as the values associated with each label. The labels are used in HB-Reg's reports (so we suggest concise labels).

Because there can be many labels and values associated with Categorical variables in HB-Reg projects, it is helpful to define these in a separate **my_labels.csv** file, using a program like Excel. You may ***Import Category Labels and Values from File*** by clicking the icon indicated with the arrow in Figure 3.3. The .csv file containing labels must have four columns: attribute#, level#, label, and value (see Figure 3.4).

Figure 3.4 shows an example labels file (from the Sample2 dataset), as displayed in Excel.



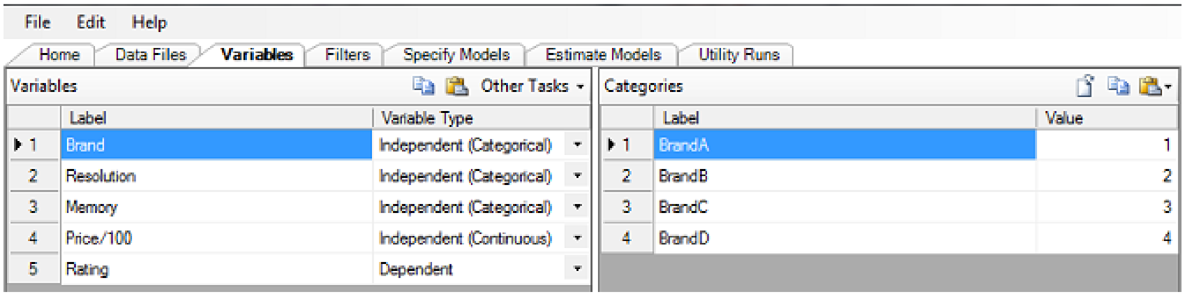
	A	B	C	D
1	1	1	Brand A	1
2	1	2	Brand B	2
3	1	3	Brand C	3
4	1	4	Brand D	4
5	2	1	Very Low Resolution	1
6	2	2	Low Resolution	2
7	2	3	High Resolution	3
8	2	4	Very High Resolution	4
9	3	1	256 MB Memory	1
10	3	2	512 MB Memory	2
11	3	3	1 GB Memory	3
12	3	4	2 GB Memory	4

For this dataset, there are three categorical independent variables (variables 1 through 3 in the file). Continuous independent variables and the continuous dependent variable should be skipped (not listed) in the labels file. In other words, the labels file only needs to list labels and values associated with Categorical Independent Variables.

Notice that the final column (values) contains values on a scale (in our situation, 1 to 4). If you plan to investigate linear and log-linear functional forms for these Categorical Independent Variables, HB-Reg software will automatically transform these to have values within the design matrix that are zero-centered with a range of 1 (to provide faster convergence during estimation). So, values for Memory could be written as metric quantities 0.256, 0.512, 1, and 2.

4.3 Selecting Independent and Dependent Variables

From the *Variables* tab, you need to indicate which variables are dependent variables and specify this in the *Variable Type* column drop-down control. Figure 3.5 shows the first three Independent Variables (Brand, Resolution, and Memory) with *Variable Type* set to "Independent (Categorical)." The next independent variable, Price/100, is a "summed (continuous) price" amount and should be treated as a Continuous Independent Variable. The final variable, Rating, is the respondent's rating for the product profile and is a Dependent Variable.



	Label	Variable Type
1	Brand	Independent (Categorical)
2	Resolution	Independent (Categorical)
3	Memory	Independent (Categorical)
4	Price/100	Independent (Continuous)
5	Rating	Dependent

	Label	Value
1	BrandA	1
2	BrandB	2
3	BrandC	3
4	BrandD	4

Figure 3.5

4.4 Missing Values

Missing Observations

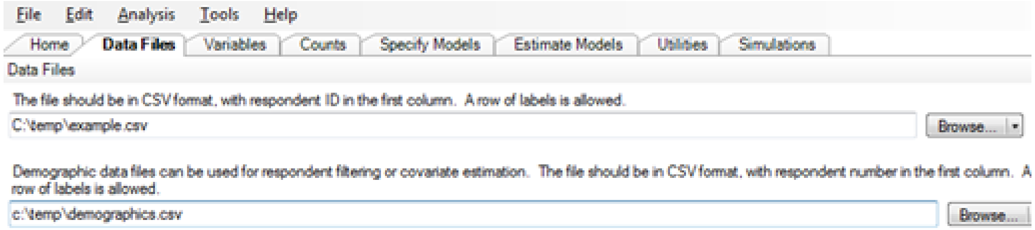
If a respondent did not provide as many observations as another (or left some observations missing), you may simply delete any missing rows from the data file. Each respondent does not need to have the same number of observations (rows) present in the data file.

Missing Independent Variables

In conjoint analysis, there can be cases involving independent variables that are only applicable to specific product concepts in the design matrix (alternative-specific effects). In these cases, you may specify a not-applicable independent variable with a code of zero. The zero carries forward into the design matrix for parameter estimation.

4.5 Segmentation Variables (Covariates)

If you have segmentation variables (for filters or covariates), these should be provided in a separate "demographics" .csv file, again with CaseID as the first column. The CaseIDs must match the CaseIDs provided in the main data file. You select the demographics file from the *Data Files* tab:



The file should be in CSV format, with respondent ID in the first column. A row of labels is allowed.

C:\temp\example.csv

Demographic data files can be used for respondent filtering or covariate estimation. The file should be in CSV format, with respondent number in the first column. A row of labels is allowed.

c:\temp\demographics.csv

Figure 3.6 (Selecting Demographics File from Data Files tab)

Missing values are not permitted in the demographics file. If you have a missing value, you should set it to a numeric code distinguishable from the other numeric values in the file.

5 Coding Independent Variables

5.1 Coding Independent Variables

Continuous Independent Variables (User-Specified)

With standard OLS regression, multiplying all values of a particular variable by any constant will result in the estimated regression coefficient being scaled by the reciprocal of that constant. But, this is not always the case with HB methods. The default priors we have set (prior covariance matrix) tend to work quite well if all your independent and dependent variables are scaled approximately in the single digits. But, if you try to use an independent or dependent variable scaled in the 100s or 1000s of units, HB estimation may fail to converge, leading to biased parameter estimates. For this reason, we recommend your continuous independent variables be coded in the magnitude of about single digits.

(Note: advanced users can specify their own prior covariance matrix to deal with situations in which some betas have quite different expected posterior variances than others.)

Categorical Independent Variables

By “categorical variables” we mean variables like color or style, for which the various possible states or categories do not have obvious numeric values. HB-Reg has a built-in provision for automatically dummy-coding or linear-coding categorical data.

Dummy-Coding

Often categorical variables are coded using “dummy variables.” This means that a separate variable is devoted to each state, scored with a one if that state is present and a zero if that state is absent. It is desirable to delete one state of each variable, since otherwise there would be colinearity, with the sum of codes for each variable being exactly unity. Omitting one state for each variable is equivalent to assuming that the regression coefficient for that state is equal to zero and that the other coefficients for that variable measure effects with respect to the omitted state. This is the simplest and most straightforward procedure. Using dummy coding for categorical variables also makes it possible to constrain any of the “explicit” levels with respect to the “omitted” variable, as the omitted variable is zero.

Two states (such as red, green)

If red, code	0
If green, code	1

Three states (such as red, green, blue)

If red, code	0, 0
If green, code	1, 0
If blue, code	0, 1

Four states (red, green, blue, yellow)

If red, code	0, 0, 0
If green, code	1, 0, 0
If blue, code	0, 1, 0
If yellow, code	0, 0, 1

The deleted state has an implied regression coefficient (zero).

Linear (and Log-Linear) Coding

You can ask HB-Reg to code a categorical independent variable as a single linear or log-linear term. Rather than dummy-coding, values are placed into a single column of the design matrix representing the different levels of the categorical variable. If the function is truly linear, this can save degrees of freedom and lead to better models. This only makes sense when the independent variable is associated with specific quantities such as speed, weight, or price.

The values you type into the *Variables* tab for a categorical independent variable can be of any magnitude you wish, such as 1000, 1100, 1200. HB-Reg automatically transforms the variable into zero-centered values that have a range of one unit, for quicker convergence in HB-Reg. You can observe the results of the transformation by clicking the ***Preview Design Matrix*** button on the Variable *Codings* tab when you are specifying models. Click here for more information on the linear and log-linear coding transformation.

Proper Prior Covariance Matrix and Dummy Coding

We have found with extremely sparse data that dummy coding can sometimes lead to improper estimates of the omitted level (with respect to the other levels). The problems are enhanced as the number of mutually exclusive levels within a factor is increased. We faced the same challenges in our CBC/HB software with choice data, where the data can become especially sparse, so in our CBC/HB software we modify the prior covariance matrix to deal more effectively with effects coding (see the appendix of the CBC/HB software manual for details).

With HB-Reg, when you use HB-Reg's capability to automatically recode categorical independent variables as "part-worth" (dummy-coding), we automatically implement the proper prior covariance coding, per the methodology described in CBC/HB software's appendix.

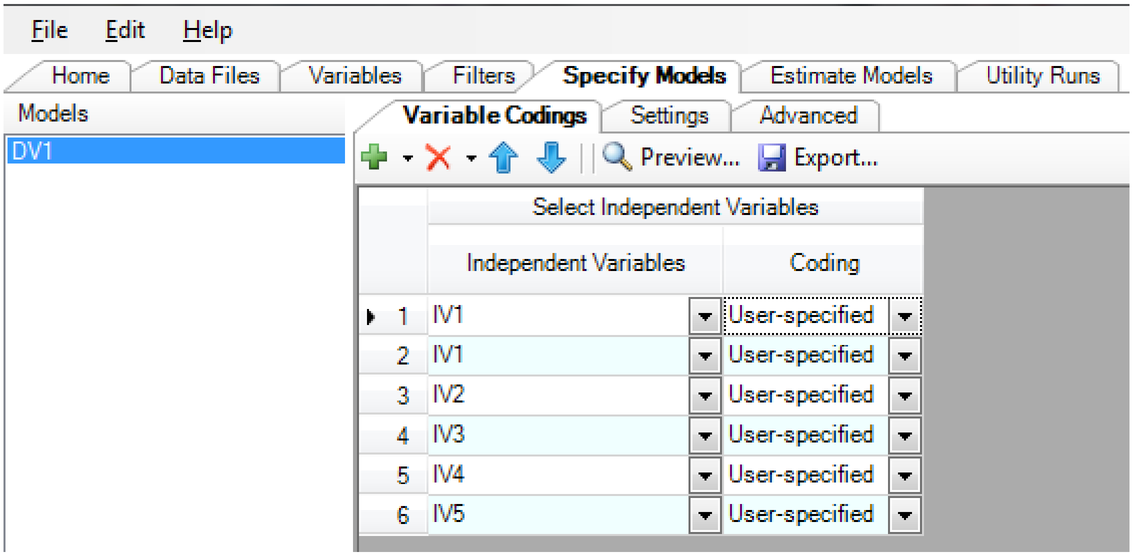
6 Building Models (Specify Models Tab)

6.1 Specify Models Tab

For each Dependent Variable included in your data file (there typically is just one), you can specify a model. Specifying a model includes selecting which independent variables to include in the HB-Reg run and specifying how they should be coded (if they are Categorical Independent variables).

To add an independent variable to the model, click the green "plus" symbol (Add a Row) icon. You can remove it by clicking the "X" icon (Remove selected rows) or by toggling the coding type to "exclude" for the independent variable.

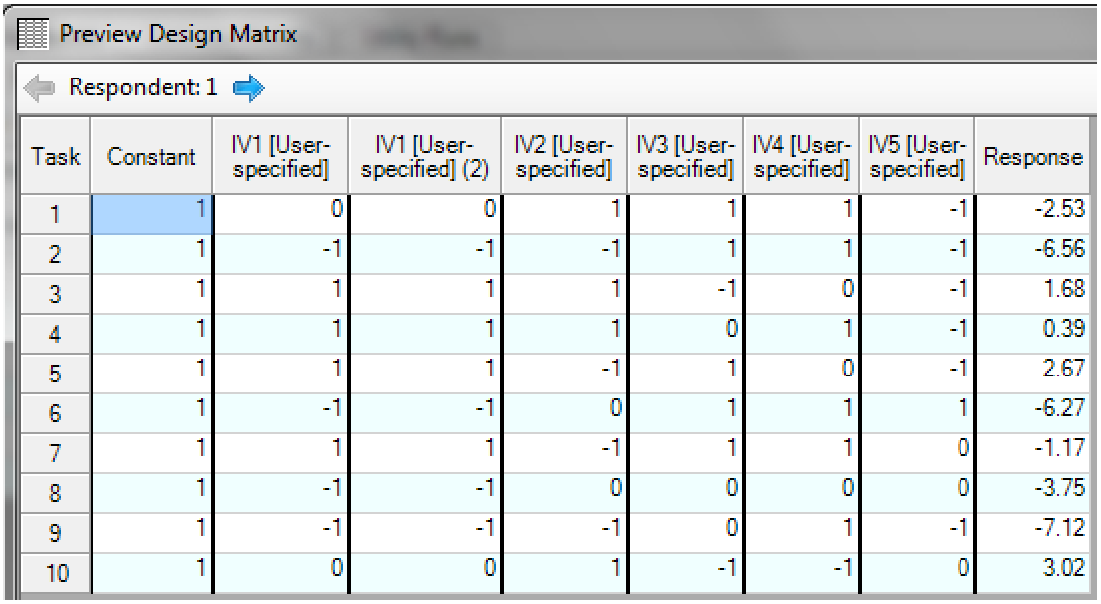
The example below uses the Sample1 data set and shows all five independent variables included in the model. Independent (Continuous) variables are also User-Specified variables, as the coding you provided in the .csv data file is the coding that will be used in the design matrix for estimation.



	Select Independent Variables	
	Independent Variables	Coding
1	IV1	User-specified
2	IV1	User-specified
3	IV2	User-specified
4	IV3	User-specified
5	IV4	User-specified
6	IV5	User-specified

The independent variables you add as rows to the table on the tab actually become columns in the independent variable (design) matrix.

To see how the specified coding above is actually implemented in the design matrix, click the ***Preview… (*Preview Design Matrix*)*** button.



Task	Constant	IV1 [User-specified]	IV1 [User-specified] (2)	IV2 [User-specified]	IV3 [User-specified]	IV4 [User-specified]	IV5 [User-specified]	Response
1	1	0	0	1	1	1	-1	-2.53
2	1	-1	-1	-1	1	1	-1	-6.56
3	1	1	1	1	-1	0	-1	1.68
4	1	1	1	1	0	1	-1	0.39
5	1	1	1	-1	1	0	-1	2.67
6	1	-1	-1	0	1	1	1	-6.27
7	1	1	1	-1	1	1	0	-1.17
8	1	-1	-1	0	0	0	0	-3.75
9	1	-1	-1	-1	0	1	-1	-7.12
10	1	0	0	1	-1	-1	0	3.02

The ability to preview the design matrix keeps you informed and in control of the mechanics of model building in HB-Reg, absent the hassle of so much data processing work. ***Preview Design Matrix*** makes the entire process transparent and reassuring, because you can see exactly how the design matrix is being coded.

In the example above, the design matrix for the first respondent is shown (Respondent #1). You can click the right arrow at the upper-left of the dialog to display the design for the next respondent, if you wish.

The rows are observations (where each respondent should have multiple observations).

The final column records the dependent variable (Response), which is a continuous variable with 2 decimal places of precision.

7 Analyzing Data Using HB-Reg

7.1 HB Settings

The *Settings* tab displays a number of settings that govern the computation:

> *Number of iterations before using results*: HB usually requires thousands of iterations to obtain good estimates of respondent utilities and of the population means and covariances. The preliminary iterations are the "burn-in" iterations that are not actually used in the final utility run, but are undertaken as a means to obtain convergence. Typically, 10000 or more iterations should be performed before convergence is assumed in HB-Reg. Only after a run is completed can one examine the history of part-worths estimated across the iterations to assess if the model indeed has converged to a relatively stable solution. The part-worth utilities should oscillate randomly around their true values, with no indication of trend. If convergence is not obtained, then you will want to re-run the HB estimation and specify a larger number of initial iterations.
>
> *Number of draws to be used for each respondent*: Each iteration, after convergence is assumed, leads to a candidate set of utilities for each respondent. The default is to use/ save 1000 draws for each respondent, which is generally more than enough for most every practical application.
>
> *Save random draws*: Clicking this box will save the draws for each respondent into the zipped archive folder. You can export the draws by highlighting a run and clicking ***Export...*** from the *Utility Runs* tab.
>
> *Skip factor for saving random draws*: The skip factor is a way of compensating for the fact that successive draws of the betas are not independent. A skip factor of k means that results will only be used for each kth iteration. It is useful to use every kth draw for each respondent to capture more completely the distribution of draws that characterize each respondent's preferences. The utilities can oscillate in rather lengthy wave-like patterns, with cycles measured in the thousands of iterations, so we recommend using a large skip factor for saving draws.
>
> *Skip factor for displaying in graph*: Indicates the skip factor employed when deciding for how many draws the results should be displayed on the graph.
>
> *Skip factor for printing in log file*: This controls the amount of detail that is saved in the Estimation_log.txt file to record the history of the iterations. You can export the Estimation_log.txt file by highlighting a run and clicking ***Export...*** from the *Utility Runs* tab.
>
> *Include a constant in the design:* This controls whether a constant column is added to the design matrix. This is also referred to as the intercept in regression analysis.
>
> *Target Acceptance:* We employ an adaptive algorithm to adjust the average jump size, attempting to keep the acceptance rate near 0.30. The proportionality factor is arbitrarily set at 0.1 initially. For each iteration we count the proportion of respondents for whom

the new candidate beta is accepted. If that proportion is less than 0.3, we reduce the average jump size by a tenth of one percent. If that proportion is greater than 0.3, we increase the average jump size by a tenth of one percent. As a result, the average acceptance rate is kept close to the target of 0.30. You may set the acceptance rate to a different value if desired.

Starting Seed: HB uses random number generators in various stages of its algorithm, so a starting seed must be specified. Setting a seed of “0” indicates to use a seed based on the computer clock, but users can specify a specific seed to use (integers from 1 to 10000), so that results are repeatable. When using different random seeds, the posterior estimates will vary, but insignificantly, assuming convergence has been reached and many draws have been used.

Advanced Settings

Prior degrees of freedom: This value is the additional degrees of freedom for the prior covariance matrix (not including the number of parameters to be estimated) and can be set from 2 to 100000. The higher the value, the greater the influence of the prior variance and more data are needed to change that prior. The scaling for degrees of freedom is relative to the sample size. If you use 50 and you only have 100 subjects, then the prior will have a big impact on the results. If you have 1000 subjects, you will get about the same result if you use a prior of 5 or 50. As an example of an extreme case, with 100 respondents and a prior variance of 0.1 with prior degrees of freedom set to the number of parameters estimated plus 50, each respondent's resulting part worths will vary relatively little from the population means. We urge users to be careful when setting the prior degrees of freedom, as large values (relative to sample size) can make the prior exert considerable influence on the results.

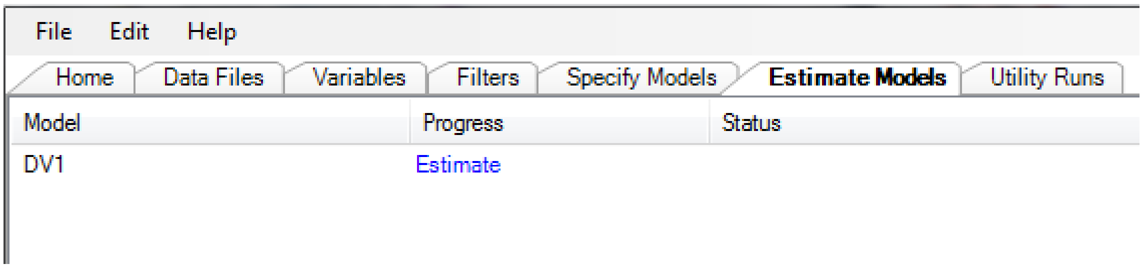
Prior variance: The default is 1 for the prior variance for each parameter, but users can modify this value. You can specify any value from 0.1 to 100. The scaling of your independent and dependent variable should guide your decision. Increasing the prior variance tends to place more weight on fitting each individual's data and places less emphasis on "borrowing" information from the population parameters. The resulting posterior estimates are relatively insensitive to the prior variance, except a) when there is very little information available within the unit of analysis relative to the number of estimated parameters, and b) the prior degrees of freedom for the covariance matrix (described above) is relatively large.

Note: we use the prior covariance matrix as recommended by Professor Peter Lenk and described in the CBC/HB software documentation, Appendix G. Part-worth functions for Categorical Independent Variables are dummy-coded and follow Lenk's recommended prior covariance structure.

7.2 Estimate Models

After you have specified *Model Settings* and *HB Settings* you are ready to estimate your models. HB-Reg datasets typically have just one dependent variable, thus there usually will just be one model to run. However, if you have included multiple dependent variables in your data file, you will see multiple models that you can run, each named by your dependent variable labels.

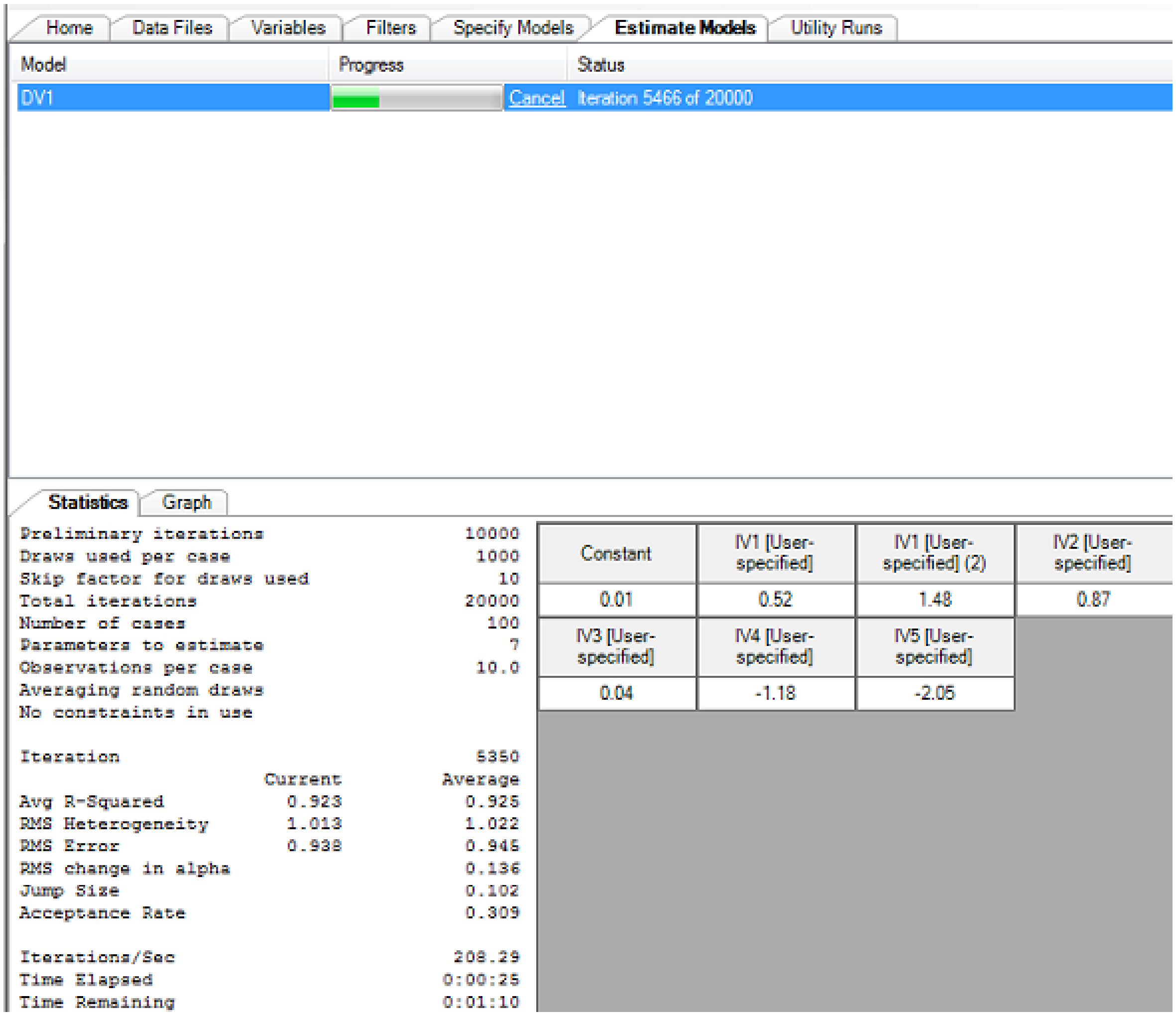
When you click the *Estimate Model* tab, you see your dependent variable(s) listed under the Model column.



Model	Progress	Status
DV1	Estimate	

For example, to run the model to predict Dependent Variable **DV1**, you would click the blue *Estimate* link shown in this table.

After you click the *Estimate* link, the estimation begins. Multiple estimations may be run simultaneously (by clicking on multiple *Estimate* links), which takes advantage of your hardware's multi-core processor.



Model	Progress	Status
DV1	🟩	Cancel Iteration 5466 of 20000

```
Preliminary iterations                     10000
Draws used per case                         1000
Skip factor for draws used                    10
Total iterations                           20000
Number of cases                              100
Parameters to estimate                         7
Observations per case                       10.0
Averaging random draws
No constraints in use

Iteration                                   5350
                           Current       Average
Avg R-Squared                0.923         0.925
RMS Heterogeneity            1.013         1.022
RMS Error                    0.938         0.945
RMS change in alpha                        0.136
Jump Size                                  0.102
Acceptance Rate                            0.309

Iterations/Sec                            208.29
Time Elapsed                             0:00:25
Time Remaining                           0:01:10
```

Constant	IV1 [User-specified]	IV1 [User-specified] (2)	IV2 [User-specified]
0.01	0.52	1.48	0.87
IV3 [User-specified]	IV4 [User-specified]	IV5 [User-specified]	
0.04	-1.18	-2.05	

A *Cancel* link is provided to cancel any run.

There are a few key statistics to pay attention to:

> *Avg R-Squared*. This abbreviation is short for the average squared correlation between each respondent's predicted and actual data for the 10 observations of the dependent variable. Because we start with estimated regression coefficients of zero, the average R-squared will be zero initially and improve throughout the early part of the computation.
>
> *RMS Heterogeneity* is short for "root mean square heterogeneity." Recall that we estimate the variances and covariances for the betas (regression coefficients) among respondents. "RMS Heterogeneity" is just the square root of the average of those variances. The SAMPLE1 and SAMPLE2 data sets were both constructed to have heterogeneity of unity, so our estimation of this quantity is quite accurate.
>
> The next statistic is "RMS Error" which is a measure of the average error in predicting each respondent's dependent variable from his/her independent variables. This is nearly the same thing as the Sigma parameter that we estimate, except that this value is computed

directly from the data, whereas Sigma is estimated by making a normal draw from an estimated distribution. The SAMPLE1 and SAMPLE2 data sets were also constructed to have RMS error of unity, so this quantity is also estimated quite accurately.

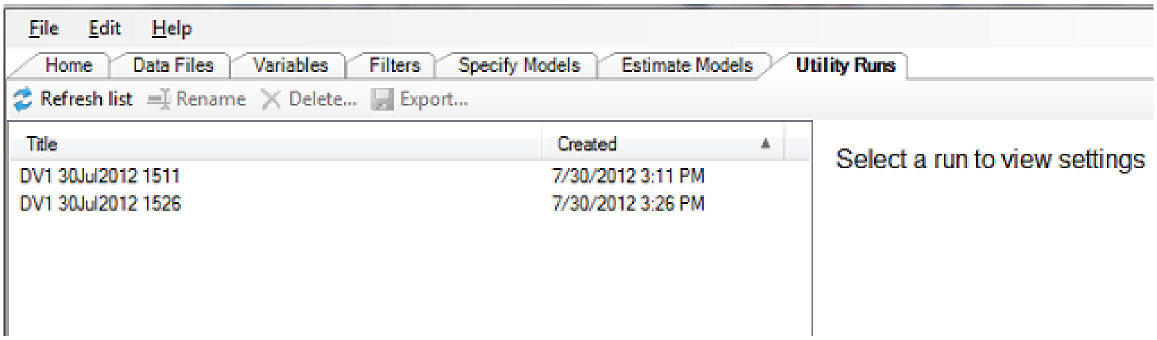
The next statistic is “RMS Change in alpha.” Alpha is the estimate of average regression coefficients for the population from which the individuals are drawn. This estimate is updated on each iteration. The RMS Change is the square root of the average squared change from one iteration to the next.

Viewing the history of Mean Betas across the burn-in and used iterations is one of the best way to assess how well the model has converged. Click the Graph tab. After a few moments the history of average betas for the sample is displayed. The first (burn-in) iterations are shown in the grey region at the left. The last (used) iterations are shown in the white region at the right. There should be some oscillation in the betas, but no remaining trend once the iterations have transitioned to the used (white) region of the history chart.

A graph is available for each of the runs you have run in the queue.

7.3 Managing Your HB Runs

Each run you estimate is stored within your project folder in a subfolder named *Utilities* and may be viewed and managed from the ***Utility Runs*** tab. The run is labeled with a combination of the Dependent Variable name along with the date the run was created (you may rename each run if you like).



Title	Created
DV1 30Jul2012 1511	7/30/2012 3:11 PM
DV1 30Jul2012 1526	7/30/2012 3:26 PM

Select a run to view settings

You can ***Rename*** and ***Delete*** runs. You can review the settings and results (average utilities, fit, variance statistics, etc.) by highlighting a run in the table. You can also export the contents of various output files by clicking ***Export...***.

When you click **Export...**, the following output files are available for exporting:

Alpha Matrix.csv
Contains unconstrained estimates of alpha for each saved iteration. Results for each saved iteration are identified by an iteration number.

Covariances.csv
Contains successive estimates (for each iteration) of the variance-covariance matrix **D** for unconstrained part worths of the population. To minimize file size, results for each iteration are saved in a single row, which contains just the part of the matrix on and above the diagonal. Thus, if there are n independent variables, each row contains n(n+1)/2 elements.

Draws.csv
If you elected to save draws, this file is available for export and contains estimates of beta for each individual for each saved iteration. The estimates are sorted together by respondent. Each estimate of beta is preceded by Case id number and 1000 times the squared correlation between predicted and actual observations.

Estimation log.txt
Contains a record of the computation. Each time you begin a run, information that appears on the screen is also saved in the log file. When you provide parameters for your run, one of those is the skip factor for printing in the log file. If that value is 100, then every 100th iteration is summarized in the log file with the same statistics that appear on the screen. Reviewing those values can help you assess whether the process has indeed converged within the assumed number of iterations. At the bottom of the description for that run is a summary of the mean and variance of the betas, together with an analysis of between-vs.-

within-respondent variance for each element of beta.

Point Estimates.csv
Contains the point estimates (averages of draws) for the parameters (regression coefficients) for each respondent. R-squared is also included (reported as r-squared * 1000).

Prior covariance matrix.csv
Contains the prior covariance matrix used in the estimation.

Standard Deviations.csv
If you did not elect to save draws, this file is available for export and contains the within-respondent standard deviations among random draws. There is one record for each respondent, consisting of respondent number, followed by the standard deviation for each parameter estimated for that respondent.

7.4 Assessing Convergence

We have already noted that one of the easiest ways to try to determine convergence is to study the pattern of average estimates for the parameters as plotted in the visual graphic displayed during iterations.

There is yet another way to assess convergence. After the computation has finished you can inspect its history by looking at a file named **Estimation Log.txt** (you export this from the *Utility Runs* tab). We specified that results of each 1,000th iteration were to be saved to the log file. We reproduce that information here:

```
Iteration  AVG  Heter  Error Change
           Rsq    RMS    RMS  Alpha
  1000   0.920  0.940  0.972  0.094
  2000   0.926  1.015  0.966  0.164
  3000   0.921  0.963  0.981  0.165
  4000   0.930  1.233  0.918  0.117
  5000   0.923  0.999  0.985  0.147
  6000   0.926  1.026  0.955  0.133
  7000   0.922  1.092  0.980  0.250
  8000   0.922  0.991  0.963  0.137
  9000   0.921  0.900  0.991  0.117
 10000   0.920  1.053  0.994  0.124
 11000   0.922  1.290  0.968  0.239
 12000   0.922  1.039  0.961  0.088
 13000   0.922  1.019  0.974  0.150
 14000   0.922  0.977  0.965  0.076
 15000   0.926  1.027  0.959  0.116
 16000   0.931  1.019  0.915  0.141
 17000   0.924  1.192  0.952  0.155
 18000   0.926  1.108  0.949  0.150
 19000   0.920  1.045  0.984  0.216
 20000   0.922  1.101  0.968  0.081
```

As you can see there is no apparent trend in any of these four columns. Apparently this process converged long before the 10,000 iterations which we had allowed for that to occur.

You can also export and inspect/plot the contents of the **Alpha Matrix.csv** file which contains each saved estimate of alpha (the vector of estimated means of the distribution of the betas).

7.5 Deciding Which Variables to Include

A simple and appropriate test for the significance of a parameter in hierarchical Bayesian estimation is to tabulate for what percent of "used" draws of alpha (those after convergence has been assumed) the parameter has the same sign. For example, if a particular parameter was greater than zero for 99% of the "used" draws of alpha, then we are 99% confident that the mean parameter weight for the population is different from zero.

Non-Bayesian Statistical Tests

The test described directly above is concerned with average parameters across the population. But, there may be parameters that are highly significant for subsets of the population that could be overlooked. We tabulate additional, more classical, statistics that help you consider not only the significance of a parameter for the population, but also gain a feel for the heterogeneity in a parameter weights across the population.

At the bottom of the **Estimation Log.txt** file is a summary which provides information about the relative influence of each independent variable. Here is the result for the SAMPLE1 run (assuming you choose to save draws):

Summary of Analysis

```
         Mean of   Mean of    t Ratio  Mean Sq   Mean Sq    F Ratio
Variable  Beta    Variance   for Mean  Between    Within    Hetero-
          Draws     Draws    of Betas    Cases    Cases     geneity
   1       1.995     0.949    20.479    767.349    0.123    6237.25
   2       0.879     0.978      8.888   773.739    0.143    5404.96
   3       0.049     1.098      0.468   794.375    0.254    3123.81
   4      -1.184     1.191   -10.849    871.575    0.253    3450.51
   5      -2.061     1.051   -20.104    828.806    0.162    5109.51
```

The first column gives point estimates for the means of the distribution of respondents' regression coefficients. This is obtained by averaging the 1000 saved values of alpha which are available in the **Alphas.csv** file. As can be seen, the values are reasonably close to the expected values of 2, 1, 0, -1, and -2, particularly considering that we are dealing with a small sample of only 100 respondents.

The second column gives the estimate of the variance of the distribution of each regression coefficient across respondents. This is obtained by averaging the 1,000 saved estimates for the variances.

The third column gives a t ratio that expresses the difference of each mean from zero. The standard error of the mean, estimated from the variance, furnishes the denominator for the t ratio. The t values give an indication of the *average* importance of each attribute in the regression equation, but reveal nothing about the value of that variable in differentiating among respondents. The importance of each variable in accounting for heterogeneity is given by the last three columns. The Mean Square Between Cases is a measure of the difference among respondents for that regression coefficient. The Mean Square Within Cases is a measure of the amount of random variation within respondents, obtained in different random draws for the same respondents. The ratio of those two variances is an F ratio, which measures the extent to which differences among respondents exceed differences within respondents. As you can see, these F ratios are all very large.

The t and F ratios from this table may be helpful in deciding on the relative importance of each variable. If a variable has a larger-than-average t ratio, or F ratio, it is probably more powerful than average in terms of accounting for variation in the dependent variable. A high t ratio indicates a variable on which individuals tend to agree and a high F ratio indicates a variable on which they disagree, but which is important to them.

Although these t and F ratios provided indications of the relative effects of variables in accounting for overall preference, we do not advocate using them for tests of statistical significance, since they would be biased upwards. The situation is similar to that of cluster analysis, where it is improper to do statistical tests on t or F statistics computed using the same data as was used to find maximally differing groups.

Following the table described above is a second table that reports the proportion of respondents for which a “pseudo-t” ratio is greater than an absolute value of 1.96. Each t value is computed by taking the point estimate of beta for each individual and dividing it by the standard deviation of the draws for that beta for that individual. Although this measure of t is somewhat *ad hoc* (not based on Bayesian theory), it has intuitive appeal.

Variable	% of respondents with t-ratio for beta absolute value > 1.96
1	93.0
2	56.0
3	26.0
4	54.0
5	94.0

A variable that has a significant t-ratio (critical value of 1.96 corresponds to 95% confidence) for very few respondents is of lesser value in the model. Most analysts would agree that variables one and five add substantial value to the model. Variable three is “significant” for 26% of the subjects. In this case, the user must make a judgment call regarding whether to include variable three in further analysis. It may be that these respondents comprise an important segment of the market for which variable three has a significant effect relative to the dependent variable.

8 How Good Are the Results?

8.1 How Good Are the Results?

In this section we present some findings from the analysis of both synthetic and real data sets. For synthetic data sets we already know the "right answers," and we assess HB-Reg's performance by measuring how well it recovers those known parameters. For real data sets we assess performance by the ability to predict holdout choices.

Before reviewing this evidence, we should point out that HB-Reg is one of four Sawtooth Software products that use HB to estimate individual coefficients. Other products are CBC/HB for use in estimating individual part-worths in choice studies using a multinomial logit formulation, ACA/HB for estimating individual part-worths from ACA data using a linear regression formulation, and CVA/HB for estimating individual part-worths for traditional conjoint analysis also using a linear regression formulation. For each of those products we have done a similar performance review, finding HB estimation to be as good or better than the alternative in every case. That evidence is available in three technical papers that can be downloaded from the sawtoothsoftware.com web site (Sawtooth Software 2002, 2003a, 2003b).

Synthetic Data Sets

Note: all the results presented in this section are based on the default settings for the prior covariance matrix as were used in HB-Reg version 2 and earlier.

We first review results from three kinds of synthetic data sets. Each contained 300 respondents, with five independent variables. Although there are some differences in details, each data set was constructed with similar steps:

Average "true" betas of (2, 1, 0, -1, -2) were chosen.

Idiosyncratic betas were chosen for each respondent by perturbing each element of the average betas by random normal heterogeneity with mean of zero and standard deviation of unity.

The independent variables were rectangularly distributed random integers in the range of 1 - 9 and were unique for each individual.

The dependent variable for each observation was constructed by summing the products of each respondent's independent variables for that observation and his/her corresponding beta, and then to that sum adding normal error with mean of zero and standard deviation of unity.

Thus, we created data representing samples of respondents from a population normally distributed with alpha (the mean of the betas) = (2,1,0,-1,-2), variances of unity and covariances of zero, and with sigma of unity.

The number of observations per respondent varied from 1 to 10 and three different data sets were made for each number of observations per respondent. Each data set was used in a separate estimation run. The three results for that number of observations were averaged for reporting.

In the second and third kinds of data sets we introduce colinearity and multiple populations. However, Table 1 provides results for the first analysis, in which the independent variables are independent of one another, and with a single population of respondents.

Table 1: Accuracy of Estimation with
No Colinearity, Single Population of Respondents

	— RMS RECOVERY —			Ind	- EST SDs FOR -	
Obs/Ind	OLS AGG	HB-AGG	HB-IND	Corr	Heter	Error
1	0.313	0.263	0.992	0.826	1.383	5.213
2	0.177	0.087	0.856	0.870	1.062	0.895
3	0.193	0.049	0.717	0.910	1.020	1.093
4	0.187	0.046	0.564	0.947	1.043	1.018
5	0.144	0.019	0.418	0.971	1.033	1.044
6	0.125	0.025	0.319	0.983	1.033	0.976
7	0.122	0.013	0.249	0.989	1.023	0.999
8	0.117	0.009	0.207	0.993	1.049	1.011
9	0.095	0.009	0.175	0.995	0.979	1.009
10	0.078	0.014	0.160	0.996	1.060	0.989

The first three columns of the table measure the success at recovering respondents' betas, individually and in aggregate. Each statistic is a "root mean square" error of estimation, so smaller values are better.

The first column provides an evaluation for ordinary least squares regression, done by pooling all respondents' observations to do aggregate regressions. The parameters that we seek to recover in that regression are the average betas.

The second column provides an evaluation for HB-Reg's ability to recover the same average values. Note that every number in the second column is smaller (more favorable) than the corresponding number in the first column. We may conclude that HB-Reg has done a better job than ordinary regression at estimating the mean of the population, probably because it has been able to separate true heterogeneity from random error. Since ordinary regression is unbiased, we would expect this superiority to decrease as the sample size increases; but it should be noted that 300 is not an unusually small number of respondents for practical research projects.

The third column provides an evaluation for HB-Reg's ability to estimate individual betas. These numbers are quite a lot larger (worse) than the corresponding aggregate numbers. However, they decrease regularly as the amount of information per respondent increases. To assess the magnitude of these errors, consider the fourth column, which gives the average correlation between actual and estimated individual betas. With as few as 3 observations per individual, the average correlation is in the .90's. Of course, success will depend on the amount of error in the data. We have arbitrarily used error of unity. If the data had a different amount of response error, the results could be either better or worse.

The fifth and sixth columns give estimates of heterogeneity and random error, for which the correct values are both unity. The precision of these estimates is already reasonably good with as few as two observations per respondent and continues to improve with more observations.

In this comparison, HB-Reg does a better job than ordinary regression at recovering the average parameters. It is able to make creditable estimates of individual parameters even with as few as three observations per respondent, an accomplishment denied ordinary regression, which requires that there be at least as many observations as parameters to be estimated per individual. However, this data set may be more favorable for HB-Reg than most real data sets would be, because it conforms to the underlying assumptions about heterogeneity and presents no problems due to colinearity.

For the second group of data sets we introduce colinearity. Another 30 data sets were produced in exactly the same way, except that a common random value with large variance was added to the independent variable values for each observation, making the average correlation among independent variables approximately .85. This is a large amount of colinearity, perhaps approximating what might occur in a typical customer satisfaction study. Table 2 provides results for this analysis.

Table 2: Accuracy of Estimation with
Colinearity, Single Population of Respondents

	— RMS RECOVERY—			Ind	-EST SD FOR-	
Obs/Ind	OLS AGG	HB AGG	HB IND	Correl	Heter	Error
1	1.521	1.296	1.668	0.564	6.955	17.766
2	0.833	0.172	0.915	0.846	0.944	2.064
3	0.666	0.067	0.791	0.890	0.941	1.354
4	0.690	0.068	0.669	0.922	0.954	1.426
5	0.487	0.038	0.507	0.957	1.036	1.077
6	0.518	0.028	0.380	0.975	1.007	1.082
7	0.682	0.016	0.312	0.984	1.064	1.004
8	0.455	0.011	0.245	0.990	1.066	1.016
9	0.602	0.014	0.208	0.993	0.968	1.003
10	0.390	0.007	0.177	0.995	1.009	1.001

This table has many similarities to the previous table, but the presence of colinearity has impeded the recovery of the true parameters.

The first column again measures the ability of ordinary regression to recover average betas for the population. The errors are about five times as large as without colinearity and again decrease uniformly as the number of observations increases.

The second column again measures the ability of HB-Reg to recover the same average betas. For a single observation per respondent its error is also about five times as large as without colinearity. But this improves dramatically as the number of observations increases. With only two observations per respondent the error is only about twice as great as without colinearity. With three or more observations per respondent its errors are less than any tabled case for ordinary regression with no colinearity.

The third column, measuring HB-Reg's recovery of individual betas, again shows much larger errors for the case of a single observation per respondent. But these also improve rapidly with increases in the number of observations per respondent. The fourth column shows that with as few as two observations per respondent, average correlations between true and estimated individual betas are in the .80s and with as few as four they are in the .90s. (Again, results would be different with different error levels in the data.)

Finally, the fifth and sixth columns show that estimates of heterogeneity and sigma are much worse than the case without colinearity, but the heterogeneity estimate is quite good with as few as two observations per respondent, and the estimate of sigma becomes reasonable when there are as many observations per individual as parameters estimated.

To summarize we see that colinearity is damaging to both ordinary regression and HB-Reg, although the impact on ordinary regression is much more severe. With as few as two observations per respondent, HB-Reg is able to produce good estimates of the population mean and of the amount of heterogeneity. With three observations per respondent, HB-Reg is able to produce reasonable estimates of individual betas.

For the final set of artificial data sets we explore HB-Reg's ability to deal with data representing a mix of two populations, rather than the single population its model assumes. A group of 30 more synthetic data sets were created which did not have colinearity, but for which half of the respondents had average betas of (2,1,0,-1,-2) and the other half had betas of opposite sign. Thus, average betas for the population were zero, although those would not describe either sub-population accurately.

Table 3: Accuracy of Estimation with
No Colinearity, but Two Populations of Respondents

	— RMS RECOVERY—			Ind	-EST SD FOR-	
Obs/Ind	OLS AGG	HB-AGG	HB-IND	Correl	Heter	Sigma
1	0.307	0.255	1.578	0.412	3.840	2.890
2	0.200	0.164	1.300	0.680	3.058	1.078
3	0.162	0.074	0.981	0.826	3.148	0.708
4	0.155	0.043	0.701	0.913	3.195	0.965
5	0.131	0.034	0.502	0.957	2.900	0.919
6	0.153	0.019	0.352	0.980	3.094	0.987
7	0.138	0.017	0.271	0.988	2.929	1.030
8	0.089	0.014	0.215	0.992	2.989	0.980
9	0.126	0.008	0.181	0.995	3.103	1.000
10	0.097	0.011	0.161	0.996	2.871	0.988

Comparison of the first and second columns of Table 3 shows that HB-Reg is again better able to recover average betas for the population, with a margin of superiority that increases with the number of observations per respondent.

Individual betas are again recovered poorly with few observations per respondent, but with as many as three observations per respondent the average correlation between true and estimated individual betas is in the .80s and with four the average correlation is in the .90s, a performance similar to its performance with colinearity (and which, again, depends on the arbitrary assumption of unit error). With the two sub-populations having betas with opposite signs, the true heterogeneity should be 3, and it is estimated quite accurately with two or more observations, as is the true value of 1 for sigma.

Summarizing these three kinds of data sets, we find that HB-Reg is consistently superior to ordinary least squares regression in its ability to recover the true population mean and its margin of superiority increases with the number of observations per respondent. We find that even in the presence of colinearity or a mixture of normal populations, HB-Reg is able to produce reasonable estimates of individual betas with as few as three observations per respondent. HB-Reg is also able to produce quite reasonable estimates of the true amount of heterogeneity among respondents with as few as two observations per respondent.

HB-Reg is able to produce reasonable estimates of individual betas even when the number of observations per individual is less than the number of parameters to be estimated. Although this is an impressive accomplishment, we believe it is equally impressive that HB-Reg is able to estimate group parameters so accurately. With ordinary regression the analyst is required to pool observations from different respondents and in doing so must confound heterogeneity and error. Recognizing the difference between these sources of variance seems to permit greater accuracy in the estimation of average betas for the population.

Real Data Sets

Note: the results presented in this section are based on the default settings for the prior covariance matrix as were used in HB-Reg version 2 and earlier.

We turn now to two small examples using real data sets, both examples of conjoint analysis.

The first is from a study reported by Orme *et al.* (1997). MBA students from three universities were respondents. The subject of the study was personal computers. Nine attributes were studied, each with two or three levels, using a total of 80 respondents. Each respondent did a full-profile card-sort in which 22 hard-copy cards were sorted into four piles based on preference, and then rated using a 100 point scale. Those ratings were converted to logits, which were used as the dependent variable, both for ordinary least squares regression and also by HB-Reg. In these regressions each respondent contributed 22 observations and a total of 16 parameters were estimated for each, including an intercept.

In addition, each respondent saw five full-profile holdout choice sets, each containing three product concepts. These choice sets were constructed randomly and uniquely for each respondent. Respondents rank ordered the concepts in each set, but the results we report here were based only on first choices. (Hit rates in the original paper are based on implied paired comparisons, whereas those reported here are based on triples and are therefore lower.) We have computed hit rates for predicting holdout choices:

Ordinary Least Squares	72.00%
HB-Reg	73.50%

Neither of these sets of part-worths was constrained so that “obviously better” levels are at least as high as “obviously worse” levels. This can be done easily simply by tying offending pairs of values. Constraining part-worths in that way usually improves their performance in predicting choice. However, since we have not imposed constraints on either set of part-worths, this is a fair comparison, and HB-Reg has a 1.5% margin of superiority.

The second data set is from a study reported by Orme and King (1998) in which 280 individuals responded to an Internet conjoint study of credit cards with four attributes, each with three levels. There were a total of 9 parameters to be estimated for each respondent (including an intercept) and each respondent saw 9 concept cards, each of which was rated for likelihood of signing up on a 5 point scale. This design provided no extra degrees of freedom for error.

Each respondent also answered 9 paired-comparison questions dealing with the same attributes, to test the relative effectiveness of single-concept full-profile conjoint analysis to paired-comparison full-profile conjoint analysis. The authors concluded that the two data collection formats had equivalent performance. We consider only the single-concept data in our comparison.

Each respondent also saw three holdout tasks at the beginning of the survey, in which the preferred concept was selected for each set. The same questions were repeated at the end of the questionnaire, with rotation of concept position. The test-retest reliability was 83%.

We again compare hit rates for predicting holdout choices:

Ordinary Least Squares	78.50%
HB-Reg	79.83%

Again, neither set of part-worths was constrained, although either set could have been. HB-Reg again has a slight margin of superiority.

9 Appendix: Technical Details

9.1 Details of Estimation

We previously attempted to provide an intuitive understanding of the HB estimation process and to avoid complexity our explanation omits some details that we shall provide here.

Gibbs Sampling

The model we wish to estimate has many parameters: an alpha vector of population means, a beta vector for each individual, a **D** matrix of population variances and covariances, and a scalar sigma squared of error variances. Estimating a model with so many parameters is made possible by our ability to decompose the problem into a collection of simpler problems.

As a simple illustration, suppose we have two random variables, x and y for which we want to simulate the joint distribution. We can do so as long as we are able to simulate the distribution of either variable conditionally, given knowledge of the other. The procedure is as follows:

(1) Draw a random value of x
(2) Draw a random value of y, given that value of x
(3) Draw a random value of x, given that value of y
(4) Repeat steps 2 and 3 many times

The paired values of x and y provide a simulation of the joint distribution of x and y. This approximation of the joint distribution by a series of simpler conditional simulations is known as Gibbs Sampling.

With our model we are interested in the joint distribution of alpha, the betas, **D**, and sigma, so our task is a little more complicated, but in principle it is like the two-variable example. We start with arbitrary estimates for each parameter. Then we estimate each of the four types of parameters in turn, conditional on the others. We do this for a very large number of iterations. Eventually the observed distribution of each parameter converges to its true distribution (assuming the model is stated correctly). Then by continuing the process and saving subsequent draws we can capture the distribution of each parameter. Since our model involves normal distributions, the point estimate for each parameter is simply the mean of those random draws.

It remains to specify how the conditional draws are made in each iteration. We differentiate between two types of data: (1) A fixed design matrix for all individuals, and (2) independent variables that can take different values for each individual. The draws of alpha, **D**, and sigma are done the same way, regardless of type of data, but draws of the betas differ. With the first type of data we use “normal draws” for the betas. For the second type of data we use a Metropolis Hastings algorithm, which is more efficient in that case.

Random Draw from a Multivariate Normal Distribution

Many times in the iterative process we must draw random vectors from multivariate normal distributions with specified means and covariances. We first describe a procedure for doing this.

Let $\boldsymbol{\alpha}$ be a vector of means of the distribution and $\mathbf{D}$ be its covariance matrix. $\mathbf{D}$ can always be expressed as the product **T T'** where **T** is a square, lower-triangular matrix. This is frequently referred to as the Cholesky decomposition of $\mathbf{D}$.

Consider two column vectors, **u** and **v** = **T u**. Suppose the elements of **u** are normal and independently distributed with means of zero and variances of unity. Since for large **n**, $1/\mathbf{n}\,\boldsymbol{\Sigma}_{n}\mathbf{u}\ \mathbf{u}'$ approaches the identity, $1/\mathbf{n}\,\boldsymbol{\Sigma}_{n}\mathbf{vv}'$ approaches **D** as shown below:

$$1/\mathbf{n}\,\boldsymbol{\Sigma}_{n}\mathbf{w}' = 1/\mathbf{n}\,\boldsymbol{\Sigma}_{n}\,\mathbf{Tu\ u'T'} = \mathbf{T}\,(1/\mathbf{n}\,\boldsymbol{\Sigma}_{n}\,\mathbf{u\ u'})\mathbf{T}' => \mathbf{T\ T'} = \mathbf{D}$$

where the symbol => means "approaches."

Thus, to draw a vector from a multivariate distribution with mean $\boldsymbol{\alpha}$ and covariance matrix $\mathbf{D}$, we perform a Cholesky decomposition of $\mathbf{D}$ to get **T**, and then multiply **T** by a vector of **u** of independent normal deviates. The vector $\boldsymbol{\alpha}$ **+ T u** is normally distributed with mean $\boldsymbol{\alpha}$ and covariance matrix **D**.

Estimation of Alpha

If there are **n** individuals who are distributed with covariance matrix $\mathbf{D}$, then their mean, $\boldsymbol{\alpha}$, is distributed with covariance matrix 1/**n** $\mathbf{D}$. Using the above procedure, we draw a random vector from the distribution with mean equal to the mean of the current betas, and with covariance matrix 1/**n** $\mathbf{D}$.

Estimation of $\mathbf{D}$

Let **p** be the number of parameters estimated for each of **n** individuals, and let **N** = **n** + **p**. Our prior estimate of $\mathbf{D}$ is the identity matrix **I** of order **p**. We compute a matrix **H** which combines the prior information with current estimates of $\boldsymbol{\alpha}$ and $\mathbf{b}_{\mathbf{i}}$

$$\mathbf{H} = \mathbf{pI} + \boldsymbol{\Sigma}_{n}(\boldsymbol{\alpha} - \boldsymbol{\beta}_{i})\,(\boldsymbol{\alpha} - \boldsymbol{\beta}_{\iota})'$$

We next compute $\mathbf{H}^{-1}$ and the Cholesky decomposition

$$\mathbf{H}^{-1} = \mathbf{T\ T'}$$

Next we generate **N** vectors of independent random values with mean of zero and unit variance, $\mathbf{u}_{\mathbf{i}}$, multiply each by **T**, and accumulate the products:

$$\mathbf{S} = \boldsymbol{\Sigma}_{N}\,(\mathbf{T\ u}_{i})\,(\mathbf{T\ u}_{i})'$$

Finally, our estimate of $\mathbf{D}$ is equal to $\mathbf{S}^{-1}$.

Estimation of Sigma

We draw a value of $\boldsymbol{\sigma}^{2}$ from the inverse Wishart distribution in a way similar to the way we draw

$\mathbf{D}$, except that $\boldsymbol{\sigma}^2$ is a scalar instead of a matrix.

Let **M** be the total number of observations fitted by the model, aggregating over individuals and questions within individual. Let **Q** be the total sum of squared differences between actual and predicted answers for all respondents. Let the scalar $\mathbf{c} = \mathbf{p} + \mathbf{Q}$, analogous to **H** above. We draw **M + p** random normal values, each with mean of zero and standard deviation of unity, multiply each by 1/sqrt(**c**), and accumulate their sum of squares, analogous to **S** above. Our estimate of $\boldsymbol{\sigma}^2$ is the reciprocal of that sum of squares.

Estimation of Betas Using a Metropolis Hastings Algorithm

We now describe the used to draw each new set of betas, done for each respondent in turn. We use the symbol $\mathbf{b_o}$ (for "beta old") to indicate the previous iteration's estimation of an individual's part-worths. We generate a trial value for the new estimate, which we shall indicate as $\mathbf{b_n}$ (for "beta new"), and then test whether it represents an improvement. If so, we accept it as our next estimate. If not, we accept or reject it with probability depending on how much worse it is than the previous estimate.

To get $\mathbf{b_n}$ we draw a random vector **d** of "differences" from a distribution with mean of zero and covariance matrix proportional to $\mathbf{D}$, and let $\mathbf{b_n} = \mathbf{b_o} + \mathbf{d}$. We regard $\mathbf{b_n}$ as a candidate to replace $\mathbf{b_o}$ if it has sufficiently high posterior probability. We evaluate each posterior probability as the product of its density (the prior) and its likelihood.

We first calculate the relative probability of the data, or "likelihood," given each candidate, $\mathbf{b_o}$ and $\mathbf{b_n}$. We do not calculate the actual probabilities, but rather simpler values that are proportional to those probabilities. We first compute the sum of squared differences between the actual answers and our predictions of them, given each set of betas. The two likelihoods are proportional to the respective quantities for $\mathbf{b_o}$ and $\mathbf{b_n}$:

$$\exp[-1/2\ (\text{sum of squared differences})/\ \mathbf{s}^2].$$

Call the resulting values $\mathbf{p_o}$ and $\mathbf{p_n}$, respectively.

We also calculate the relative density of the distribution of the betas corresponding to $\mathbf{b_o}$ and $\mathbf{b_n}$, given current estimates of parameters **a,** $\mathbf{D}$**,** and **s**. Again, we do not compute actual probabilities, but rather simpler values that are proportional to the desired probabilities. This is done by evaluating the following expression for each candidate:

$$\exp[-1/2\ (\text{sum of squared differences})/\ \boldsymbol{\sigma}^2].$$

Call the resulting values $\mathbf{d_o}$ and $\mathbf{d_n}$, respectively.

Finally we then calculate the ratio:

$$\mathbf{r = p_n\ d_n\ / p_o\ d_o}$$

From Bayes' theorem, the posterior probabilities are proportional to the product of the likelihoods times the priors. The values $\mathbf{p_n}$ and $\mathbf{p_o}$ are proportional to the likelihoods of the data given parameter estimates respectively. The values $\mathbf{d_n}$ and $\mathbf{d_o}$ are proportional to the probabilities of drawing those values of $\mathbf{b_n}$ and $\mathbf{b_o}$, respectively, from the distribution of betas, and play the role of priors. Therefore, **r** is the ratio of posterior probabilities of $\mathbf{b_n}$ and $\mathbf{b_o}$, given current estimates of **a**, **D**, and **s**, as well as information from the data.

If **r** is greater than or equal to unity, $\mathbf{b_n}$ has posterior probability greater than or equal to that of $\mathbf{b_o}$, and we accept $\mathbf{b_n}$ as our next estimate of beta for that individual. If **r** is less than unity, then $\mathbf{b_n}$ has posterior probability less than that of $\mathbf{b_o}$. In that case we use a random process to decide whether to accept $\mathbf{b_n}$ or retain $\mathbf{b_o}$ for at least one more iteration. We accept $\mathbf{b_n}$ with probability equal to **r**.

As can be seen, two influences are at work in deciding whether to accept the new estimate of beta. If it fits the data better than the old estimate, then $\mathbf{p_n}$ will be larger than $\mathbf{p_o}$, which will tend to produce a larger ratio. However, the relative densities of the two candidates also enter into the computation, and if one of them has a higher density with respect to the current estimates of **a** and **D**, and **s**, then that candidate has an advantage.

If the densities were *not* considered, then betas would be chosen solely to maximize likelihoods. This would be similar to estimating for each individual separately, and eventually the betas for each individual would converge to a distribution that fits his/her data, without respect to any higher-level distribution. However, since densities are considered, and estimates of the higher-level distribution change with each iteration, there is considerable variation from iteration to iteration. Even after the process has converged, successive estimations of the betas are still quite different from one another. Those differences contain information about the amount of random variation in each individual's betas that best characterizes them.

We mentioned that the vector **d** of differences is drawn from a distribution with mean of zero and covariance matrix proportional to **D**, but we did not specify the proportionality factor. In the literature the distribution from which **d** is chosen is called the "jumping distribution," because it determines the size of the random jump from $\mathbf{b_o}$ to $\mathbf{b_n}$. This scale factor must be chosen well because the speed of convergence depends on it. Jumps that are too large are unlikely to be accepted, and those that are too small will cause slow convergence.

Gelman, Carlin, Stern, and Rubin (p 335) state: "A Metropolis algorithm can also be characterized by the proportion of jumps that are accepted. For the multivariate normal distribution, the optimal jumping rule has acceptance rate around 0.44 in one dimension, declining to about 0.23 in high dimensions … This result suggests an *adaptive* simulation algorithm."

We employ an adaptive algorithm to adjust the average jump size, attempting to keep the acceptance rate near 0.30. The proportionality factor is arbitrarily set at 0.1 initially. For each iteration we count the proportion of respondents for whom $\mathbf{b_n}$ is accepted. If that proportion is less than 0.3, we reduce the average jump size by a tenth of one percent. If that proportion is greater than 0.3, we increase the average jump size by a tenth of one percent. As a result, the average acceptance rate is kept close to the target of 0.30.

The iterative process has two stages. During the first stage, while the process is moving toward convergence, no attempt is made to save any of the results. During the second stage we assume the process has converged, and results for hundreds or thousands of iterations are saved to the hard disk. For each iteration there is a separate estimate of each of the parameters. We are particularly interested in the betas, which are estimates of individuals' betas. We produce point estimates for each individual by averaging the results from many iterations. We can also estimate the variances and covariances of the distribution of respondents by averaging results from the same iterations.

Readers with solid statistical background who are interested in further information about the Metropolis Hastings Algorithm may find the article by Chib and Greenberg (1995) useful.

9.2 Prior Covariance Matrix for HB Estimation

In our work with the CBC/HB system, we have found that for categorical independent variables, there can be a bias in the utility estimate of the "reference" level under HB estimation. The greater the number of levels of the independent variable, and the fewer choice tasks available, the greater the bias (Orme and Lenk, 2004, Lenk and Orme, 2009). Peter Lenk provided a solution that we have used very successfully within CBC/HB, and it involves some adjustments to the Prior Covariance Matrix. We employ those same adjustments for the dummy-coded independent variables within the HB-Reg software (only implemented when you specify Categorical Independent Variables and permit HB-Reg to automatically perform the dummy-coding). This happens automatically, and we refer the interested reader to the relevant documentation on the prior covariance matrix within the CBC/HB documentation.

9.3 Coding Linear and Log-Linear Functions

HB-Reg software can use independent variables that you have pre-specified to use "as-is" in the design matrix (Continuous Independent Variables), or can use Categorical Independent Variables. Some Categorical Independent Variables have nominal meaning for the values, such as red=1, green=2, blue=3. Other Categorical Independent Variables can be treated as metric, with values associated with each level, such as 10lbs = 1, 15lbs = 2, 20lbs = 3.

By default, Categorical Independent Variables are dummy-coded automatically by HB-Reg. But, if the levels truly convey metric meaning (such as weights or prices), then you can request that HB-Reg code them as continuous independent variables with either linear or log-linear functions. If the true shape of the preference function is essentially linear or log-linear, then coding the function in this manner reduces the number of independent variables to be estimated in the model, cuts the time to estimate models, can reduce the likelihood of observing reversals, and can lead to models with higher predictive validity.

Estimating linear or log-linear functions leads to a single parameter (beta coefficient) to be estimated rather than multiple parameters (dummy-coded part-worth functions) that estimate the utility at each price level. In the case of a linear function, the prices for the menu items might be entered as a single column of values in the design matrix, with values such as 1.50, 1.75, 2.00, and 2.25 (HB-Reg further transforms such price values as described further below). For log-linear functions, the natural log of the price values may be used in the design matrix. HB-Reg manages the design matrix for the user, so the data processing and natural log transformations are done automatically (with further zero-centering and normalization steps described below).

In standard regression or logit models, the absolute scaling of independent variables is not consequential to the convergence, fit of the model, or simulator predictions. Whether we use price values of 10, 20, or 30 in the independent variable matrix or 1.0, 2.0, or 3.0, the fit of the model will be the same; only the magnitude of the coefficient will be different, by a factor of 10. But, when using HB, the absolute scaling of the independent variables will affect the convergence of the model and accuracy of the coefficients. Because our HB routine assumes prior means of 0 and prior variances that are uniform across the parameters, it makes it difficult for HB to converge properly if the coded values for independent variables have vastly different variance. So that you can obtain consistently good results and relatively fast convergence, HB-Reg software automatically rescales the linear and log-linear values in the independent variable matrix to be zero-centered, with a range of 1.

For example, consider an independent variable (price) that was shown to respondents as $10000, $20000, and $30000. Let's assume the user has specified that this price variable is a Categorical Independent Variable. The user may specifies that certain *Level Values* are to be associated with those prices. Let's assume the user has associated the values 10000, 20000, 30000 with the three levels of price.

Linear Coding Example

Level #	Shown to Respondents	Assigned Level Values	Transformed Values for IV Matrix
1	$10,000	10000	-0.5

2	$20,000	20000	0.0
3	$30,000	30000	0.5

When the internal files are prepared (built) for HB analysis, values of -0.5, 0.0, and 0.5 are used in the independent variable (IV) matrix rather than 10000, 20000, and 30000. (Accomplished by zero-centering the values and giving them a range of 1.0.) This will obtain faster convergence and unbiased estimates for HB.

Below, we show the final transformed values used in the independent variable matrix when log-linear coding is selected:

Log-Linear Coding Example

Level #	Shown to Respondents	Assigned Level Values	Transformed Values for IV Matrix
1	$10,000	10000	-0.544
2	$20,000	20000	0.088
3	$30,000	30000	0.457

The transformed values used in the independent variable matrix are computed as follows. First, we take the natural log of the assigned level values:

Assigned Level Values	**Natural Log**
10000	**9.210**
20000	**9.903**
30000	**10.309**

Next, we zero-center the transformed values, by subtracting the mean (9.808) from each of the values:

Assigned Level Values	Natural Log	**Centered Values**
10000	9.210	**-0.597**
20000	9.903	**0.096**
30000	10.309	**0.501**

Last, we have found through years of working with HB on CBC datasets that our HB routine converges quicker if the range of the independent variables is a small number, such as 1.0. Currently, the range is 0.501- -0.597 = 1.099. So, we can obtain the desired range if we multiply the centered values above by a factor of 1.0/1.099, leading to the following Final Rescaled Values:

Assigned Level Values	Natural Log	Centered Values	**Final Rescaled Values**
10000	9.210	-0.597	**-0.544**
20000	9.903	0.096	**0.088**
30000	10.309	0.501	**0.457**

The Final Rescaled Values remain zero-centered (sum to zero), and they have a range of 0.457 - -0.544 = 1.0. (There is some rounding error in representing these with 3 decimal places of precision.)

Using transformed values with a range exactly of 1.0 isn't an absolute requirement. In our work, any range scaled in the single digits works reasonably well within our HB estimation (where we've assumed means of zero and uniform prior variance across the parameters). A range of 1.0 seems logical, since any dummy-coded parameters also range from 0 to 1 and have a range of 1.0.

10 References

10.1 References

Allenby, G. M., Arora, N., and Ginter, J. L. (1998) “On the Heterogeneity of Demand,” *Journal of Marketing Research*, 35, (August) 384-89.

Allenby, G. M. and Ginter, J. L. (1995) “Using Extremes to Design Products and Segment Markets,” *Journal of Marketing Research*, 32, (November) 392-403.

Chib, S. and Greenberg, E. (1995) “Understanding the Metropolis-Hastings Algorithm,” American Statistician, 49, (November) 327-35.

Gelman, A., Carlin, J. B., Stern H. S. and Rubin, D. B. (1995) “Bayesian Data Analysis,” Chapman & Hall, Suffolk.

Green, P. E., Krieger, A. M., and Agarwal, M. K. (1991) “Adaptive Conjoint Analysis: Some Caveats and Suggestions,” *Journal of Marketing Research*, 28 (May), 215-22.

Huber, J., Orme B. K., and Miller, R. (1999) “Dealing with Product Similarity in Conjoint Simulations,” *Sawtooth Software Conference Proceedings*, Sawtooth Software, Sequim, 253-66.

Lenk, P. J., DeSarbo, W. S., Green P. E. and Young, M. R. (1996) “Hierarchical Bayes Conjoint Analysis: Recovery of Partworth Heterogeneity from Reduced Experimental Designs,” *Marketing Science*, 15, 173-91.

Orme, B. K., Alpert, M. I. & Christensen, E. (1997) “Assessing the Validity of Conjoint Analysis – Continued,” *Sawtooth Software Conference Proceedings*, Sawtooth Software, Sequim, 209-26.

Orme, B. K. and King, W. C. (1998) “Conducting Full-Profile Conjoint Analysis over the Internet,” Sawtooth Software Technical Paper, accessible from sawtoothsoftware.com web site.

Rossi, P. E., Zvi, G., and Allenby, G. M. (1999) “Overcoming Scale Usage Heterogeneity: A Bayesian Hierarchical Approach.” Paper read at ART Forum, American Marketing Association.

Sawtooth Software (2002) “The CVA/HB Technical Paper,” Technical Paper accessible from sawtoothsoftware.com web site.

Sawtooth Software (2003a) “The CBC/HB Module for Hierarchical Bayes Estimation,” Technical Paper accessible from sawtoothsoftware.com web site.

Sawtooth Software (2003b) “The ACA/HB Module for Hierarchical Bayes Estimation,” Technical Paper accessible from sawtoothsoftware.com web site.

Index