R installation and mplus.R source

1. Download R for Windows from CRAN at http://cran.r-project.org/bin/windows/base/ and follow the installation instructions.
2. Download the R source code, mplus.R.
3. Open R. In Windows, go to Start -> Programs -> R.
4. Under the **File** menu, choose the **Source R code…** option. Browse to the folder with the mplus.R source code. Select the mplus.R file and click on the Open button. In the R Console window, the following lines will appear:

```
> source("C:\\MplusWork\\Develop\\Rdev\\mplus.R")
[1] "Loaded rhdf5 package"
```

The Mplus R functions and the rhdf5 package

The Mplus R functions use the rhdf5 package from Bioconductor to read the Mplus GH5 file. The package should automatically load after sourcing the mplus.R file. You can see a list of Mplus R functions added from the mplus.R source file by typing the following command:

```
> ls()
```

All functions above require that the name of the Mplus GH5 file be specified as the first argument. The full path to the file must be given. An alternative to this is to change the working directory in R and then refer to the filename instead of the full path of the file. To change the working directory in R in Windows, go the **File** menu and choose the **Change dir…** option. On the Mac, go to the **Misc** menu and choose the **Change Working Directory…** option. In the dialog box, browse to the desired directory or create a new folder. The R **setwd()** command can also be used to change the working directory. When specifying the Mplus GH5 file, be sure to surround the filename (or full path) with quotes.
The remaining sections of the tutorial will describe the various R functions to get various plots. Following is the list of sections:

View plots

The function mplus.view.plots lists all available functions for viewing various plots.

- mplus.view.plots(filename)

```
> mplus.view.plots('ex3.1.gh5')

Plot functions:
 - mplus.plot.histogram('ex3.1.gh5',variable,bins)
 - mplus.plot.scatterplot('ex3.1.gh5',xvar,yvar)

Plot data extraction functions:
 - mplus.list.variables('ex3.1.gh5')
 - mplus.get.data('ex3.1.gh5',variable)
```

Histograms and scatterplots

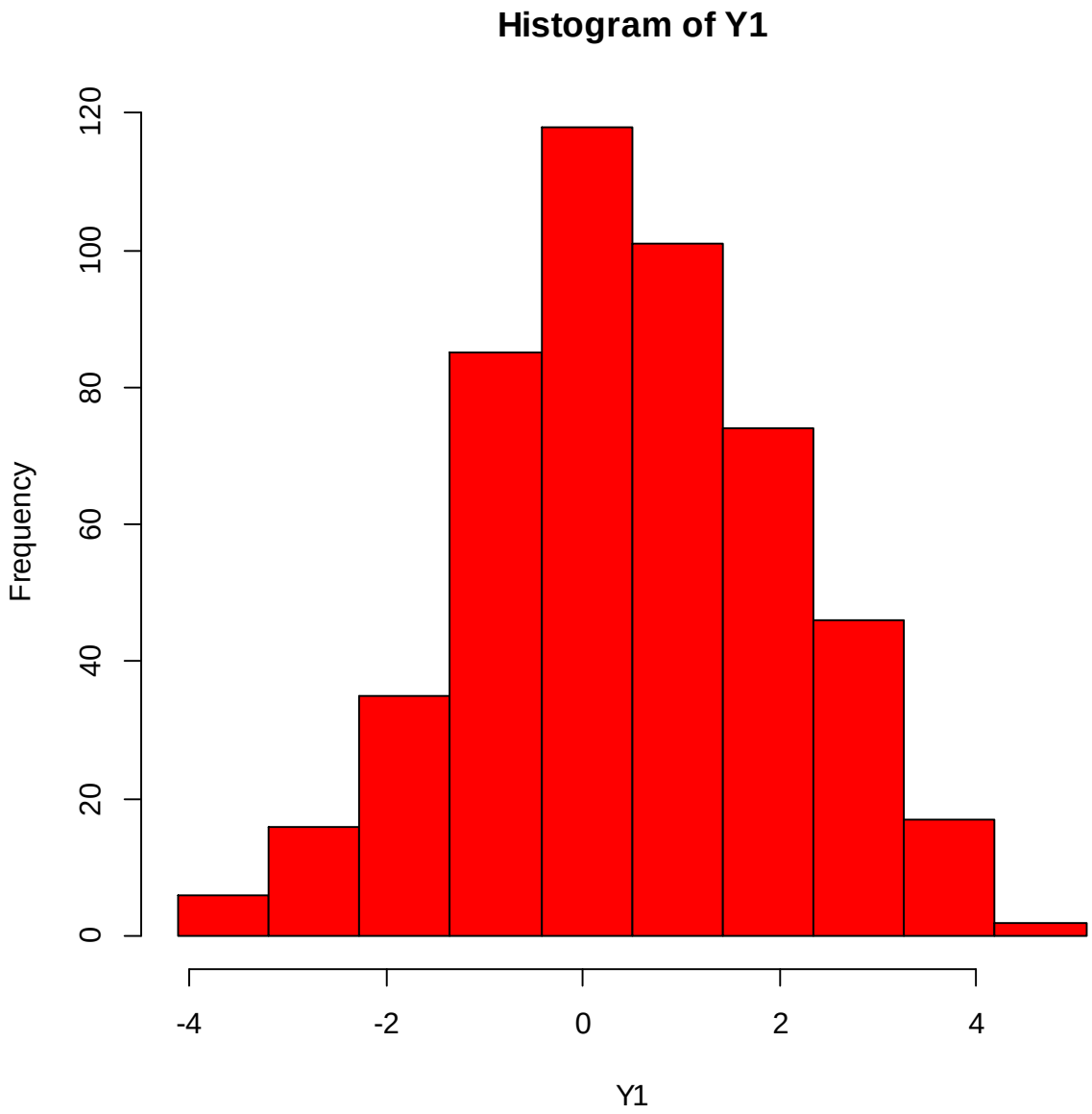
The following functions are used to view basic plots such as histograms and scatterplots. These plots require that the settings PLOT1 or PLOT3 are requested for the TYPE option in the PLOT command in the Mplus input.

- mplus.plot.histogram(filename,variable,bins)
- mplus.plot.scatterplot(filename,xvariable,yvariable)

The variable argument for the mplus.plot.histogram function refers to the name of a variable for which a histogram is to be plotted. The xvariable argument in the mplus.plot.scatterplot function refers to the name of the variable for the x-axis in the scatterplot; the yvariable argument refers to the name of the variable for the y-axis in the scatterplot. To see a list of variable names that can be given for these two functions, use the function mplus.list.variables.

- mplus.list.variables(filename)

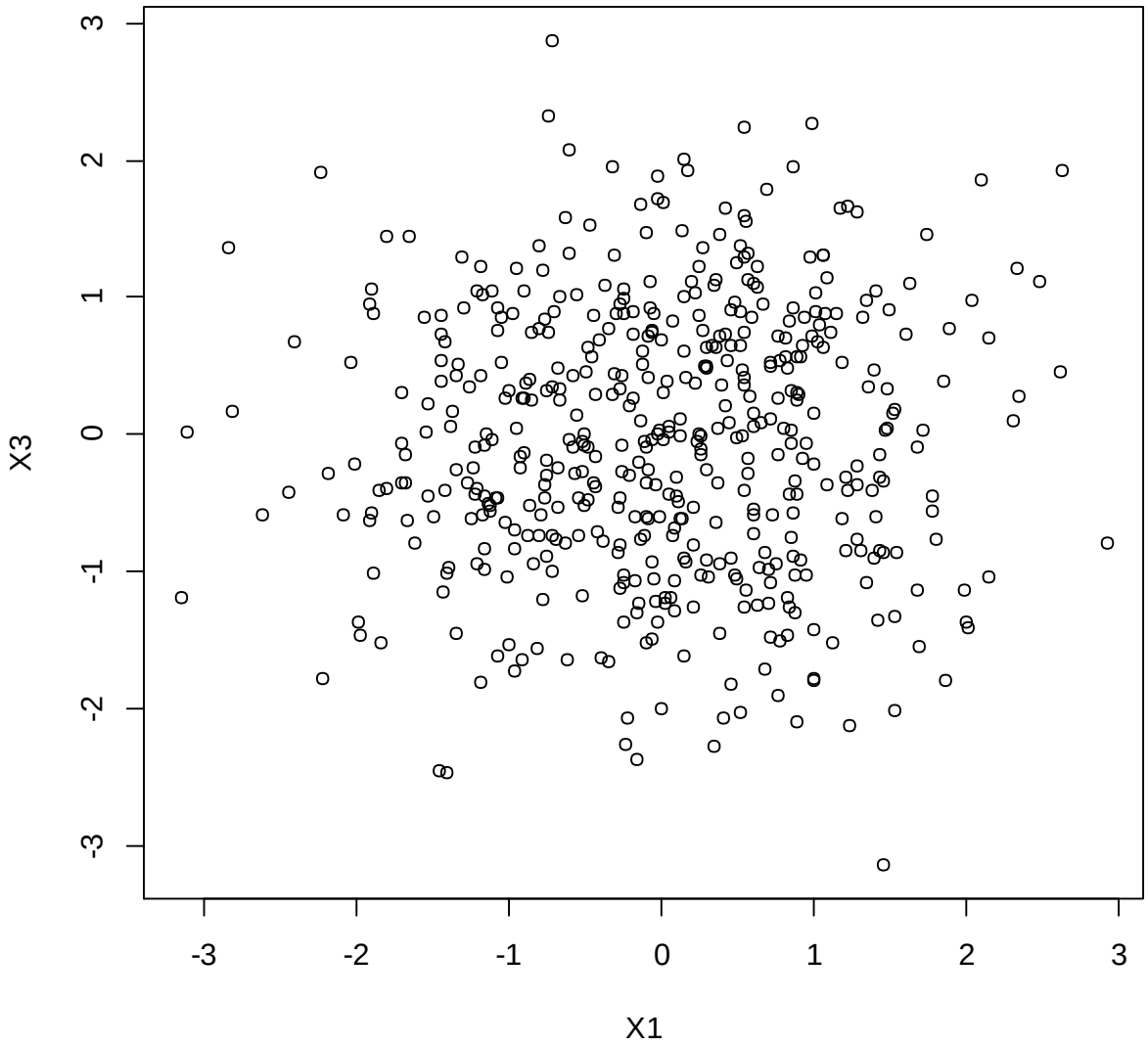
```
> mplus.list.variables('ex3.1.gh5')

List of variable names to use in the following functions:
 - mplus.plot.histogram
 - mplus.plot.scatterplot
 - mplus.get.data

Variables:
[1] "Y1" "X1" "X3"

> mplus.plot.histogram('ex3.1.gh5','y1')
```



The default number of bins is 10. To change the number of bins, give the desired number of bins as the last argument to the function. This is useful for categorical variables where the number of bins would be the number of categories.

When a plot function is used, an R Graphics window will open showing the plot. If you type a new function, R does not open up another graphics window. Instead, the previous plot will be replaced by the current plot.

```
> mplus.plot.scatterplot('ex3.1.gh5','x1','x3')
```



Plots in R can be saved or copied. Right-click on the R Graphics window to bring up a context menu giving the options to copy the plot as metafile or bitmap or to save as metafile or postscript.

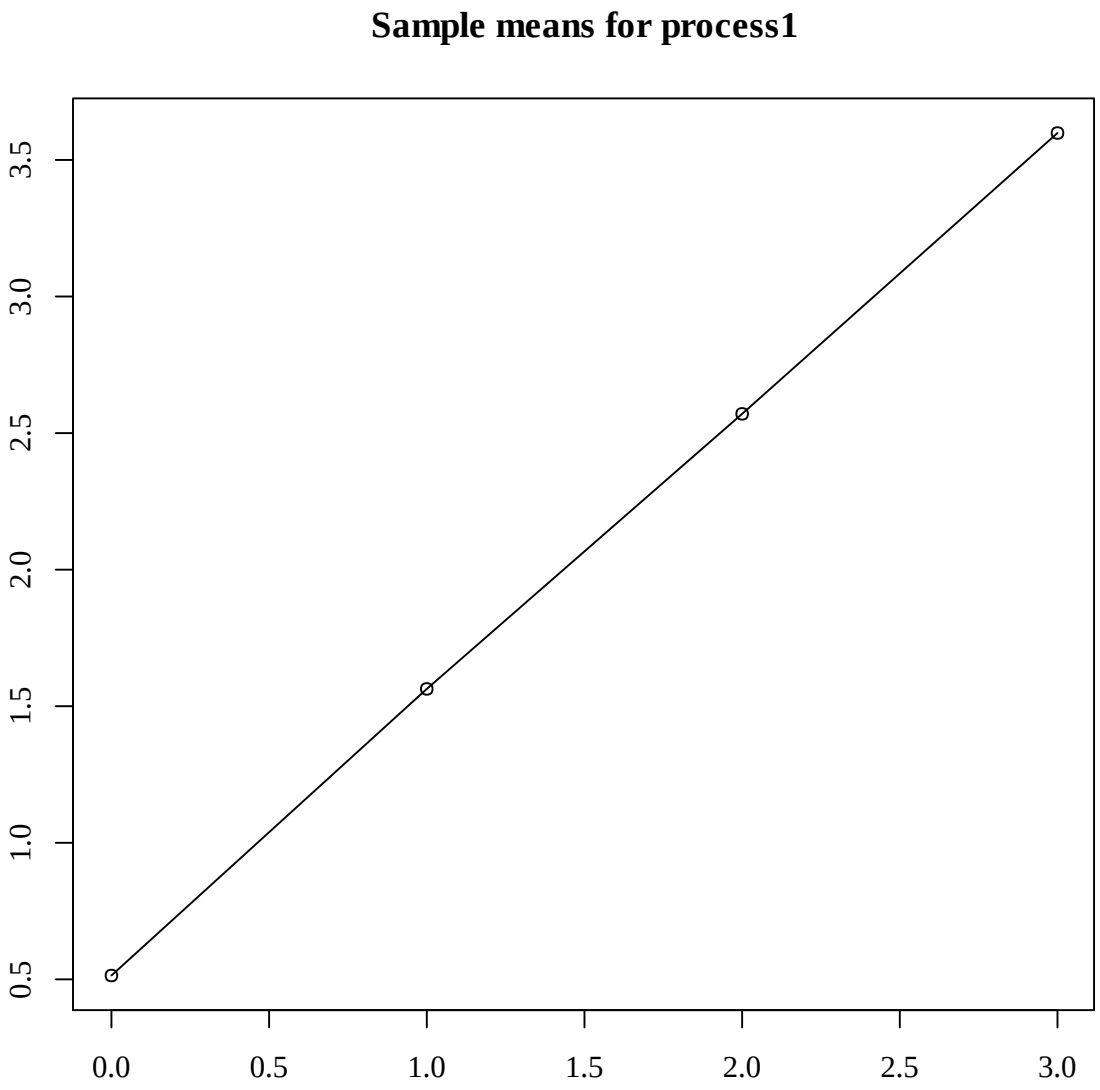
Means and probabilities

The following functions are used to get line plots of sample and estimated means for continuous variables. These plots require that the SERIES option is specified in the PLOT command and PLOT2 or PLOT3 setting is given for the TYPE option.

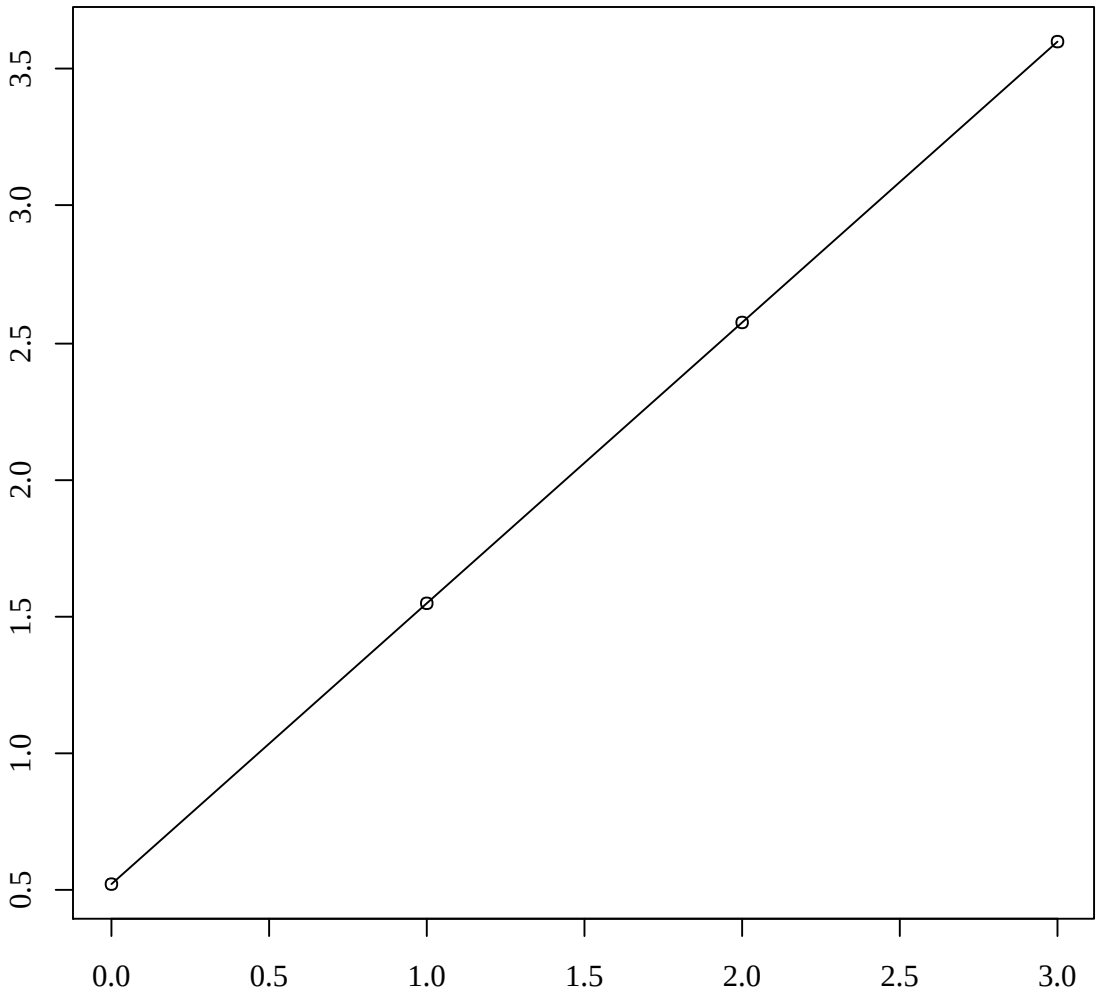
- mplus.plot.sample_means(file, process)
- mplus.plot.estimated_means(file, process)
- mplus.plot.sample_and_estimated_means(file, process)

The process argument refers to the process name given for each set of variables given in the SERIES option. The process names are “process1”, “process2”, etc. The process name is a character string and must be quoted when given as an argument. The process argument is not required. The default is “process1”.

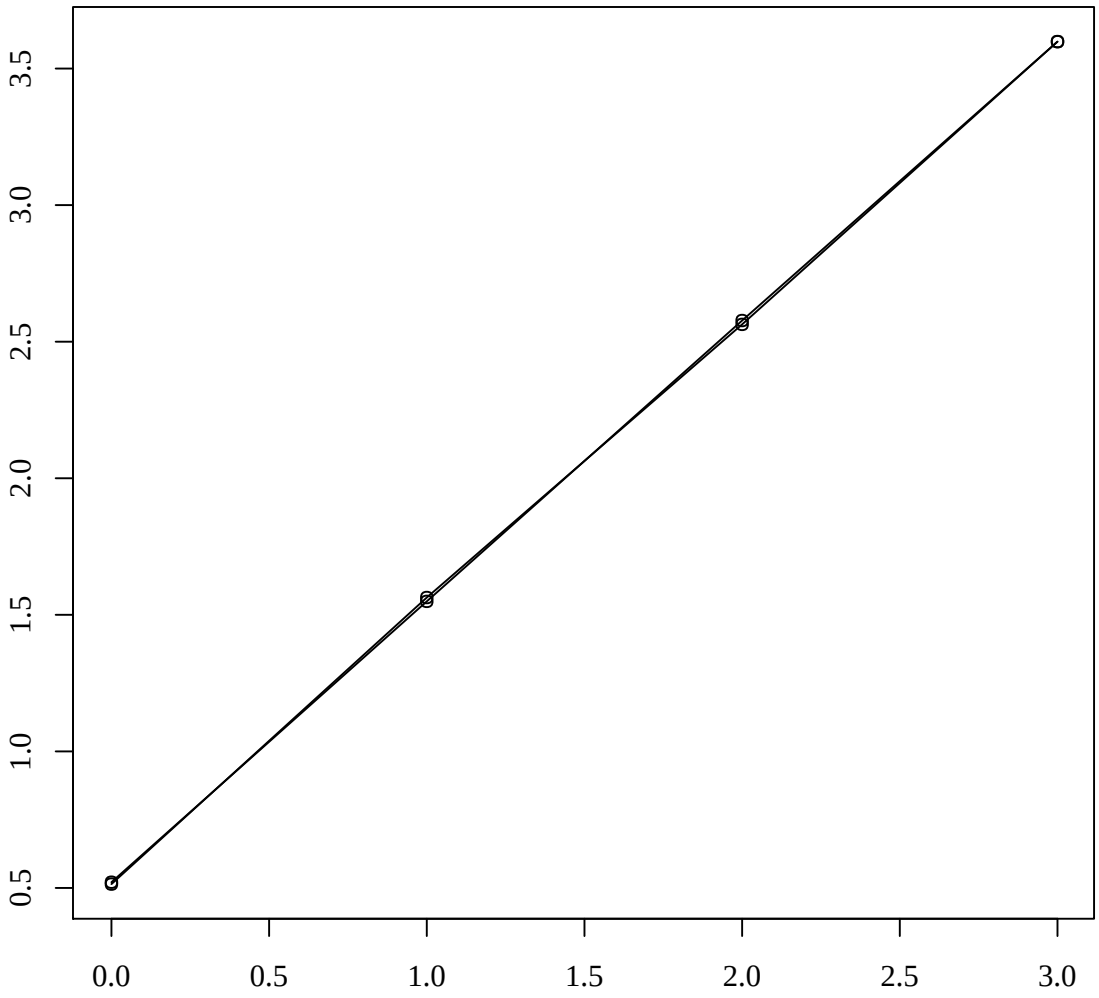
```
> mplus.plot.sample_means('ex6.1.gh5')
> mplus.plot.sample_means('ex6.1.gh5', 'process1')
```




```
> mplus.plot.estimated_means('ex6.1.gh5')
> mplus.plot.estimated_means('ex6.1.gh5','process1')
```




```
> mplus.plot.sample_and_estimated_means('ex6.1.gh5')
> mplus.plot.sample_and_estimated_means('ex6.1.gh5','process1')
```



The following functions can be used to get the data plotted in these plots.

- mplus.get.sample_means(file, process)
- mplus.get.estimated_means(file, process)
- mplus.get.time_scores(file, process)

The process argument is as described above for functions such as mplus.plot.sample_means. The function mplus.get.time_scores is used to get the time scores given in the SERIES option or estimated time scores plotted as the x-values.

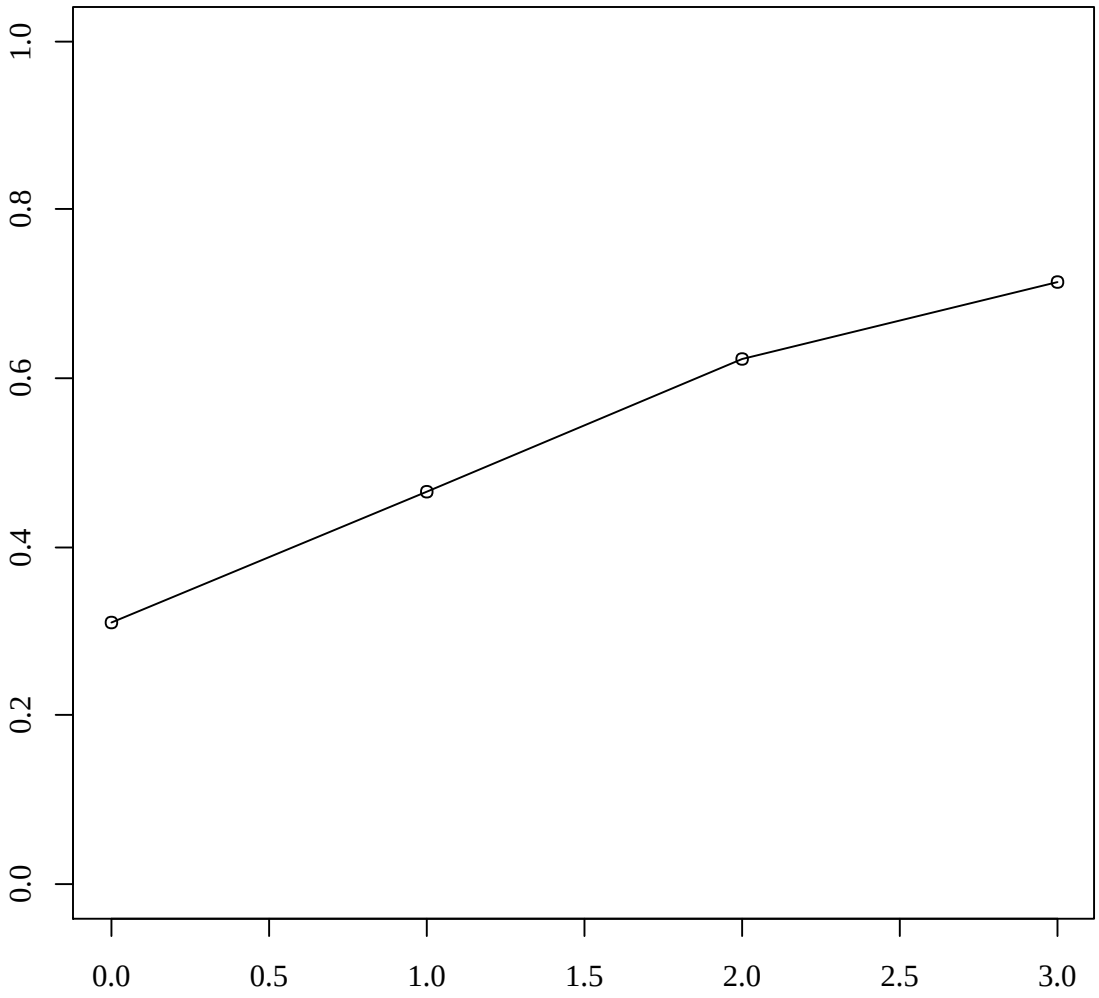
Similar to means for continuous variables, there are functions to get line plots of sample proportions and estimated probabilities for categorical variables. Again, the SERIES option must be specified in the PLOT command. The following functions provide these plots:

- mplus.plot.sample_proportions(file, process, cat1, cat2)
- mplus.plot.estimated_probabilities(file, process, cat1, cat2)

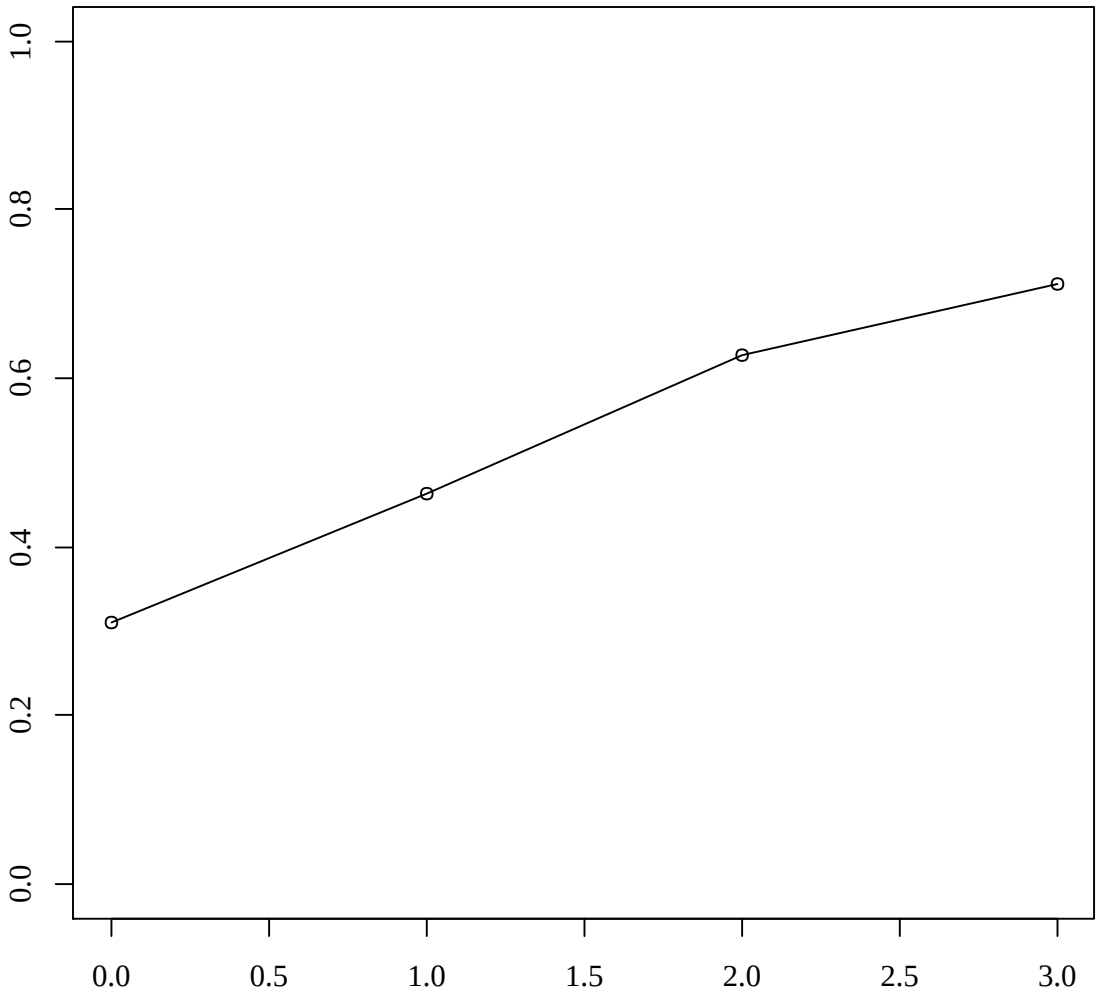
The process argument is as described above for functions such as mplus.plot.sample_means. The cat1 and cat2 arguments refer to the categories to be plotted. The plot can show probabilities for one category or a sum of the probabilities for a range of categories. The cat1 and cat2 arguments are not required. If both are not given, the default is the first category (1). If cat1 is given and cat2 is not given, then cat2 will be the same as cat1.

```
> mplus.plot.sample_proportions('ex6.4.gh5')
> mplus.plot.sample_proportions('ex6.4.gh5',cat1=1)
> mplus.plot.sample_proportions('ex6.4.gh5','process1',1)
> mplus.plot.sample_proportions('ex6.4.gh5','process1',1,1)
```

The four function calls above are equivalent and shown in the plot below.




```
> mplus.plot.estimated_probabilities('ex6.4.gh5')
> mplus.plot.estimated_probabilities('ex6.4.gh5',cat1=1)
> mplus.plot.estimated_probabilities('ex6.4.gh5','process1',1)
> mplus.plot.estimated_probabilities('ex6.4.gh5','process1',1,1)
```

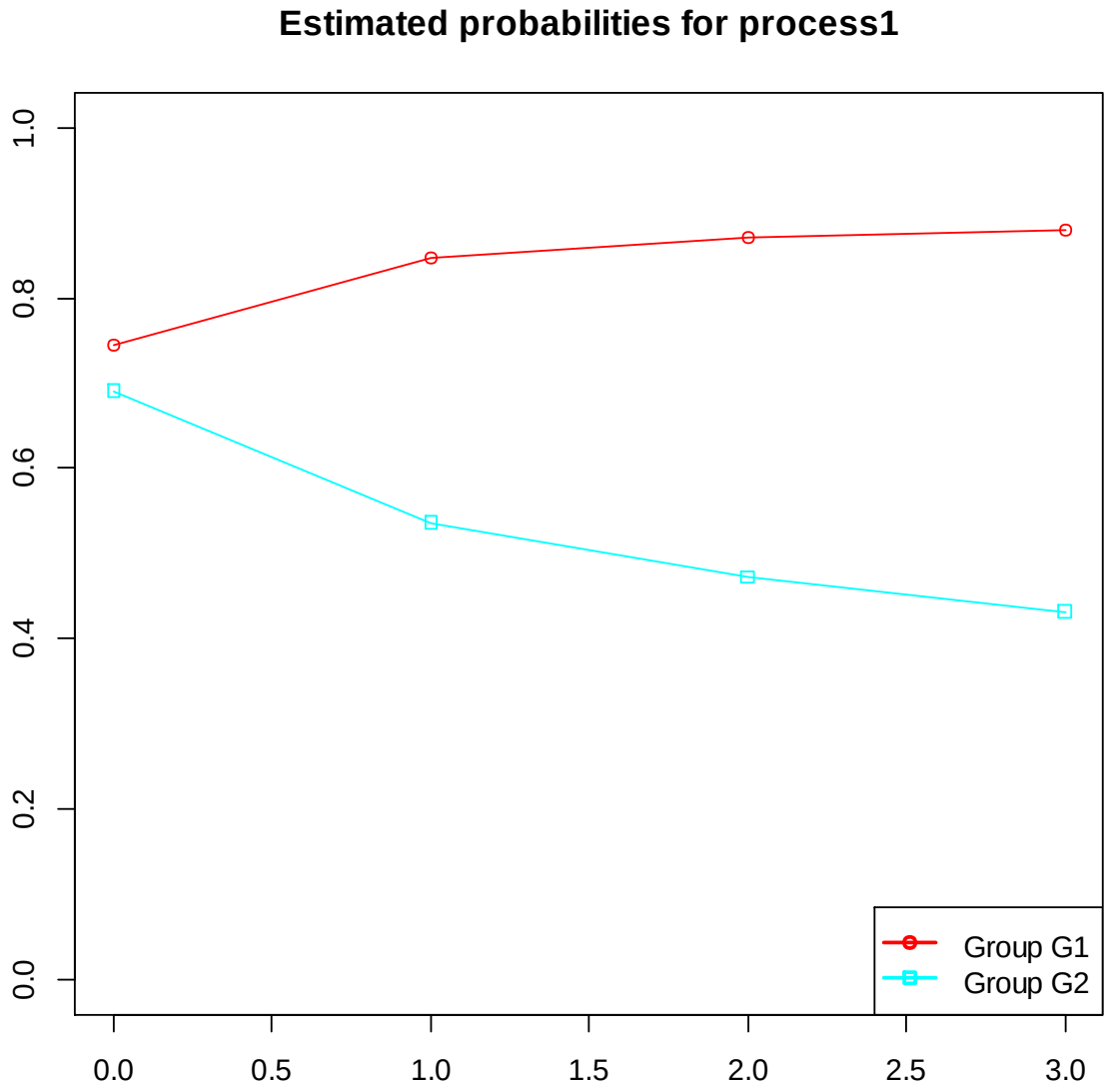


The following function plots the sum of the estimated probabilities for category 1 to category 3. The catpoly example is a 2-group example so the plot contains 2 line curves for each of the groups.

```
> mplus.plot.estimated_probabilities('catpoly.gh5',cat1=1,cat2=3)
```

The following functions are used to get the sample proportions and estimated probabilities shown in the plots:

- mplus.get.sample_proportions(file, process, cat1, cat2)
- mplus.get.estimated_probabilities(file, process, cat1, cat2)
- mplus.get.time_scores(file, process)



All arguments are as described above for functions such as mplus.plot.sample_proportions. Note that these functions give the information for all groups in a multiple group analysis or all classes in a Mixture analysis. The function mplus.get.time_scores is described in the section for getting sample and estimated means.

IRT plots

The following functions are used to get plots of item characteristic curves and item information curves. The settings PLOT2 or PLOT3 must be given for the TYPE option in the PLOT command. Models with IRT plots will be automatically determined by Mplus.

- mplus.plot.irt.icc(file,group,xvar,uvar,cat,cat2,covariates,xrange,xstep,lloc)

- mplus.plot.irt.iic(file,group,xvar,uvar,covariates,xrange,xstep,lloc)
- mplus.plot.irt.tic(file,group,xvar,uvar,covariates,xrange,xstep)
- mplus.list.irt.xvariables(file)
- mplus.list.irt.variables(file)

The group argument specifies the group number to plot. For Mixture analysis, this is the class number. By default, the group argument is 1.

The xvar argument for the three plot functions specifies the variable that should appear on the x-axis. To view a list of possible variables for the x-axis, use the function mplus.list.irt.xvariables. These are the required arguments. The default variable for the x-axis is the first variable in the list.

The uvar argument refers to the item or items for which the IRT curves are plotted. If the uvar argument is not given, then all items are plotted. The uvar argument can be one variable or variable index or a list of variables or variable indices. To view a list of item variables, use the function mplus.list.irt.variables. When the variable index is used, the index refers to the variable ordering in this list.

The function mplus.plot.irt.icc can be used to plot all, a specific category, or the sum of a range of categories for a particular item. The cat argument refers to the first category in a range. The cat2 argument refers to the last category in the range. If no categories are specified, all categories will be plotted. If only cat argument is specified, then only that category is plotted. If cat and cat2 arguments are given, then the plot shows the sum of probabilities from cat to cat2.

The xrange argument is used to specify the range of the variable on the x-axis. The xrange argument must be a value between 1 and 6. The default is xrange=3. The following is the range for each of the values:

- xrange=1: -1 s.d to +1 s.d of the x-axis variable
- xrange=2: -2 s.d to +2 s.d of the x-axis variable
- xrange=3: -3 s.d to +3 s.d of the x-axis variable
- xrange=4: -4 s.d to +4 s.d of the x-axis variable
- xrange=5: -5 s.d to +5 s.d of the x-axis variable
- xrange=6: -6 s.d to +6 s.d of the x-axis variable

The xstep argument is used to specify the step increments for the x-axis range. The xstep argument must be a value between 1 and 11. The default is xstep=7. The following is the step for each of the values:

- xstep=1: 1.0
- xstep=2: 0.5
- xstep=3: 0.1
- xstep=4: 0.05
- xstep=5: ½ s.d of x-axis variable
- xstep=6: ¼ s.d of x-axis variable
- xstep=7: 1/5 s.d of x-axis variable
- xstep=8: 1/10 s.d of x-axis variable
- xstep=9: 1/20 s.d of x-axis variable
- xstep=10: 1/50 s.d of x-axis variable
- xstep=11: 1/100 s.d of x-axis variable

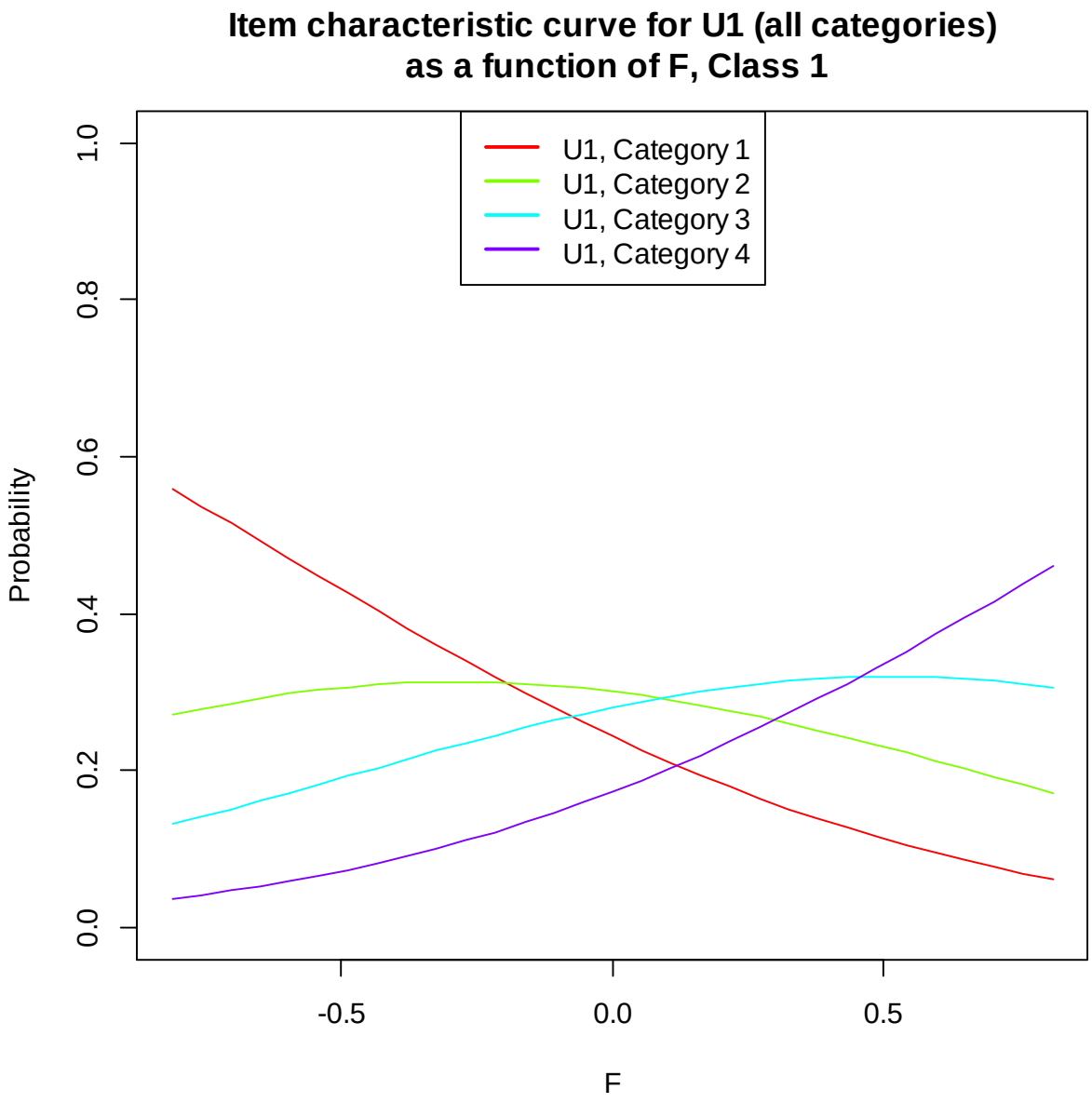
The lloc argument is used to specify the placement of the legend in plots that have a legend. The lloc argument is a quoted string and must be one of the predefined values for the legend function in R. The default is lloc="top". The following are possible settings for the lloc argument:

- lloc="top": The legend is centered horizontally at the top of the plot area.
- lloc="bottom": The legend is centered horizontally at the bottom of the plot area.
- lloc="left": The legend is centered vertically and starts from the left edge of the plot area.
- lloc="right": The legend is centered vertically and starts from the right edge of the plot area.
- lloc="center": The legend is centered in the plot area.
- lloc="topleft"
- lloc="topright"
- lloc="bottomleft"
- lloc="bottomright"

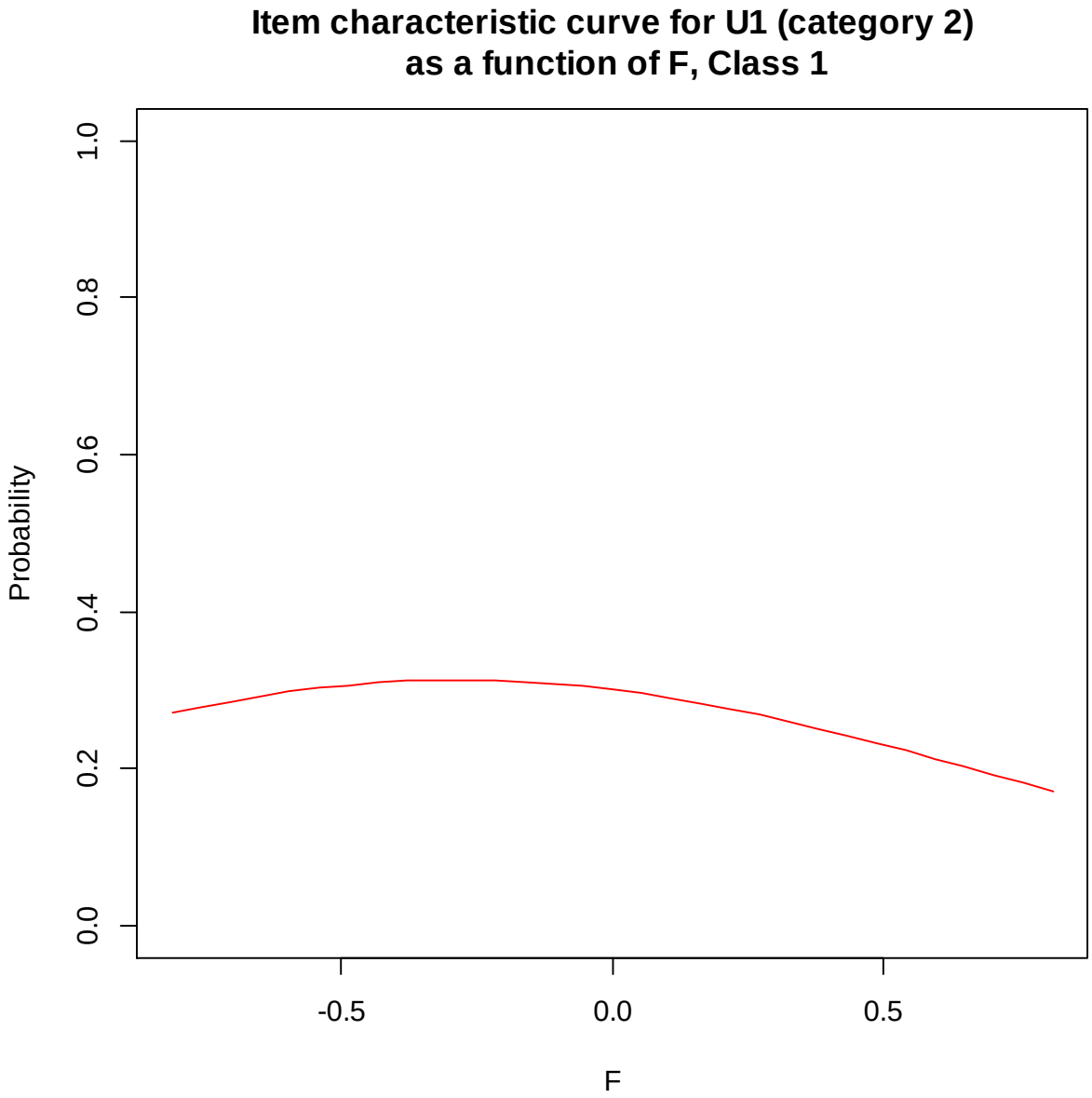
When there are more than one covariates in the model, it may be desirable to set the values of the covariates that is not the covariate on the x-axis to specific values. By default, estimated means for the covariates not on the x-axis are used.

Examples of usage of function mplus.plot.irt.icc:

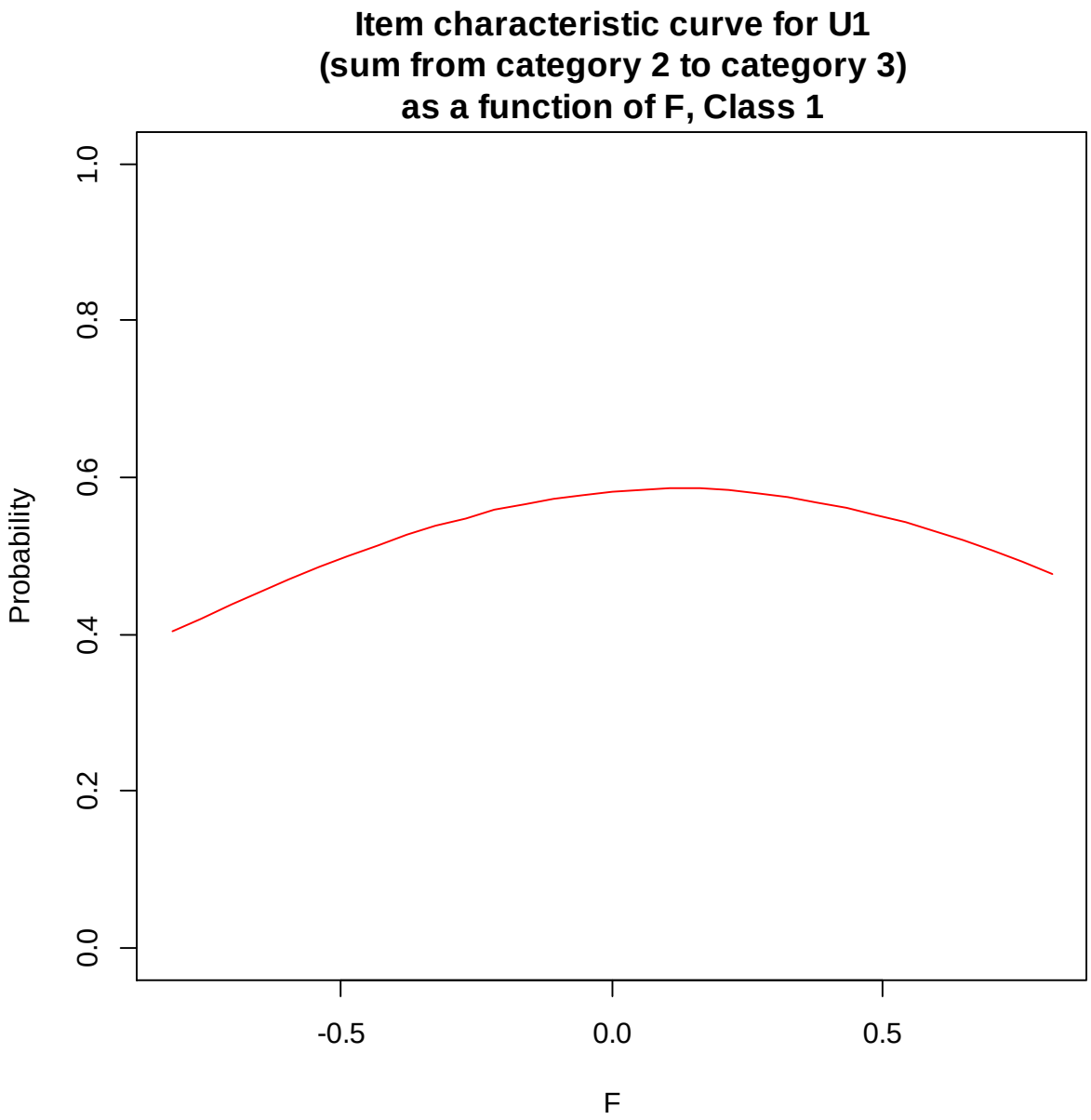
- Plot all categories for a single item
  > mplus.plot.irt.icc('catpoly2.gh5',uvar='u1')



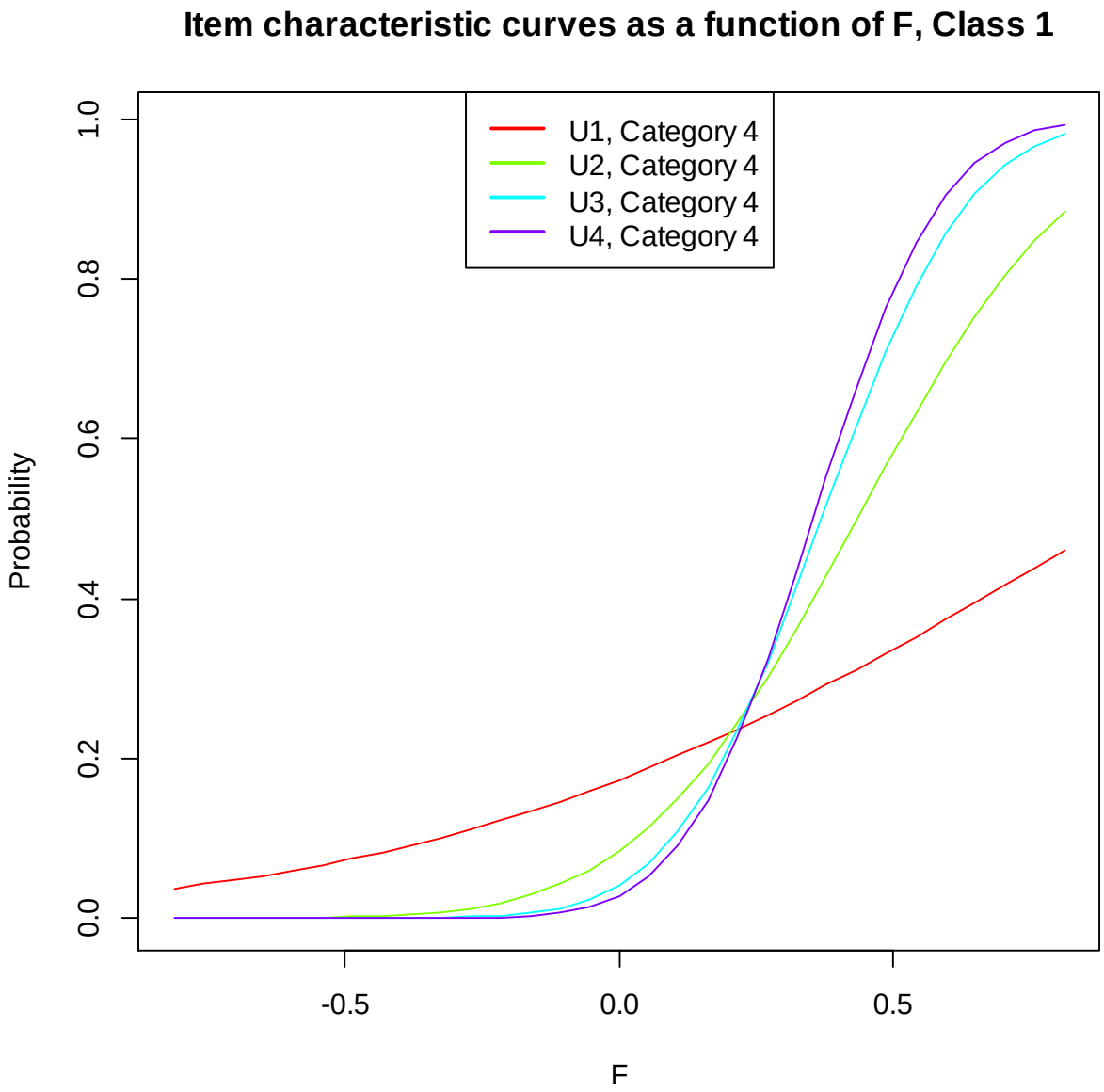
- Plot a specific category for a single item
  > mplus.plot.irt.icc('catpoly2.gh5',uvar='u1',cat=2)



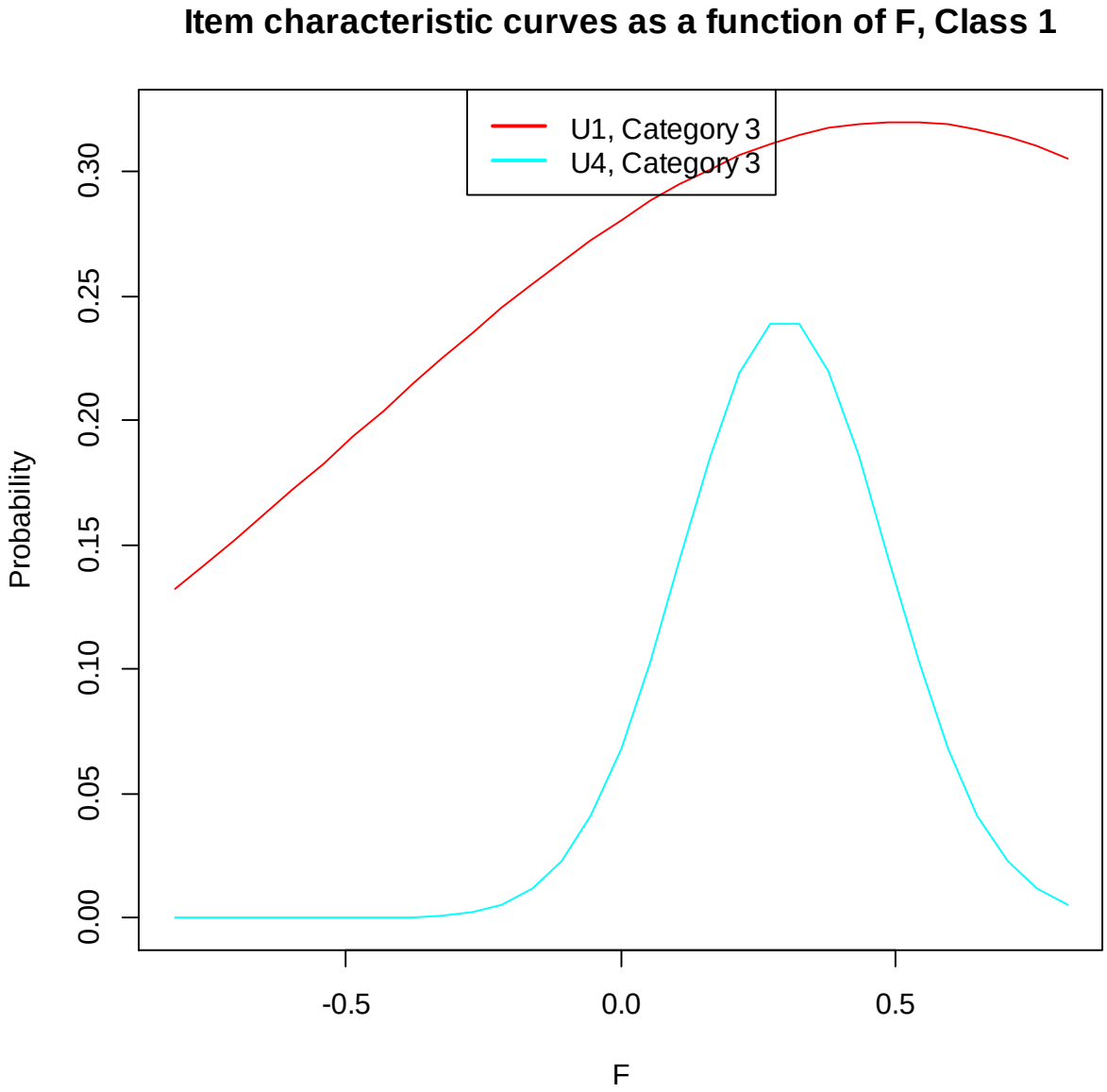
- Plot a sum of categories for a single item
  > mplus.plot.irt.icc('catpoly2.gh5',uvar='u1',cat=2,cat2=3)



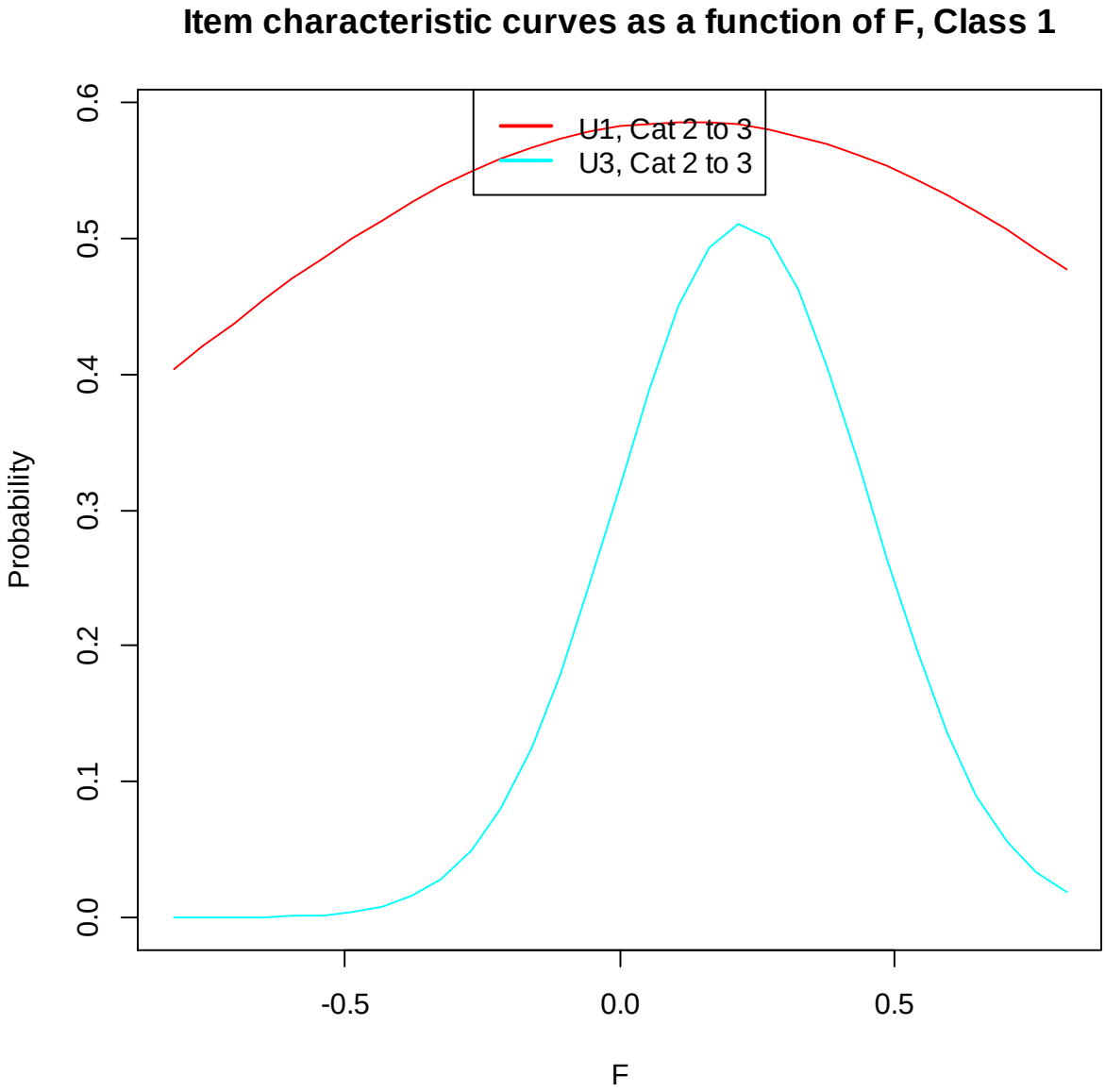
- Plot a specific category for all items
  > mplus.plot.irt.icc('catpoly2.gh5',cat=4)

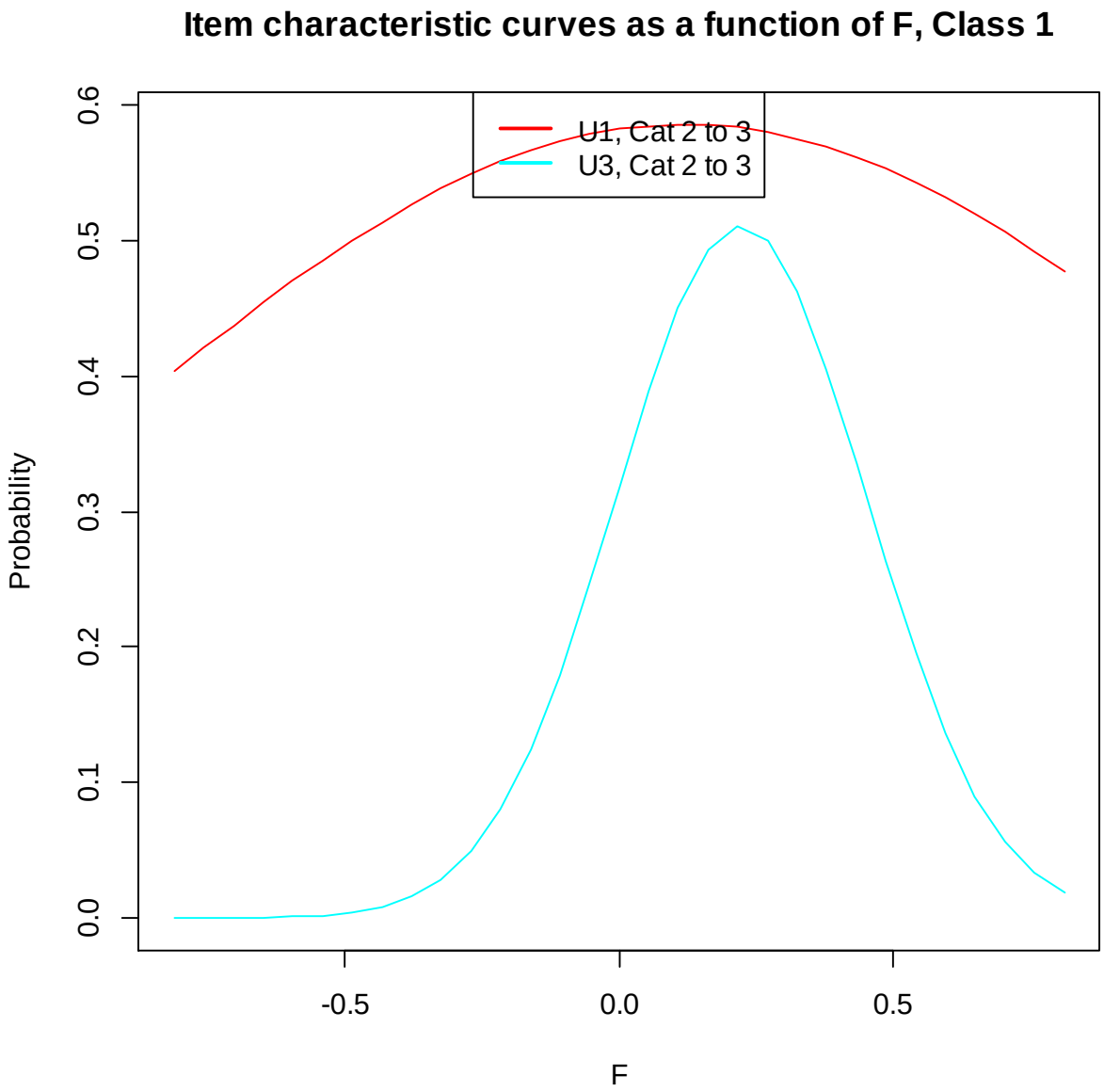


- Plot a specific category for selected items
  > mplus.plot.irt.icc('catpoly2.gh5',uvar=c('u1','u4'),cat=3)

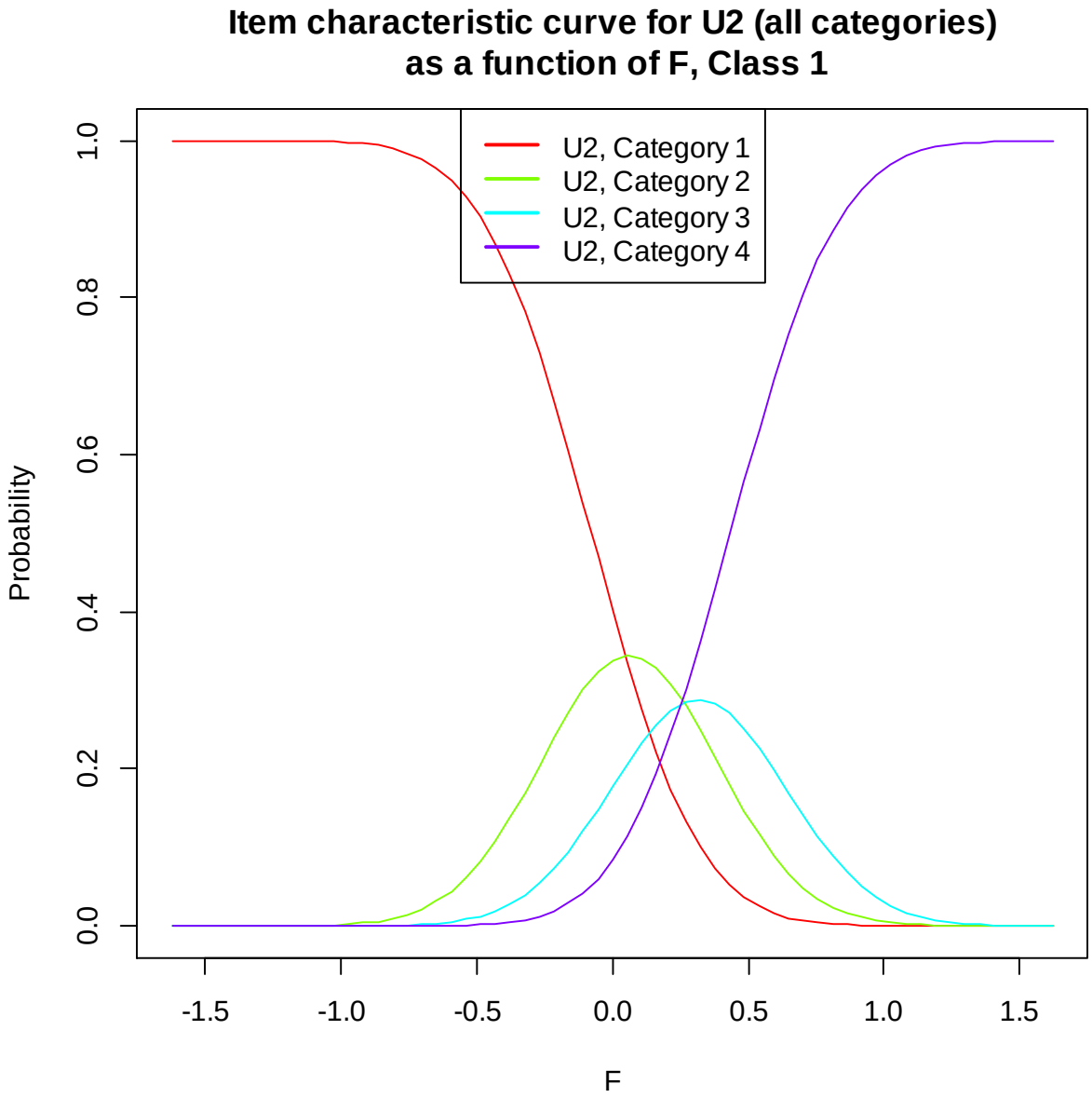


- Plot a sum of categories for selected items
  > mplus.plot.irt.icc('catpoly2.gh5',uvar=c('u1','u3'),cat=2,cat2=3)

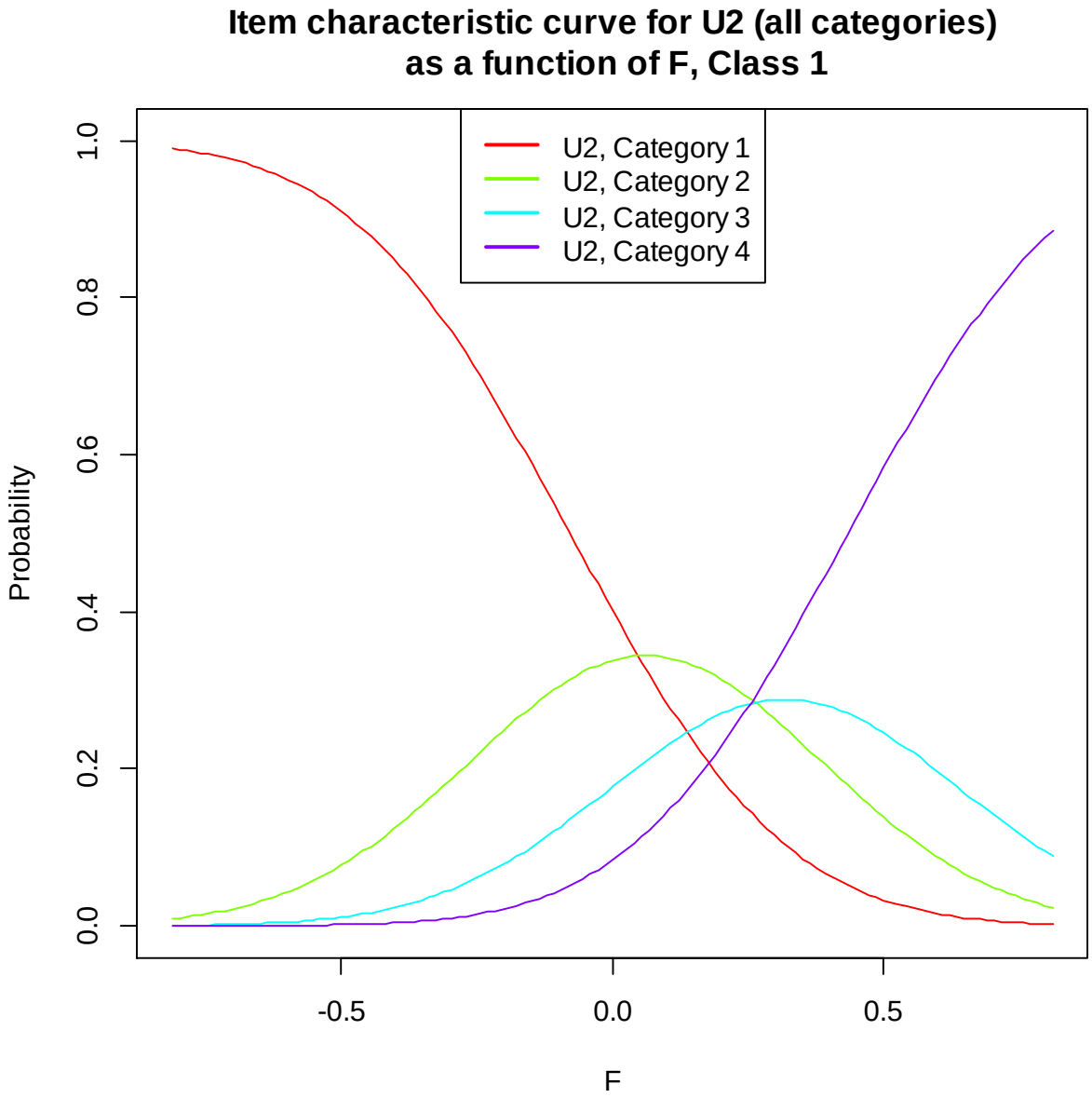

Item characteristic curves as a function of F, Class 1
U1, Cat 2 to 3
U3, Cat 2 to 3
Probability
0.0
0.1
0.2
0.3
0.4
0.5
0.6
-0.5
0.0
0.5
F



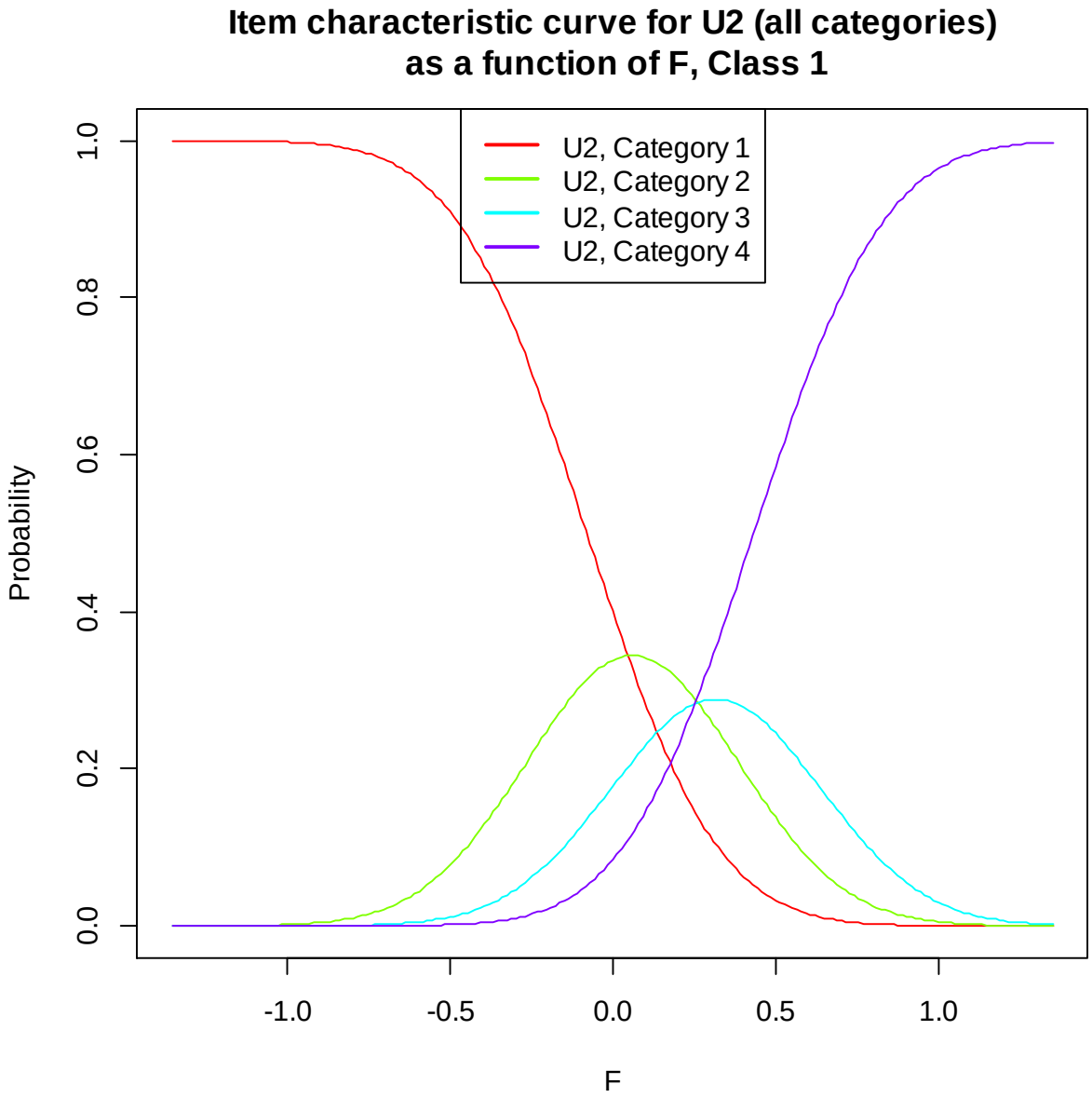
- Plot all categories for a single item using a specific range for the x-axis variable
  > mplus.plot.irt.icc('catpoly2.gh5',uvar='u2',xrange=6)



- Plot all categories for a single item using a specific step for the x-axis variable
  > mplus.plot.irt.icc('catpoly2.gh5',uvar='u2',xstep=9)

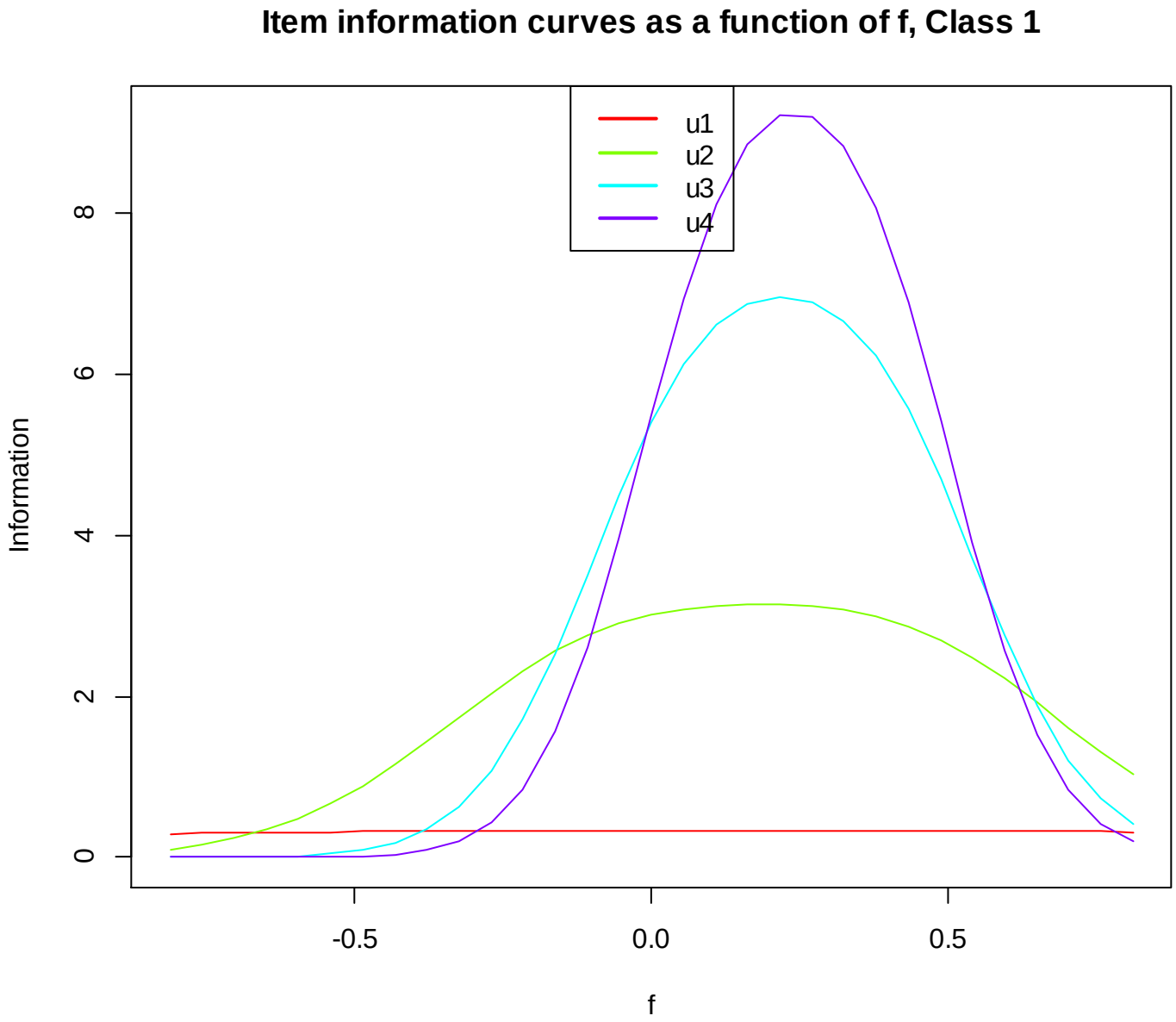


- Plot all categories for a single item using a specific range and step for the x-axis variable
  > mplus.plot.irt.icc('catpoly2.gh5',uvar='u2',xstep=9,xrange=5)

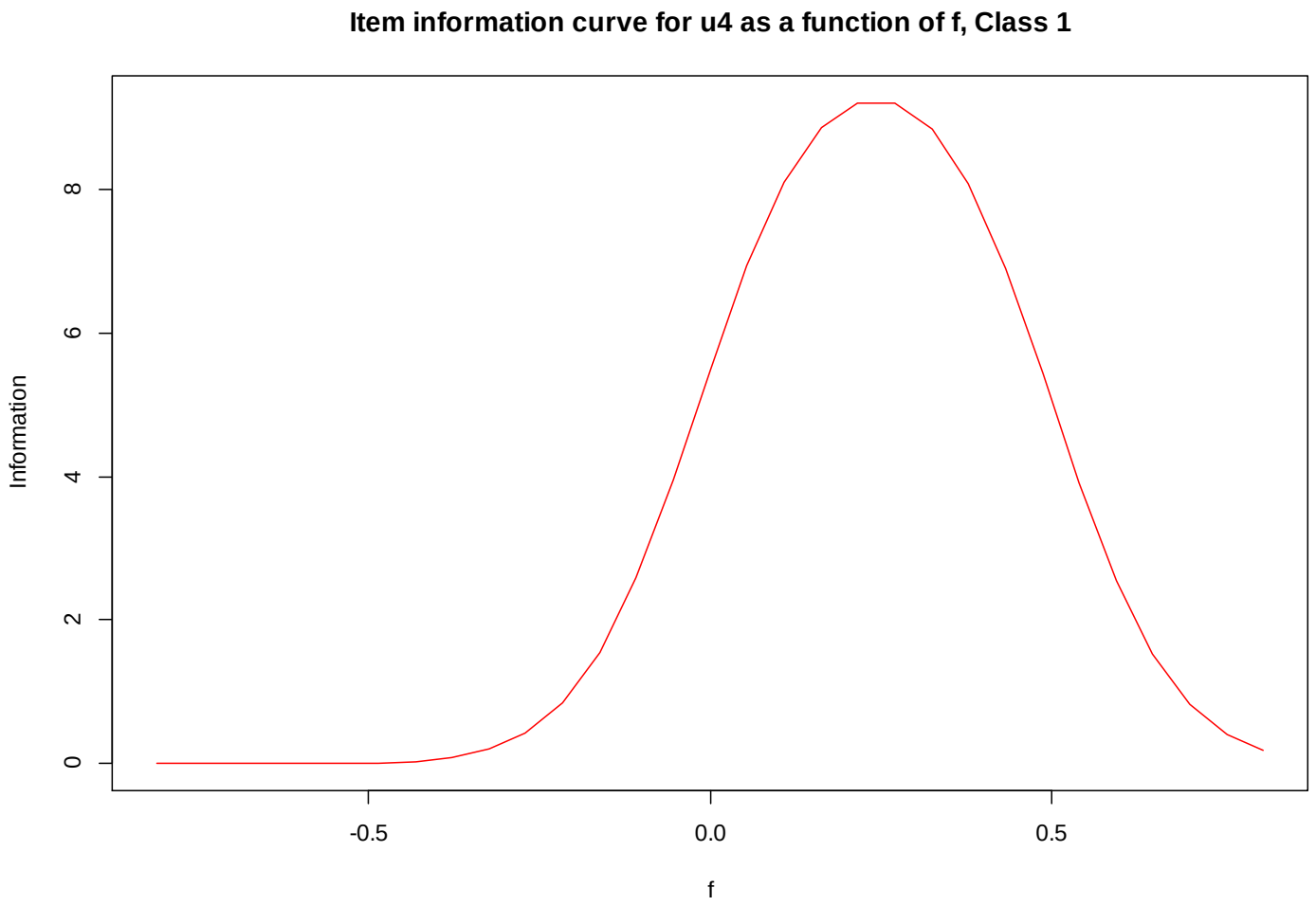

Item characteristic curve for U2 (all categories)
as a function of F, Class 1
U2, Category 1
U2, Category 2
U2, Category 3
U2, Category 4
Probability
0.0
0.2
0.4
0.6
0.8
1.0
-1.0
-0.5
0.0
0.5
1.0
F

Examples of usage of function mplus.plot.irt.iic:

- Plot all items
  > mplus.plot.irt.iic('catpoly2.gh5')

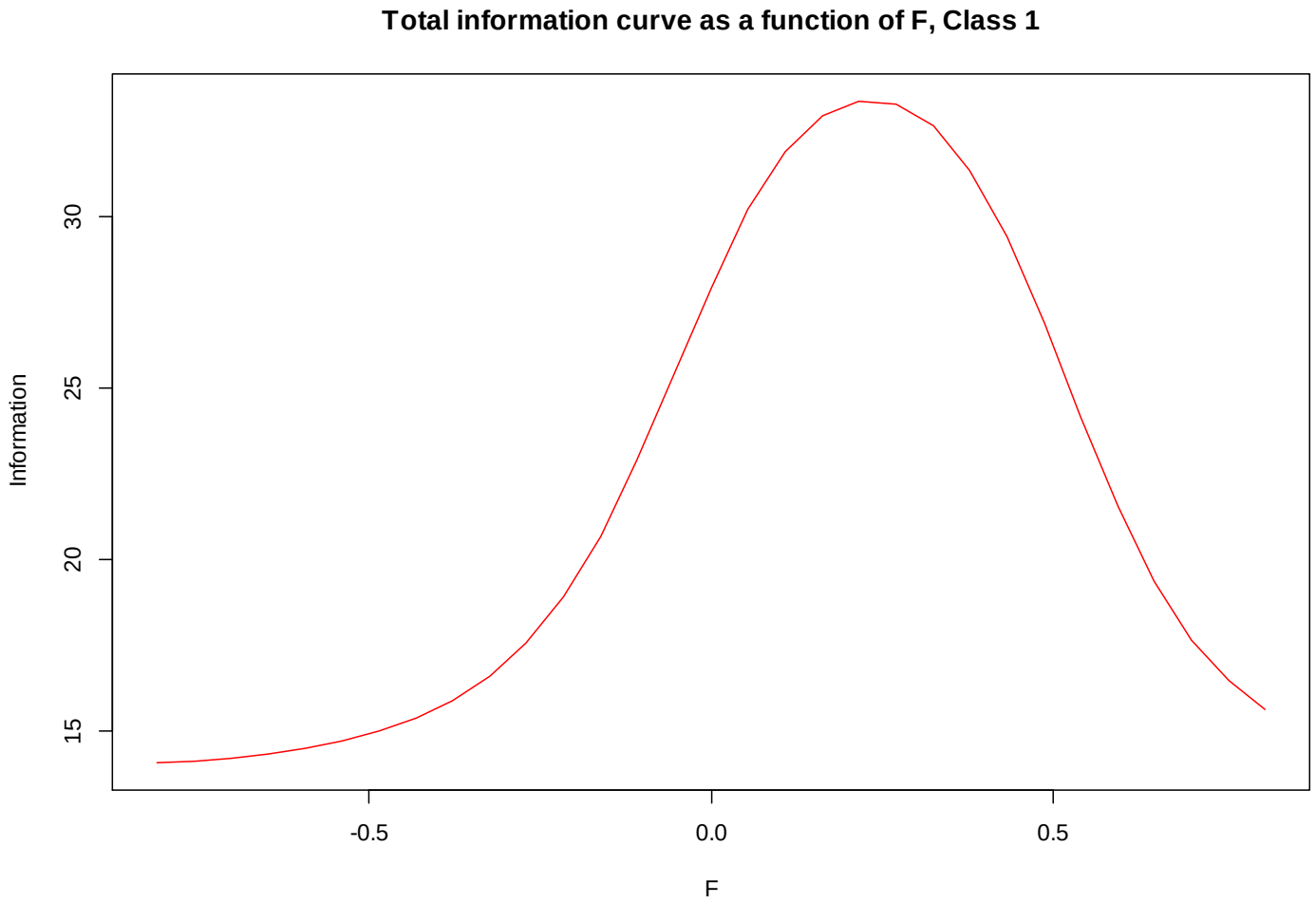


- Plot a single item
  > mplus.plot.irt.iic('catpoly2.gh5',uvar=4)
  > mplus.plot.irt.iic('catpoly2.gh5',uvar='u4')

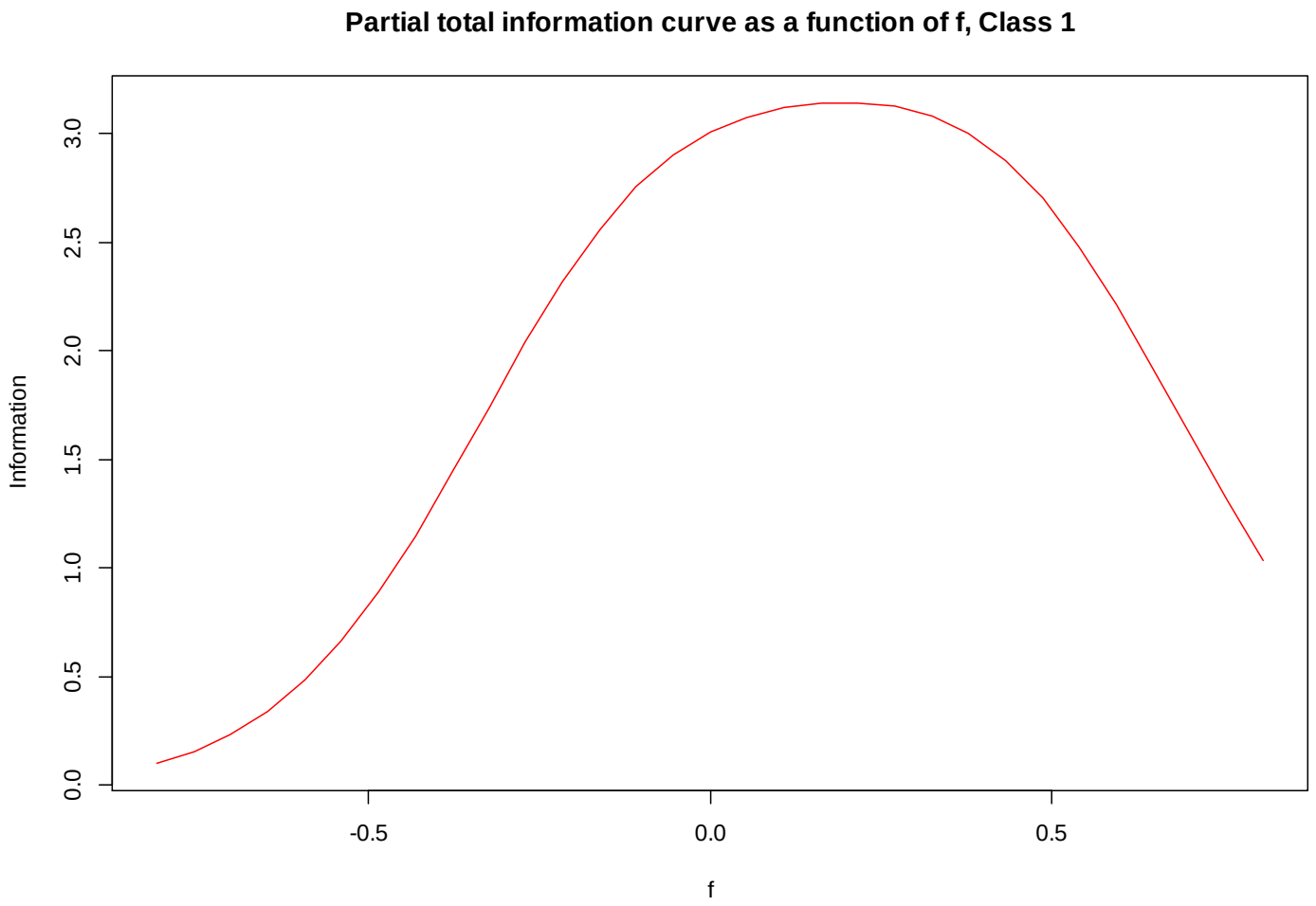


Examples of usage of function mplus.plot.irt.tic:

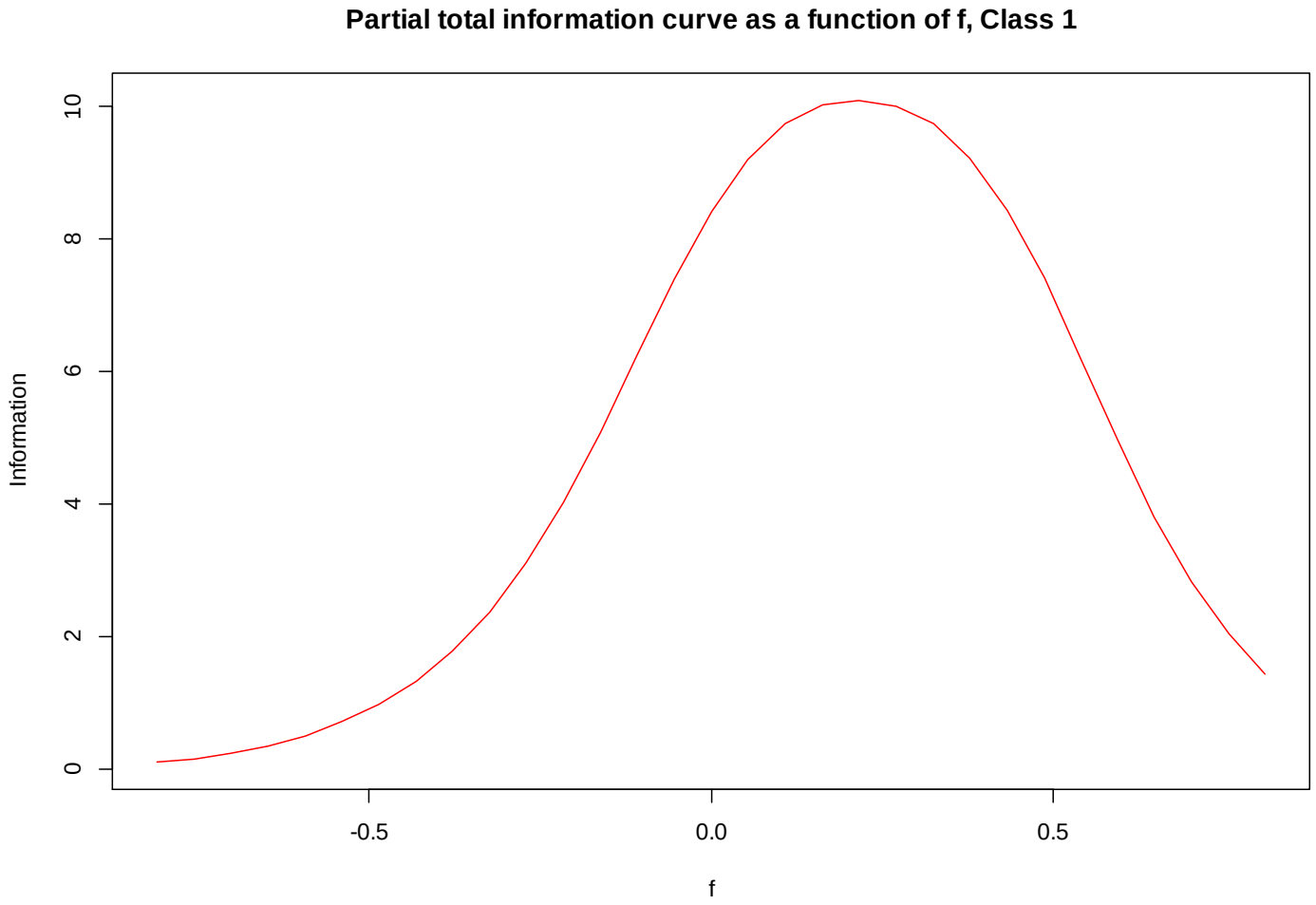
- Plot all items
  > mplus.plot.irt.tic('catpoly2.gh5')



- Plot partial information curve for a single item
  > mplus.plot.irt.tic('catpoly2.gh5',uvar=2)
  > mplus.plot.irt.tic('catpoly2.gh5',uvar='u2')



- Plot partial information curve for several items
  > mplus.plot.irt.tic('catpoly2.gh5',uvar=c(2,3))
  > mplus.plot.irt.tic('catpoly2.gh5',uvar=c('u2','u3'))



Survival curves

The following functions are used to view various survival curves.

- mplus.plot.survival.kaplanmeier('ex7.30.gh5',survvar,classnum)
- mplus.plot.survival.baseline('ex7.30.gh5',survvar,classnum)
- mplus.plot.survival.basehazard('ex7.30.gh5',survvar,classnum)
- mplus.plot.survival.sample.logcumulative('ex7.30.gh5',survvar,classnum)
- mplus.plot.survival.estimated.logcumulative('ex7.30.gh5',survar,classnum)
- mplus.plot.survival.kaplanmeier.vs.baseline('ex7.30.gh5',survvar,classnum)
- mplus.plot.survival.sample.vs.estimated.logcumulative('ex7.30.gh5',survvar,classnum)
- mplus.list.survival.variables('ex7.30.gh5')

The survvar argument refers to the name or index of the survival variable. The index refers to the ordering given in the list of survival variables by the function. The survvar argument defaults to the first survival variable.

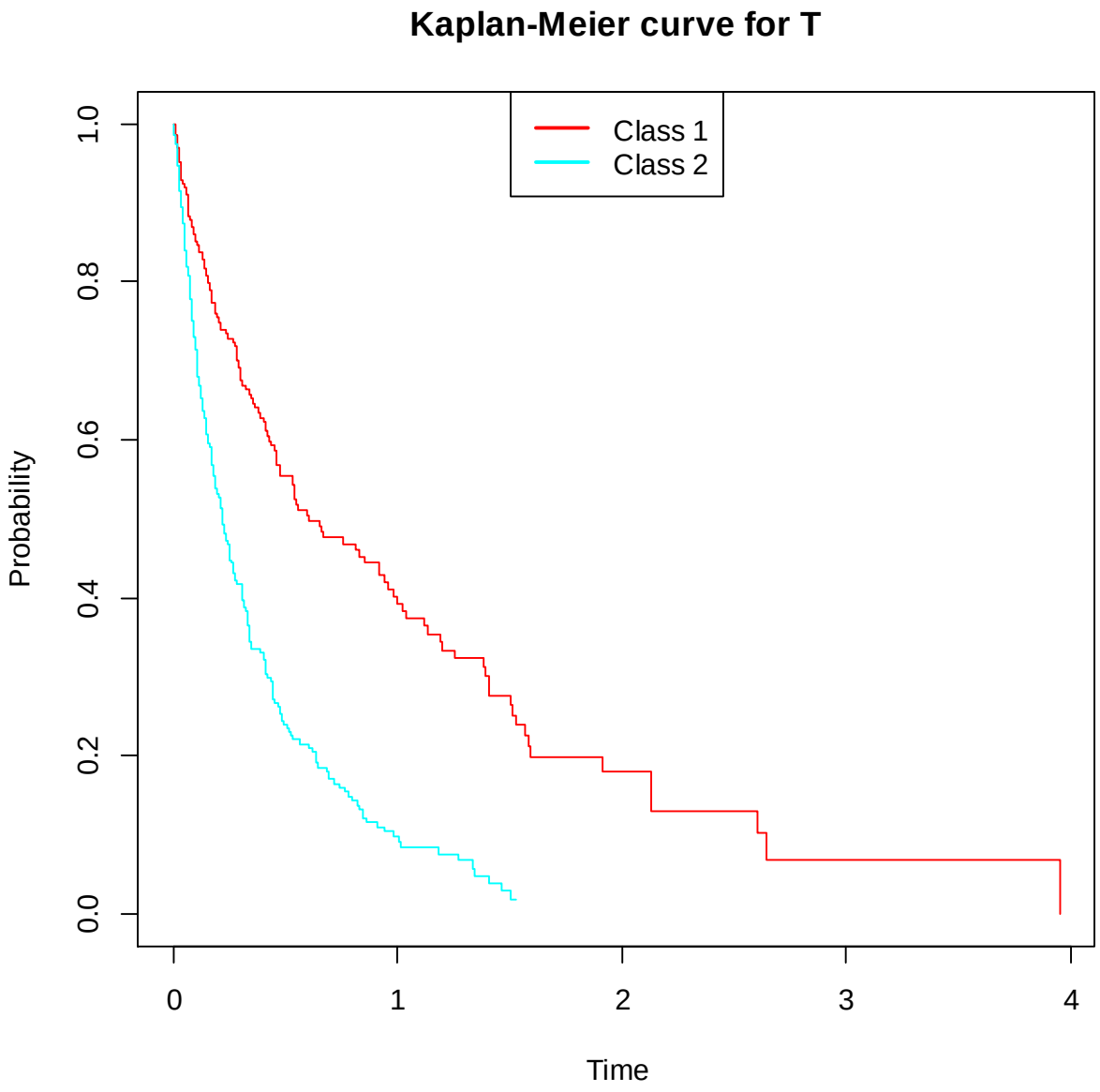
The classnum argument refers to the class number for Mixture analysis. If the classnum argument is not given, then the plot will show all classes. For non-mixture analysis, this argument is not required.

The following functions are provided for getting the data plotted in the various survival curves:

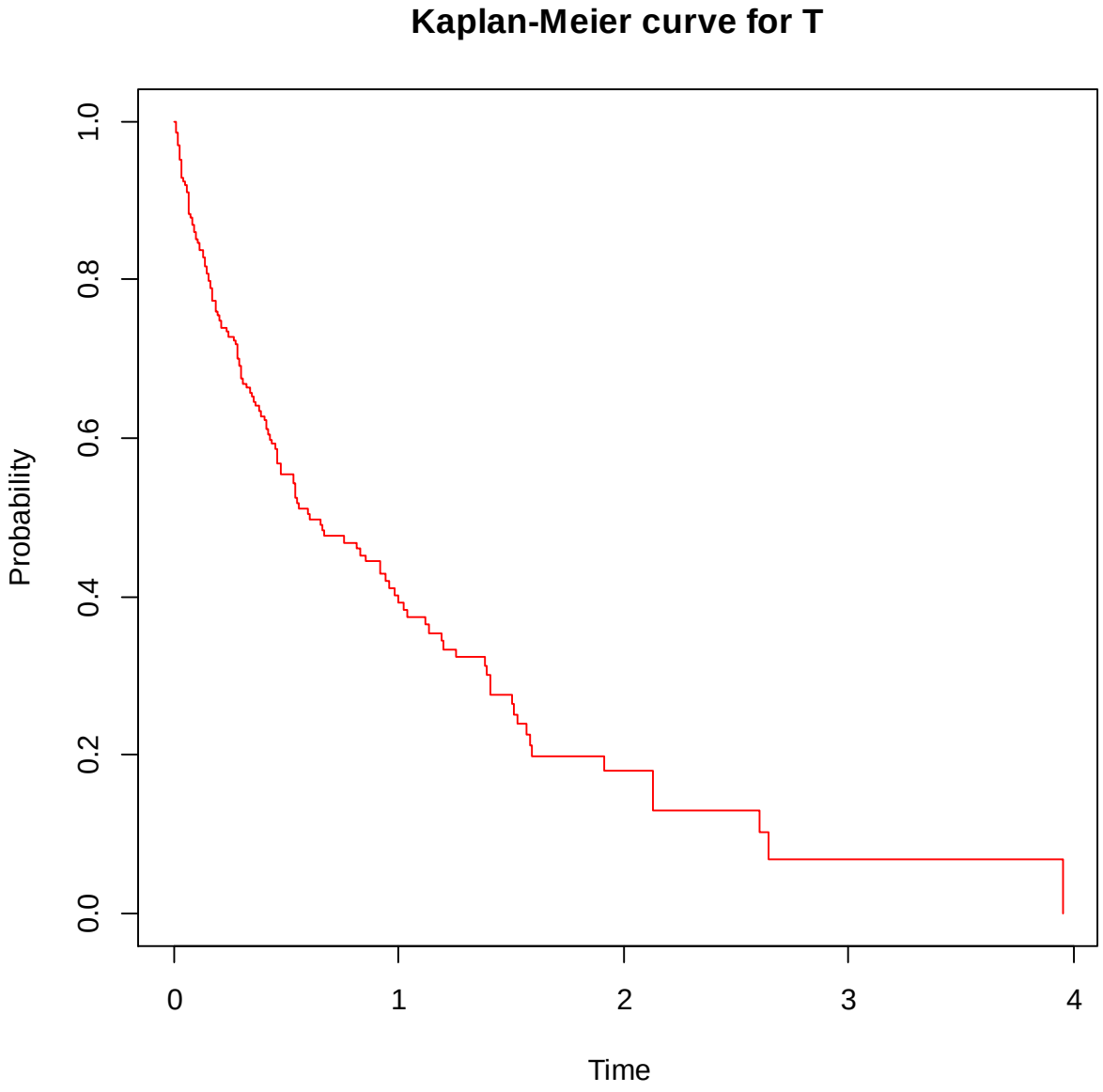
- mplus.get.survival.kaplanmeier.values('ex7.30.gh5',survvar,classnum,time)
- mplus.compute.survival.sample.logcumulative.values('ex7.30.gh5',survvar,classnum,time)
- mplus.get.survival.baseline.values('ex7.30.gh5',survvar,survvar2,clasnum,time)
- mplus.compute.survival.estimated.logcumulative.values('ex7.30.gh5',survvar,classnum,time)
- mplus.get.survival.basehazard.values('ex7.30.gh5',file,survvar,classnum,time)

The time argument is used to request the x values (time) used in the plots. If the time argument is not specified, the y values are given. All other arguments are as described above for the plot functions.

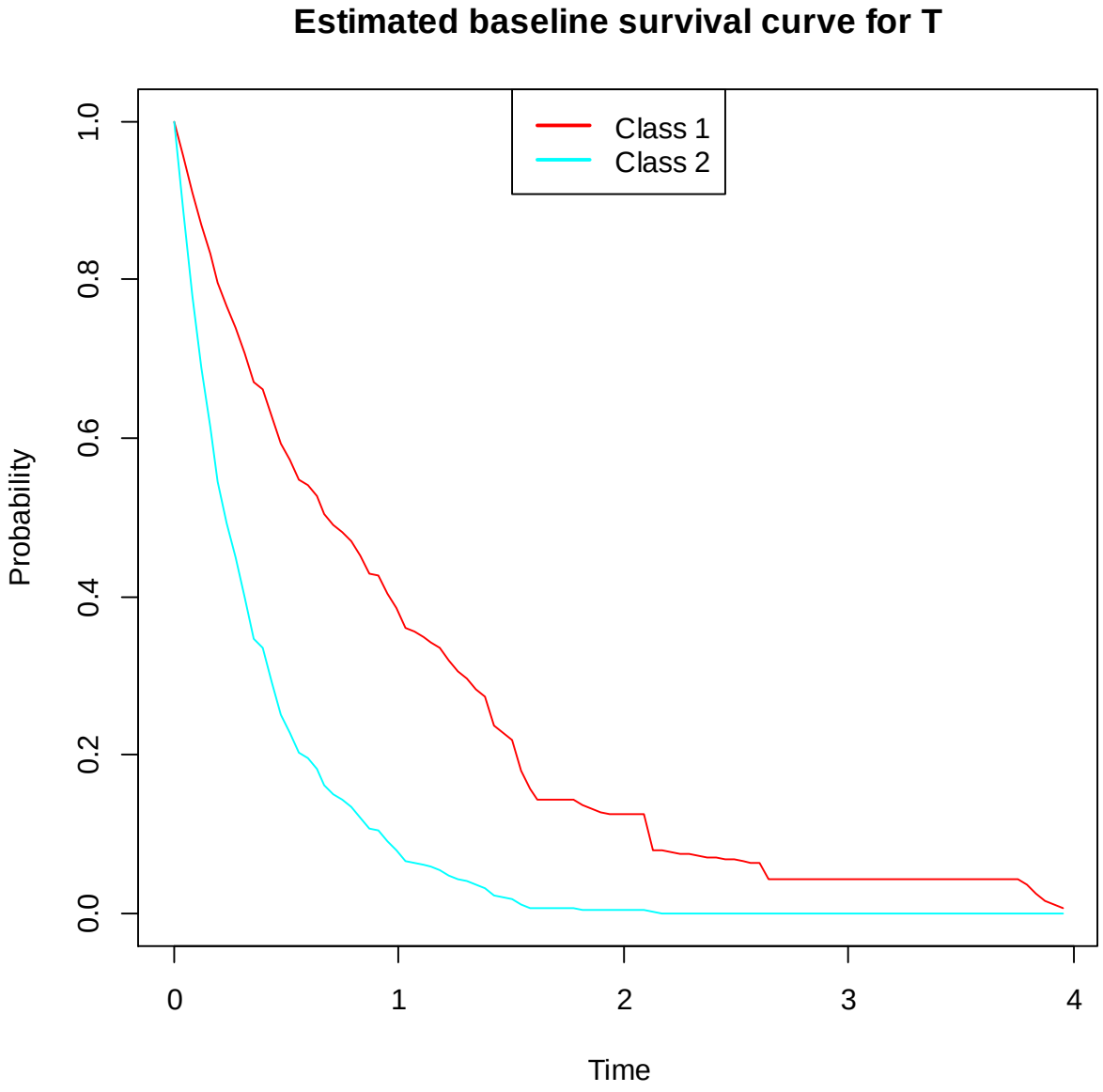
```
> mplus.plot.survival.kaplanmeier('ex7.30.gh5')
```



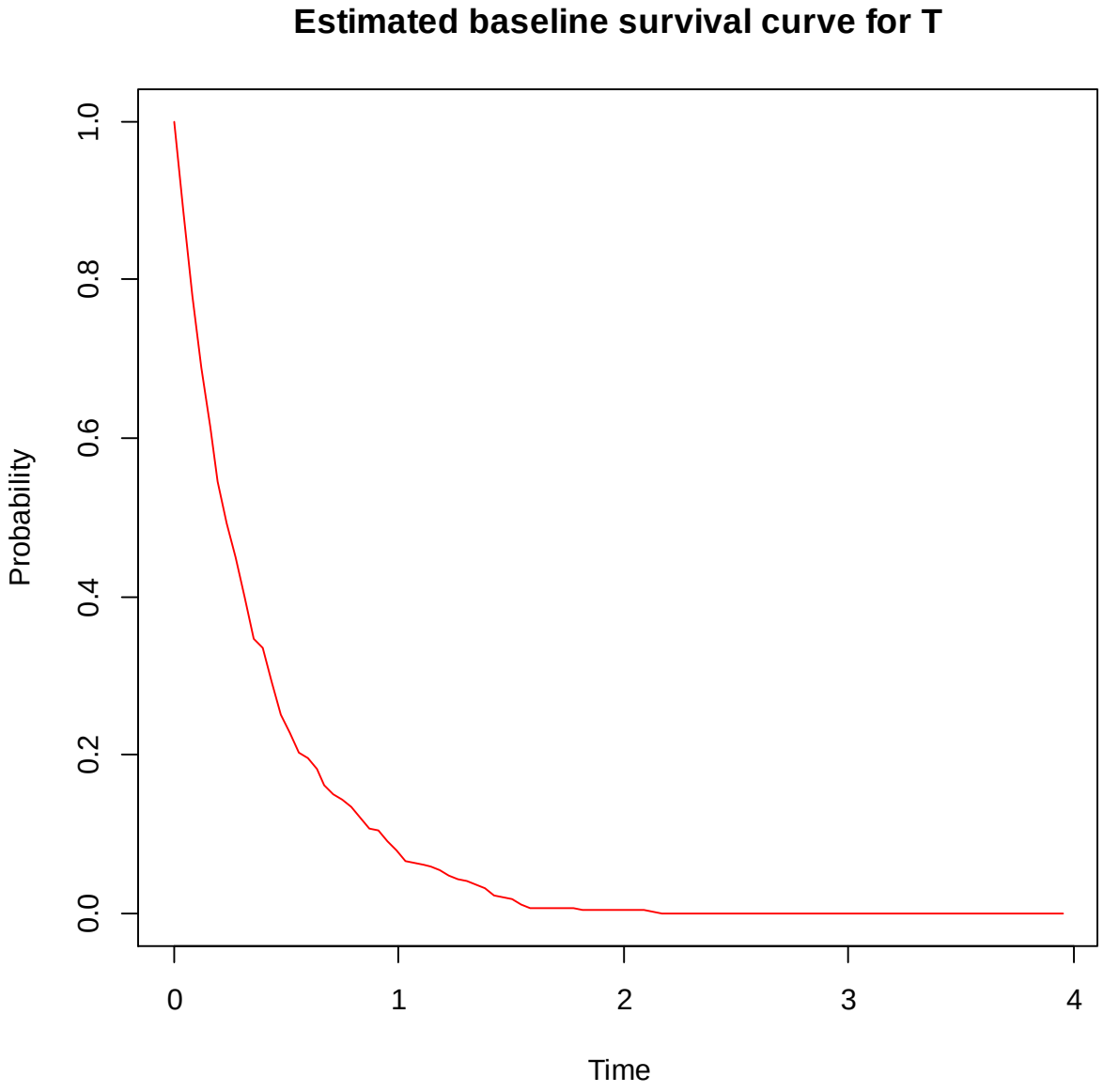
> mplus.plot.survival.kaplanmeier('ex7.30.gh5',classnum=1)



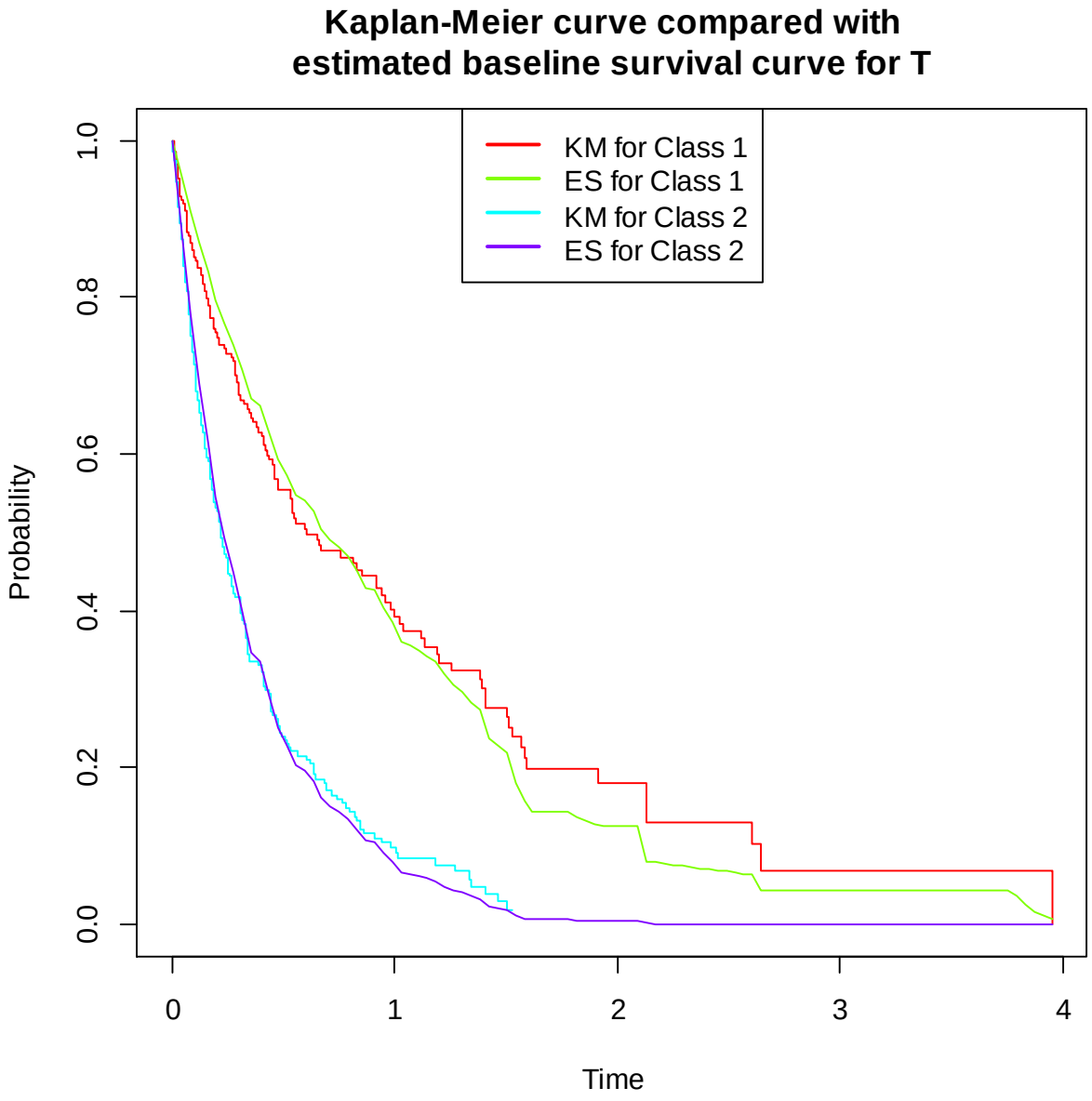
> mplus.plot.survival.baseline('ex7.30.gh5')



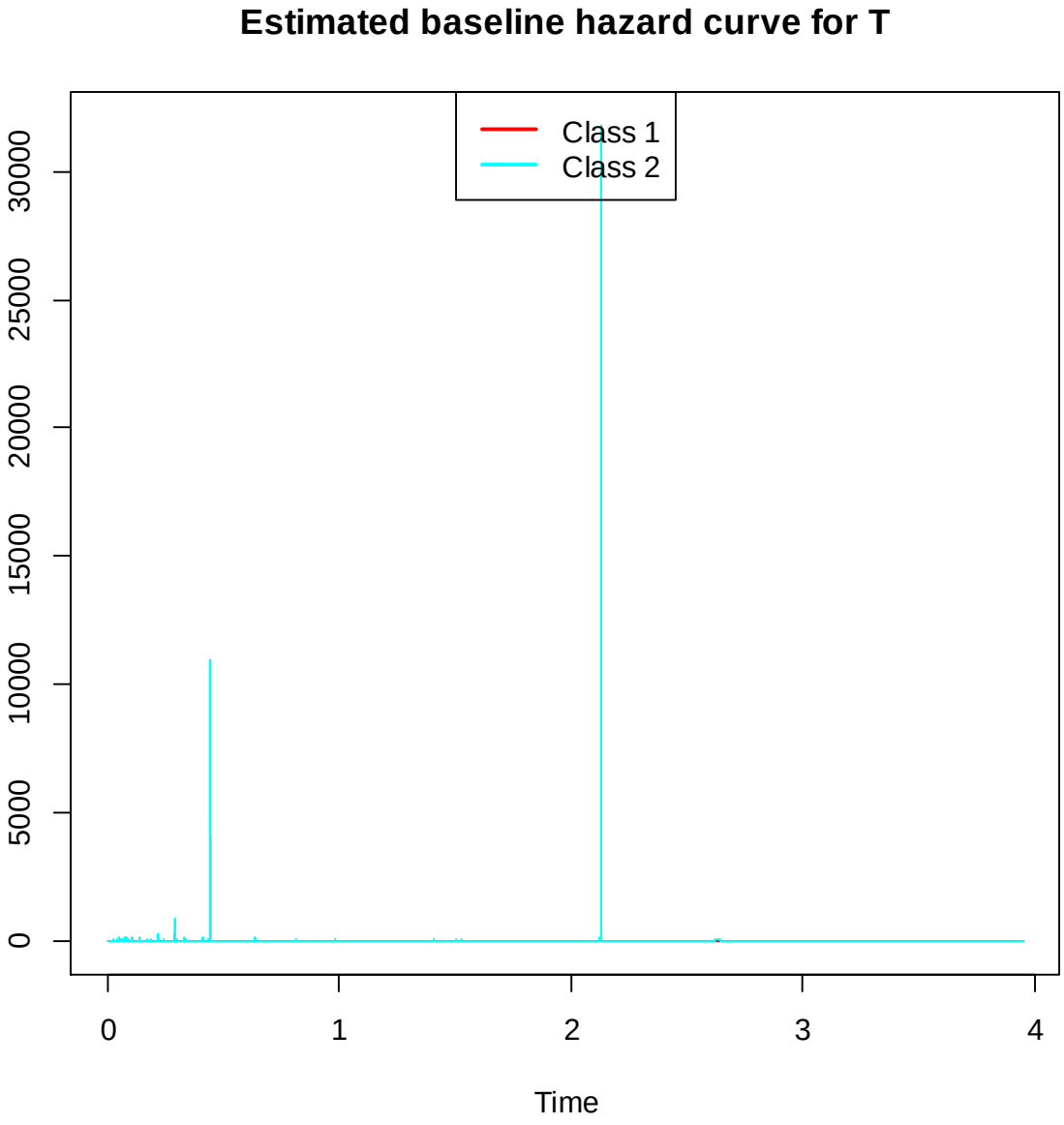
> mplus.plot.survival.baseline('ex7.30.gh5',classnum=2)



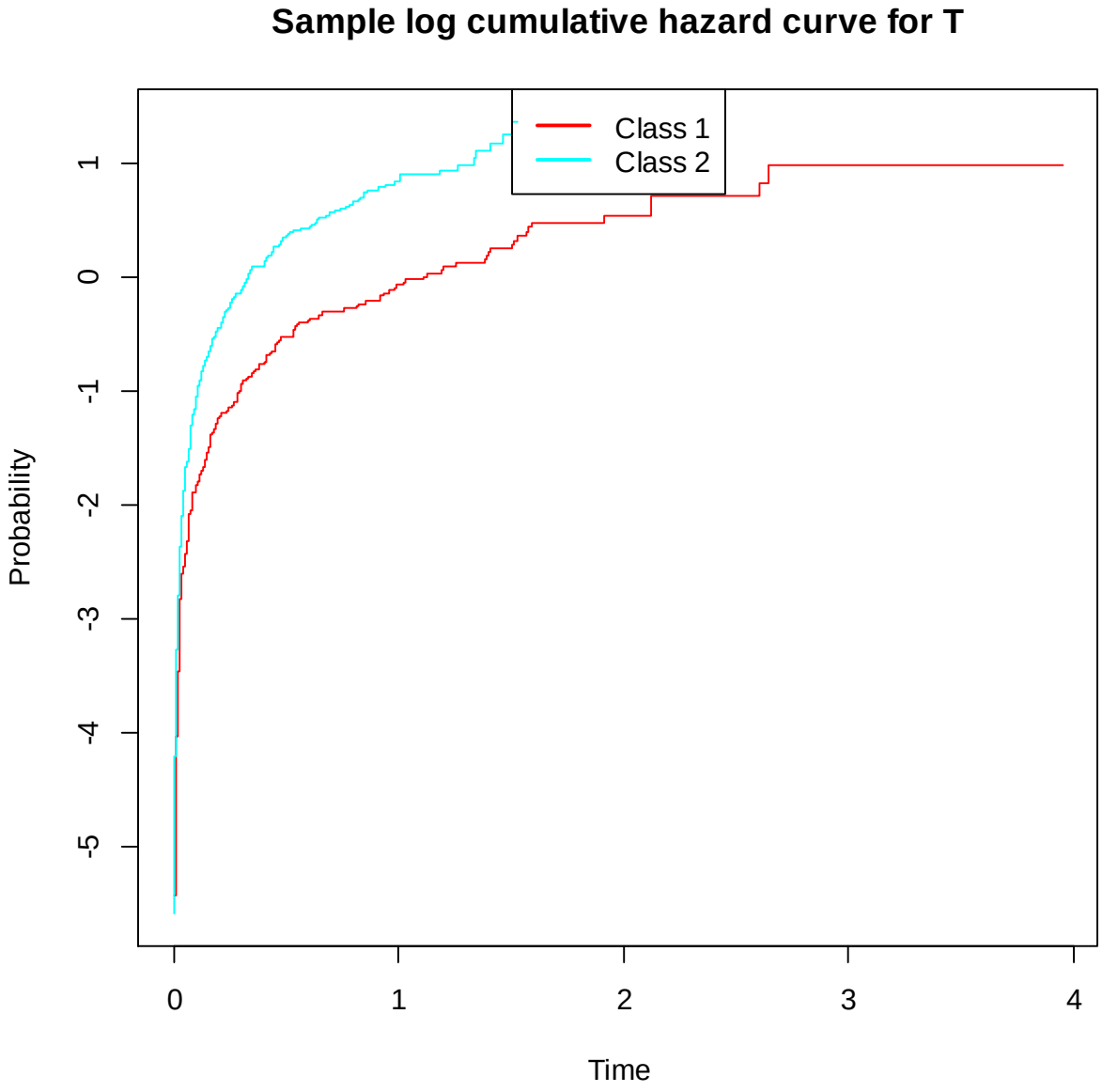
> mplus.plot.survival.kaplanmeier.vs.baseline('ex7.30.gh5')



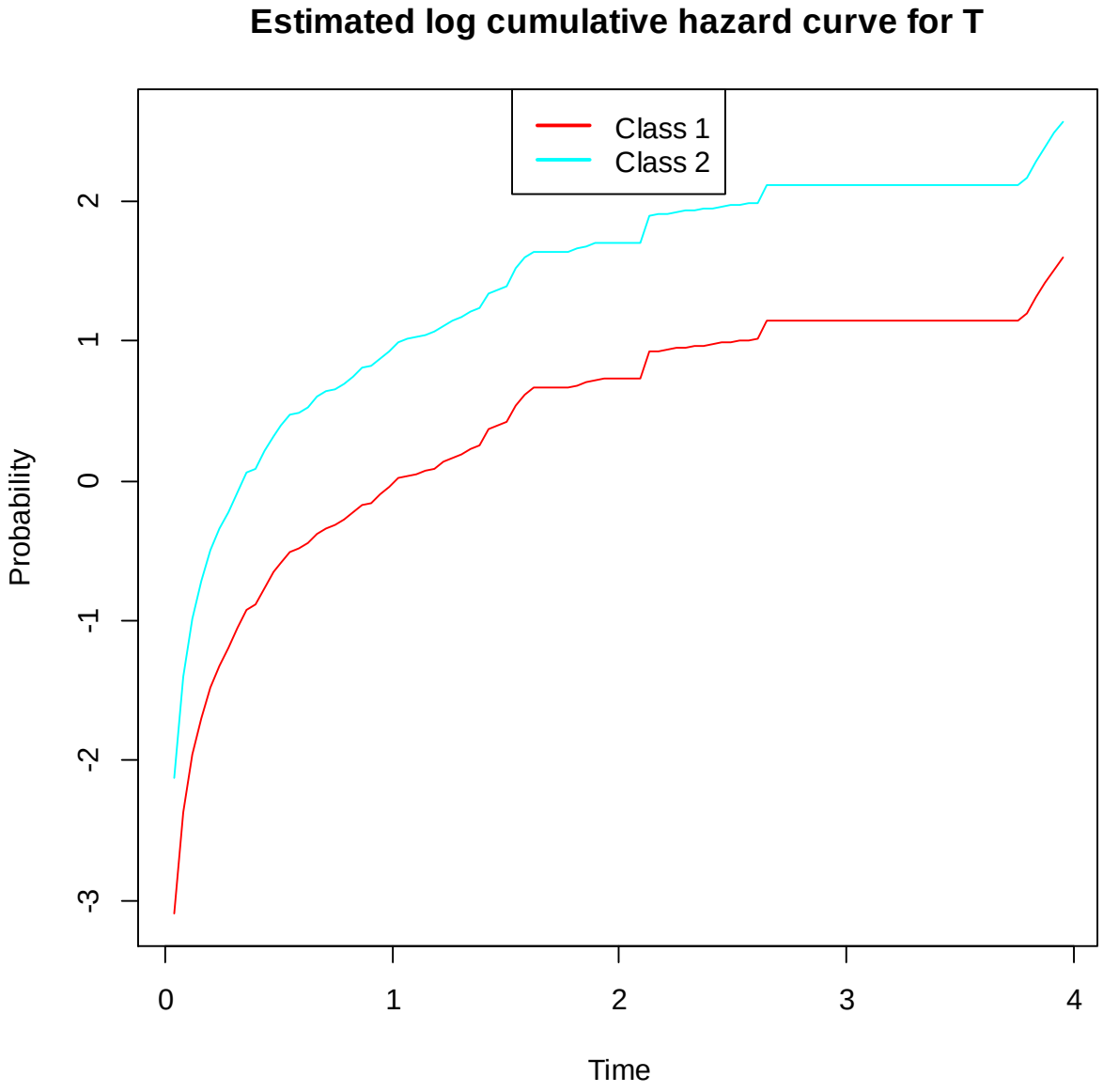
> mplus.plot.survival.basehazard('ex7.30.gh5')



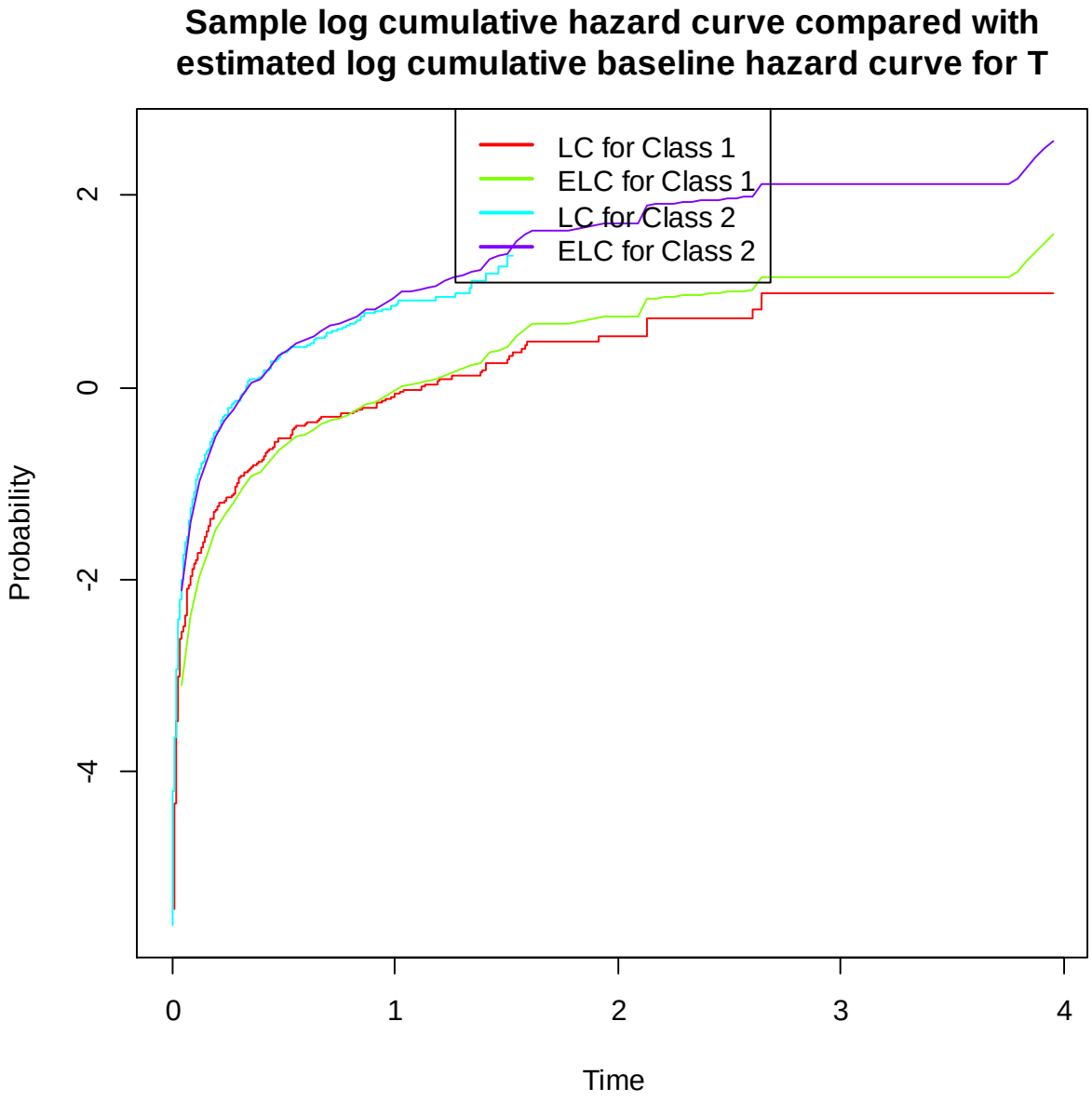
> mplus.plot.survival.sample.logcumulative('ex7.30.gh5')



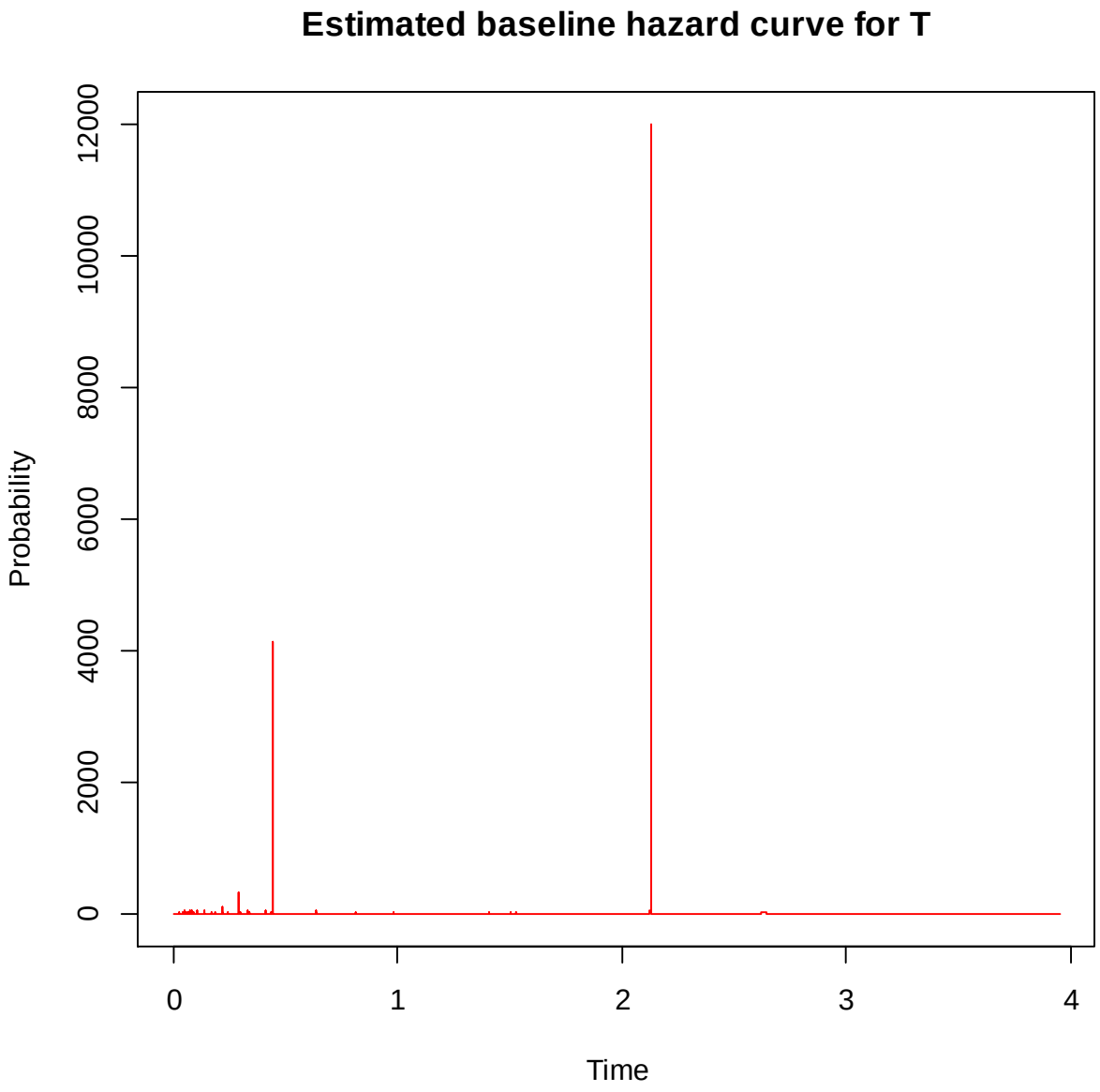
> mplus.plot.survival.estimated.logcumulative('ex7.30.gh5')



> mplus.plot.survival.sample.vs.estimated.logcumulative('ex7.30.gh5')



> mplus.plot.survival.basehazard('ex7.30.gh5',classnum=1)


Estimated baseline hazard curve for T
Probability
0
2000
4000
6000
8000
10000
12000
0
1
2
3
4
Time

Discrete survival curves

The following functions are used to view various survival curves.

- mplus.plot.discrete.survival.kaplanmeier('ex6.19.gh5',survvar,classnum)
- mplus.plot.discrete.survival.baseline('ex6.19.gh5',survvar,classnum)
- mplus.plot.discrete.survival.kaplanmeier.vs.baseline('ex6.19.gh5',survvar,classnum)
- mplus.list.discrete.survival.variables('ex6.19.gh5')

The survvar argument refers to the name or index of the survival variable. The index refers to the ordering given in the list of survival variables by the function. The survvar argument defaults to the first survival variable.

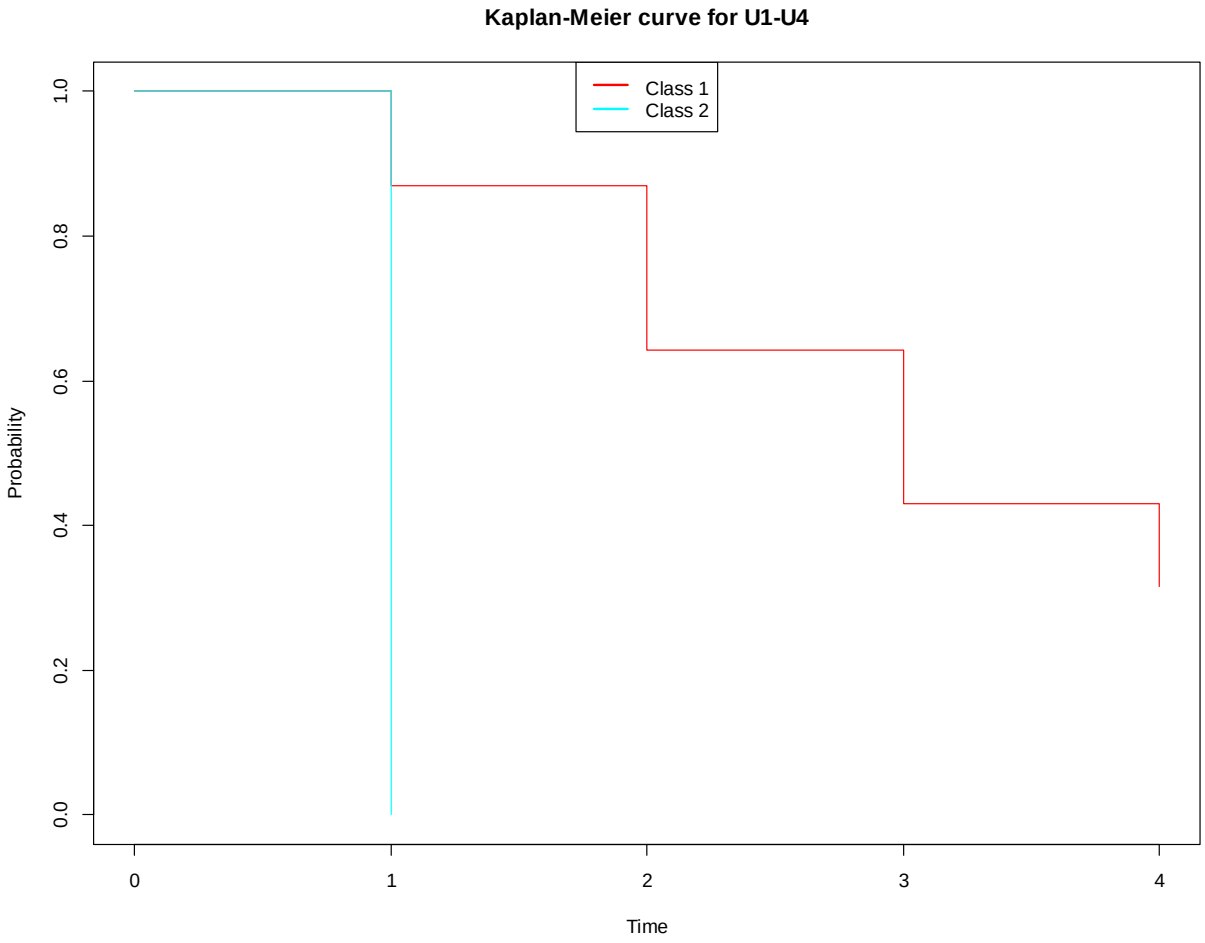
The classnum argument refers to the class number for Mixture analysis. If the classnum argument is not given, then the plot will show all classes. For non-mixture analysis, this argument is not required.

The following functions are provided for getting the data plotted in the various survival curves:

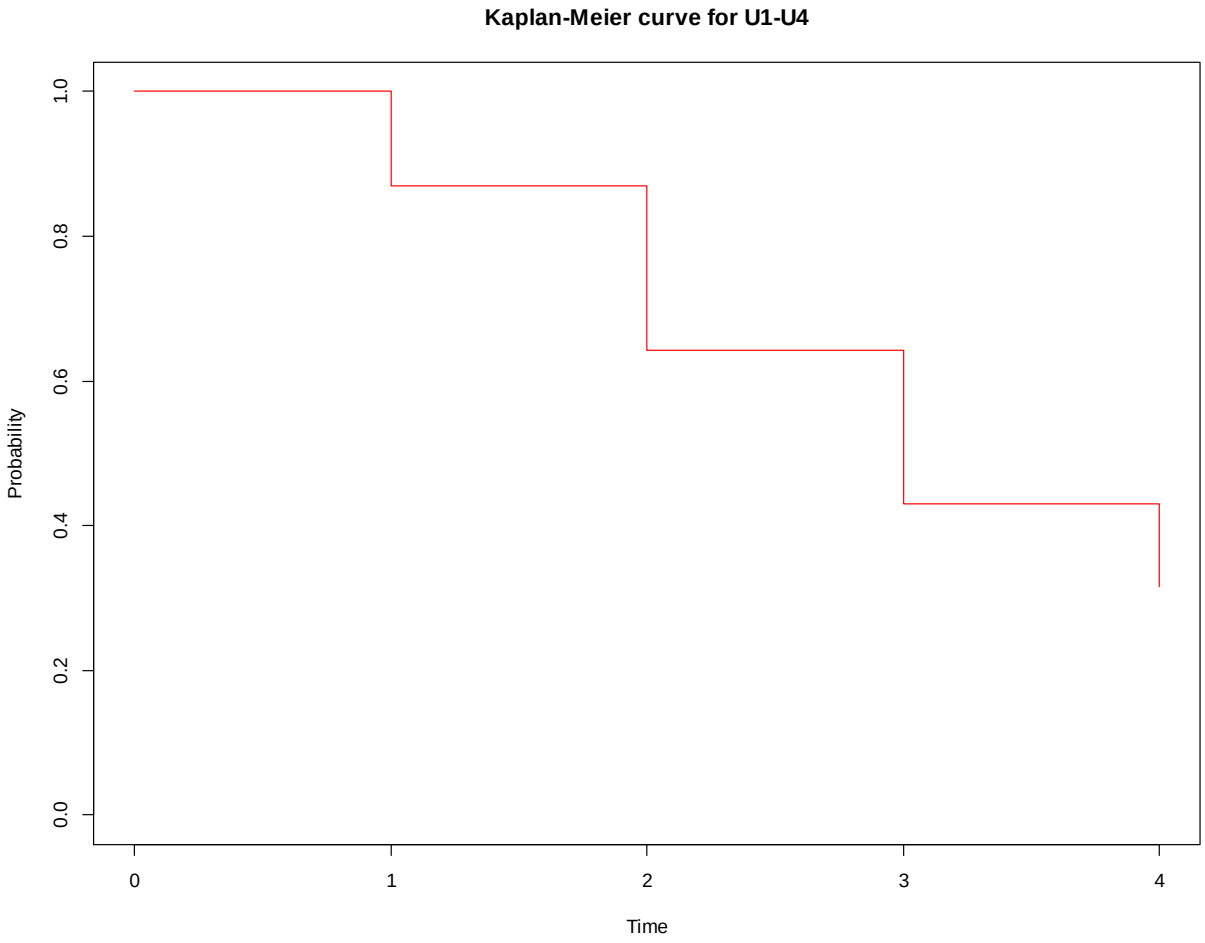
- mplus.get.discrete.survival.kaplanmeier.values('ex6.19.gh5',survvar,classnum,time)
- mplus.get.discrete.survival.baseline.values('ex6.19.gh5',survvar,survvar2,classnum,time)

The time argument is used to request the x values (time) used in the plots. If the time argument is not specified, the y values are given. All other arguments are as described above for the plot functions.

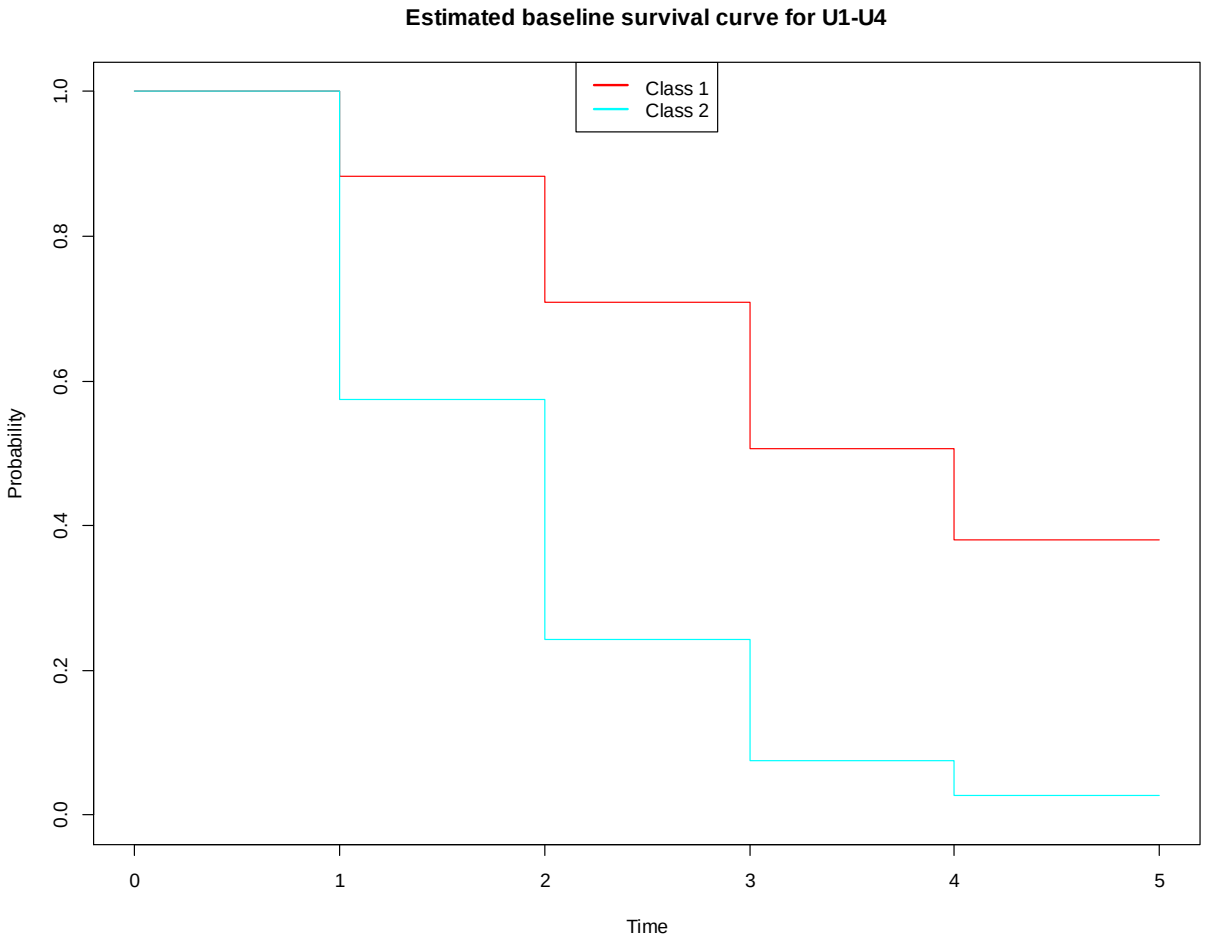
> mplus.plot.discrete.survival.kaplanmeier('ex6.19mix.gh5')



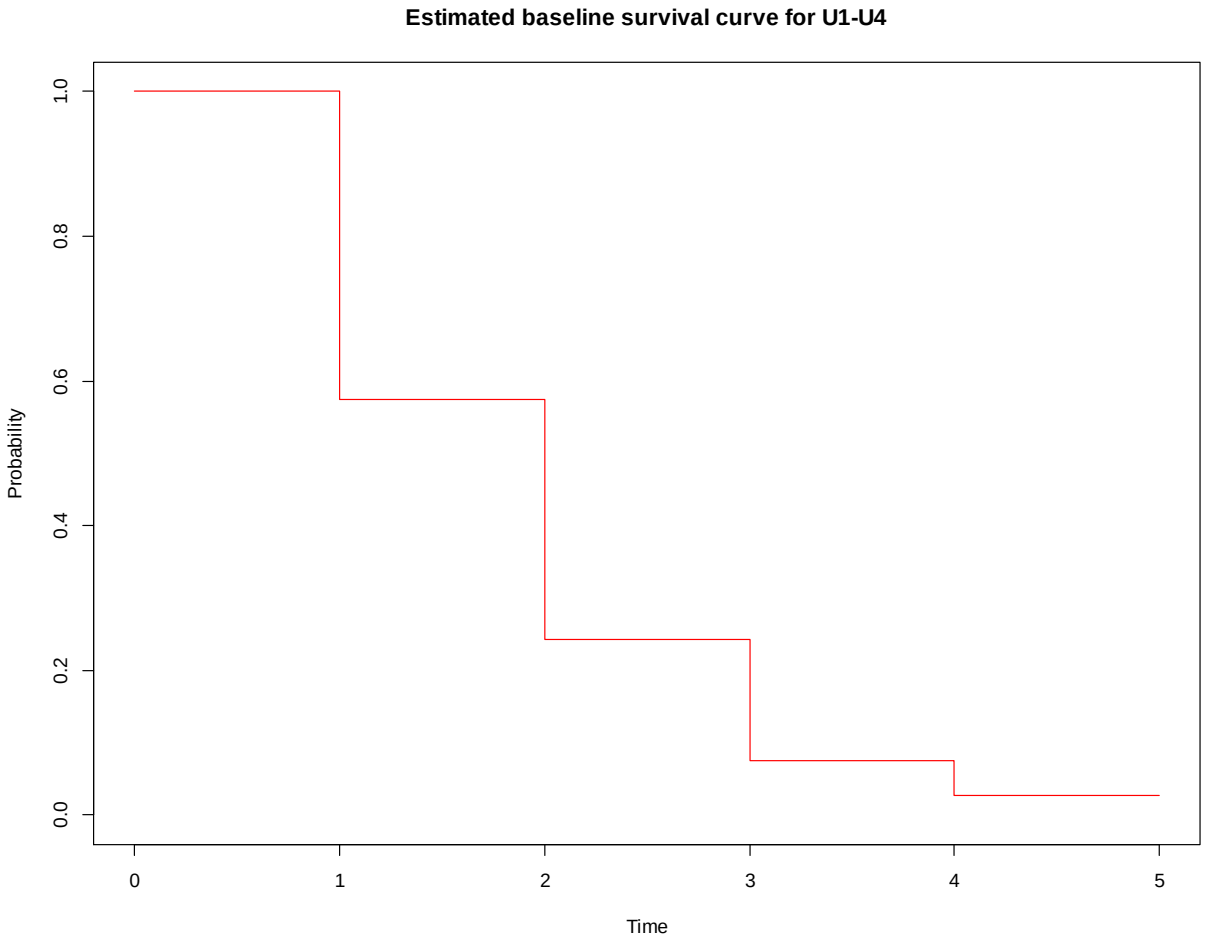
> mplus.plot.discrete.survival.kaplanmeier('ex6.19mix.gh5',classnum=1)



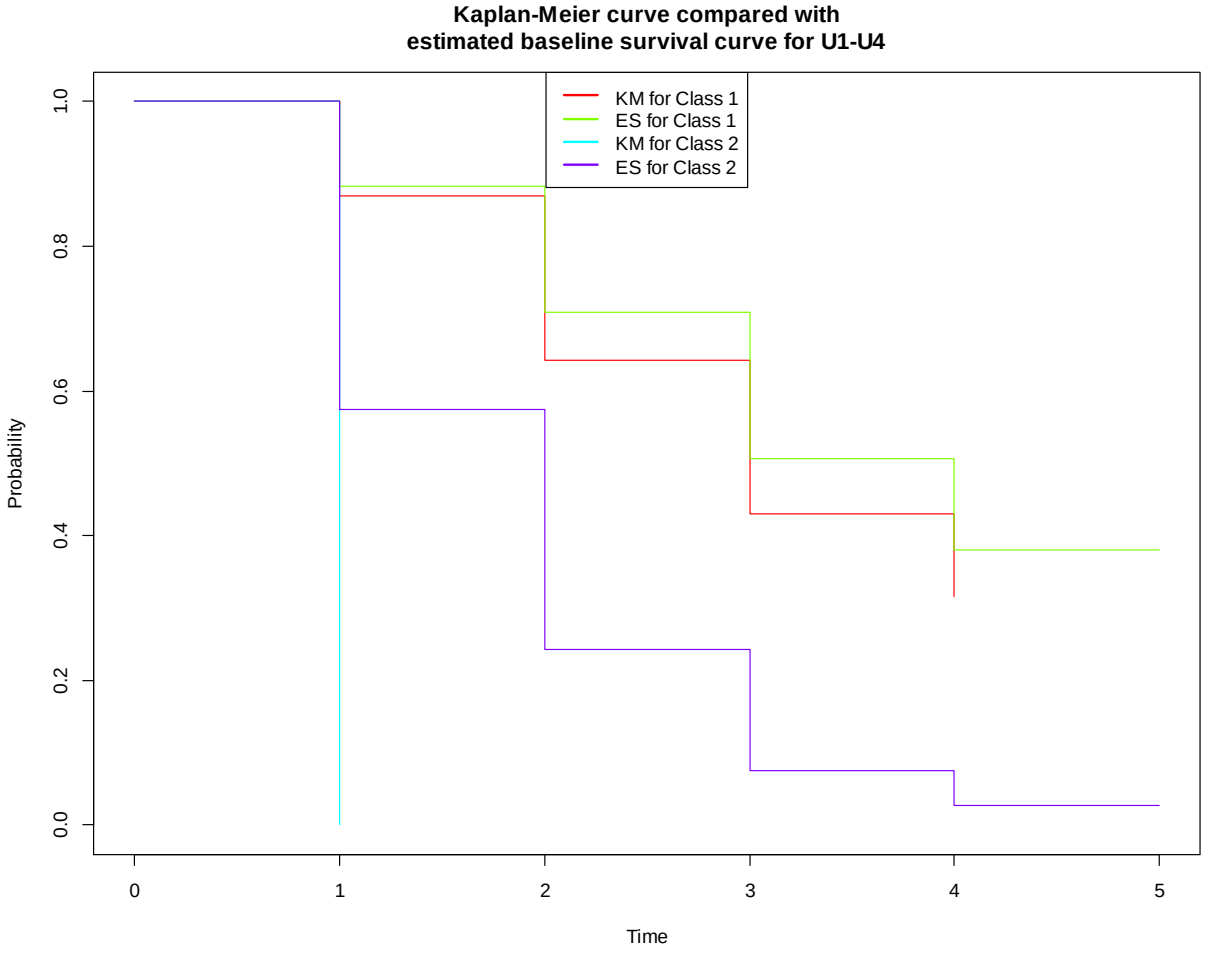
> mplus.plot.discrete.survival.baseline('ex6.19mix.gh5')



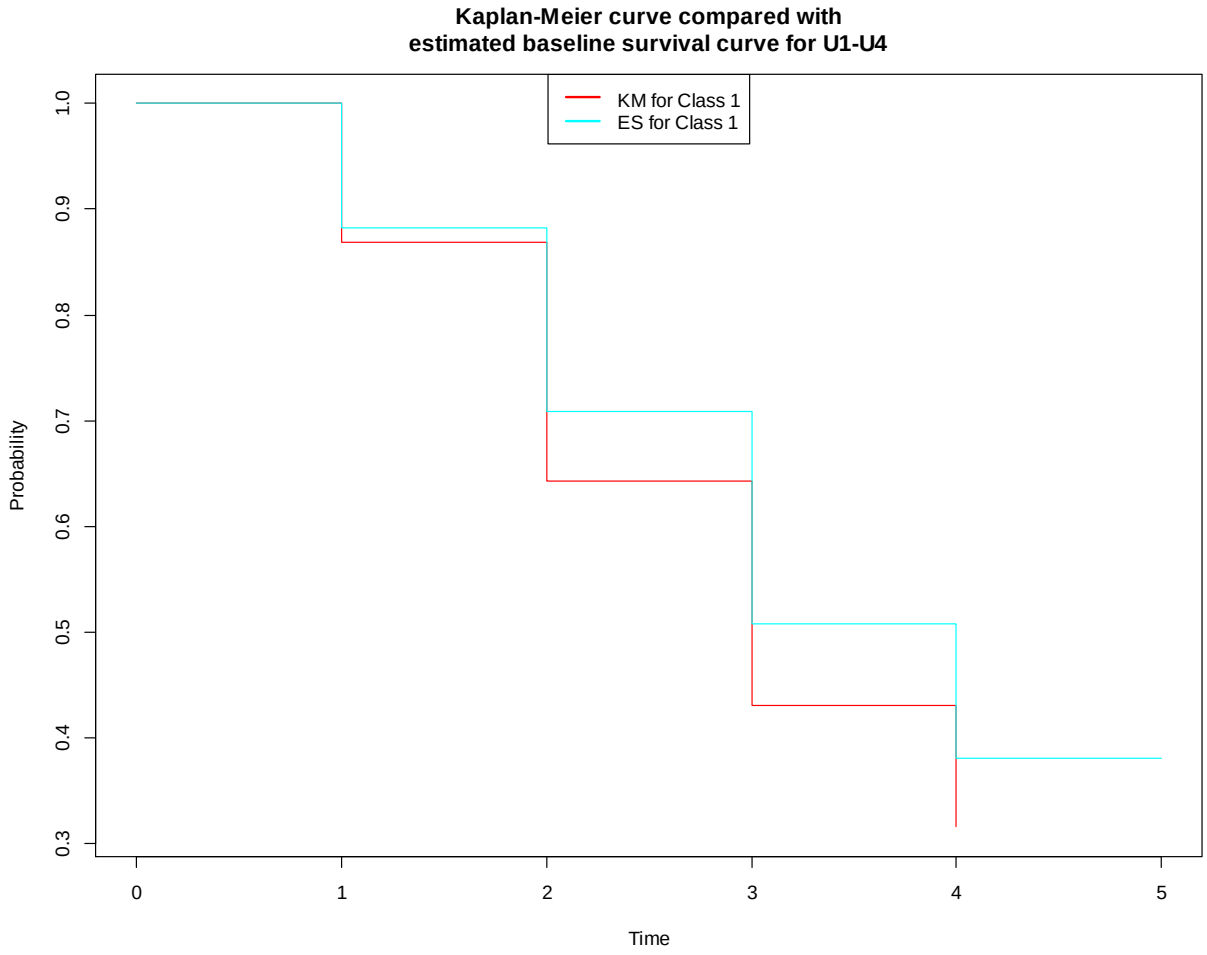
> mplus.plot.discrete.survival.baseline('ex6.19mix.gh5',classnum=2)



> mplus.plot.discrete.survival.kaplanmeier.vs.baseline('ex6.19mix.gh5')



> mplus.plot.discrete.survival.kaplanmeier.vs.baseline('ex6.19mix.gh5',classnum=1)



Loop plots

The following function is used to get Mplus loop plots. These plots require that PLOT2 or PLOT3 settings are given for the TYPE option of the PLOT command.

Following are functions for loop plots:

- mplus.list.loop.labels(file)
- mplus.plot.loop(file, label, showgrid)

The file argument is the name of the Mplus GH5 file.

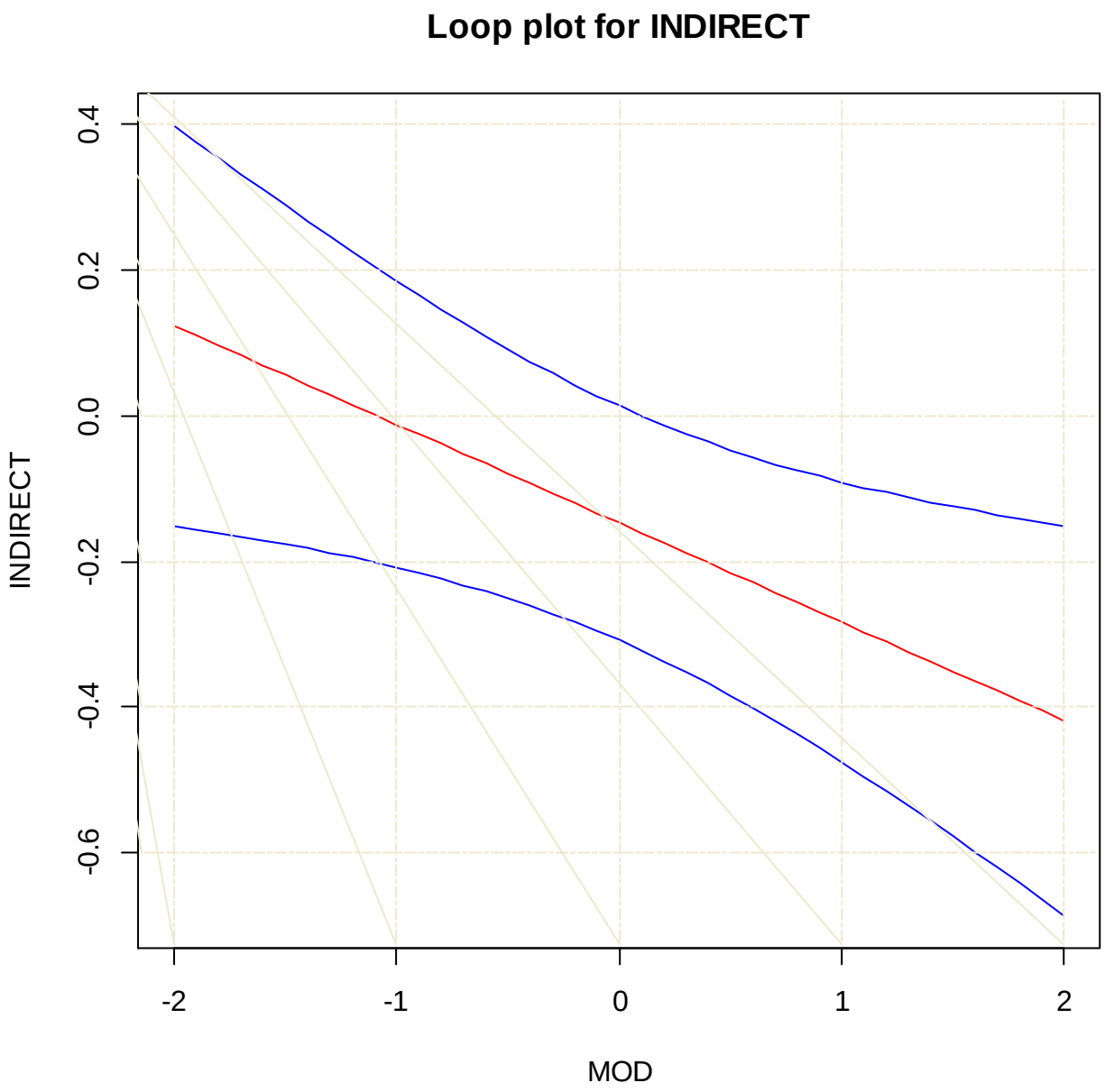
The label argument refers to the label given in the PLOT function of the MODEL CONSTRAINT command. To view a list of labels, use the function mplus.list.loop.labels. The label argument can be the actual label, an index of the label in the list, a list of labels or a list of label indices. This argument is not required. If no label is given, the default will be the first label in the list.

```
> mplus.list.loop.labels(FILE)
```

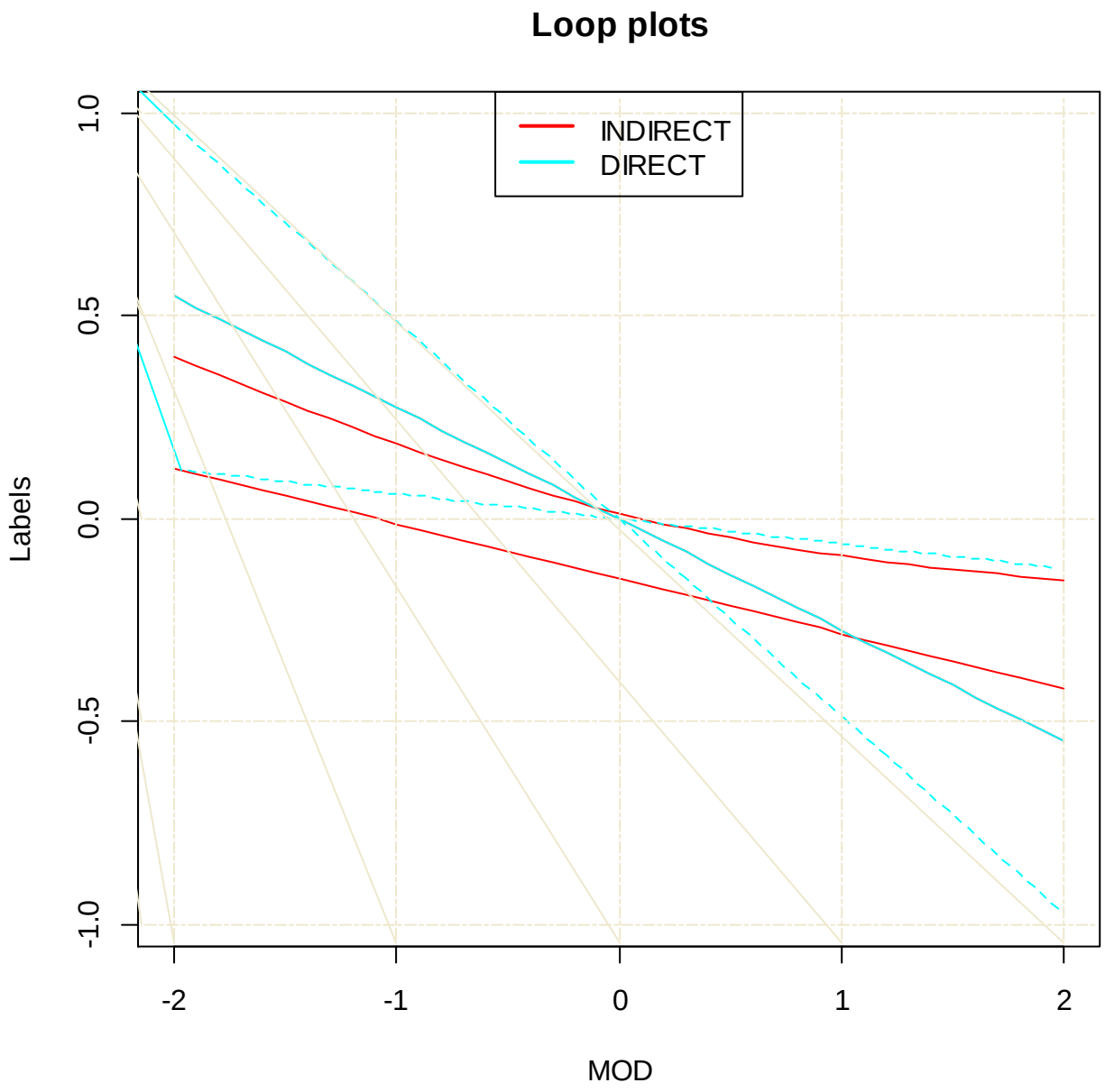
List of loop labels to use in the following functions:
- mplus.plot.loop
- mplus.get.loop.estimates
- mplus.get.loop.lowerci
- mplus.get.loop.upperci
- mplus.get.loop.xvalues

Loop labels:
[1] INDIRECT
[2] DIRECT

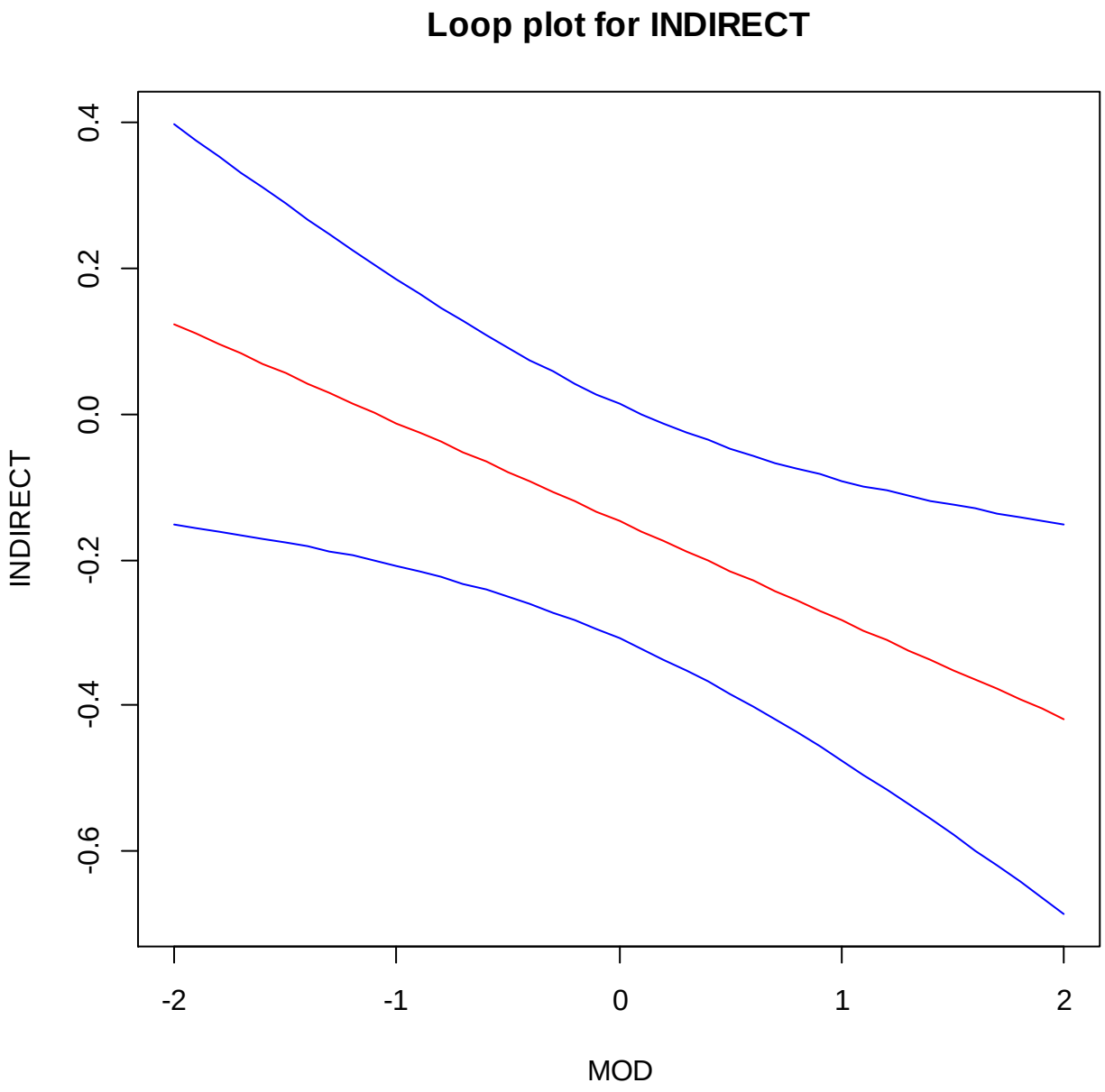
```
> mplus.plot.loop(FILE,'indirect')
> mplus.plot.loop(FILE,1)
```



```
> mplus.plot.loop(FILE,c('indirect', 'direct'))
```



```
> mplus.plot.loop(FILE,1,showgrid=FALSE)
```



The following functions can be used to get the values shown in the loop plots.

- mplus.get.loop.estimates(file,label)
- mplus.get.loop.lowerci(file,label)
- mplus.get.loop.upperci(file,label)
- mplus.get.loop.xvalues(file)

The label argument for the functions above is the same as described above for the mplus.plot.loop function. The function mplus.get.loop.xvalues is used to retrieve the values for the x-axis. These values are the values determined by the LOOP function in MODEL CONSTRAINT.

Moderation plots

Moderation plots require that PLOT2 or PLOT3 settings are given for the TYPE option of the PLOT command and the MOD option with four arguments is specified in the MODEL INDIRECT command. Here is an example of the MOD option with four arguments:

MODEL INDIRECT:
y MOD m z (-1 1 0.1) mz x;

Following are functions for moderation plots:

- mplus.list.moderation.labels(file)
- mplus.plot.moderation(file,label,group,allgroups,showgrid,lloc)

The file argument is the name of the Mplus GH5 file.

The label argument refers to the label of the indirect effects given for the MOD option in the MODEL INDIRECT command. To view a list of labels, use the function mplus.list.moderation.labels. The label argument can be the actual label or an index of the label in the list. This argument is not required. If no label is given, the default will be the first indirect effect label in the list.

Moderation plots are available for each group in a multiple group analysis. The group argument refers to the group index. The group argument defaults to 1. The allgroups argument specifies when moderation plots for all groups of a given indirect effect should be plotted together. To plot all groups together, set allgroups to TRUE. The default for the allgroups argument is FALSE. When moderation plots for all groups are plotted together, a legend is shown to indicate the curves for the different groups. The lloc argument is used to specify the placement of the legend. The lloc argument is a quoted string and must be one of the predefined values for the legend function in R. The default is lloc="top". The following are possible settings for the lloc argument:

- lloc="top": The legend is centered horizontally at the top of the plot area.
- lloc="bottom": The legend is centered horizontally at the bottom of the plot area.
- lloc="left": The legend is centered vertically and starts from the left edge of the plot area.
- lloc="right": The legend is centered vertically and starts from the right edge of the plot area.
- lloc="center": The legend is centered in the plot area.
- lloc="topleft"
- lloc="topright"
- lloc="bottomleft"
- lloc="bottomright"

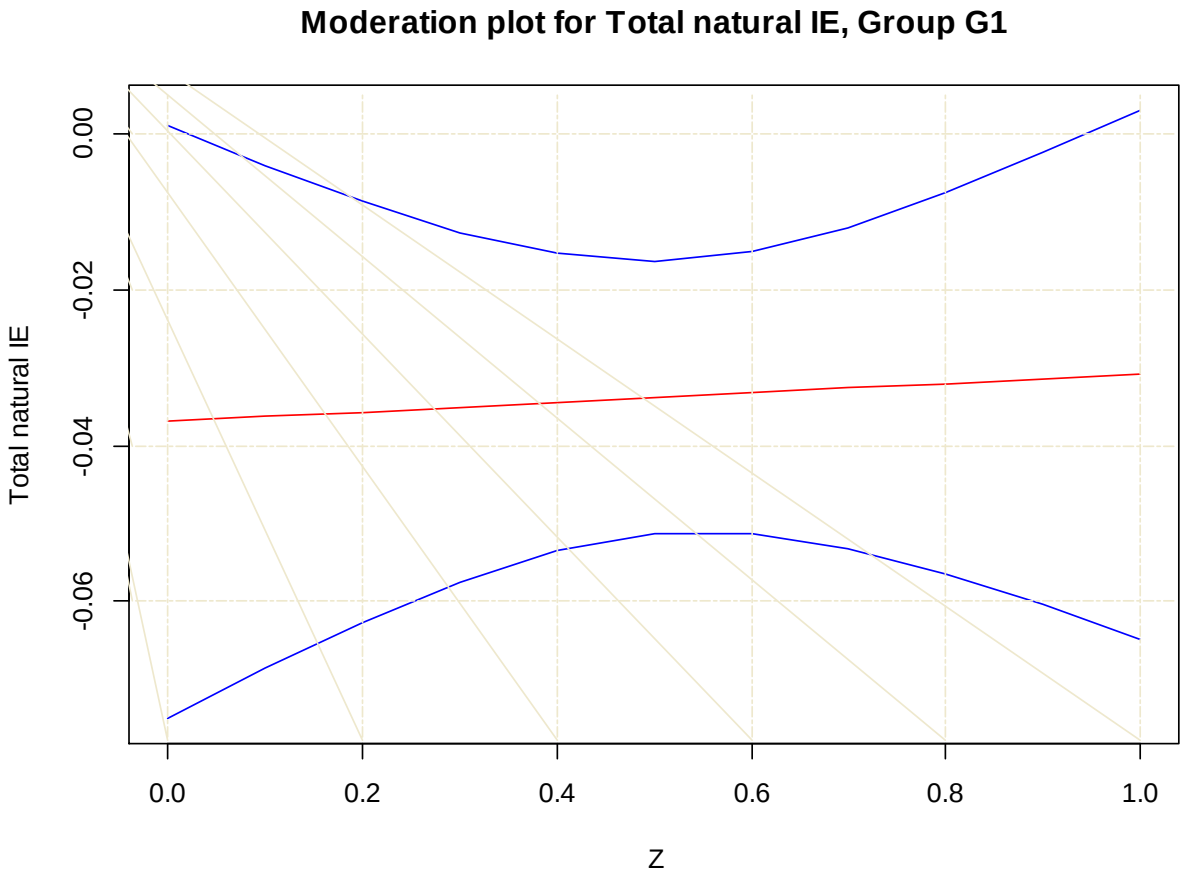
By default, the sensitivity plot is shown with grid lines. The showgrid argument is used to change the appearance of grid lines. To turn off grid lines, specify showgrid=FALSE.

```
> mplus.list.moderation.labels(FILE)
```

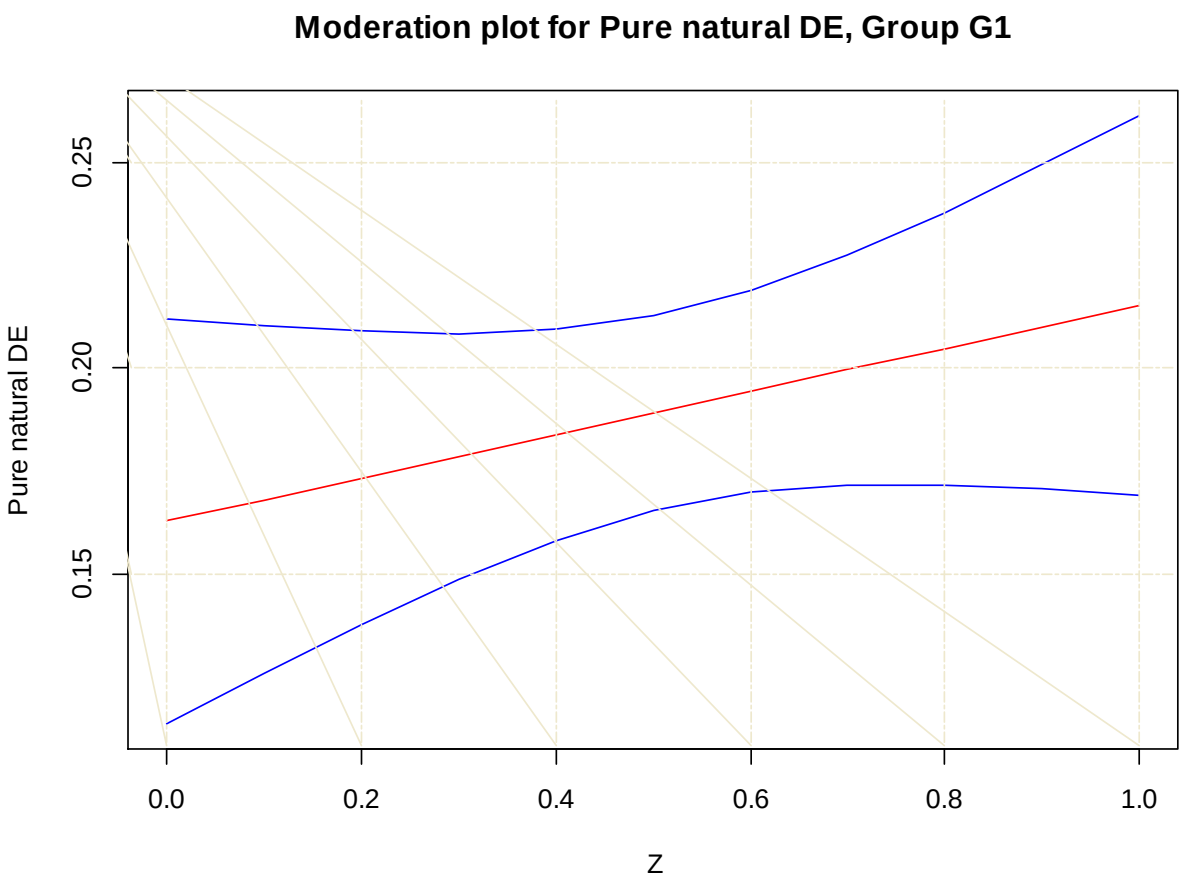
List of moderation labels to use in the following functions:
- mplus.plot.moderation
- mplus.get.moderation.estimates
- mplus.get.moderation.lowerci
- mplus.get.moderation.upperci
- mplus.get.moderation.xvalues

Moderation labels:
[1] Total natural IE
[2] Pure natural DE
[3] Total effect
[4] Pure natural IE
[5] Total natural DE

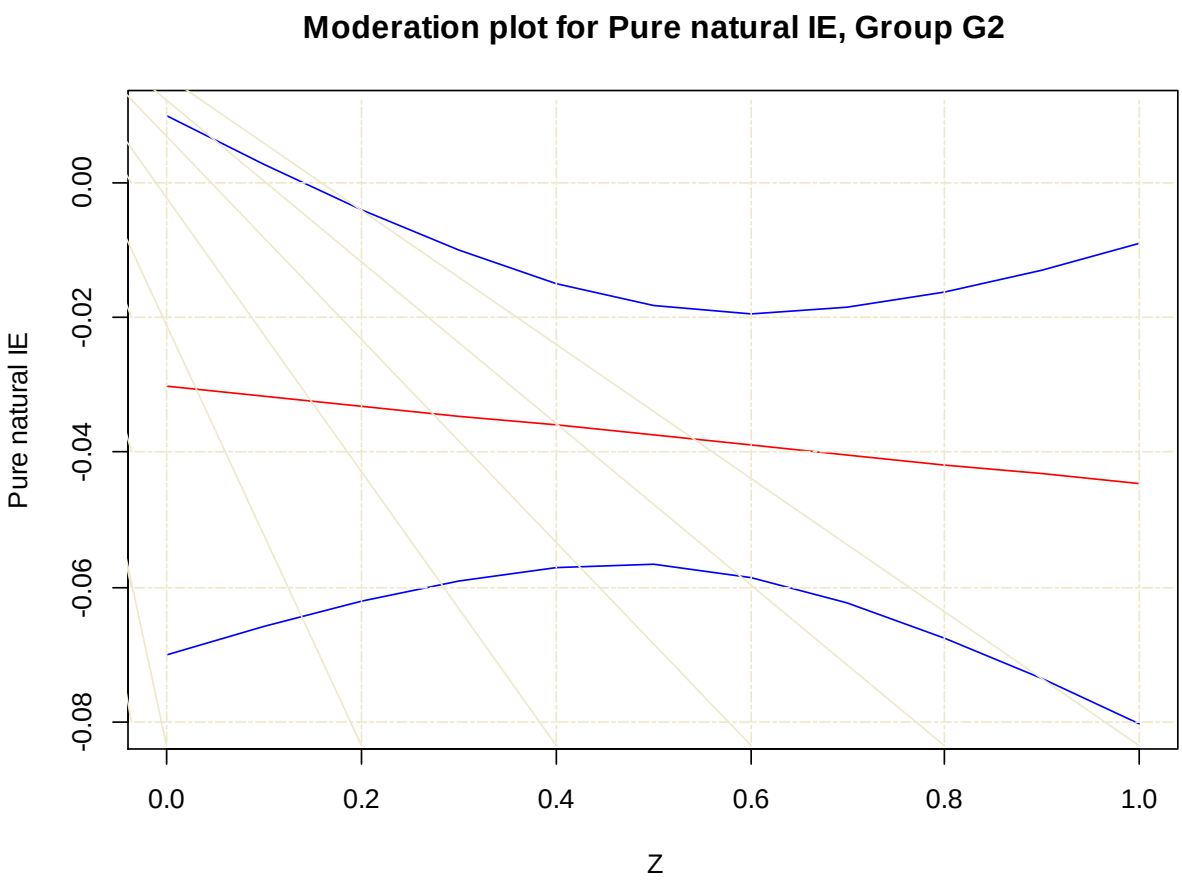
```
> mplus.plot.moderation(FILE)
> mplus.plot.moderation(FILE,1)
> mplus.plot.moderation(FILE,'Total natural IE')
```



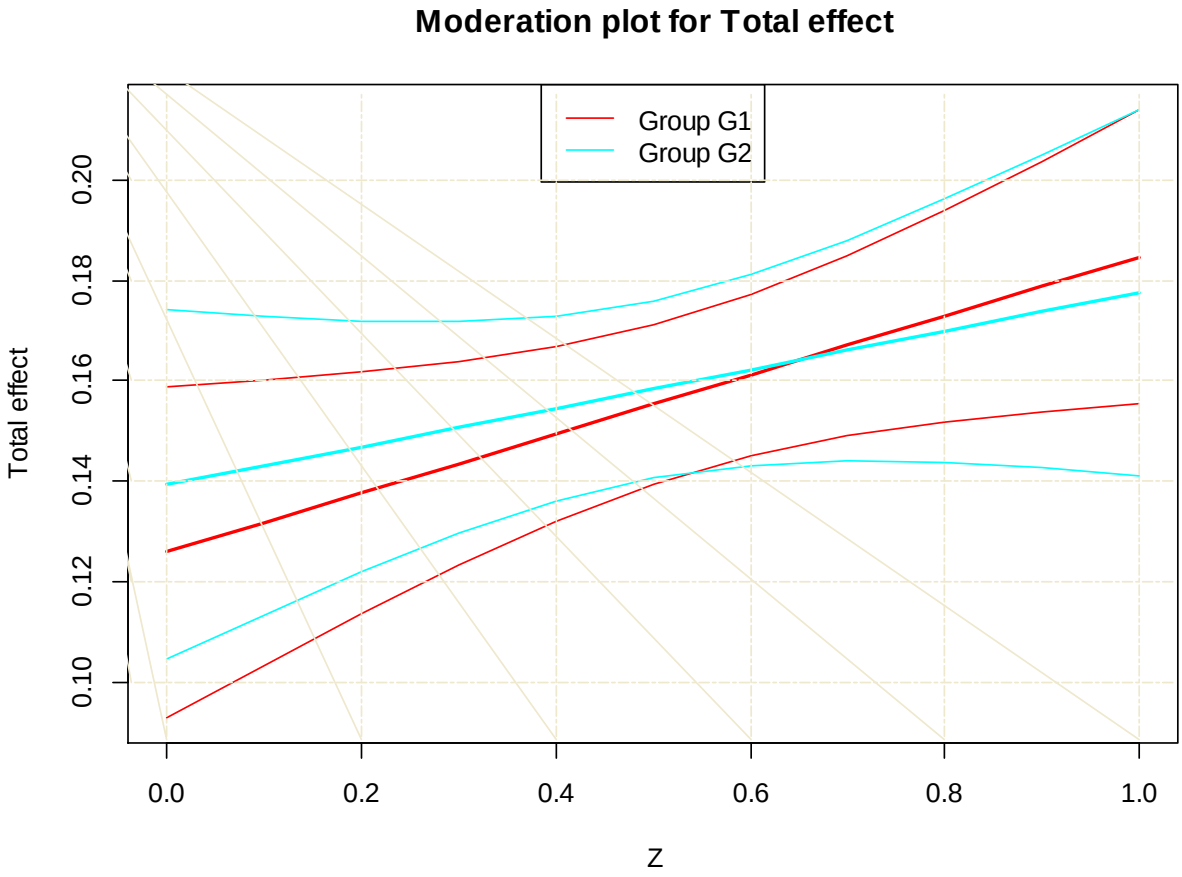
```
> mplus.plot.moderation(FILE,2)
> mplus.plot.moderation(FILE,'Pure natural DE')
```



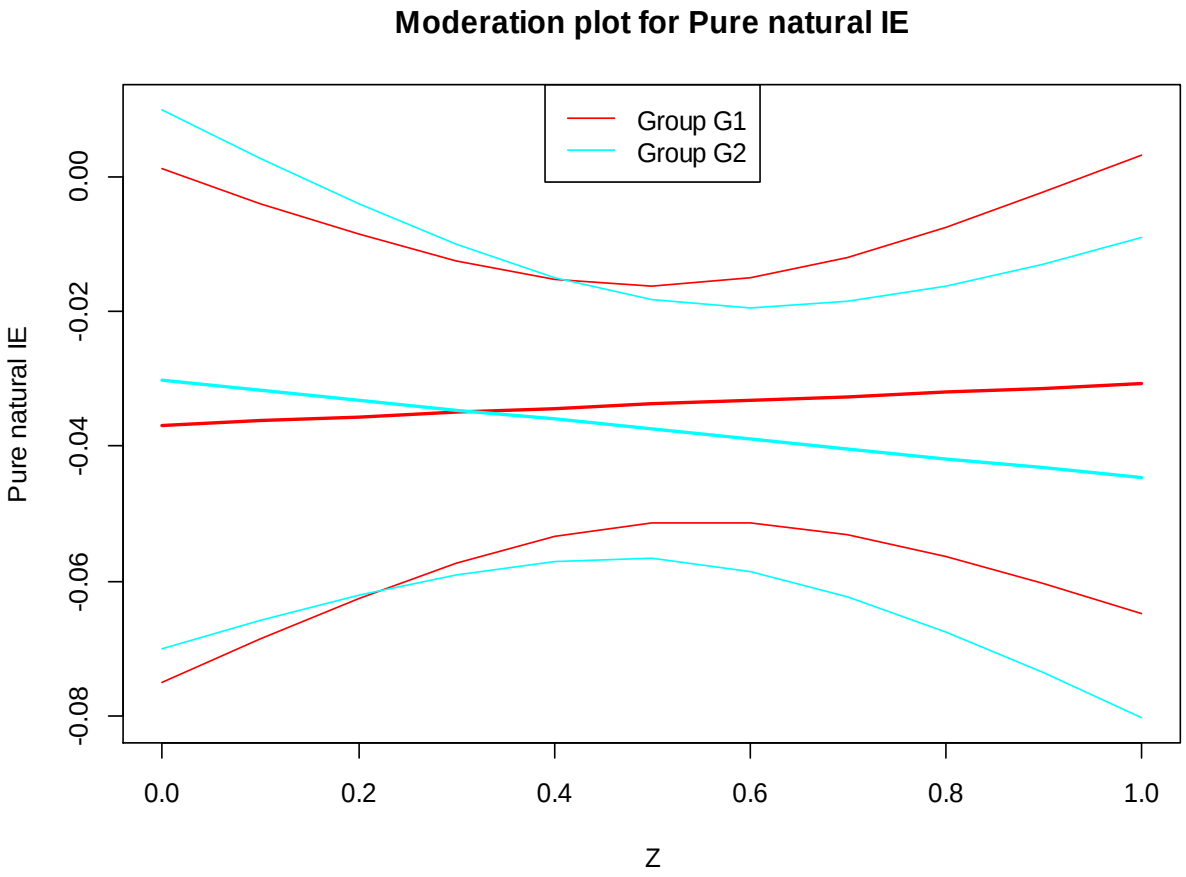
```
> mplus.plot.moderation(FILE,4,group=2)
> mplus.plot.moderation(FILE,'Pure natural IE',group=2)
```



```
> mplus.plot.moderation(FILE,3,allgroups=TRUE)
> mplus.plot.moderation(FILE,'Total effect',allgroups=TRUE)
```



> mplus.plot.moderation(FILE,label=4,allgroups=TRUE,showgrid=FALSE)



The following functions can be used to get the values shown in the moderation plots.

- mplus.get.moderation.estimates(file,label)
- mplus.get.moderation.lowerci(file,label)
- mplus.get.moderation.upperci(file, label)
- mplus.get.moderation.xvalues(file)

The label argument for the functions above is the same as described above for the mplus.plot.moderation function. The function mplus.get.sensitivity.xvalues is used to retrieve the values for the x-axis. These values are the values given for the Z variable in MODEL INDIRECT.

Sensitivity plots

Sensitivity plots are available for certain indirect effects in MODEL INDIRECT. Sensitivity plots require that SENSITIVITY setting is specified in conjunction with PLOT2 or PLOT3 settings for the TYPE option of the PLOT command.

Following are functions for sensitivity plots:

- ❖ mplus.list.sensitivity.labels(file)
- ❖ mplus.plot.sensitivity(file,label,zvalue,group,allgroups,xrange,showgrid,lloc,zdecimals)

The file argument is the name of the Mplus GH5 file.

The label argument refers to the label of the indirect effects with sensitivity plot. To view a list of labels, use the function mplus.list.sensitivity.labels. The label argument can be the actual label or an index of the label in the list. This argument is not required. If no label is given, the default will be the first indirect effect label in the list.

For indirect effects that are given for the MOD statement with four arguments in the MODEL INDIRECT command, sensitivity plots are given for each of the Z values. The zvalue argument is used specify the value of Z for the indirect effect. The default is the first value of Z. By default, the zvalue argument and all values of Z are rounded to 3 decimals to avoid numerical imprecision when finding the right plot information. If more precision is needed, use the zdecimals argument. For example, to round to the 4th decimals, specify zdecimals=4.

Sensitivity plots are available for each group in a multiple group analysis. The group argument refers to the group index. The group argument defaults to 1. The allgroups argument specifies when sensitivity plots for all groups of a given indirect effect should be plotted together. To plot all groups together, set allgroups to TRUE. The default for the allgroups argument is FALSE. When sensitivity plots for all groups are plotted together, a legend is shown to indicate the curves for the different groups. The lloc argument is used to specify the placement of the legend. The lloc argument is a quoted string and must be one of the predefined values for the legend function in R. The default is lloc="top". The following are possible settings for the lloc argument:

- lloc="top": The legend is centered horizontally at the top of the plot area.
- lloc="bottom": The legend is centered horizontally at the bottom of the plot area.
- lloc="left": The legend is centered vertically and starts from the left edge of the plot area.
- lloc="right": The legend is centered vertically and starts from the right edge of the plot area.
- lloc="center": The legend is centered in the plot area.
- lloc="topleft"
- lloc="topright"
- lloc="bottomleft"
- lloc="bottomright"

Sensitivity plots are plotted with an x-axis range from -1 to 1. The xrange argument is used to specify a custom x-axis range. For example, to plot an x-axis range of -0.9 to 0.9, specify xrange=c(-0.9,0.9).

By default, the sensitivity plot is shown with grid lines. The showgrid argument is used to change the appearance of grid lines. To turn off grid lines, specify showgrid=FALSE.

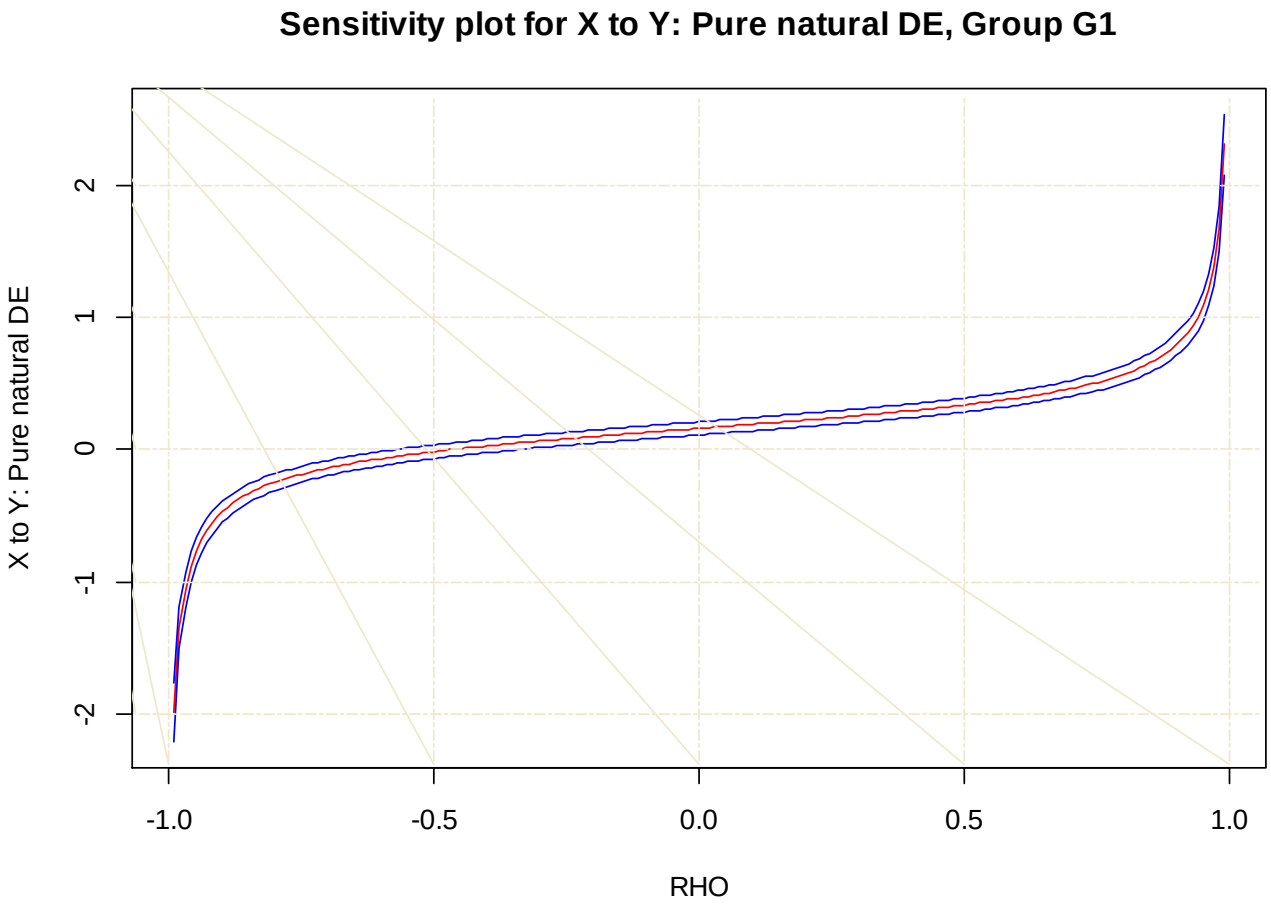
```
> mplus.list.sensitivity.labels(FILE)
```

List of sensitivity labels to use in the following functions:
- mplus.plot.sensitivity

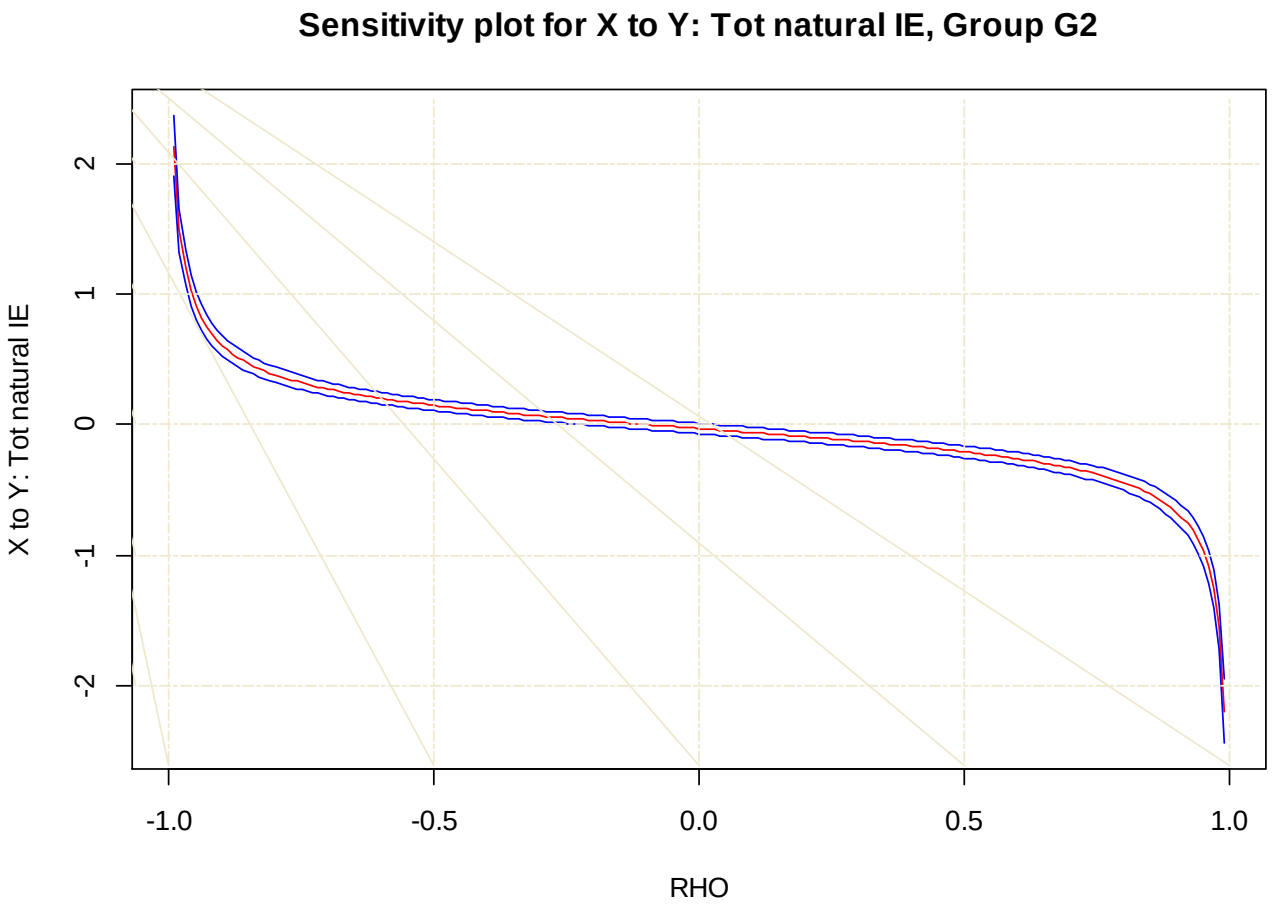
- mplus.get.sensitivity.estimates
- mplus.get.sensitivity.lowerci
- mplus.get.sensitivity.upperci
- mplus.get.sensitivity.xvalues

Sensitivity labels:
[1] X to Y: Pure natural DE
[2] X to Y: Tot natural IE
[3] X to Y for Z: Tot natural IE
[4] X to Y for Z: Pure natural DE

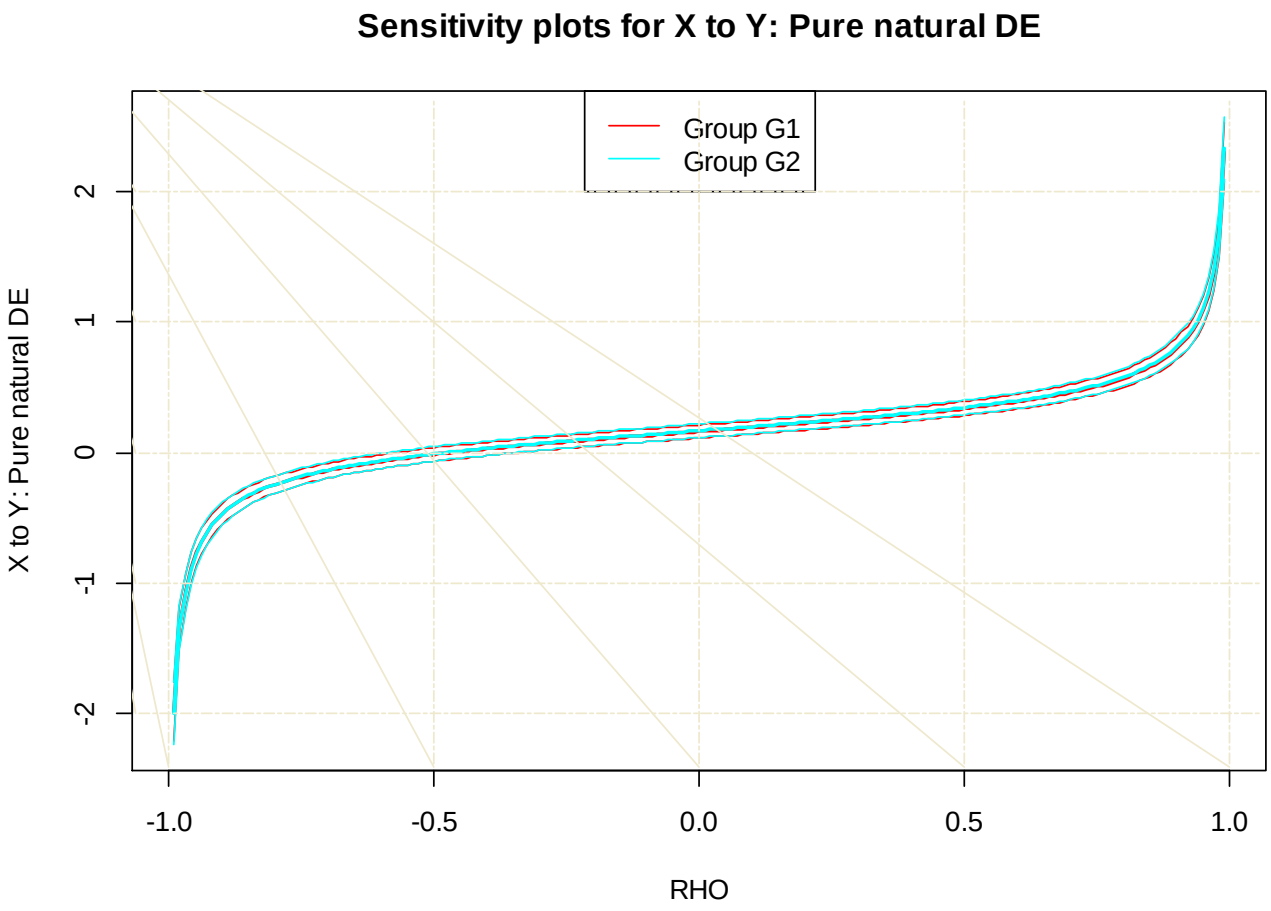
```
> mplus.plot.sensitivity(FILE)
> mplus.plot.sensitivity(FILE,1)
> mplus.plot.sensitivity(FILE,'X to Y: Pure natural DE')
```



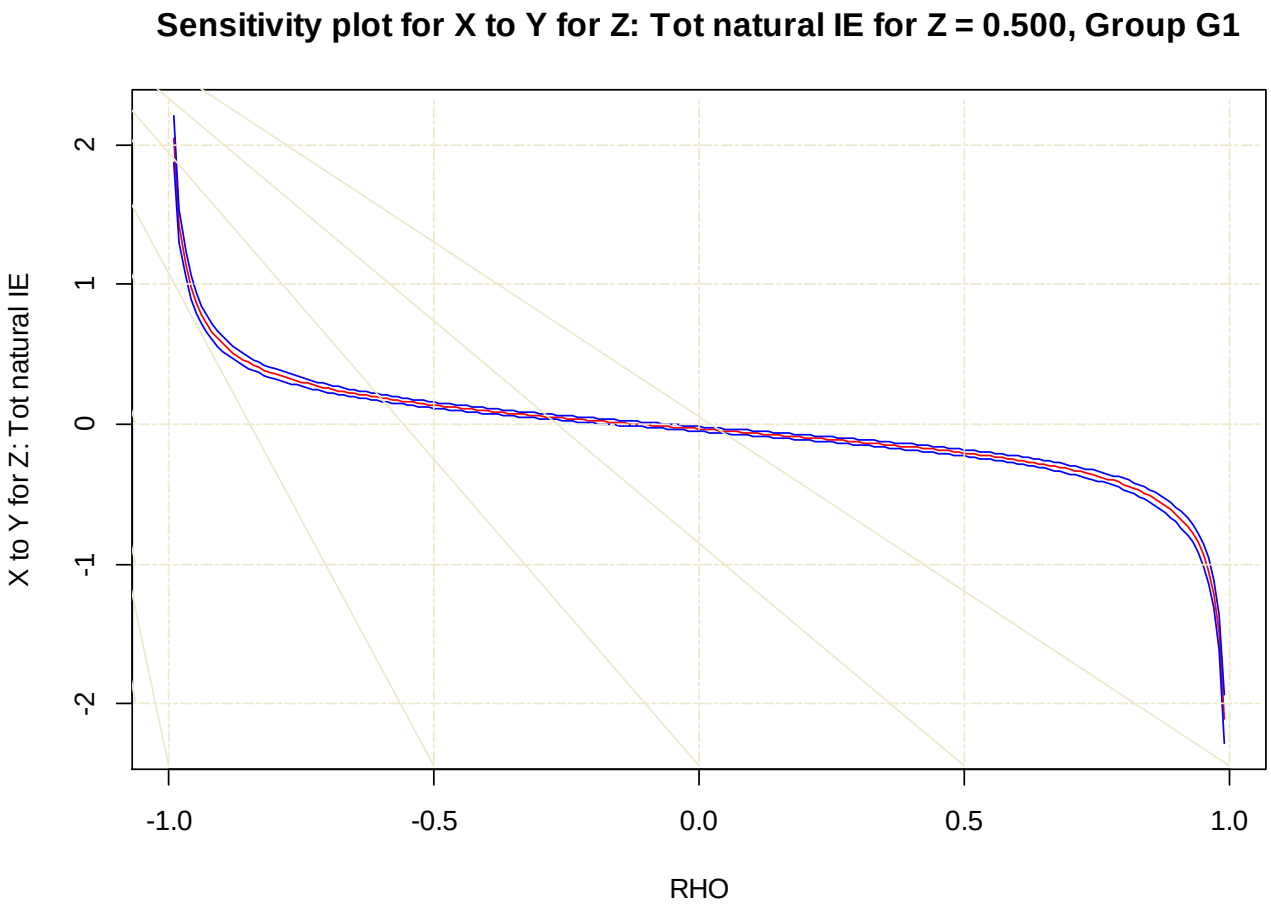
```
> mplus.plot.sensitivity(FILE,label=2,group=2)
> mplus.plot.sensitivity(FILE,'X to Y: Tot natural IE',group=2)
```



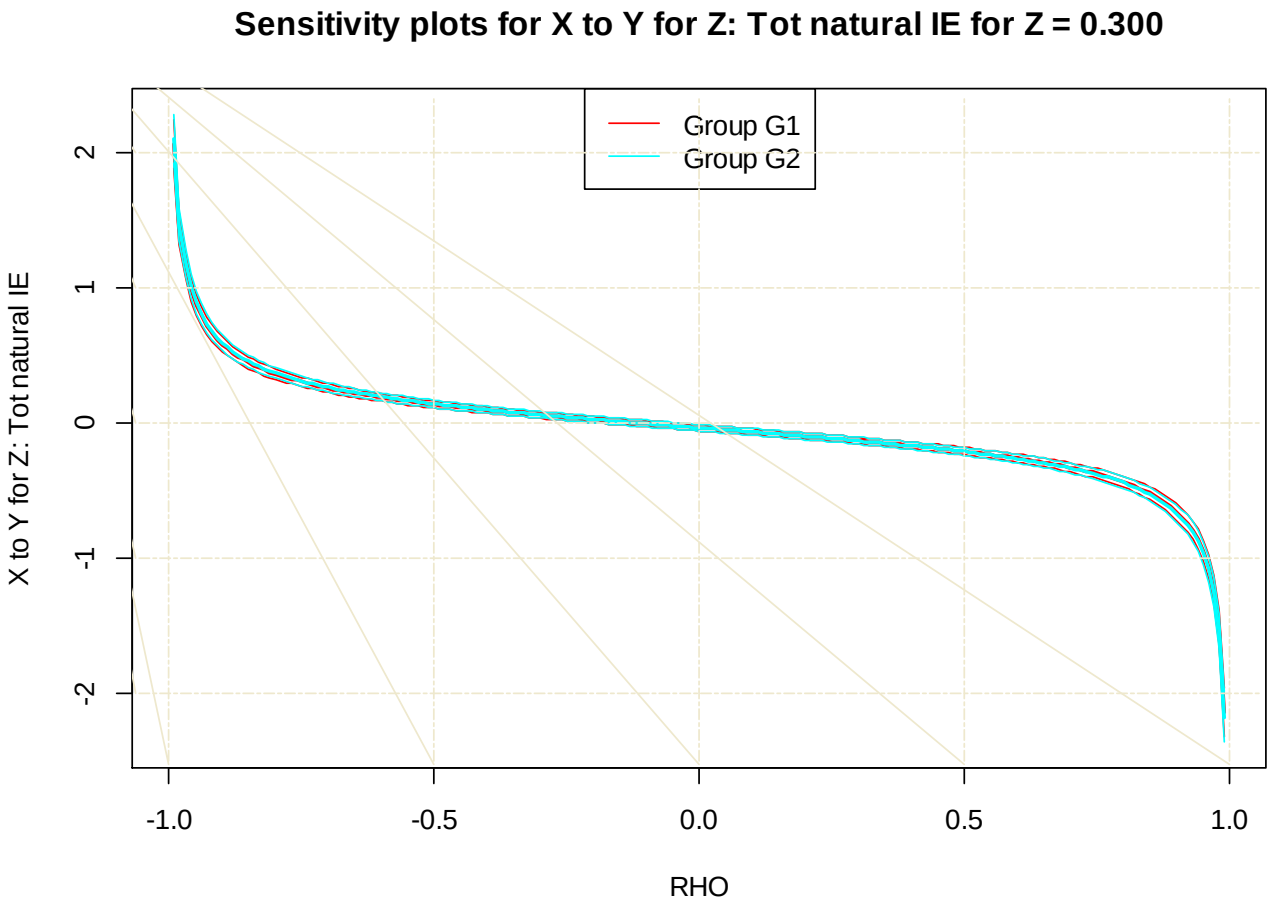
```
> mplus.plot.sensitivity(FILE,1,allgroups=TRUE)
> mplus.plot.sensitivity(FILE,'x to y: pure natural de',allgroups=TRUE)
```



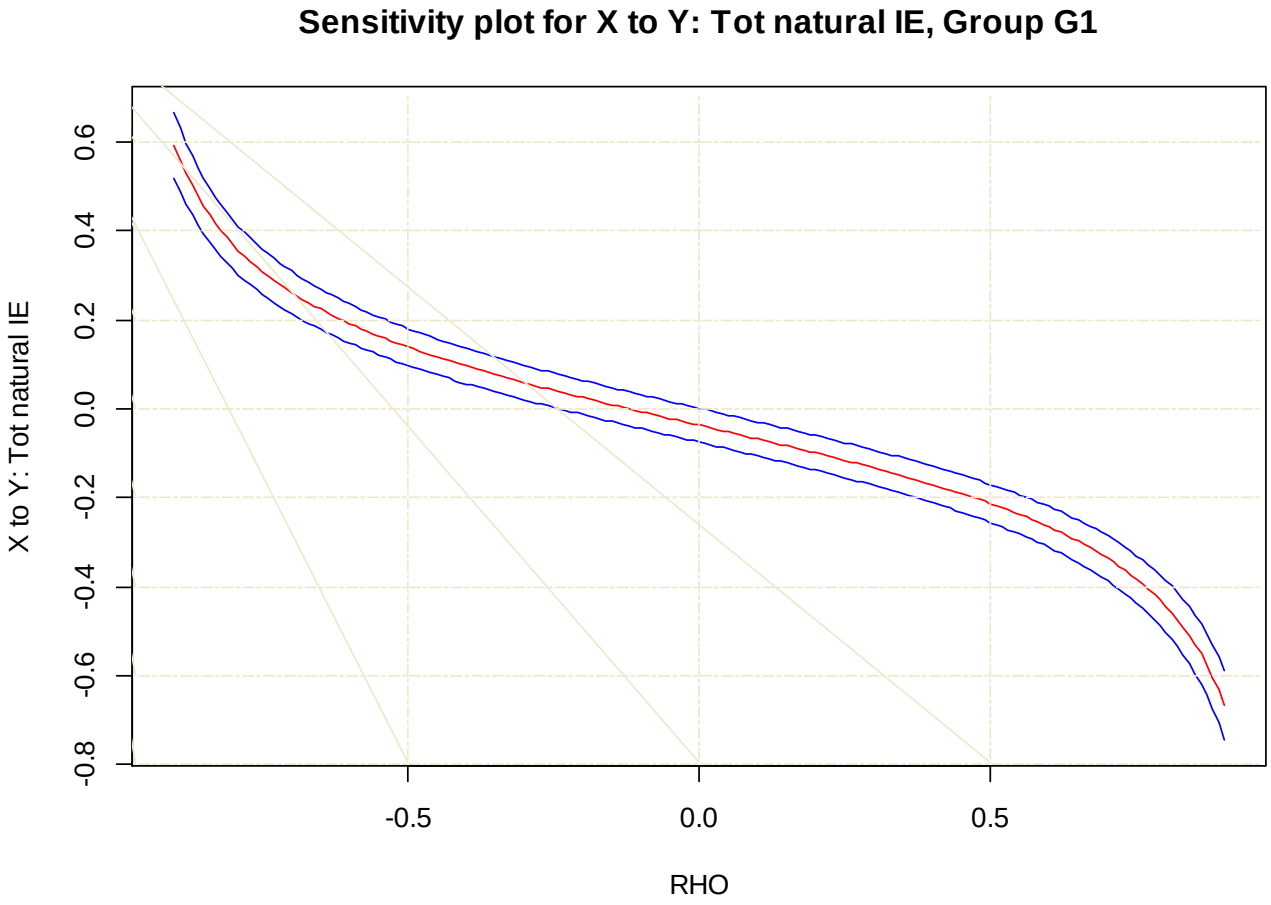
```
> mplus.plot.sensitivity(FILE,3,0.5)
> mplus.plot.sensitivity(FILE,3,0.5,1)
> mplus.plot.sensitivity(FILE,label=3,zvalue=0.5,group=1)
```



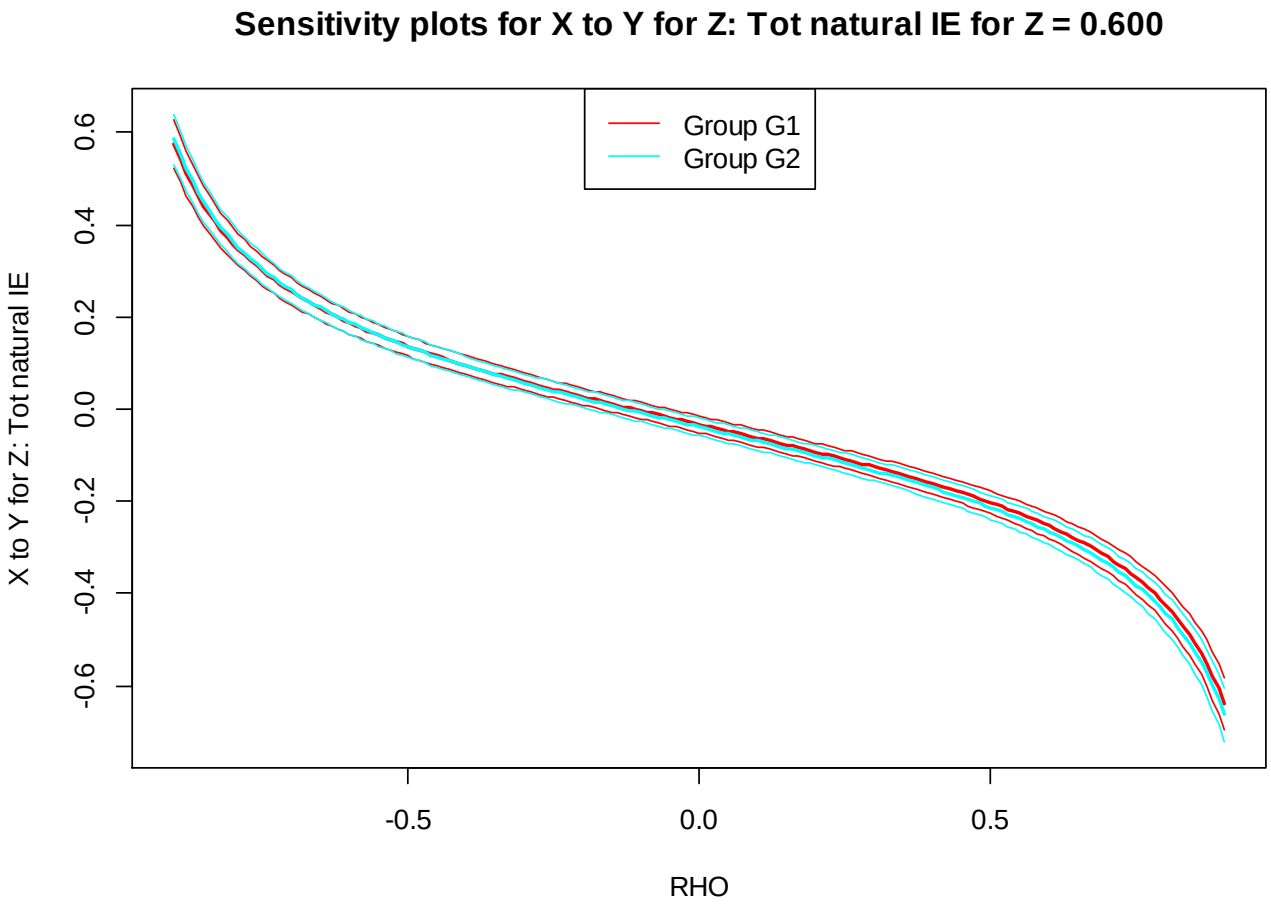
```
> mplus.plot.sensitivity(FILE,label=3,zvalue=0.3,allgroups=TRUE)
```



```
> mplus.plot.sensitivity(FILE,2,xrange=c(-0.9,0.9))
```



> mplus.plot.sensitivity(FILE,3,zvalue=0.6,allgroups=TRUE,showgrid=FALSE,xrange=c(-0.9,0.9))



The following functions can be used to get the values shown in the sensitivity plots.

- mplus.get.sensitivity.estimates(file,label,zvalue,group,zdecimals)
- mplus.get.sensitivity.lowerci(file,label,zvalue,group,zdecimals)
- mplus.get.sensitivity.upperci(file,label,zvalue,group,zdecimals)
- mplus.get.sensitivity.xvalues(file)
- mplus.list.sensitivity.zvalues(file)

The label argument for the functions above is the same as described above for the mplus.plot.moderation function. The function mplus.get.sensitivity.xvalues is used to retrieve the values for the x-axis. Mplus computes sensitivity analysis using the range -0.99 to 0.99. The function mplus.list.sensitivity.zvalues show the list of values for Z given in MODEL INDIRECT.

Bayesian plots

The following functions are used to view Bayesian plots.

- mplus.plot.bayesian.traceplot(file,parameter)
- mplus.plot.bayesian.distribution(file,parameter,bins)
- mplus.plot.bayesian.prior.distribution(file,parameter,bins)
- mplus.plot.bayesian.autocorrelation(file,parameter,chain)
- mplus.plot.bayesian.predictive.scatterplot(file,parameter)
- mplus.plot.bayesian.predictive.distribution(file,parameter,bins)
- mplus.plot.bayesian.plausible.distribution(file,plauslabel,obs,bins)
- mplus.list.bayesian.parameters(file,parameter)
- mplus.list.bayesian.predictive.labels(file)
- mplus.list.bayesian.plausible.labels(file)

The function mplus.list.bayessian.parameters gives the list of parameter labels. The complete name of the parameter label or the index of the parameter label in this list can be given for the parameter argument in the various functions. The bins argument is the number of bins to use in plotting the distribution. The bins argument is optional. The default is 100 bins.

For the function mplus.plot.bayesian.autocorrelation, the chain argument refers to the chain number. The chain argument is optional. By default, chain 1 is displayed.

The plauslabel argument for the function mplus.plot.bayesian.plausible.distribution refers to the plausible label or the index of the label. The index refers to the ordering of the plausible labels given by the function mplus.list.bayesian.plausible.labels. The obs argument refers to the observation number. If no observation number is given or if 0 is given for the obs argument, then the overall distribution is shown for the plausible label. Otherwise, the distribution is given for the specific observation. The bins argument refers to the number of bins used to display the distribution. The default is 100 bins for the overall distribution and 10 bins for the distribution of specific observations.

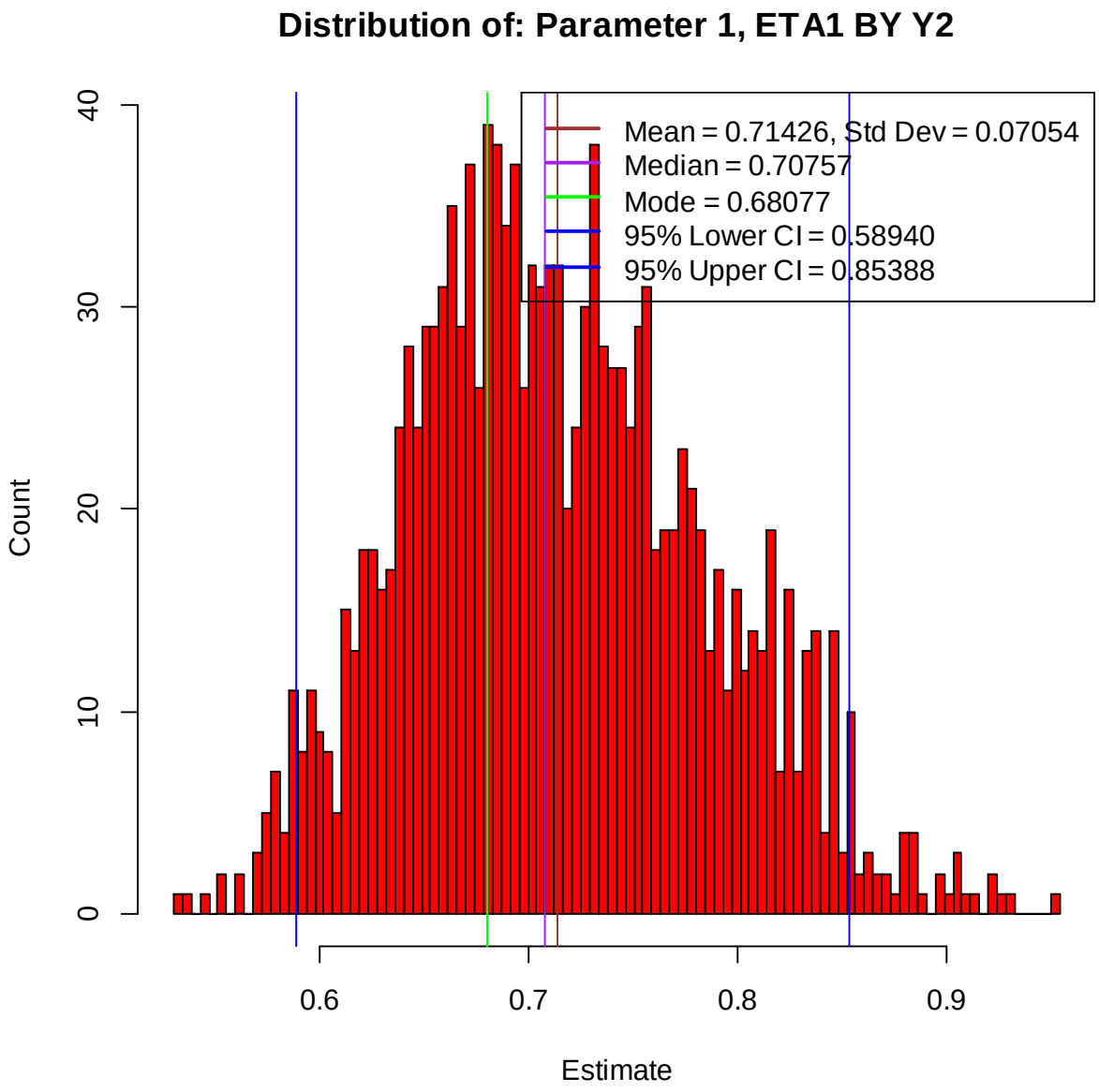
The plabel argument for the functions mplus.plot.bayesian.predictive.distribution and mplus.plot.bayesian.predictive.scatterplot refers to the predictive label or the index of the predictive label. The function mplus.list.bayesian.predictive.labels give the list of predictive labels. The function mplus.plot.bayesian.predictive.scatterplot plots the observed values vs. the replicated values. The function mplus.plot.bayesian.predictive.distribution plots the distribution of the observed minus replicated values. The bins argument specifies the number of bins to use in the plot. The default is 100 bins.

Various functions are available to extract the different plot data in the various Bayesian plots.

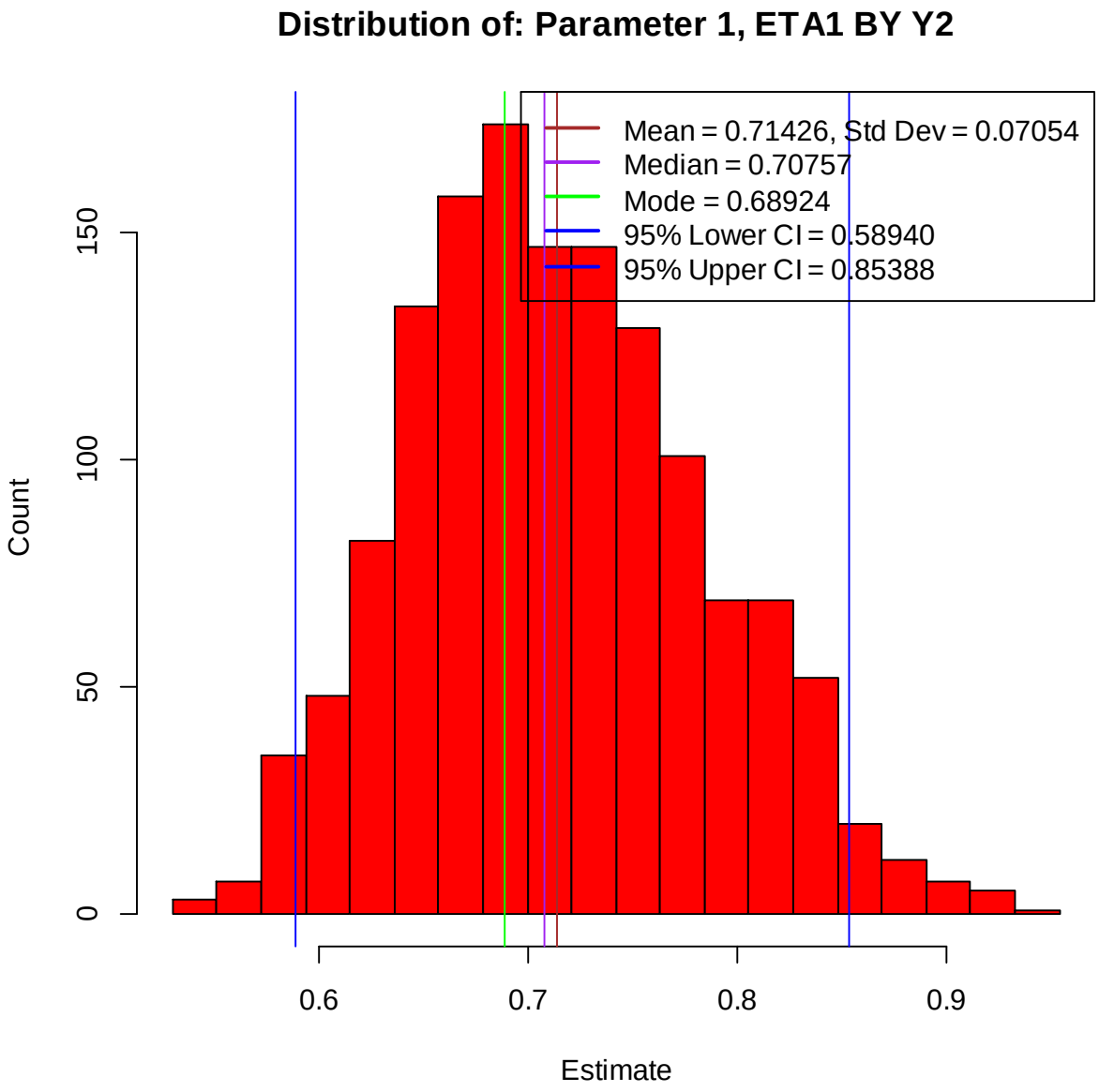
- mplus.get.bayesian.parameter.data('bcat.gh5',parameter,chain)
- mplus.get.bayesian.prior.parameter.data('bcat.gh5',parameter)
- mplus.get.bayesian.autocorrelation('bcat.gh5',parameter,chain)
- mplus.get.bayesian.predictive.observed('bcat.gh5',plabel)
- mplus.get.bayesian.predictive.replicated('bcat.gh5',plabel)
- mplus.get.bayesian.predictive.lowerci('bcat.gh5',plabel)
- mplus.get.bayesian.predictive.upperci('bcat.gh5',plabel)
- mplus.get.bayesian.predictive.pvalue('bcat.gh5',plabel)
- mplus.get.bayesian.predictive.pvalue('bcat.gh5',plabel)
- mplus.get.bayesian.plausible.data('bcat.gh5',plauslabel,obs)

The arguments are the same as described above for the related plot functions.

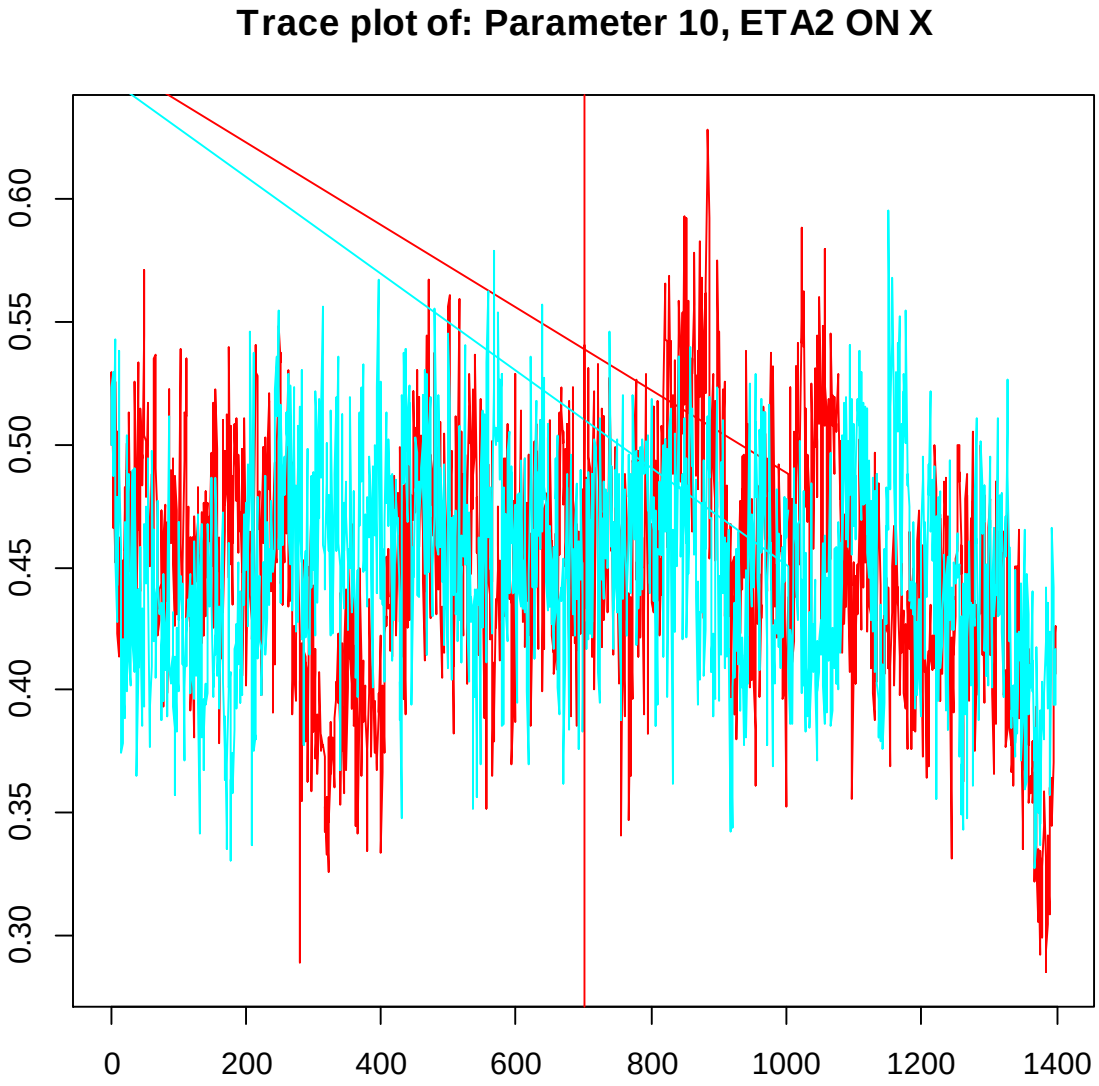
```
> mplus.plot.bayesian.distribution('bcat.gh5',1)
> mplus.plot.bayesian.distribution('bcat.gh5','parameter 1, eta1 by y2')
```



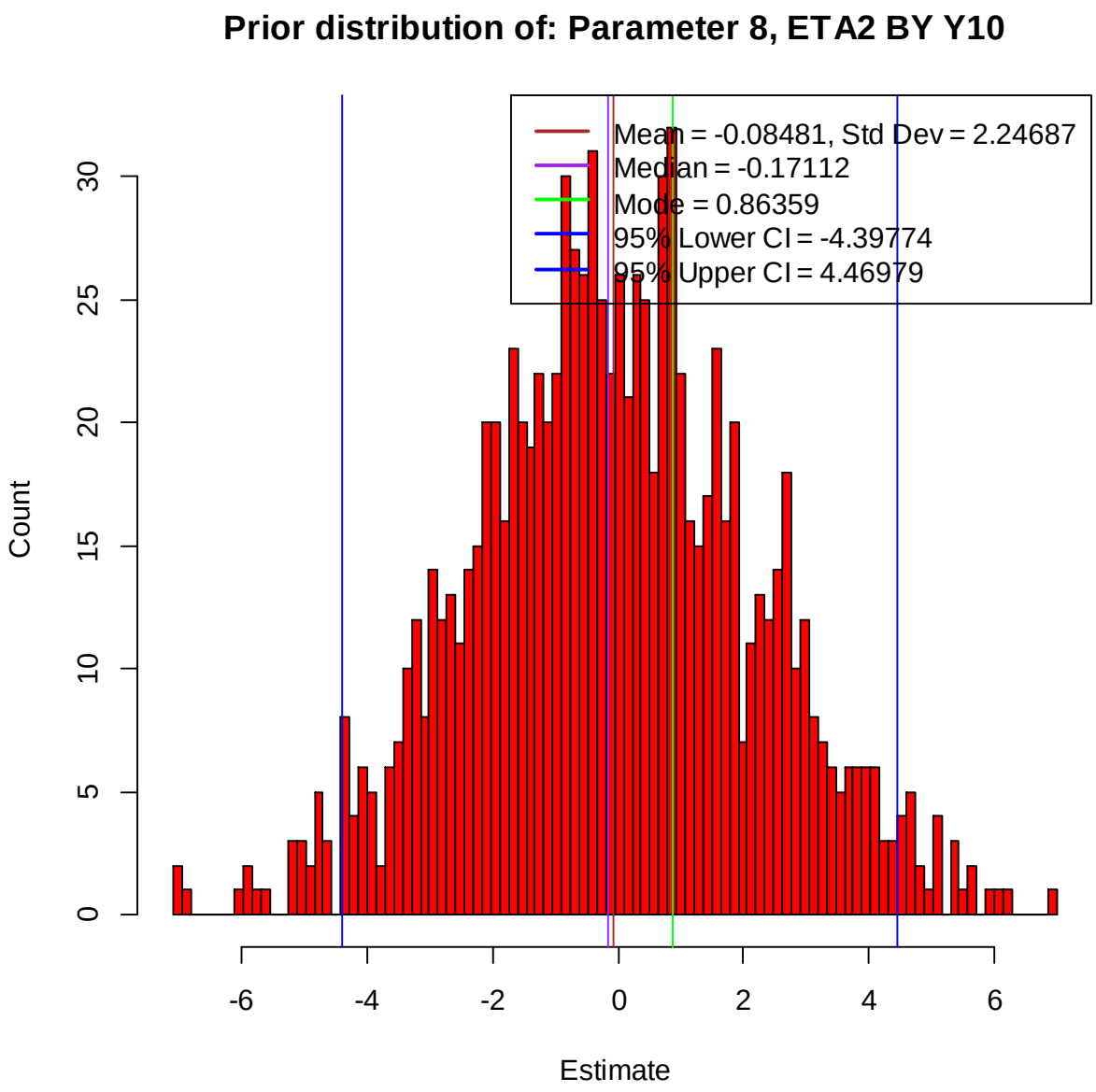

```
> mplus.plot.bayesian.distribution('bcat.gh5',1,20)
```



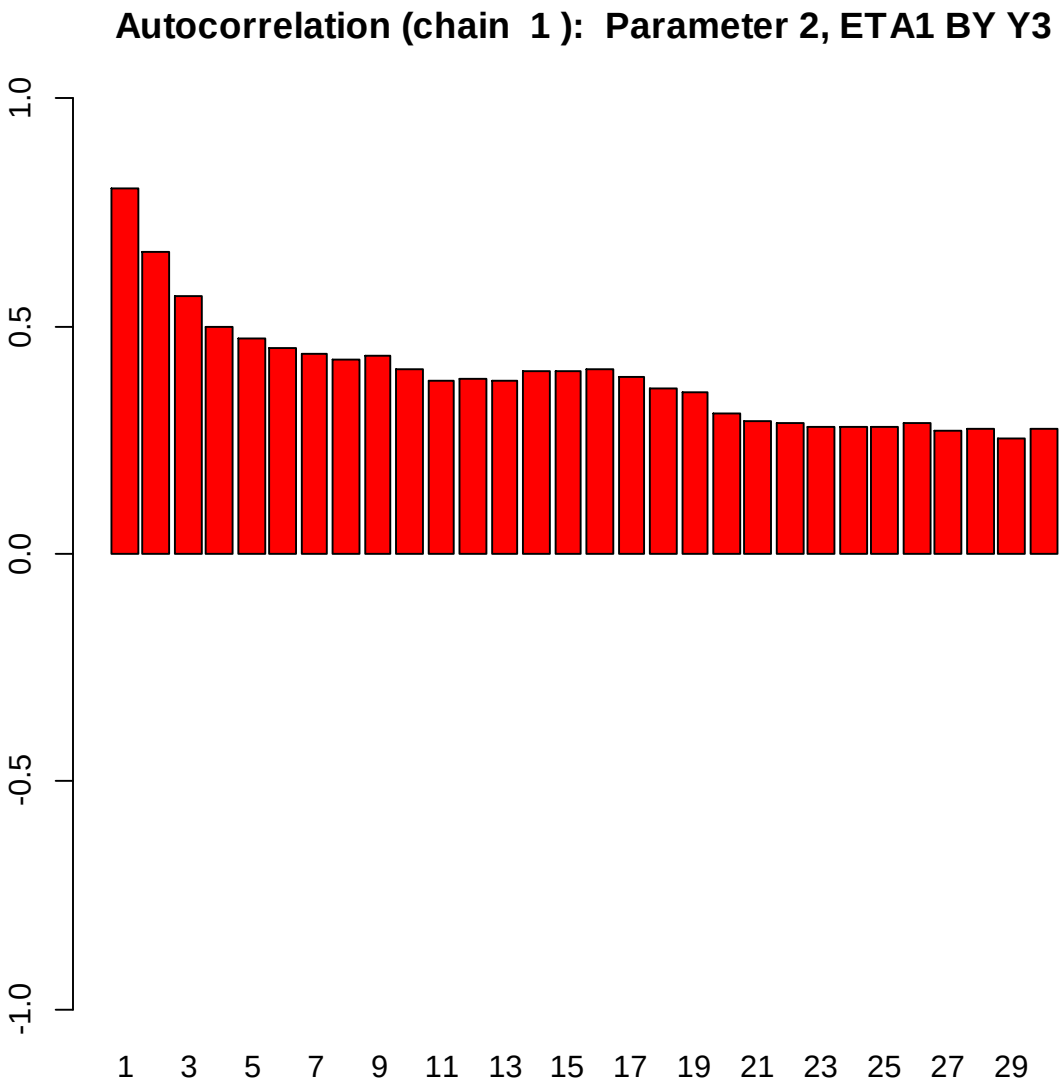

```
> mplus.plot.bayesian.traceplot('bcat.gh5',10)
> mplus.plot.bayesian.traceplot('bcat.gh5','parameter 10, eta2 on x')
```



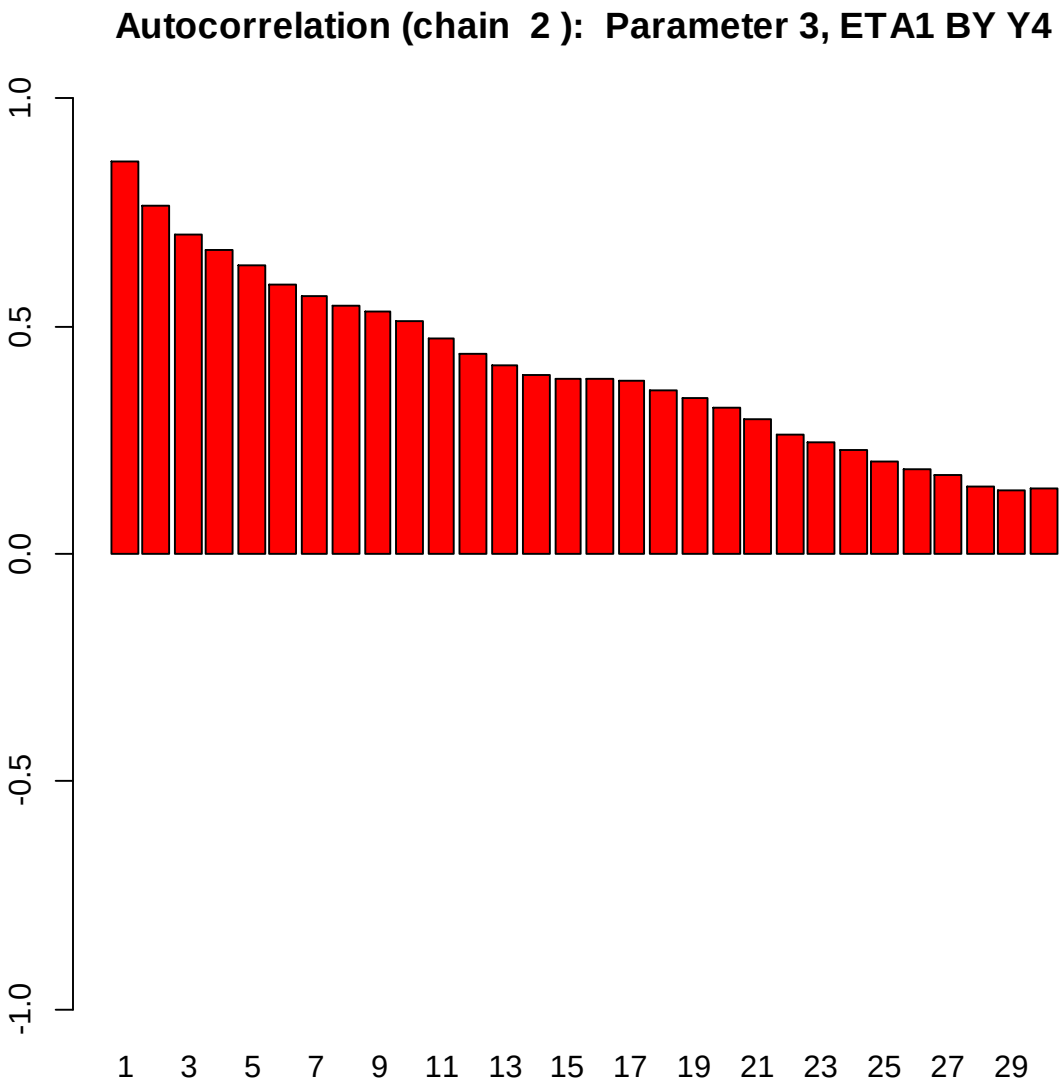
> mplus.plot.bayesian.prior.distribution('bcat.gh5',8)
> mplus.plot.bayesian.prior.distribution('bcat.gh5','parameter 8, eta2 by y10')



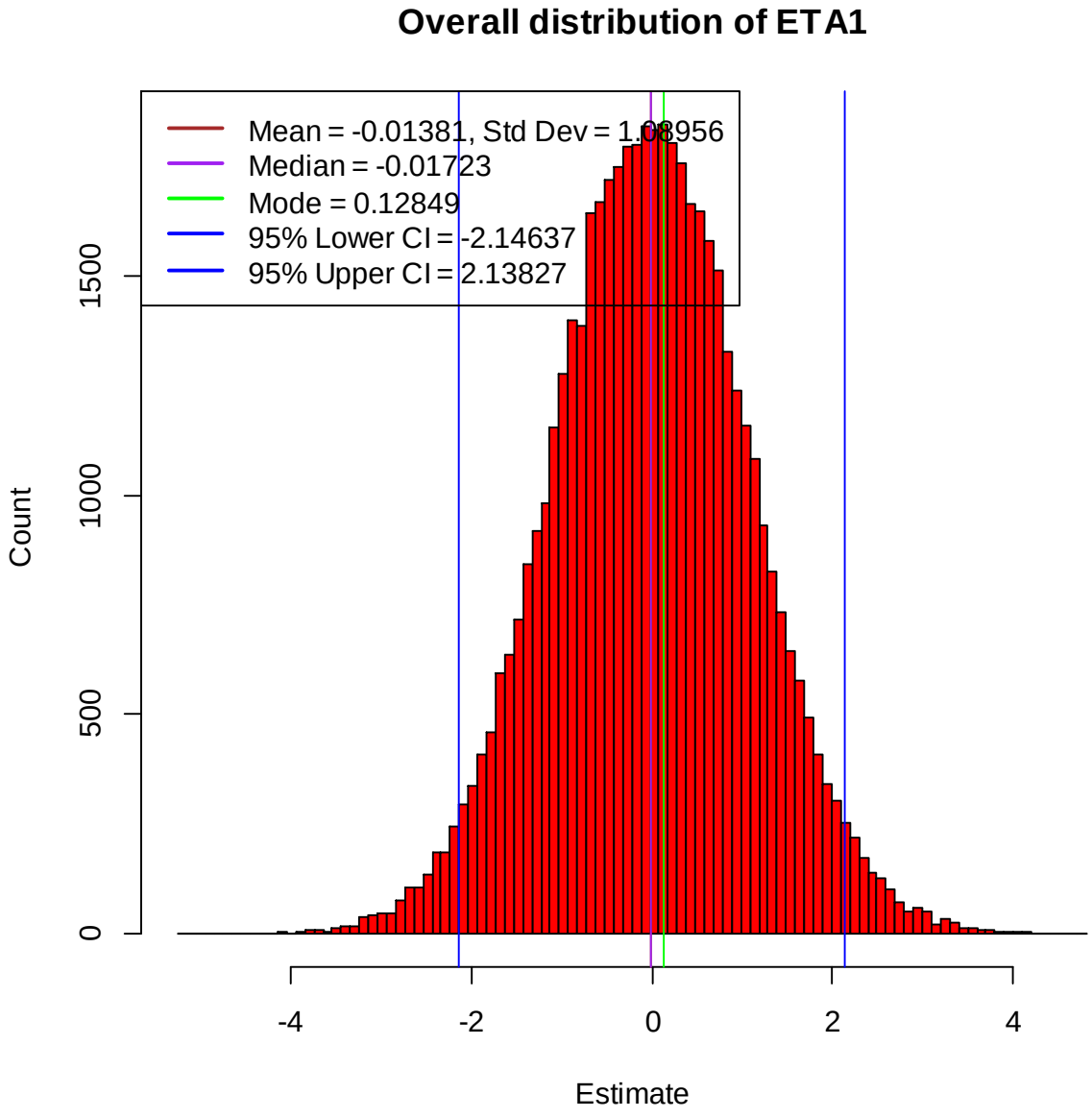

```
> mplus.plot.bayesian.autocorrelation('bcat.gh5',2)
> mplus.plot.bayesian.autocorrelation('bcat.gh5','parameter 2, eta1 by y3')
> mplus.plot.bayesian.autocorrelation('bcat.gh5','parameter 2, eta1 by y3',1)
```



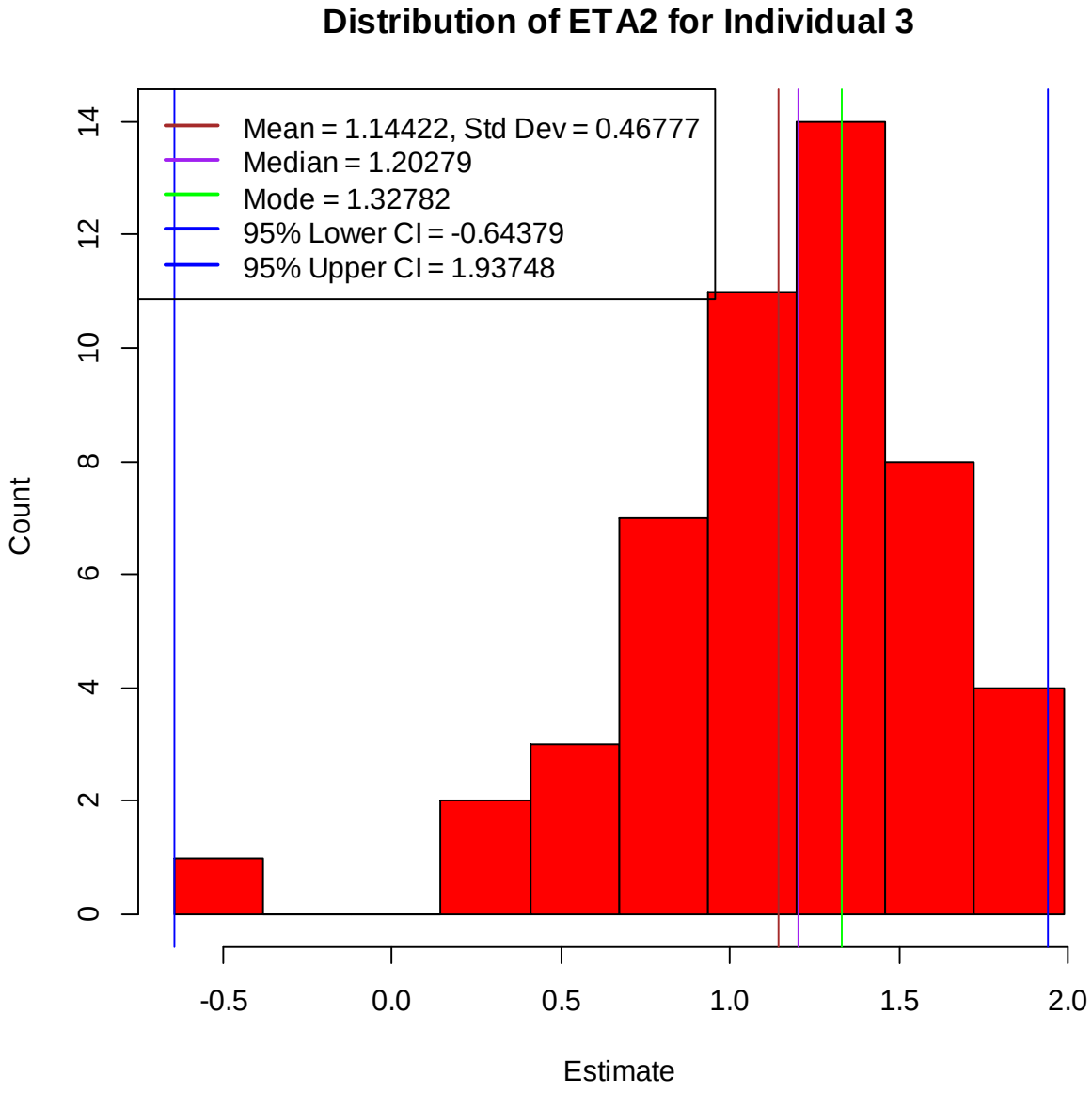
> mplus.plot.bayesian.autocorrelation('bcat.gh5',3,2)
> mplus.plot.bayesian.autocorrelation('bcat.gh5','parameter 3, eta1 by y4',2)



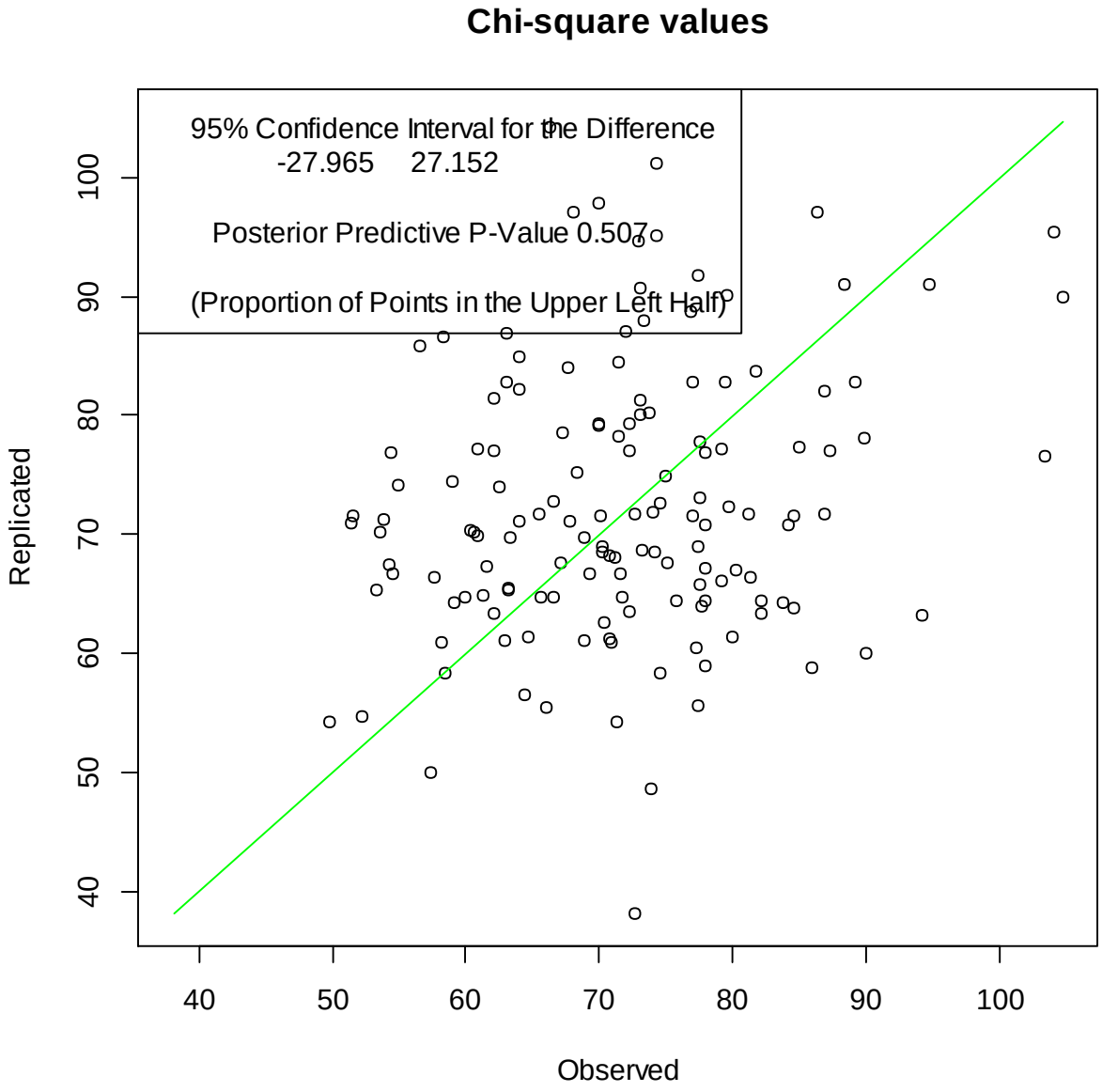
> mplus.plot.bayesian.plausible.distribution('bcat.gh5',1)
> mplus.plot.bayesian.plausible.distribution('bcat.gh5','eta1')



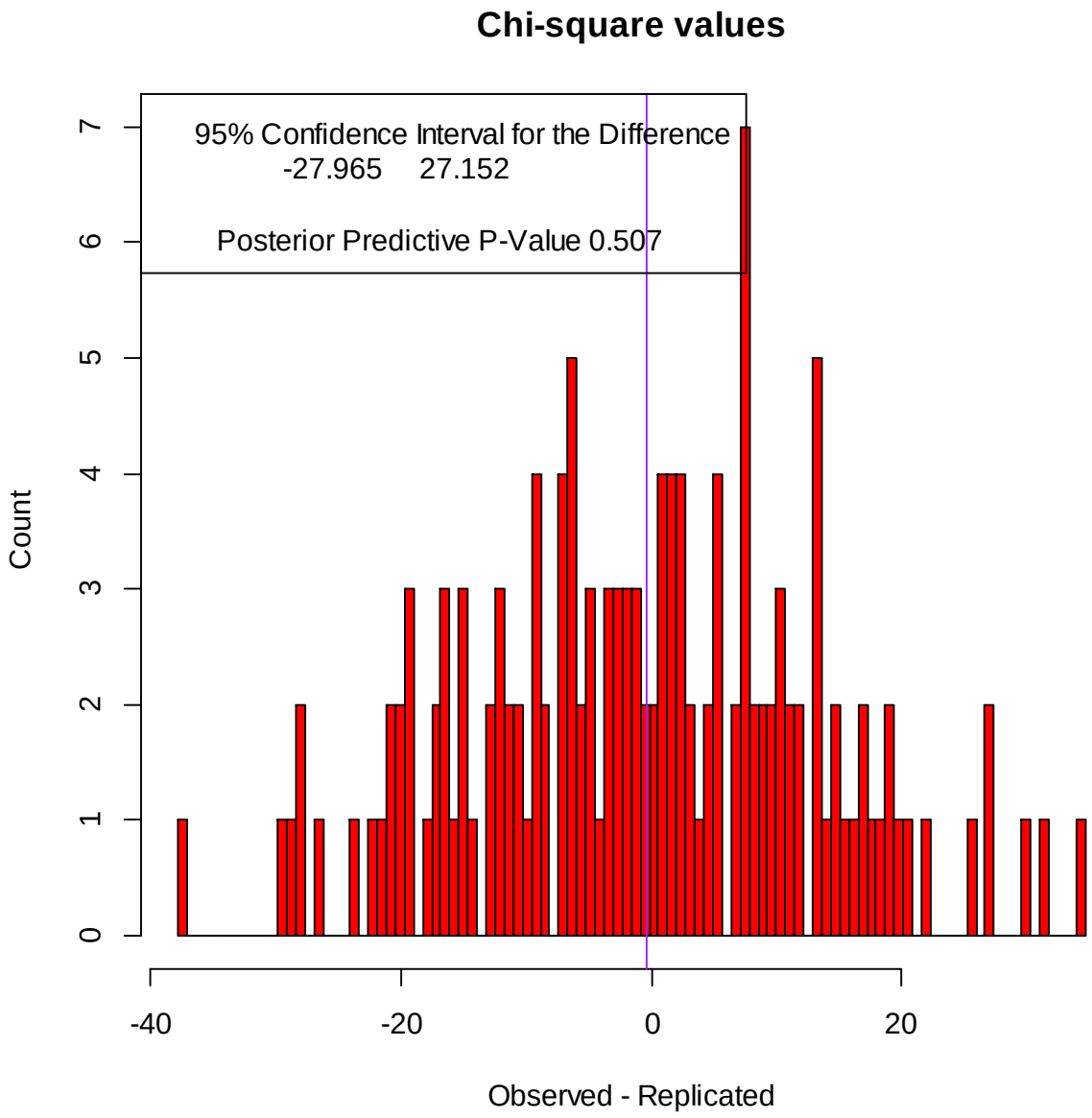

```
> mplus.plot.bayesian.plausible.distribution('bcat.gh5',2,obs=3)
> mplus.plot.bayesian.plausible.distribution('bcat.gh5','eta2',obs=3)
```




```
> mplus.plot.bayesian.predictive.scatterplot('bcat.gh5',1)
> mplus.plot.bayesian.predictive.scatterplot('bcat.gh5','chi-square values')
```




```
> mplus.plot.bayesian.predictive.distribution('bcat.gh5',1)
> mplus.plot.bayesian.predictive.distribution('bcat.gh5','chi-square values')
```



Bootstrap distributions

The following functions are used to view plots of bootstrap distributions.

- mplus.list.bootstrap.parameters(file)
- mplus.plot.bootstrap.distribution(file, parameter, bins, colest, colmed, colci, lloc, hcol)

The function mplus.list.bootstrap.parameters gives the list of parameter labels. The complete name of the parameter label or the index of the parameter label in this list can be given for the parameter argument in the various functions. The bins argument is the number of bins to use in plotting the bootstrap distribution. The bins argument is optional. The default is 100 bins.

By default, the plot of bootstrap distribution is shown with vertical lines marking the point estimate, the median of the distribution, and the lower and upper confidence interval. The default colors for the lines are green for the point estimate, purple for the median, and blue for the lower and upper confidence intervals. To change these colors, use the arguments colest (point estimate), colmed (median) and colci (confidence intervals). To remove any line from the plot, specify 'none' for its color argument. For example, specify colmed='none' to remove the line for the median.

When there is at least one line marker in the plot, a legend will show the color of the marker along with the value it is marking. The lloc argument is used to specify the placement of the legend. The lloc argument is a quoted string and must be one of the predefined values for the legend function in R. The default is lloc="right". The following are possible settings for the lloc argument:

- lloc="top": The legend is centered horizontally at the top of the plot area.
- lloc="bottom": The legend is centered horizontally at the bottom of the plot area.
- lloc="left": The legend is centered vertically and starts from the left edge of the plot area.
- lloc="right": The legend is centered vertically and starts from the right edge of the plot area.
- lloc="center": The legend is centered in the plot area.
- lloc="topleft"
- lloc="topright"
- lloc="bottomleft"
- lloc="bottomright"

Histogram bars are red by default. The hcol argument can be used to specify a different color for the histogram.

Various functions are available to extract the different plot data in the various Bayesian plots.

- mplus.get.bootstrap.distribution(file, parameter)
- mplus.get.bootstrap.point.estimate(file, parameter)

The parameter argument is the same as described above for the plot function.

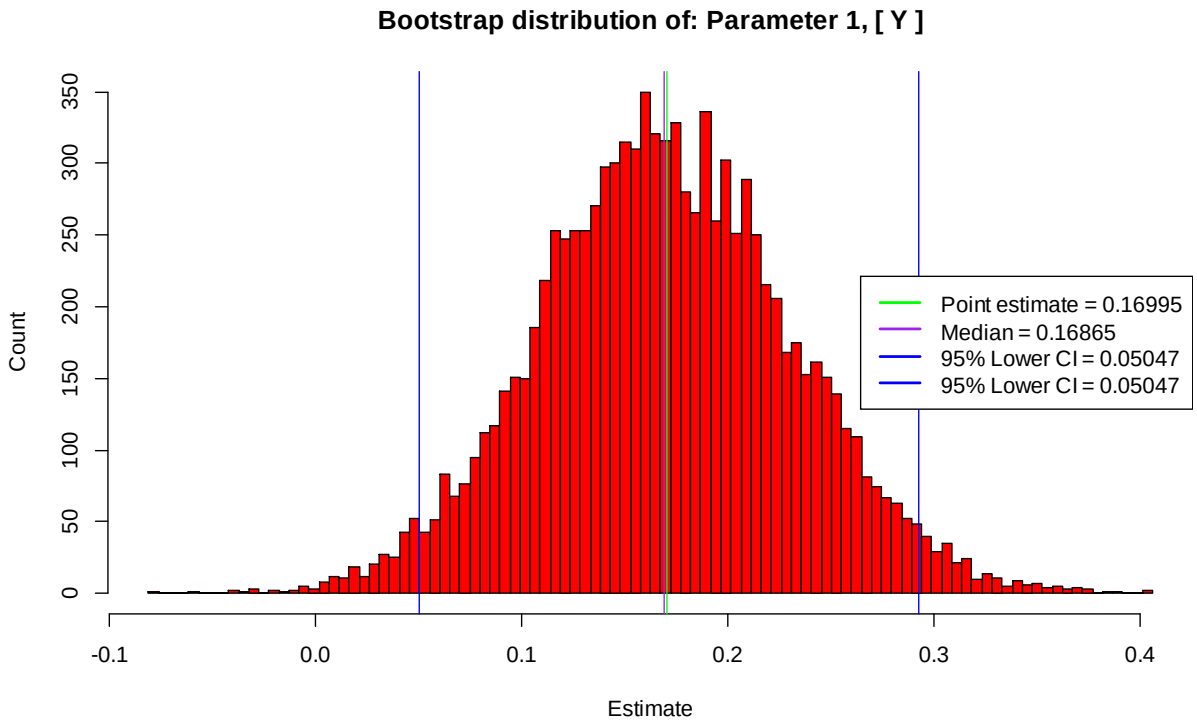
```
> mplus.list.bootstrap.parameters(FILE)
```

List of parameters to use in the following functions:
- mplus.plot.bootstrap.distribution
- mplus.get.bootstrap.distribution
- mplus.get.bootstrap.point.estimate

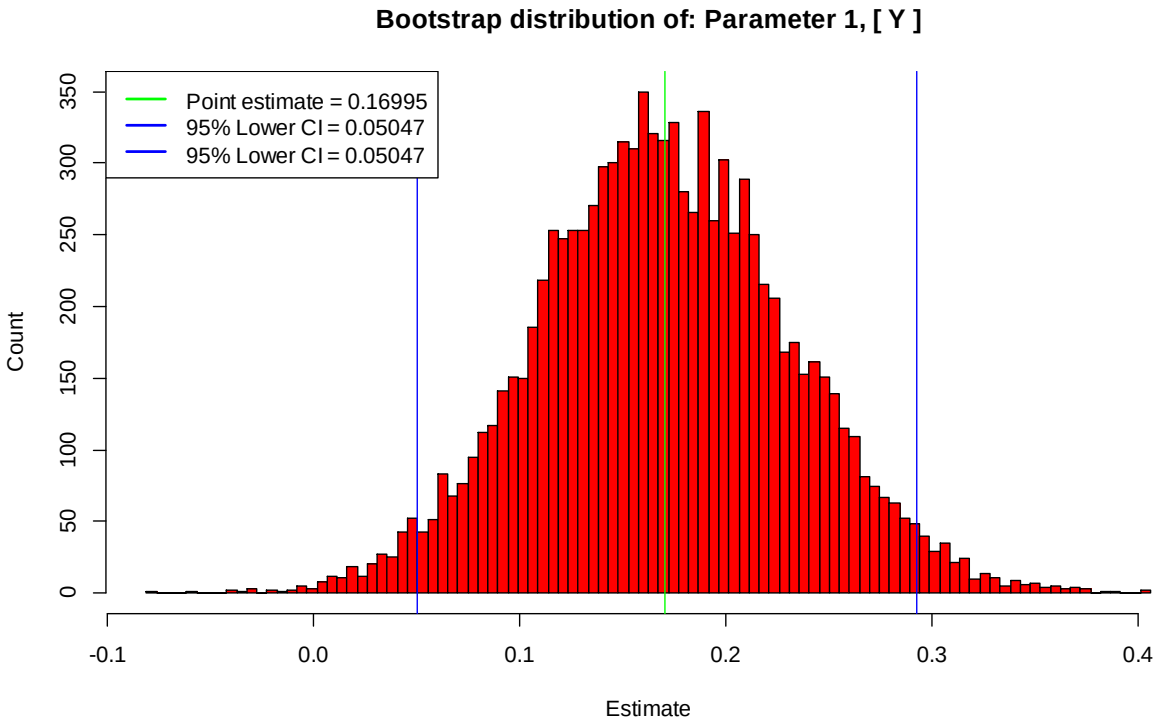
Parameters:
[1] Parameter 1, [Y]
[2] Parameter 2, [M] (equality/label)
[3] Parameter 3, Y ON M (equality/label)
[4] Parameter 4, Y ON X (equality/label)
[5] Parameter 5, Y ON Z (equality/label)
[6] Parameter 6, Y ON XZ (equality/label)

[7] Parameter 7, Y ON MZ (equality/label)
[8] Parameter 8, M ON X (equality/label)
[9] Parameter 9, M ON Z (equality/label)
[10] Parameter 10, M ON XZ (equality/label)
[11] Parameter 11, Y
[12] Parameter 12, M (equality/label)

```
> mplus.plot.bootstrap.distribution(FILE)
> mplus.plot.bootstrap.distribution(FILE,1)
> mplus.plot.bootstrap.distribution(FILE,'Parameter 1, [ Y ]')
```



```
> mplus.plot.bootstrap.distribution(FILE,1,colmed='none',lloc='topleft')
```



```
> mplus.plot.bootstrap.distribution(FILE,11,hcol='blue',colest='red',colmed='green',colci='none')
> mplus.plot.bootstrap.distribution(FILE,'parameter 11, y', hcol='blue', colest='red', colmed='green',
  colci='none')
```