

“Non-Equivalence of Measurement in Latent Variable Modeling of Multigroup Data: A Sensitivity Analysis” by Jouni Kuha and Irini Moustaki (*Psychological Methods*)

Computing instructions

This note describes the files provided as supplementary materials for the article, and explains how they can be used to carry out sensitivity analysis of latent trait and latent class models. For any questions, please contact Jouni Kuha at j.kuha@lse.ac.uk.

1. Software

The following software packages are needed:

- R (see <http://cran.r-project.org/>). Three add-on packages in R are also needed:
 - *MplusAutomation* (<http://cran.r-project.org/web/packages/MplusAutomation/index.html>) for passing information between R and Mplus. This is crucial to the procedure described here. Please see the documentation of *MplusAutomation* if you want more information on the package.
 - *cubature* (<http://cran.r-project.org/web/packages/cubature/index.html>)
 - *mvtnorm* (<http://cran.r-project.org/web/packages/mvtnorm/index.html>)
- Mplus (<https://www.statmodel.com/>). The free Mplus demo version should also work, but only for examples which are sufficiently small for its constraints.
- Microsoft Excel, or something else for editing .csv files.

The data and analyses travel between software as follows:

Excel → R → Mplus → R

R controls the process, templates for the sensitivity scenarios are edited in Excel, and Mplus is used to fit the models.

The R functions for the sensitivity analysis which are included in the supplementary materials draw on a larger suite of functions (also included) for fitting multigroup latent variable models for real data with Mplus and R. These were written by us for a research project called *Latent Variable Modelling of Categorical Data: Tools of Analysis for Cross-National Surveys* (LCAT). Please see <http://stats.lse.ac.uk/lcat/> (under Resources/Computing there) for an explanation of these functions.

2. Files and folders

There are six files in the supplementary materials (plus these instructions). Place them in folders on your computer as follows¹:

- Choose any folder to be the master folder where the R workspace will be located. For this example, suppose that this is called *Sensitivity*. Place the following files there:
 - *nonequivalence_sensitivity_analysis.R* (an R script for an example analysis)
 - *noneqSensitivity_functions.R* (an R script with the necessary functions and objects)
- Create a subfolder called *Templates* (i.e. in this example *Sensitivity/Templates*). Here you place .csv files which provide templates for the sensitivity analyses. The supplementary materials have two such files:
 - *Template_Case_2L8.csv*. This is the template for the latent trait case labelled “2L8” in the article.

¹ We have tested these functions on a Windows computer, but they should also work on a Mac. In case of any problems with file references, please see the R function *sens.lcat* which controls the reading of the template file from Excel.

- *Template_Lclass_2_5.csv*. This is the template for a scenario for a latent class model with four latent classes and five binary items (two of which are really non-equivalent).
- Create a subfolder called *Mplus* (i.e. in this example *Sensitivity/Mplus*). In it, create subfolders corresponding to each of the template files, with the same name as the template (omitting the “Template_”). Place Mplus input files corresponding to these templates into the corresponding folders. For the supplementary materials, this means two files in two folders:
 - *Case_2L8.inp*, placed in subfolder *Mplus/Case_2L8*
 - *Lclass_2_5.inp*, placed in subfolder *Mplus/Lclass_2_5*

3. The template file (.csv)

This specifies the true models in the sensitivity scenarios. It is a comma-limited (.csv) file which can be edited with (for example) Excel. The file is placed in the *Templates* subfolder as explained above. The name of the file is of the form *Template_AAA.csv*, where the AAA stands for the name of the template.

To explain the structure of the template file, consider *Template_Case_2L8.csv* as an example:

- The name of the template (here *Case_2L8*) is given in the [2,2] (2nd row, 2nd column) cell of the template.
- The model type is in [3,3]. For a latent trait model it is *trait*.
- The rows with Y in the first column give information for items, and those with G on groups (countries). The number in the second column is the number of an item or a group. “N” gives within-group sample sizes.

Most of the rest of the table gives parameter values for the true model for the data. If a single number is given, this value is fixed for all cases considered for that template. If a cell has text beginning with “R”, this means parameter values are varied according to a specified rule, as explained below.

For latent trait models, this parameter specification is as follows:

- Each of the items is taken to be binary. On the Item rows of the template, columns 4 and 5 specify the response probability $P(Y_j=1|\eta)$ of each item at two values of the latent trait η (as specified by *traitat*; here $\eta = -1.5$ and $\eta = 1.5$).
- The numbers in the G rows are the means (in column 4) and variances (in column 5) of the latent trait in each of the groups.

For latent class models, the parameter specification is as follows (see *Template_Lclass_2.5.csv* for an example):

- The model type in [3,3] is *class*.
- There is one column per latent class. The numbers in the Item row give (consecutive) numbers for the classes.
- The numbers in the Y rows are probabilities for categories of each item given the classes. Here the items do not need to be binary, and these measurement probabilities are for all but the first level (we call it level 1 here) of an item. So for binary items only one row each, and the numbers are the probabilities $P(Y_j = 2 | \text{Class})$. If, for example, item number 1 has 3 levels, include two rows with “1” in the second column. The probabilities are then $P(Y_1 = 2 | \text{Class})$ and $P(Y_1 = 3 | \text{Class})$. Make sure these probabilities sum to ≤ 1 .
- The numbers in the G rows are the probabilities of latent classes given group. These should sum to 1 along each row.

The cell(s) where the entry is of the form *Rm,<parameters>* indicate that the parameter value corresponding to this cell should be varied according to “rule” number *m*, with parameters of the rule defined by *<parameters>*. One template may have one or more parameters for which a rule like this is defined. The parameters that are varied would normally be measurement parameters. This thus creates a range of true models where the measurement model is non-equivalent to varying extents.

The rules are defined in the object *sens.lcat.rules* in R. Currently defined rules and their parameters are listed at the end of this note. Both template examples in the supplementary materials, and all the scenarios in the article, employ rule 1.

4. Command files in Mplus (.inp)

These give the Mplus commands for fitting the model of full measurement equivalence corresponding to the scenarios in a template. The number and type of the items and the type of the model (latent trait model or class model with the specified number of classes) should match the template.

Each command file is placed in a separate subfolder of the *Mplus* folder. The name of the subfolder and the name of the input (.inp) file should match the name of the template, as explained above.

Please see the files in the supplementary materials for examples of the structure of these input files. In the Names option of the Variable command, it is crucial that the variables are listed in the order that they will be included in the data file exported from R:

- For latent trait models:
 - *group* variable, followed by the items (in the example, *y1-y5*) and frequency weight *freq*
- For latent class models:
 - *group* variable, followed dummy variables for all but the first group (in the example, *d2-d3*), the items (in the example, *y1-y5*) and frequency weight *freq*

5. Fitting the models through R

The sensitivity analysis itself is carried out from R. To see how this is done, open R and then open in R the script file *nonequivalence_sensitivity_analysis.R* and set the working directory to be the folder under which you have created the *Template* and *Mplus* subfolders (the script shows an example of how to do this). Run the commands in the beginning of the script to load (and if needed, install) the add-on packages and to load the functions from the file *noneqSensitivity_functions.R*.

The rest of the script contains examples of how the sensitivity analysis is done, using the two example scenarios provided in the supplementary materials (the first of them, *Case_2L8*, replicates one of the scenarios in the article).

You should only need to use two R functions directly (the rest of the functions are called by them):

- *sens.lcat*: This does most of the work
- *sens.lcat.print*: This prints and plots selected results

The function *sens.lcat* has 3 arguments:

```
> args(sens.lcat)
function (template, n.values = 20, runMplus = TRUE)
```

- *template*: Compulsory. Name of a template, as a string. The corresponding template and Mplus input file must exist in their correct folders. For example, "Case_2L8"
- *n.values*: the number of distinct values over which varying parameters will be varied. For example, in all the examples in the article this is 20 (see the part (c) of Table 1). The default value is 20.
- *runMplus*: logical value, default TRUE (or T). If FALSE (F), the function just reads in the template and prints out it and the values of the varying parameters. This is useful for checking the values.

If *runMplus=T*, the function reads in the template, creates the sequence of parameter values specified for the varying parameters, creates a set of pseudo data which corresponds to each of these settings of parameter values, sends the pseudo data to Mplus to get the parameter values to which a fitted equivalence model will converge, and reassembles the results in R. It is best to save the result in a named R object (this is called *Case.2L8* in the examples below, but it does not need to have the same name as the template).

An example session:

```
Case.2L8 <- sens.lcat("Case_2L8",n.values=20,runMplus=F)
```

- This prints the template for checking, without actually running the analysis.

```
Case.2L8 <- sens.lcat("Case_2L8",n.values=20,runMplus=T)
```

- This runs the analysis and saves the results in the R object *Case.2L8*.
- When doing this for the first time, it is advisable to use a smaller value for *n.values* to check that the connection to Mplus works and to see how long the analysis takes. See the script file for an example.

The resulting object is a list which contains a full description of the true models and the values to which estimates of the fitted equivalence model will converge in each case. The script file shows how you can take a look at the contents of this object. However, usually the best way to extract results from it is by using the function *sens.lcat.print*. It has the following arguments:

```
> args(sens.lcat.print)
function (input, rows = NULL, columns = NULL, plot = F, assigned.classes = F,
  lwd = NULL, lty = NULL, plcol = NULL, legend = T, x = NULL,
  y = NULL, xscale = NULL, xref = NULL, plot.value.at.eq = T,
  ...)
```

- *input*: Compulsory. This is the name of an object produced by *sens.lcat*, e.g. *Case.2L8*.
- *rows*, *columns*: the row and column numbers identifying the parameters to be printed and/or plotted. The numbers are the same as in the template (also printed out by *sens.lcat*). The default values are NULL, which means that all parameter rows and/or columns are used. Otherwise, numbers are specified as usual in R, e.g. *c(6,8,10)* or *6:10*. All combinations of these rows and columns are used; e.g. *rows=6:7*, *columns=4:5* means the cells (6,4), (6,5), (7,4), (7,5).
- *plot*: logical; if TRUE, a plot of the parameter values is drawn
- *assigned.classes*: logical. This applies only if the parameters that are requested are class probabilities in a latent class model. If *assigned.classes=F* (the default), the printed (or plotted) parameters are the corresponding estimated parameters. If *assigned.classes=T*, what is printed are the proportions of different classes when units are assigned to posterior modal classes based on the estimated model.
- The rest of the named arguments affect the appearance of the plot; see examples in the script.
- ...: other parameters, which are passed on to the plotting function. For example, *ylim=c(0,1)* will change the vertical scale to go from 0 to 1.

sens.lcat.print returns a matrix which contains the parameters requested (in the example below, this is assigned the name *r.tmp*). This may be then used for further plots or exported.

For example:

```
r.tmp <- sens.lcat.print(Case.2L8,17:19,4, plot=T, x=1, y=-0.05)
```

Please see the script file for other examples of how this function is used.

Appendix: Template rules defined in the R object `sens.lcat.rules`

- *R1,p,start,end*
 - In Group 1, parameter value is p
 - If there are M groups, in group j the parameter value varies between $a \cdot p + b \cdot \text{start}$ and $a \cdot p + b \cdot \text{end}$ where $a = (M-j) / (M-1)$ and $b = (j-1) / (M-1)$
 - In other words, in Group 1 the parameter is always p , in Group M it varies from *start* to *end*, and in other groups it is ranges between the values for groups 1 and M , with 2 being closest to 1 and $M-1$ closest to M
 - *Note*: *start* does not need to be smaller than *end*, so the sequence can be increasing or decreasing.
 - The number of values for each sequence is set by the argument *n.values* of the R function *sens.lcat*; the default is 20
 - For example, $p = 0.8$, *start* = 0.1, *end* = 0.9 and $M = 3$:
 - in group 1, parameter is always 0.8
 - in group 3, parameter takes the values 0.1, ..., 0.9
 - in group 2, parameter takes the values 0.45 [half-way between 0.1 and 0.8], ..., 0.85
 - if (because of the value of the argument *n.values*) none of the values in this sequence for groups 3 and 2 is equal to 0.8, the nearest value is changed to 0.8; in other words, the sequences always contain one point where measurement equivalence holds

Values in this example, with parameters ranging over 11 values:

```
(*1*): R1,.8,.1,.9
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
G1 0.80 0.80 0.80 0.80 0.80 0.80 0.80 0.80 0.80 0.8 0.80
G2 0.45 0.49 0.53 0.57 0.61 0.65 0.69 0.73 0.77 0.8 0.85
G3 0.10 0.18 0.26 0.34 0.42 0.50 0.58 0.66 0.74 0.8 0.90
```

- *R2,p,start,end,m*
 - If there are M groups,
 - in each of the first $M-m$ groups of them parameter value is p
 - in each of the last m groups, the parameter value varies between *start* and *end*
 - *Note*: *start* does not need to be smaller than *end*, so the sequence can be increasing or decreasing.
 - The number of values for each sequence is set by the argument *n.values* of the R function *sens.lcat*; the default is 20
 - For example, $p = 0.8$, *start* = 0.1, *end* = 0.9, $m=2$ and $M = 6$:
 - in groups 1-4, parameter is always 0.8
 - in groups 5-6, parameter takes the values 0.1, ..., 0.9
 - if (because of the value of the argument *n.values*) none of the values in this sequence for groups 5 and 6 is equal to 0.8, the nearest value is changed to 0.8; in other words, the sequences always contain one point where measurement equivalence holds

Values in this example, with parameters ranging over 11 values:

```
(*1*): R2,.8,.1,.9,2
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
G1 0.8 0.80 0.80 0.80 0.80 0.8 0.80 0.80 0.80 0.8 0.8
G2 0.8 0.80 0.80 0.80 0.80 0.8 0.80 0.80 0.80 0.8 0.8
G3 0.8 0.80 0.80 0.80 0.80 0.8 0.80 0.80 0.80 0.8 0.8
G4 0.8 0.80 0.80 0.80 0.80 0.8 0.80 0.80 0.80 0.8 0.8
G5 0.1 0.18 0.26 0.34 0.42 0.5 0.58 0.66 0.74 0.8 0.9
G6 0.1 0.18 0.26 0.34 0.42 0.5 0.58 0.66 0.74 0.8 0.9
```

- *R3,alpha,start,end,beta*
 - NOTE: This rule only makes sense if it is applied to all latent classes for a given item, each with the same values of alpha (the intercept), start and end, and with different values of beta
 - This parametrises a measurement probability as $1/(\exp[-(\alpha+\beta)])$
 - In Group 1, value of the intercept term alpha is alpha
 - If there are M groups, in group j the value of the intercept varies between $a*\alpha+b*start$ and $a*\alpha+b*end$ where $a = (M-j) / (M-1)$ and $b = (j-1) / (M-1)$
 - In other words, in Group 1 the intercept is always alpha, in Group M it varies from *start* to *end*, and in other groups it is ranges between the values for groups 1 and M, with 2 being closest to 1 and M-1 closest to M
 - In each case the value of the “slope term” beta is the same in all groups. It may, on the other hand, vary between classes.
 - Note: start does not need to be smaller than end, so the sequence can be increasing or decreasing.
 - The number of values for each sequence is set by the argument *n.values* of the R function *sens.cat*; the default is 20
 - See below for a table which can help to choose values for alpha and beta, to get appropriate class probabilities

An example, with 4 classes. Here the parameter alpha is 0 in Group 1, and in other groups ranges over 11 values, in the last group from -1.25 to +1.25. Note that here beta is +1.39, +0.41, -0.41, -1.39 in classes 1-4 respectively, resulting in the measurement probabilities for Group 1 as shown on the first rows below:

```
(*1*): R3,0,-1.25,1.25,1.39
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
G1 0.80 0.80 0.80 0.80 0.80 0.8 0.80 0.80 0.80 0.80 0.80
G2 0.76 0.77 0.78 0.78 0.79 0.8 0.81 0.82 0.82 0.83 0.84
G3 0.71 0.73 0.75 0.77 0.78 0.8 0.82 0.83 0.84 0.86 0.87
G4 0.65 0.69 0.72 0.75 0.78 0.8 0.82 0.84 0.86 0.88 0.89
G5 0.60 0.64 0.69 0.73 0.77 0.8 0.83 0.86 0.88 0.90 0.92
G6 0.53 0.60 0.65 0.71 0.76 0.8 0.84 0.87 0.89 0.92 0.93

(*2*): R3,0,-1.25,1.25,0.41
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
G1 0.60 0.60 0.60 0.60 0.60 0.6 0.60 0.60 0.60 0.60 0.60
G2 0.54 0.55 0.56 0.58 0.59 0.6 0.61 0.62 0.64 0.65 0.66
G3 0.48 0.50 0.53 0.55 0.58 0.6 0.62 0.65 0.67 0.69 0.71
G4 0.42 0.45 0.49 0.53 0.56 0.6 0.64 0.67 0.70 0.73 0.76
G5 0.36 0.40 0.45 0.50 0.55 0.6 0.65 0.69 0.73 0.77 0.80
G6 0.30 0.36 0.42 0.48 0.54 0.6 0.66 0.71 0.76 0.80 0.84

(*3*): R3,0,-1.25,1.25,-0.41
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
G1 0.40 0.40 0.40 0.40 0.40 0.4 0.40 0.40 0.40 0.40 0.40
G2 0.34 0.35 0.36 0.38 0.39 0.4 0.41 0.42 0.44 0.45 0.46
G3 0.29 0.31 0.33 0.35 0.38 0.4 0.42 0.45 0.47 0.50 0.52
G4 0.24 0.27 0.30 0.33 0.36 0.4 0.44 0.47 0.51 0.55 0.58
G5 0.20 0.23 0.27 0.31 0.35 0.4 0.45 0.50 0.55 0.60 0.64
G6 0.16 0.20 0.24 0.29 0.34 0.4 0.46 0.52 0.58 0.64 0.70

(*4*): R3,0,-1.25,1.25,-1.39
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
G1 0.20 0.20 0.20 0.20 0.20 0.2 0.20 0.20 0.20 0.20 0.20
G2 0.16 0.17 0.18 0.18 0.19 0.2 0.21 0.22 0.22 0.23 0.24
G3 0.13 0.14 0.16 0.17 0.18 0.2 0.22 0.23 0.25 0.27 0.29
G4 0.11 0.12 0.14 0.16 0.18 0.2 0.22 0.25 0.28 0.31 0.35
G5 0.08 0.10 0.12 0.14 0.17 0.2 0.23 0.27 0.31 0.36 0.40
G6 0.07 0.08 0.11 0.13 0.16 0.2 0.24 0.29 0.35 0.40 0.47
```

- *R4,alpha,start,end,beta,m*
 - NOTE: This rule only makes sense if it is applied to all latent classes for a given item, each with the same values of alpha (the intercept), start, end and m, and with different values of beta
 - This is like Rule 3 in that it specifies non-equivalence in the intercept term.
 - This is like Rule 2 in that the first M groups have the baseline measurement model, and the last m groups a different model, same for all of them

An example, with 4 classes. Here the parameter alpha is 0 in Groups 1-4, and in other Groups 5-6 it ranges over 11 values, from -1.25 to +1.25. Note that here beta is +1.39, +0.41, -0.41, -1.39 in classes 1-4 respectively, resulting in the measurement probabilities for Groups 1-4 as shown on the first rows below:

```
(*1*): R4,0,-1.25,1.25,1.39,2
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
G1 0.80  0.8 0.80 0.80 0.80  0.8 0.80 0.80 0.80  0.80  0.80
G2 0.80  0.8 0.80 0.80 0.80  0.8 0.80 0.80 0.80  0.80  0.80
G3 0.80  0.8 0.80 0.80 0.80  0.8 0.80 0.80 0.80  0.80  0.80
G4 0.80  0.8 0.80 0.80 0.80  0.8 0.80 0.80 0.80  0.80  0.80
G5 0.53  0.6 0.65 0.71 0.76  0.8 0.84 0.87 0.89  0.92  0.93
G6 0.53  0.6 0.65 0.71 0.76  0.8 0.84 0.87 0.89  0.92  0.93

(*2*): R4,0,-1.25,1.25,0.41,2
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
G1 0.6 0.60 0.60 0.60 0.60  0.6 0.60 0.60 0.60  0.6 0.60
G2 0.6 0.60 0.60 0.60 0.60  0.6 0.60 0.60 0.60  0.6 0.60
G3 0.6 0.60 0.60 0.60 0.60  0.6 0.60 0.60 0.60  0.6 0.60
G4 0.6 0.60 0.60 0.60 0.60  0.6 0.60 0.60 0.60  0.6 0.60
G5 0.3 0.36 0.42 0.48 0.54  0.6 0.66 0.71 0.76  0.8 0.84
G6 0.3 0.36 0.42 0.48 0.54  0.6 0.66 0.71 0.76  0.8 0.84

(*3*): R4,0,-1.25,1.25,-0.41,2
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
G1 0.40  0.4 0.40 0.40 0.40  0.4 0.40 0.40 0.40  0.40  0.4
G2 0.40  0.4 0.40 0.40 0.40  0.4 0.40 0.40 0.40  0.40  0.4
G3 0.40  0.4 0.40 0.40 0.40  0.4 0.40 0.40 0.40  0.40  0.4
G4 0.40  0.4 0.40 0.40 0.40  0.4 0.40 0.40 0.40  0.40  0.4
G5 0.16  0.2 0.24 0.29 0.34  0.4 0.46 0.52 0.58  0.64  0.7
G6 0.16  0.2 0.24 0.29 0.34  0.4 0.46 0.52 0.58  0.64  0.7

(*4*): R4,0,-1.25,1.25,-1.39,2
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
G1 0.20 0.20 0.20 0.20 0.20  0.2 0.20 0.20 0.20  0.2 0.20
G2 0.20 0.20 0.20 0.20 0.20  0.2 0.20 0.20 0.20  0.2 0.20
G3 0.20 0.20 0.20 0.20 0.20  0.2 0.20 0.20 0.20  0.2 0.20
G4 0.20 0.20 0.20 0.20 0.20  0.2 0.20 0.20 0.20  0.2 0.20
G5 0.07 0.08 0.11 0.13 0.16  0.2 0.24 0.29 0.35  0.4 0.47
G6 0.07 0.08 0.11 0.13 0.16  0.2 0.24 0.29 0.35  0.4 0.47
```

- *R5,alpha,start,end,beta,x*
 - This is the same as Rule 3, but applied to latent trait models. The extra argument x specifies the value of the trait at which the measurement probabilities are evaluated.

For rules 3 and 4:

To select values of alpha and beta, the following tables may be useful. The numbers inside the table are probabilities

	beta									
alpha	-2.94	-2.20	-1.73	-1.39	-1.10	-0.85	-0.62	-0.41	0.20	0.00
-2.25	0.01	0.01	0.02	0.03	0.03	0.04	0.05	0.07	0.08	0.10
-2.00	0.01	0.01	0.02	0.03	0.04	0.05	0.07	0.08	0.10	0.12
-1.75	0.01	0.02	0.03	0.04	0.05	0.07	0.09	0.10	0.12	0.15
-1.50	0.01	0.02	0.04	0.05	0.07	0.09	0.11	0.13	0.15	0.18
-1.25	0.01	0.03	0.05	0.07	0.09	0.11	0.13	0.16	0.19	0.22
-1.00	0.02	0.04	0.06	0.08	0.11	0.14	0.17	0.20	0.23	0.27
-0.75	0.02	0.05	0.08	0.11	0.14	0.17	0.20	0.24	0.28	0.32
-0.50	0.03	0.06	0.10	0.13	0.17	0.21	0.25	0.29	0.33	0.38
-0.25	0.04	0.08	0.12	0.16	0.21	0.25	0.30	0.34	0.39	0.44
0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
0.25	0.06	0.12	0.19	0.24	0.30	0.35	0.41	0.46	0.51	0.56
0.50	0.08	0.15	0.23	0.29	0.35	0.41	0.47	0.52	0.57	0.62
0.75	0.10	0.19	0.27	0.35	0.41	0.48	0.53	0.58	0.63	0.68
1.00	0.13	0.23	0.33	0.40	0.48	0.54	0.59	0.64	0.69	0.73
1.25	0.16	0.28	0.38	0.47	0.54	0.60	0.65	0.70	0.74	0.78
1.50	0.19	0.33	0.44	0.53	0.60	0.66	0.71	0.75	0.79	0.82
1.75	0.23	0.39	0.50	0.59	0.66	0.71	0.76	0.79	0.82	0.85
2.00	0.28	0.45	0.57	0.65	0.71	0.76	0.80	0.83	0.86	0.88
2.25	0.33	0.51	0.63	0.70	0.76	0.80	0.84	0.86	0.89	0.90

	beta									
alpha	0.00	0.20	0.41	0.62	0.85	1.10	1.39	1.73	2.20	2.94
-2.25	0.10	0.11	0.14	0.16	0.20	0.24	0.30	0.37	0.49	0.67
-2.00	0.12	0.14	0.17	0.20	0.24	0.29	0.35	0.43	0.55	0.72
-1.75	0.15	0.18	0.21	0.24	0.29	0.34	0.41	0.50	0.61	0.77
-1.50	0.18	0.21	0.25	0.29	0.34	0.40	0.47	0.56	0.67	0.81
-1.25	0.22	0.26	0.30	0.35	0.40	0.46	0.53	0.62	0.72	0.84
-1.00	0.27	0.31	0.36	0.41	0.46	0.52	0.60	0.67	0.77	0.87
-0.75	0.32	0.37	0.42	0.47	0.52	0.59	0.65	0.73	0.81	0.90
-0.50	0.38	0.43	0.48	0.53	0.59	0.65	0.71	0.77	0.85	0.92
-0.25	0.44	0.49	0.54	0.59	0.65	0.70	0.76	0.81	0.88	0.94
0.00	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
0.25	0.56	0.61	0.66	0.70	0.75	0.79	0.84	0.88	0.92	0.96
0.50	0.62	0.67	0.71	0.75	0.79	0.83	0.87	0.90	0.94	0.97
0.75	0.68	0.72	0.76	0.80	0.83	0.86	0.89	0.92	0.95	0.98
1.00	0.73	0.77	0.80	0.83	0.86	0.89	0.92	0.94	0.96	0.98
1.25	0.78	0.81	0.84	0.87	0.89	0.91	0.93	0.95	0.97	0.99
1.50	0.82	0.85	0.87	0.89	0.91	0.93	0.95	0.96	0.98	0.99
1.75	0.85	0.88	0.90	0.91	0.93	0.95	0.96	0.97	0.98	0.99
2.00	0.88	0.90	0.92	0.93	0.95	0.96	0.97	0.98	0.99	0.99
2.25	0.90	0.92	0.93	0.95	0.96	0.97	0.97	0.98	0.99	0.99