European
Social
Survey

automtmm Documentation ESS JRA-3 deliverable 9

Release 1

Daniel Oberski

May 26, 2009

CONTENTS

CHAPTER

ONE

INTRODUCTION

This document describes `automtmm`, a suite of computer programs that automatizes parts of the process of analyzing MTMM experiments.

The program is developed in the context of the European Social Survey's Joint Research Activities 3 project (JRA-3), which aims to estimate and improve the quality of the questionnaire.

JRA-3 has two goals: control and improvement of the translation of the source questionnaire, and control and improvement of the survey questions themselves. `automtmm` has been developed to assist this second goal. It is described in the work description of the project as deliverable 9.

The quality of survey questions is defined in terms of their reliability and internal validity, and these quantities are estimated. Those estimates themselves then become the subject of a 'meta-analysis' predicting the quality of a question from many of its characteristics such as the number of categories, syllables, language, and so forth [1].

Obtaining the estimates for all countries, rounds, experiments and questions is not trivial. In the third round of the ESS alone there are 23 countries in which experiments were performed, four experiments, and twelve questions within each experiment. There are therefore 1104 validities and 1104 reliabilities to be estimated: a total of 2208 parameters of interest. For the subsequent meta-analysis additional technical data on these parameters is needed that, until the development of `automtmm`, was not provided by any existing computer program.

Because of the difficulties associated with preparing the data, estimating the parameters, and storing them together with additional information that is not given directly by existing computer programs, a new program was developed. Besides assisting with the JRA-3 group's analysis of the currently available data, this program also has the following advantages:

- Subsequent rounds can be quickly analyzed as new data become available;
- The quality prediction program SQP, used by the CCT in the evaluation process of the ESS rotating modules, will be improved based on the extra information obtained by `automtmm` that cannot be obtained in other ways;
- Other researchers that may wish to analyze the same experiments with different models can quickly obtain the results for all of their analyses.

Overall, therefore, this collection of programs greatly improves the computational efficiency of the process of quality control and improvement by MTMM experiments.

The document consists of two chapters. The first chapter explains the goals and the background of the program, as well as its workings in a manner meant for a broad audience. The second chapter, *Documentation*, documents the Python and R modules, classes, and functions that make up the program. It is intended as reference material for those who want to use or change the program.

[1] For a full overview of this approach and the characteristics used, please see [saris2007].

CHAPTER

TWO

GOALS AND BACKGROUND

This chapter describes the need for quality evaluation in the ESS and cross-national surveys in general, our approach to quality evaluation, complications arising from this approach, and consequently the motivation for the development of a new collection of computer programmes.

It also explains the functions of the different modules that make up `automtmm`. The explanation found in this chapter is meant for practitioners and methodologists who are not developers.

The following section briefly discusses different approaches to quality control of questionnaires, as well as the set-up of the experiments and our approach in analyzing them. The next section motivates the need for the development of `automtmm`. In the third section of this chapter the general structure and methodological background of the programme will be explained.

2.1 Evaluation of the quality of survey questions by MTMM experiments

2.1.1 The quality of survey questions

There are different ways to study the quality of survey questions. Most prominent among them are cognitive interviewing, behaviour coding, and psychometric assessment. Cognitive interviewing is a collection of techniques that are meant to reveal how respondents arrive at their answers [willis2005]. In behaviour coding, formal codes for the interaction between interviewer and respondent are developed and the entire interview is coded. This then reveals 'incorrect' behaviour on the part of interviewer or respondent [zouwen1998]. Finally, psychometric assessment is a statistical analysis of the answers to different questions that should be related according to a measurement model [lord1968].

The three approaches are complementary to each other; psychometric models alone can estimate the extent of random and systematic errors in the questions and are therefore indispensable. But they cannot reveal anything about the interpretation that is given to the questions by different respondents. Cognitive interviewing gives insight into these interpretations, and can provide suggestions for improvement in the concepts behind the questions. Behavior coding can identify more precisely what is going wrong in the survey interview that may be causing errors.

The ESS has recently performed a cross-national cognitive interviewing project [fitzgerald2009], which was successful in showing "...how respondents understood some of the questions and how they went about answering [them, and in] identifying and classifying the nature of any problems found..." (p. 12). In the ESS Dutch pilot study some interviews were recorded on audio and coded, revealing parts of the questionnaire that yield problematic interactions between interviewer and respondent [ongena2003].

The ESS has also performed many experiments that allow for the psychometric assessment of the reliability and internal validity of questions, and `automtmm` has been developed to aid in this procedure. Therefore the rest of this document discusses the need for psychometric assessment and how it is done in more detail.

2.1.2 Why estimate the quality?

The European Social Survey aims to allow governments, policy analysts, scholars and members of the public to interpret how people in different countries and at different times see themselves and the world around them.

The reliability and validity of the survey questions have several influences on this goal:

1. The lower the quality of a question, **the larger the amount of variation in that variable that has no interpretation**. This means that real differences between countries or across time will take larger sample sizes to detect than would otherwise be the case. Since larger sample sizes cost money, it follows that measurement error costs money.
2. Besides increasing the variance of variables users analyse, low quality questions will also have lower correlations with other variables. The consequence is that regression coefficients, cross-tables, and other measures of relationship will be biased. This bias can be upwards as well as downwards [fuller1986].
3. If the reliability and validity are different in different countries or times, measures of relationship will be biased differently. Thus, for measures of relationship **differences between countries and times arise that have no interpretation, or true differences are suppressed**. Previous studies have shown that there is considerable variation in quality across countries [oberski2004].

These arguments clarify both the need to create questions that are as good as possible and the need to estimate the extent of the errors so that corrections can be made afterwards.

After the estimation, the second need is immediately satisfied, because the estimates of the quality can be used directly to correct estimates of regression coefficients.

The improvement of questions is more complicated, because it involves identifying precisely which methods of asking a question are better or worse. For this purpose the programme SQP was developed, which suggests improvement of questions based on their characteristics [sqp2007]. It also provides estimates of the quality of the question so that survey estimates can be corrected.

This programme, however, currently has limited use for cross-national comparisons in the ESS, because it does not provide estimates for all participating countries and languages. Moreover, with the great wealth of information available from the ESS experiments, it could be greatly improved.

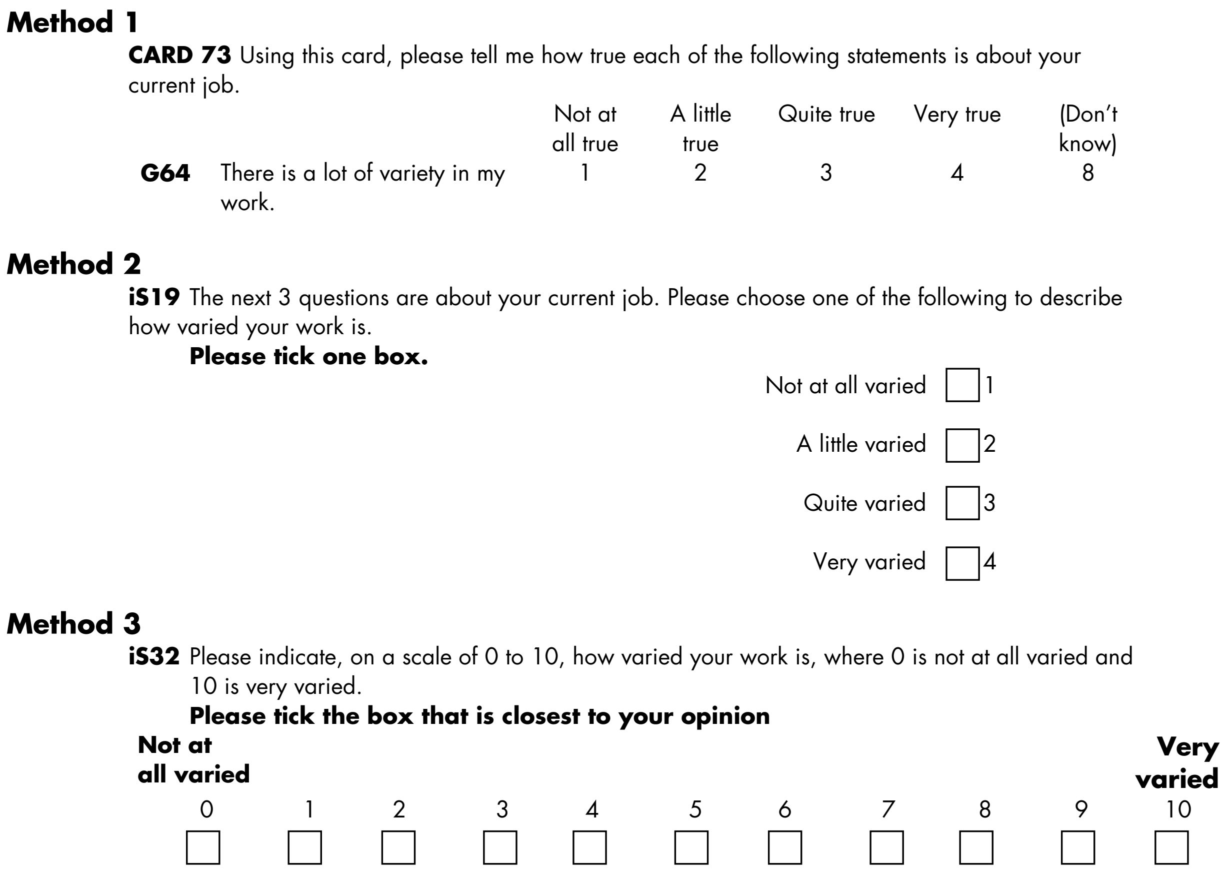
Therefore, the multitrait-multimethod experiments conducted in the European Social Survey:

- Contribute to the improvement of the ESS questionnaire, yielding higher efficiency and a greater power for governments, policy analysts, scholars and members of the public to detect important phenomena;
- Contribute to the improvement of survey questions in general;
- Contribute academic knowledge about the way people answer different kinds of questions in different cultures and languages;
- Allow for the correction of the effect measurement errors have on estimates and the cross-national comparison of those estimates.

The following section briefly explains the rationale and implementation of these experiments.

2.1.3 A short explanation of MTMM

In every round of the European Social Survey experiments are built in to evaluate the quality of the questions by statistical estimation. Each of these experiments consist of repeating at least three different questions in at least three different ways (see figure *A trait measured by three different methods from round two of the ESS.*). This way one can separate how much of the variance of each ESS question in the experiment is due to the 'trait' to which the question



Method 1

CARD 73 Using this card, please tell me how true each of the following statements is about your current job.

		Not at all true	A little true	Quite true	Very true	(Don't know)
G64	There is a lot of variety in my work.	1	2	3	4	8

Method 2

iS19 The next 3 questions are about your current job. Please choose one of the following to describe how varied your work is.

Please tick one box.

Not at all varied	☐ 1
A little varied	☐ 2
Quite varied	☐ 3
Very varied	☐ 4

Method 3

iS32 Please indicate, on a scale of 0 to 10, how varied your work is, where 0 is not at all varied and 10 is very varied.

Please tick the box that is closest to your opinion

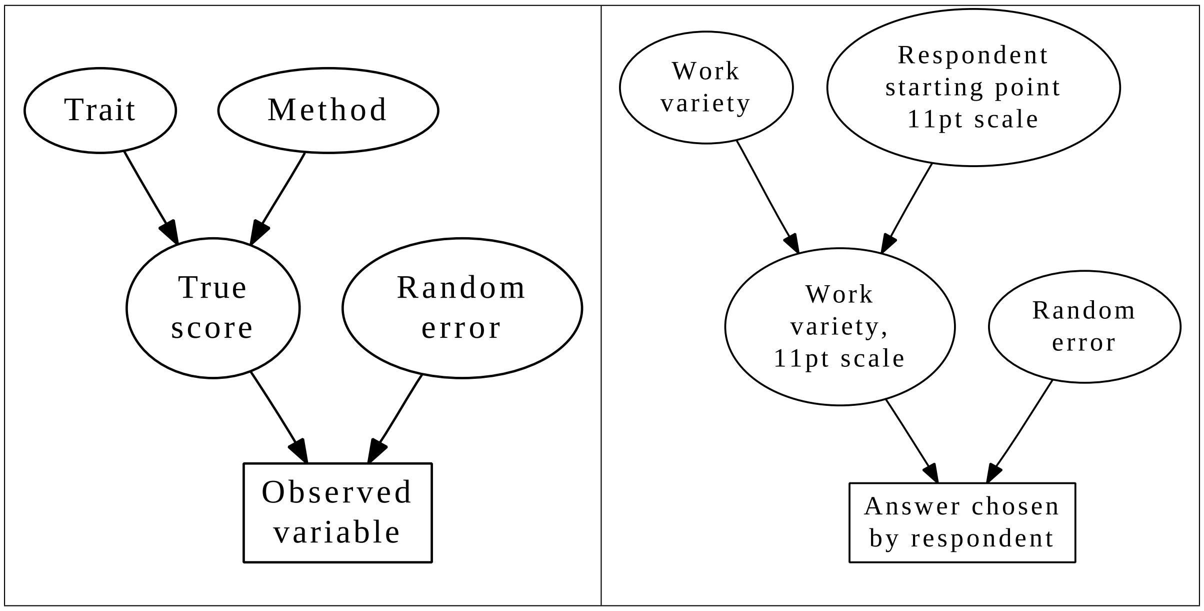
Not at all varied										**Very varied**
0	1	2	3	4	5	6	7	8	9	10
☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Figure 2.1: A trait measured by three different methods from round two of the ESS.

is supposed to refer, and how much is due to the 'method' of asking it. Therefore the type of experiment is also called 'multitrait-multimethod' [campbell1959].

Our basic response model is shown on the left-hand side of the figure below. It can be seen that the observed variable is a combination of random measurement error and a so-called 'true score'. This score represents the opinion of the respondent expressed in the particular method of asking the question, free of random errors. But the true score is not completely free of error: it is itself a combination of *systematic* measurement errors in the form of a method factor on the one hand, and the true opinion or 'trait' on the other. The method factor represents the fact that different respondents arrive at an answer to questions asked by this specific method in their own way that is stable across questions.

The right-hand side of the figure shows this basic response model applied to the variety of work example. In this picture, one would have liked to have measured 'work variety', but obtained only the observed scores. Although true 'work variety' can never be observed, it can be estimated how strong the relationship between this variable and the observed answers is.



The path diagrams in the figure can also be expressed in equations as

$$y = \tau + \epsilon$$
$$\tau = m.M + v.T$$

where y is the observed variable, τ the true score, ϵ the random error component, M the method factor, and T the trait, and m and v are scaling coefficients. This model implies that the deviation scores of the observed variable are unbiased for the true score, and that there are no unique components in the true score besides method and trait variance.

It can then be shown that the proportion of random error in the observed variable equals *var(e)/var(y)*, which we call the 'reliability' of the question. The proportion of variance in the true score due to the trait we want to measure is called the internal 'validity' and equals $v.[var(T)/var(\tau)]$.

The model shown is clearly not identified as there is only one observed variable and four unobserved variables. However, in the ESS experiments there are three traits and three or four methods, yielding nine or twelve observed variables. Then the model is identified under the following assumption:

$$\mathrm{Cov}(\mathrm{T_i}, \mathrm{M_j}) = \mathrm{Cov}(\mathrm{T_i}, \epsilon_j) = \mathrm{Cov}(\mathrm{M_j}, \epsilon_i) = \mathrm{Cov}(\epsilon_i, \epsilon_j) = 0, \forall\, i \neq j;$$

that is, there are no correlations between the traits and methods, among the methods, among the random errors, or between either traits or methods and random errors.

Furthermore in the first instance the following assumptions are also made but relaxed when necessary:

1. Cov(ϵ_{i}, ϵ_{j}) = 0 (uncorrelated error terms);
2. m_{ij}= *1* for all *i* and *j*. (equal scaling coefficients for the method factors)

The goal of the MTMM experiments is to estimate how much random and systematic error each question contains. That is, it is to estimate the validity and reliability of the questions. These two quantities can be expressed as the standardised coefficients of a structural equation model (SEM) [saris1991]. We then label these coefficients the *validity coefficient* and the *reliability coefficient*. Their product, the *quality coefficient*, is the square root of the proportion of variation in the observed variable that is explained by the trait we want to measure.

For instance, in the work variety example above, if the quality coefficient for an 11 point scale in Norway were to equal *0.8*, then $0.8^2(100) = 64\%$ of the variation in the observed answers to the question “Please indicate, on a scale of 0 to 10, how varied your work is...” is due to the respondents’ perception of variety in their work, while 36% is random and systematic measurement error. In round 3 of the ESS, the median quality coefficient found was *0.7*.

2.1.4 Complications and the solution: split ballot MTMM

The model shown above has the advantage of being able to evaluate the reliability and validity of any survey question, and of being able to identify which methods of asking it provide the highest quality.

However, it also entails asking the same question of the same respondent three or even four times. This increases the burden on the respondent quite a lot, as well as increasing the likelihood that the respondents remember their previous answer (memory effects). Therefore in the ESS a different design has been implemented: the split ballot MTMM design [saris2004].

In the split ballot MTMM design, each respondent is asked the question once in the format of the main questionnaire, and again one more time at the end of the interview in a different format. The table below shows the resulting design applied to the variety of work example shown earlier.

Method	Group 1	Group 2
4pt horizontal	×	×
4pt vertical	×	
11pt horizontal		×

The table shows that the combination of “4pt vertical” and “11pt horizontal” is never observed for the same person. However, if the groups are assigned completely at random the MTMM model formulated above is still identified even given this planned missing data [1].

The split-ballot MTMM design makes the MTMM setup feasible in practice. However, it also complicates the analysis. Instead of a single group analysis, a multiple group analysis has to be run, where care is taken that the correct order of variables is maintained and the correct across-group constraints that lead to identification of the model are made.

2.2 The need for a new programme

Why develop a new collection of modules if software for structural equation modelling is already available?

In each round of the ESS, there are approximately 20-26 countries. For each country, four or five experiments were conducted, each of which contains two or three split-ballot groups with each nine or twelve variables (depending on the round). For each of these variables, there are several parameters, from which two parameters of interest need to be computed. Furthermore, there are several rounds, with a new round available every two years.

For each of the experiments in each round and country:

[1] The crucial assumption here is that of no correlation between the method factors for these two methods.

1. The data for each split-ballot group need to be extracted and prepared for analysis;
2. The model needs to be programmed, estimated, and adjusted to fit the data (currently LISREL and its syntax is used);
3. The output (unstandardised parameter estimates and their variance-covariance matrix) needs to be written to files that can be read by another programme;
4. The standardised estimates need to be computed from the unstandardised ones. For each question a reliability and a validity coefficient is computed;
5. The variance-covariance matrix of these reliability and validity coefficients is needed for the further meta-analysis.

Finally, the reliability and validity coefficients, along with their variance-covariance matrices and information about the round, country, experiment, and question, need to be collected into a database to be used for the meta-analysis.

These steps are laborious to do by hand. Although the process of model formulation and adjustment cannot currently be automated, the other steps can be, and this is why a new computer programme was needed. The programme saves time, and also allows one to re-run the entire analysis if a new round of data becomes available, or if new insights require that the models need to be adjusted.

2.3 Functions of `automtmm`

`automtmm` is not a single programme, but a suite of independently operating modules.

Each of these modules performs or facilitates one of the five steps listed in the previous section.

Data preprocessing

Preprocessing of the ESS data is done by a collection of `R` scripts that read in the official ESS main and supplementary questions datafile, merge the two, recode missing values, split the sample on country, experiment and split ballot groups, and write the result to a directory tree with different directories for countries, and experiments. The output is different files containing the covariances and means for each split-ballot group in each such directory.

Currently these `R` scripts are not sufficiently general to allow running them on new data sets without some adjustments. However, they do allow for re-running of the first three ESS rounds and provide an easily adjusted template for other datasets.

Model formulation and adjustment

The basic starting model is the same for each analysis, but sometimes adjustments are needed. It may be the case that the method scaling factors are not equal for the different topics, for instance. This can happen when one of the questions was negatively formulated if respondents' answering strategies are not exactly symmetrical. An example of this was found in the experiment on opinion about the position of women in round two of the ESS. Adjustment may also be needed if some random errors terms turn out to be correlated.

In order to detect which parts of the model may be misspecified, the procedure suggested by Saris, Satorra and van der Veld is used as implemented by the program JRule for LISREL [sarisfrth]. This may yield different versions of the same model file, as the model is adapted. The versions may even branch into non-nested models. For this reason a module is in development that uses the version control system `git` [git] to keep track of different versions of the model and their outcomes in terms of the reliability and validity estimates. We aim to add this module to `automtmm` before September 2009.

Extracting results and calculating necessary quantities

LISREL provides as output the estimates of the model parameters as well as the standardized reliability and validity coefficients. It also provides standard errors and a full variance-covariance matrix for the free (unstandardised) parameters.

However, what LISREL does not provide is standard errors and/or a variance-covariance matrix for the standardized estimates (the SEM program Mplus does provide standard errors, but not the full variance matrix).

This matrix is, however, needed for the subsequent meta-analysis. If it is not available, sampling variation in the reliability and validity estimates and dependence among estimates obtained from the same experiment cannot be taken into account. This results in predictions that have too-narrow intervals and in severe cases even in bias. In order to improve the predictions made by the program SQP, therefore, this information is needed.

There are different ways of obtaining the variance of standardized estimates. We have chosen an analytic approximation to the variance matrix by applying the delta method [oehlert1992]. This is the same approach as used by Mplus to obtain standard errors for standardized coefficients [2].

The validity and reliability coefficients can be viewed as a function *g(θ)* of the free parameters of the model across all split-ballot groups **m**, **v**, Var(**T**), Var(**M**), and Var(ϵ).

For example, the reliability coefficient of the question measured with trait 1 and method 1 equals:

$$g_{v1}(\theta) = [m^2 \text{ var}(M_1) + v_1{}^2 \text{ var}(T_1)]^{1/2}[\text{var}(\epsilon_1) + (m^2 \text{ var}(M_1) + v_1{}^2 \text{ var}(T_1))]^{-1/2}.$$

Since for each of the coefficients such a formula exists, the function *g(θ)* is vector-valued.

If all free parameters of the multi-group model are collected in a vector θ, while the validity and reliability coefficients are collected in a vector **q**, so that **q** = *g(θ)*, the delta method states that their variance covariance matrix will be:

$$\text{Var}(\mathbf{q}) = \nabla g(\theta)' \text{ Var}(\theta) \nabla g(\theta),$$

where ∇ denotes the gradient of the function with respect to the free parameters.

Moreover, if θ converges in distribution to a normal distribution with variance Var(θ), **q** will converge in distribution to a normal distribution with Var(**q**):

$$\sqrt{n}\,(q - Q) \rightarrow N(0, \text{Var}(\mathbf{q})) \text{ iff } \sqrt{n}\,(\theta - \Theta) \rightarrow N(0, \text{Var}(\theta)).$$

Thus, the sampling distribution of the validity and reliability coefficients can be obtained if the gradient of the vector-valued function *g(θ)* with respect to the vector θ is calculated, together with the variance-covariance matrix of θ.

`automtmm` contains a module which accomplishes these tasks. The first task, obtaining the derivatives, is accomplished by reading in algebraic formulas for the derivatives from a file. This file is generated by a batch script for the Maxima computer algebra system [maxima2009]. The variance-covariance matrix is obtained from a LISREL output file. It can therefore be computed in any way that is supported by LISREL, such as from the observed or expected information matrix, or applying the Satorra correction for non-normality [satorra1992]. The information about which parameter number in this file refers to which parameter in the model is read directly from the 'parameter specification' part of the LISREL output.

The module then reads in the parameter estimates, and, based on these estimates, calculates the validity and reliability coefficients, as well as the matrix of derivatives. It automatically selects the parameters and coefficients from the correct split-ballot group [3]. The formula for Var(**q**) is then applied.

[2] See <http://www.statmodel.com/download/StandardizedCoefficients.pdf>.

[3] The validity and reliability coefficients for the first method are available in all split-ballot groups. In this case the estimates for the last group are chosen.

Gathering estimates and constructing a database

The module `walk_and_run` applies the above procedures to a directory tree of files, usually one created by the ESS data preprocessing module. This directory tree contains information about the country and experiment, as well as the input files for the analyses.

The analyses are run one by one and the module that calculates validity and reliability coefficients and their variance matrix is applied. This variance matrix is written to a separate text file in the same directory, along with a file containing the names of the coefficients. These files are readable by other programs such as `R`.

The module also stores the coefficient estimates along with the information found in the directory structure in a MySQL database table. The variable numbers are stored as well, so that the estimates can later be linked to a database of questions. By default the database name is ‘automtmm’ and the table name ‘estimates’.

CHAPTER

THREE

DOCUMENTATION

3.1 Module that preprocesses the ESS data files for MTMM analysis

This module consists of various `R` scripts that perform the following tasks:

- Read in the ESS SPSS data sets for the main and supplementary questionnaires, select variables, change some labels, and merge the files
- Given the merged dataset, write covariance matrices and means of the experimental variables to files for each country, experiment, and split-ballot group. This results is <number of countries> directories, each containing <number of experiments> subdirectories, each containing <number of split-ballot groups> files with the covariances and the same number of files for the means.

3.2 LISREL input and output processing

3.2.1 Module that reads and interprets LISREL input and output files

The `parse_lisrel` module contains classes and functions to deal with LISREL input and output files.

The `LisrelInput` class provides functions for the following tasks:

- Modifying a LISREL input file such that it will write the (unstandardized) estimates and variance-covariance matrix of the free parameters to a temporary directory when run;
- Running LISREL on an input file;
- Getting the form (symmetric, full, diagonal, ...), dimensions (number of rows and columns), and contents (actual estimates) of the different matrices in each group;
- Reading in the variance-covariance matrix of the free parameters, and figuring out from the LISREL output what name (e.g. 'GA 1 1') belongs to each numbered free parameter in that matrix;
- Calculating the completely standardized coefficients from the unstandardized estimates;
- Calculating the variance-covariance matrix of the _standardized_ coefficients from the unstandardized estimates and their variance- covariance matrix via the Delta method.

It also provides some utility functions such as retrieving the number of groups, making a lower diagonal matrix symmetric, etc.

class `LisrelInput` (*path='', text=''*)

Functions to get information about number of groups and matrix form from a LISREL self.input_text file.

`get_derivs`(*groupnum=2*)
Calculate the variance-covariance matrix of the standardized estimates

`get_dimensions`()
Retrieve number of x, y, eta, and ksi variables for each group. Returns a <ngroups> list of dictionaries.

`get_free_params`(*outpath=''*)
Retrieve the free parameters of the model from the output path.

`get_matrices`(*path=''*)
Read matrices from output files <MAT>.out, taking into account the form of the matrix. Returns dict of <ngroups> list of NumPy.matrices read from the files.

`get_matrix_forms`()
Retrieve 'form' (full, symmetrix, zero, diagonal, ... and free or fixed) Of matrices for LISREL model for each group. Returns an <ngroups> list of dictionaries.

`get_matrix_shape`(*matname, groupnum*)
Returns the shape of a given LISREL matrix <matname> in group <groupnum>.

`get_modified_input`()
Modifies input to write matrix results to files, and converts relative paths in LISREL input to absolute paths. Returns string.

`get_ngroups`()
Retrieve number of groups. Fails if file contains stacked analyses.

`get_var_standardized`(*path='temp', select=(('ly 4 4', 'ly 5 5', 'ly 6 6', 'ga 4 1', 'ga 5 2', 'ga 6 3'), ('ly 7 7', 'ly 8 8', 'ly 9 9', 'ga 7 1', 'ga 8 2', 'ga 9 3'), ('ly 10 10', 'ly 11 11', 'ly 12 12', 'ly 1 1', 'ly 2 2', 'ly 3 3', 'ga 10 1', 'ga 11 2', 'ga 12 3', 'ga 1 1', 'ga 2 2', 'ga 3 3'))*)
Calculate the analytical variance-covariance matrix of the standardized estimates via the Delta method.

The delta method obtains the variance matrix of a (vector-valued) function of (a vector of) random variables θ by the formula

D[f(θ)]' V(θ) D[f(θ)],

where D[.] is the gradient and V(.) the variance matrix. In our case the function f(.) is the calculation of standardized coefficients from unstandardized coefficients (e.g. Bollen 1989:350).

Therefore the derivative (gradient) matrix of this calculation with respect to the unstandardized parameters is obtained from Maxima via the `read_maxima` module.

select a tuple of tuples that determines for each group which standardized coefficients' derivatives are to be selected from the derivative matrix. By default the coefficients for the first MTMM method are taken from the last group.

Returns a tuple of the names of the selected standardized coefficients, and a symmetric matrix of variances and covariances between the various standardized estimates in the same order.

`get_varcov_params`(*matname='EC', path='temp'*)
read in the EC (Vcov matrix of the parameters).

`lisrel_science_to_other`(*string*)
Finds numbers of the form 0.12D-04 and converts them to a list of Python floats.

LISREL Scientific notation is different from the regular one. Normally an e or E is used to mean 'multiplied by 10 to the power of...'. LISREL uses a D, confusing Python and R.

`run_lisrel`(*temp_path=''*)
Run LISREL executable on the input file. The output is written to temp_path. Raises an exception if the exit code of the LISREL executable is not 0.

`standardize_matrices`(*path='temp'*)
Returns the same kind of list as get_matrices, but standardizes some of them. See Bollen 1989:350-1.

`write_to_file`(*new_text, path=''*)
Write the input text to a file. Overwrites the original (making a backup) if no argument is given.

nrows_symm(*n*)
Calculate number of rows a symmetric matrix must have had if len(vech(matrix)) equals n

nullify_diagonal(*mat*)
Replaces negative numbers on the diagonal by 0.0.

Sometimes variances of latent variables that are locally unidentified have been left free. These variances can become negative, making it impossible to take the sqrt(). One solution provided here is to set these variances to zero.

symmetrize_matrix(*mat*)
mat is a NumPy.matrix or .array. Copies the lower diagonal elements to the upper diagonal elements.

class LisrelInput(*path=", text="*)
Functions to get information about number of groups and matrix form from a LISREL self.input_text file.

get_derivs(*groupnum=2*)
Calculate the variance-covariance matrix of the standardized estimates

get_dimensions()
Retrieve number of x, y, eta, and ksi variables for each group. Returns a <ngroups> list of dictionaries.

get_free_params(*outpath="*)
Retrieve the free parameters of the model from the output path.

get_matrices(*path="*)
Read matrices from output files <MAT>.out, taking into account the form of the matrix. Returns dict of <ngroups> list of NumPy.matrices read from the files.

get_matrix_forms()
Retrieve 'form' (full, symmetrix, zero, diagonal, ... and free or fixed) Of matrices for LISREL model for each group. Returns an <ngroups> list of dictionaries.

get_matrix_shape(*matname, groupnum*)
Returns the shape of a given LISREL matrix <matname> in group <groupnum>.

get_modified_input()
Modifies input to write matrix results to files, and converts relative paths in LISREL input to absolute paths. Returns string.

get_ngroups()
Retrieve number of groups. Fails if file contains stacked analyses.

get_var_standardized(*path='temp', select=(('ly 4 4', 'ly 5 5', 'ly 6 6', 'ga 4 1', 'ga 5 2', 'ga 6 3'), ('ly 7 7', 'ly 8 8', 'ly 9 9', 'ga 7 1', 'ga 8 2', 'ga 9 3'), ('ly 10 10', 'ly 11 11', 'ly 12 12', 'ly 1 1', 'ly 2 2', 'ly 3 3', 'ga 10 1', 'ga 11 2', 'ga 12 3', 'ga 1 1', 'ga 2 2', 'ga 3 3'))*)
Calculate the analytical variance-covariance matrix of the standardized estimates via the Delta method.

The delta method obtains the variance matrix of a (vector-valued) function of (a vector of) random variables θ by the formula

D[f(θ)]' V(θ) D[f(θ)],

where D[.] is the gradient and V(.) the variance matrix. In our case the function f(.) is the calculation of standardized coefficients from unstandardized coefficients (e.g. Bollen 1989:350).

Therefore the derivative (gradient) matrix of this calculation with respect to the unstandardized parameters is obtained from Maxima via the `read_maxima` module.

select a tuple of tuples that determines for each group which standardized coefficients' derivatives are to be selected from the derivative matrix. By default the coefficients for the first MTMM method are taken from the last group.

Returns a tuple of the names of the selected standardized coefficients, and a symmetric matrix of variances and covariances between the various standardized estimates in the same order.

get_varcov_params(*matname='EC', path='temp'*)
read in the EC (Vcov matrix of the parameters).

lisrel_science_to_other (*string*)
Finds numbers of the form 0.12D-04 and converts them to a list of Python floats.

LISREL Scientific notation is different from the regular one. Normally an e or E is used to mean 'multiplied by 10 to the power of...'. LISREL uses a D, confusing Python and R.

run_lisrel (*temp_path="*)
Run LISREL executable on the input file. The output is written to temp_path. Raises an exception if the exit code of the LISREL executable is not 0.

standardize_matrices (*path='temp'*)
Returns the same kind of list as get_matrices, but standardizes some of them. See Bollen 1989:350-1.

write_to_file (*new_text, path="*)
Write the input text to a file. Overwrites the original (making a backup) if no argument is given.

3.2.2 Module that calculates analytical derivatives of the standardized coefficients with respect to the free parameters of the model

This module consists of two parts: one written in the Maxima Computer Algebra System language <http://maxima.sourceforge.net/>, and the other a Python script to read in and translate the results obtained from Maxima to Python.

In the Maxima script, the standardized gamma and lambda coefficients are first defined in terms of the unstandardized parameters. These parameters *might*, but need not be, free parameters of the model for a given MTMM analysis.

The analytical derivatives of the standardized coefficients are then taken for each parameter. This yields a (number of parameters) x (number coefficients) size matrix of analytical expressions in terms of the parameters.

This matrix is read in by the Python script and used by the `parse_lisrel` module to calculate the variance-covariance matrix of the standardized coefficients.

See also the `get_var_standardized` function in `parse_lisrel`.

get_derivs (*maxpath='derivmatrix.txt'*)
Use pythonize_maxima() to obtain a list of lists of strings to be evaluated. Returns a list 'derivs'.

Each list in derivs refers to the first derivatives wrt to one free parameter. (the order is in paramdict)

Each string in each of these lists refers to one standardized coefficient. (the order is in scoefdict and also in the handy-to-read scoefs.names file which can be read by R scan())

The string can be evaluated in an environment where the relevant matrices are present as NumPy.matrices or arrays. Also 'from math import sqrt' is needed.

pythonize_maxima (*maxpath='derivmatrix.txt'*)
Read in Maxima file which contains the analytic expressions for the first derivatives of the standardized estimates w.r.t. the parameters and convert it into a Python expression that can be evaluated. Returns string that will evaluate to a list of strings to be evaluated.

3.3 Module that runs the LISREL input files and gathers the results

The `walk_and_run` module works by following these steps:

1. Walk a directory tree, looking for LISREL input files;
2. If an input file is encountered:
    (a) Modify the input so that the needed estimates will be
    written to files;

(a) Run this modified input using the LISREL executable

(application);

(a) Retrieve the standardized estimates and perform an 'action'

on them. Currently available 'actions' are:

- Selecting the MTMM validity and reliability coefficients from the appropriate group, inserting them into a MySQL database table, retrieving their variance-covariance matrix and writing them to a file (this matrix is needed in order to perform meta-analyses);
- Just printing the standardized estimates.

Most of the above procedures use the module `parse_lisrel`.

`default_action`(*matrix, **kwargs*)
By default the standardized matrices encountered by walk_and_run are just printed

`entry_exists`(*input_path, experiment, country, study, varnum*)
Checks whether the entry with the given characteristics is already in the database.

`retrieve_mtmm`(*matrix, **kwargs*)
An 'action' that collects the reliabilities, validities, and method effects and writes them to one data file.

`save_estimates`(*input_path, experiment, country, study, val, rel, met, varnum, update_or_insert='insert'*)
Writes one row of estimates data to the database. Returns True if all is well, False if an exception occurs (exception is not thrown directly to ensure that the database connection is closed).

`solution_obtained`(*outpath*)
Checks whether the output file contains errors or non-convergence messages.

`walk_and_run`(*top_dir, tempdir='', action=<function default_action at 0x985cb1c>*)
recurse through directory structure, looking for .LS8 files. Each .LS8 file is run.

`write_tuple_to_file`(*tup, path, sep='r'*)
takes a (multidimensional) tuple and writes it to a file at path

CHAPTER

FOUR

BIBLIOGRAPHY

BIBLIOGRAPHY

[campbell1959] Campbell, D. T. and Fiske, D. W. (1959). Convergent and discriminant validation by the multitrait-multimethod matrix. *Psychological bulletin*, 56 (2).

[fitzgerald2009] Fitzgerald R., S. Widdop, M. Gray, and D. Collins (2009). "Testing for equivalence using cross-national cognitive interviewing". *Centre for Comparative Social Surveys Working Paper Series*, paper no. 01.

[fuller1986] Fuller, W. A. (1986). *Measurement error models*. New York: Wiley.

[git] Torvalds, L., J. C. Hamano and others (2009). *''git'', the fast version control system*. <http://git-scm.com/>.

[lord1968] Lord, F. M. and Novick, M. R. (1968). *Statistical theories of mental test scores*. Reading MA: Addison-Welsley Publishing Company.

[maxima2009] The Maxima Group (2009). *Maxima, a Computer Algebra System*. Version 5.18.1 <http://maxima.sourceforge.net/>

[oehlert1992] Oehlert, G. W. (1992). A Note on the Delta Method. *The American Statistician*, 46 (1), pp. 27-29.

[oliphant2006] Oliphant, T. E. (2006). *Guide to NumPy*. <http://numpy.scipy.org/>.

[oberski2004] Oberski, D. L., W. E. Saris, and J. A. P. Hagenaars (2004). "Why are there Differences in Measurement Quality across Countries?" in: *Measuring meaningful data in social research*, G. Loosveldt, M. Swyngedouw, and B. Cambré (eds.). Leuven: Acco.

[ongena2003] Ongena, Y. (2003). *Pre-testing the ESS-questionnaire using interaction analysis*. <http://europeansocialsurvey.org/?option=com_docman&task=doc_download&gid=181&itemid=80>

[R2005] R Development Core Team (2005). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing, <http://www.R-project.org>.

[rossum2009] Rossum, G. van, and F. L. Drake Jr. (ed.) (2009). *The Python Language Reference*. Hampton, NH: Python Software Foundation.

[saris1991] Saris, W. E. and F. Andrews (1991). "Evaluation of measurement instruments using a structural modeling approach". In: *Measurement errors in surveys*, Biemer, P. P. et al. (eds.). New York: Wiley.

[saris2004] Saris, W. E., A. Satorra and G. Coenders (2004). A new approach to evaluating the quality of measurement instruments: the split-ballot MTMM design. *Sociological methodology*, 34(1), pp. 311-347.

[saris2007] Saris, W. E. and I. Gallhofer (2007). *Design, Evaluation, and Analysis of Questionnaires for Survey Research*. New York: Wiley.

[sarisfrth] Saris, W. E., A. Satorra and W. van der Veld (2007) *Testing structural equation models or detection of misspecifications?* Forthcoming.

[sqp2007] Saris, W. E., D. Oberski, and S. Kuipers. *SQP: Survey Quality Predictor* (computer programme). <http://www.sqp.nl/>

[satorra1992] Satorra, A. Asymptotic Robust Inferences in the Analysis of Mean and Covariance Structures. *Sociological Methodology*, 22, pp. 249-278

[willis2005] Willis, B. G. (2004). *Cognitive interviewing: a tool for improving questionnaire design*. London: Sage.

[zouwen1998] Zouwen, J. van der and W. Dijkstra (1998). Het vraaggesprek onderzocht. Wat zegt het verloop van de interactie in survey-interviews over de kwaliteit van de vraagformulering? [The survey interview dissected. What does the course of interaction in survey-interviews tell us about the quality of the question formulation?] *Sociologische Gids*, 45(6), 387-403.

MODULE INDEX

INDEX