PROFILEG
A Computer Program for Profile Analysis

Norman D. Verhelst
Eurometrics, The Netherlands

Introduction

Profile analysis is a powerful tool to detect systematic deviations from the predictions of a measurement model for binary or partial credit items. It can be applied at the individual level as well as at the group level. The program PROFILEG is aimed at detecting deviations at the group level.

The rationale of this analysis is based on the following principles:
1. The total sample is partitioned into a number (≥ 2) of *groups* of respondents;
2. The items of the test are partitioned into a number (≥ 2) of *categories* of items;
3. For each respondent and each category of items, the subscore in the category is compared to the expected subscore in the category. The difference between observed and expected subscore is called a deviation score, and the deviation scores, considered jointly for all categories, is called a *deviation profile*. These deviation profiles are the basic quantities on which the reported results are based. In the case of a group oriented analysis, the deviation profiles are aggregated in two different ways for all respondents belonging to the same group, and the final analysis is based on a comparison of the aggregated results across groups.
4. The expected subscore is a conditional expectation: it is the expected subscore given the *total score* on the test. This conditional expectation is a function of the item parameters only. The value of these parameters is usually fixed at their estimated values. So, profile analysis is to be carried *after* the calibration of the items.

The present manual is composed of two main sections. In the first section, a theoretical exposé is given on the features of profile analysis. In the second section, the manual for using the software is described in detail.

Section 1: Profile analysis

The concept of a score

A score on a test is a technical term whose definition depends on the measurement model used. One can consider the following cases:

- If the measurement model is the Rasch model, the score is the *raw* score, the number of correct responses. In this model, all the information about the respondent about the construct one wants to measure is contained in this score. Conversely, if one does not use explicitly a measurement model from IRT, but one *defines* the score as the raw score, then either one *implicitly* accepts the Rasch model or one ignores part of the useful information in the test performance.
- If one uses the two-parameter logistic model (2PLM), the score (in the sense of the quantity containing all information about the latent ability of the respondent) is the *weighted score*, where the weights are the discrimination parameters of the items. In the general 2PLM, however, the weights can be any positive numbers, and in general (i.e., with a few exceptions) all response patterns will result in a different weighted score. But this means that given the weighted score on the test, there is in general only

one response pattern possible that yields this weighted sum, and this makes the 2PLM in general not suited for profile analysis.

- One can, however, capitalize on the exceptions mentioned in the previous paragraph. Such an exception is the 2PLM where the discrimination parameters are *integer valued* (and do not differ too much from each other). This is the approach taken in the so-called One Parameter Logistic Model (OPLM, see Verhelst & Glas, 1995). Here is an example with 5 items, having discrimination parameters 1, 1, 2, 2, and 3, respectively. The two extreme scores (zero or perfect score) of course can occur only in one way (having zero items and all items correct, respectively), but all other weighted scores (from 1 up to 8) can each come about in more than one way. Notice that in this model the test score is always integer valued.
- In the Three Parameter Logistic Model (3PLM) there is no such thing as a score, so this model is not suited for profile analysis as treated here.
- If some of the items in the test are polytomous or partial credit items, raw and weighted scores are simple generalizations of their counterparts for the case of binary items:
    - If one uses the Partial Credit Model, the score is the *raw* score, defined as the sum (across items) of the partial credits obtained.
    - If one uses the Generalized Partial Credit Model, where items may have different discriminations, the score is the *weighted* score defined as the weighted sum of the partial credits, where the weights are the discrimination parameters. If the weights are restricted to integer values, a special case arises, which is suited for profile analysis. For calibration purposes this case is also suited for the software package OPLM (Verhelst, Glas & Verstralen, 1995) .
- For all other known models (probit models, the graded response model, models using latent classes and multidimensional models) there is no concept of a score, and therefore these models are not suited for performing a profile analysis as described here.

Profiles

Observed Profile

Suppose a tests consists of k items, on which a *categorization* has been defined, i.e., for a limited number m of *categories* (say, 2, 3 or 4), each item has been assigned to one (and only one) of these categories. Now, for a given respondent, with a given test score equal to s, one can compute the subscore for the m subtests defined by the categorization. The ordered m-tuple $(s_1, s_2,\ldots,s_m)$ of the m subscores is called the *observed profile*. Notice that for every observed profile it holds that the sum of the subscores equals the test score s.

Profiles as Multivariate Random Variables

If the test score is given, then in general, there can exist several observable profiles which yield the same test score, and their number is finite. The software package PROFILE-G carries out essentially the following two tasks:

- It enumerates (for a given score s) all possible observable profiles;
- For each possible profile (given s), it computes the probability that it will occur (under the measurement model used). This probability only depends (in a quite complicated way) on the item parameters.

The mathematical and procedural details on how these tasks are carried out are described in Verhelst (2011), and will not be repeated here.

Expected Profile

Once the distribution is completely known, it is simple to compute any summary statistic of this distribution. The mean vector is such a summary statistic and it is called the *expected profile.* Since the distribution is a conditional distribution, given the test score, the mean vector is also a conditional mean. For the conditional means it also holds that their sum equals the test score *s*.

Deviation Profile

The deviation profile of a respondent is defined as the difference between his/her observed profile and the conditional expected profile given the test score. A simple example is given in Table 1 for a categorization with two categories. The test score is 6. The expected profile is computed from the item parameters (not given here), and usually the expected subscores are not integer numbers.

Table 1: An example of an observed, an expected and a deviation profile

	Category *A*	Category *B*	Sum
Observed profile	4	2	6
Expected profile	4.406	1.594	6
Deviation profile	-0.406	+0.406	0

From this example, we notice some features which hold generally:

- The sum of the numbers in the deviation profile is zero. This holds for all profiles, independent of the number of categories. Of course, if there are only two categories, the two deviations must have the same absolute value and opposite algebraic sign.
- The interpretation of the values in the deviation profile is clear: it is a comparison (a difference) between what is observed and what is expected under the measurement model used. In the example of Table 1, the respondent performs worse than expected on items of category *A*, and better than expected on items of category *B*.
- From the deviation profile one **cannot** derive
    - the test score;
    - the number of items in the test;
    - whether there are more or less category *A* items in the test than category *B* items;
    - whether the category *A* items are on the average easier or more difficult than the category *B* items.

The fact that the deviation profile hides so many things is not a weakness, but it is its strength. One can construct a deviation profile for any respondent, irrespective of his/her test score, and the profile will tell us if the respondent was doing better than expected or worse than expected on category *A* (or *B*).

But there are stronger results. Suppose that the categorization of the items is based on the item format, category *A* being multiple choice items and category *B* open ended items, and that one has administered several forms of the test, each form containing some items of each category. Then for every respondent one can construct a table like Table 1, and decide for every respondent if he/she performs better or worse than expected on category *A* (or *B*). More in general, this means that profile analysis can be carried out in incomplete designs. The only restriction is that the item parameters can be estimated from all data jointly, which in practice means that the incomplete design has to be linked. Notice, that in such a case, the construction of the deviation profile only uses the parameters of the items contained in the test form the respondent has answered to.

Aggregation of deviation profiles

Although very useful, the analysis of profiles cannot suffice with the construction of a deviation profile for every respondent in the sample. Suppose one has test data from 5,000 students. Then the primary output from the software consists of 5,000 deviation profiles, and the problem will become apparent immediately: what can one do with these 5,000 deviation profiles? Is there a reasonable way to summarize them in such a way that substantial conclusions can be drawn?

The basic rationale for profile analysis is a comparison of the deviation profiles across predetermined groups of respondents. In the example with multiple choice and open ended items, one might have the hypothesis that depending on the testing culture in some countries, in one country the relative performance on multiple choice items will be better in one country than in the other. Therefore, the problem is how to summarize deviation profiles for all respondents in each of the predetermined groups.

The program PROFILE-G provides two summaries, one consisting of a contingency table per group; the other is the average deviation profile per group. These two approaches are discussed next. It will also be indicated how differences can be tested statistically.

Aggregation 1: Contingency tables

For each predefined group a two dimensional contingency table is constructed. The first dimension is the type of profile, the second profile is the triviality of the deviation profile.

Profile Types

If there are two categories in the categorization, two types of deviation profiles can be distinguished: either the respondent performs better than expected on category *A* items (and hence, worse than expected on category *B* items) or the other way around. (The case where the deviation profile contains a zero value can be ignored, as observed scores are integer valued but expected scores will in general be fractional numbers.) The former type will be indicated as '*CategoryA+*' and the latter one as '*CategoryB+*'. The deviation profile given in Table 1 is of type '*CategoryB+*'

With three categories six different types can be distinguished, which can be indicated as follows (+ – –), (– + –), (– – +), (– + +), (+ – +) and (+ + –), where a '+' means 'better than expected' and a '–' 'worse than expected'. The two types (+ + +) and (– – –) cannot occur since the sum of the deviations is zero. This means that in every deviation profile there are two deviations with the same algebraic sign and one with the opposite algebraic sign. In the output of the program, these types will be indicated with the label of the category followed by a '+' or a '–'. For example '*CategoryC–*' means that the deviation for the category with label '*CategoryC*' is negative, while the other two are positive. If '*CategoryC*' is the third category, the type '*CategoryC–*' corresponds to (+ + –).

With more than three categories, the number of types grows rapidly, and therefore contingency tables are not computed in these cases[1].

Triviality

The deviation profile given in Table 1 is not identical zero (which is impossible since the expected subscores are fractional numbers), but one can ask whether the deviations are important in some sense. To answer this question, we need to proceed in two steps: the first step is to develop a summary quantity for the deviation profile and the second step is to judge on the magnitude of this quantity. These two steps are discussed next.

The Disparity Index D^2

The disparity index is a chi-square distance between the observed and the expected profile. In Table 2, it is illustrated how this index is constructed. Suppose a test where the maximum obtainable score[2] is 36, and where for the sake of a profile analysis, three categories are distinguished. The maximum subscore for each category is displayed in the bottom row of the table. The observed and the expected profile are displayed in the first row of the table, expected subscores between parentheses. The second row is just the complement of the first row with respect to the maximum subscore.

Table 2. Construction of the disparity index D^2

	Category *A*	Category *B*	Category *C*	Total
Score obtained	7 (4.55)	4 (7.84)	8 (6.61)	19
Score not obtained	3 (5.45)	10 (6.16)	4 (5.39)	17
Maximum score	10	14	12	36

The disparity index D^2 is defined as

$$D^2 = \sum_i \frac{(O_i - E_i)^2}{E_i} = 7.35$$

where the summation runs over all six cells in the interior of Table 2, O_i stands for observed score and E_i for expected score. The value of D^2 for the profile given in Table 2 is 7.35.

The Conditional Distribution of D^2

To judge on the importance of the profile, we will make a judgment on the magnitude of D^2, but in such a way that this judgment is consistent across test scores and across test forms. Since the software computes the complete conditional distribution of all profiles compatible with a given score and a given test form, we consider these distributions.

An example is given in Table 3, which corresponds to the profile given in Table 1. If the test score is 6 and there are two categories, then there are no more than 7 observable profiles:

---

[1] If there are *C* categories, the number of types is 2^C-2. For four categories this is 14 and for $C = 5$, the number of types is 30.

[2] This does not mean that there are 36 items in the test: there may be less, because some discrimination items may be larger than one and/or there are some partial credit items in the test.

(0,6), (1,5),…,(6,0). For each profile, the software computes the probability and the disparity index D^2. In Table 3, this distribution is displayed, but the observable profiles are sorted in increasing order of their disparity index. The column 'Cum. Prob.' is the cumulative probability and the column 'Type' gives the type of the deviation profile (the expected profile is (4.406, 1.594), see Table 1).

Table 3. The distribution of D^2

profile	D^2	Prob.	Cum. Prob.	Type
(4,2)	0.160	0.348	0.348	*B*+
(5,1)	0.348	0.361	0.709	*A*+
(3,3)	1.933	0.141	0.850	*B*+
(6,0)	2.497	0.122	0.972	*A*+
(2,4)	5.668	0.026	0.998	*B*+
(1,5)	11.36	0.002	1.000	*B*+
(0,6)	19.02	<0.001	1.000	*B*+

A deviation profile is called **trivial** if its associated D^2 is less than or equal to the **median** of the conditional distribution, otherwise it is called **non-trivial**. The median value of the disparity index is defined as the largest value whose cumulative probability does not exceed 0.5. In Table 3, the cumulative probability of $D^2 = 0.160$ is 0.348; the next observable value is 0.348 but its cumulative probability is 0.709 > 0.5. Therefore the median is 0.160, and (4,2) is the only trivial profile[3].

Contingency Tables

Tables 1 and 3 contain all the information to construct a very simple contingency table (with a total sample size of $n = 1$). It is displayed as Table 4. The row 'observed' gives the number of observations. The deviation profile from Table 1 is classified as 'trivial' and '*B*+'. The expected frequency in each cell is just the sum of all probabilities (from Table 3) to which the classification applies. Example: there are two profiles of type '*A*+' in Table 3 and both are 'non-trivial'; the sum of their probabilities is 0.361 + 0.122 = 0.483.

Table 4. A simple contingency table

	Non-trivial		Trivial	
	A+	*B*+	*A*+	*B*+
observed	0	0	0	1
expected	0.483	0.169	0.000	0.348

These simple contingency tables are not very interesting by themselves, but they become interesting when they are summed across all respondents belonging to the same group: they give per group observed and expected frequencies for a cross-classification of type and triviality. In table 5, an example is given from an analysis of the PISA data for Reading (cycle 2000). In the profile analysis, the groups were defined as the participating countries. In table 5, the contingency tables for two countries are displayed. The item categories are 'open ended' (OE) and 'multiple choice' (MC).

[3] If the probability of the smallest observable disparity index is larger than 0.5, the median is defined to be zero, and all profiles are defined to be non-trivial.

Table 5. Two contingency tables

		non-trivial		trivial	
		OE+	MC+	OE+	MC+
country 1	obs.	1062	1014	468	488
	exp.	973.6	981.5	538.6	538.4
country 2	obs.	1260	1592	694	699
	exp.	1366.9	1372.5	774.5	731.1

Further Analyses

Here ends the contribution of the program PROFILEG for the contingency tables. If the number of categories is two, tables like Table 5 are output, where the number of groups is arbitrary. If the number of categories is three, similar tables are output, but they have 12 numerical columns: 6 types times 2 values for triviality. Further analyses of these tables are left to the user.

By way of example, a number of considerations and analyses are presented here in connection with Table 5.

For each country separately, a simple chi-square test can be carried out. The number of degrees of freedom is 3. Notice, that the expected frequencies as displayed in the table have to be used in computing the chi-square statistic. For both countries from Table 5, the tests give highly significant results: the test statistic for country 1 is 23.06 and for country 2 it is 53.25.

A closer look at Table 5, however, will reveal that in both countries the number of observed trivial profiles is less than the expected number. This is most probably caused by the fact that the measurement model (in this case the Rasch model) cannot explain all the variation in the observed responses. But this is not very interesting as one already suspects when starting the present analysis: if the Rasch model would be valid (for all data, irrespective of the country), then 5% of all possible tests would be significant at the 5% level. By doing the present analysis, however, one suspects that the performance in the category MC, say, might systematically differ between countries, and if this is so, the Rasch model can not be valid for all countries taken together.

By doing this analysis (with the categorization defined as it is), one suspects that there might be systematic differences between countries with respect to the categorization chosen. So one might to do some statistical testing where only the contrast between MC and OE items is involved.

What one must not do, is to apply a chi-square test on the non-trivial profiles alone, because the sum of the observed frequencies and of the expected frequencies is not equal in the subtable of non-trivial profiles. The only way to do a sensible analysis is to collapse Table 5 along the dimension triviality. These collapsed tables are displayed in Table 6. The chi-square statistics (with 1 degree of freedom) are 0.42 for country 1, and 33.1 for country 2. The latter one is highly significant, showing that in country 2 more students than expected have a deviation profile in favour of the multiple choice items, or what amounts to the same: multiple choice items are easier in country 2 than in country 1.

Table 6. Table 5 collapsed along the triviality dimension

		OE+	MC+
country 1	obs.	1530	1502
	exp.	1512.2	1519.8
country 2	obs.	1954	2291
	exp.	2141.4	2103.6

Aggregation 2: mean profiles

Since for each observed score, the distribution is completely known, one can not only compute the expected profile but also the variance-covariance matrix of the observed profile (considered as a random variable). This is done by the computer program PROFILEG.

Denote the covariance matrix associated with test score s in test form f by Σ_{fs} and the number of respondents having obtained score s in test form f by n_{fs}, then the variance covariance matrix of the mean deviation profile is given by

$$\frac{\sum_f \sum_s n_{fs} \Sigma_{fs}}{\left(\sum_f \sum_s n_{fs} \right)^2} \qquad (1)$$

In Table 7 an example is given for a categorization into two categories, labelled as A and B[4]. The analysis is run with three groups. The table contains the mean profiles and the sample sizes of the groups and the variances

Table 7. Mean profiles of three groups

group	size	*A*	*B*	Variance[5]
group 1	457	-0.742	0.742	5.6603E-02
group 2	1733	0.142	-0.142	1.3379E-02
group 3	635	0.148	-0.148	3.4091E-02

Here are some comments going with Table 7:

- It is seen that for each group the sum of mean deviations equals zero, just as is the case for individual profiles. This also holds for three or more categories.
- The variance-covariance matrix of the means is in the example with two categories a 2 x 2 matrix, but since the sum of the deviations is zero, the sum of each row (and each column) of the matrix is zero. The means that necessarily the two variances must be equal and the covariance must equal minus the variance. The zero-sum feature of the matrix also holds in case of more than two categories, but the variances do not have to be equal and the covariances cannot be predicted from the variances alone. An example of such a matrix is given in Table 8.
- The variance displayed in the rightmost column is the variance of the mean, which is not the same as the mean variance: the variance of the mean is the mean variance divided by the sample size. This explains the square in the denominator of formula (1).

[4] At the moment of writing this report, the results of this analysis has to be published yet. Therefore no details about the concrete hypotheses are given.

[5] To have always the same number of significant digits for the variance, it is displayed in scientific notation. The numbers 1.2345E-2 and 1.2345E+3 represent the numbers 0.012345 and 1234.5, respectively, in common decimal notation. The number following the 'E' in this notation indicates how many places the decimal point has to be shifted. To the left if the number is negative and to the right if it is positive.

- The variances across the groups are not equal. This is so because of two reasons:
    - The sample sizes are not equal across groups;
    - The variance-covariance matrix for an individual profile depends on the score and on the specific test form that is used for this individual.
- A feature which is not immediately obvious from Table 7 is this: for each category, the weighted mean of the deviations (with weights equal to the sample sizes) will often be close to zero. It equals exactly zero when the following two conditions are fulfilled:
    - The total group entering the profile analysis is the calibration sample;
    - Item parameters are estimated by conditional maximum likelihood (CML).
- Notice that in Table 7, the distinction between trivial and non-trivial profiles is ignored.

Table 8. A variance-covariance matrix for three categories
(example based on real data)

	A	***B***	***C***
A	2.5837E-03	-1.8396E-03	-7.4414E-04
B	-1.8396E-03	4.8041E-03	-2.9646E-03
C	-7.4414E-04	-2.9646E-03	3.7087E-03

Further Analyses

The program PROFILEG outputs the mean deviation profiles and their variance-covariance matrices. If the number of categories equals two, statistical tests are carried out by the program PROFILEG. If the number of categories is larger than two, further analyses are left to the user. By way of an example, some analyses are presented next. They are all based on Table 7. This table, enhanced with two extra rows and an extra column is displayed as Table 9.

Table 9. Some analyses based on Table 7

group	size	*A*	*B*	Variance	z
group 1	457	-0.742	0.742	5.6603E-02	**-3.117**
group 2	1733	0.142	-0.142	1.3379E-02	1.231
group 3	635	0.148	-0.148	3.4091E-02	0.799
group (2+3)		0.145	-0.145	2.4121E-02	0.934
gr.1 - gr.(2+3)		-0.887	0.887	8.0724E-02	**-3.121**

Here are the comments going with table 9:

- The mean profiles of group 2 and group 3 are quite similar; so it may increase the power of the statistical tests if these two groups are combined. Of course one can run another analysis with a modified definition of the groups, but in this case, mean profile and the variance of the mean can be calculated quite easily.
    - The mean of the combined group is the weighted mean of both means; the weights are the sample sizes.
    - The variance of the combined mean is the weighted mean of both variances **plus** the variance of the means.
    - The results of these calculation are displayed in Table 9 in the row 'group (2+3)'.
- The null hypothesis is the measurement model, and for the mean deviations, this implies that in each group, the null hypothesis is that the mean is zero. To test this hypothesis, it is sufficient to compute the ratio between the observed mean and the square root of its variance. Using the central limit theorem, this ratio is standard normally distributed. Its

value is displayed in the added column with column head ‘**z**’. As can be seen from Table 9, the average deviation (for category *A*) is significantly smaller than zero, whereas for the other two groups and for the combined group, no significance is reached.

- A more powerful test, however, can be reached by considering contrasts between groups. It is possible that for two groups the mean deviations do not differ significantly from zero, but that the difference between the means does reach significance. In the bottom line the difference between the means is displayed; the variance of this difference is just the **sum** of the two variances: 5.6603E-02 + 2.4121E-02 = 8.0724E-02. The difference is significantly different from zero, meaning that group 1 reacts differently to the items than the combined group.

If there are only two categories, a statistical test is carried out as in the first three lines of Table 9. Moreover, if there are only two groups and two categories, a contrast between the two groups is tested. The contrast is the difference between the means on the first category.

Section 2. The program PROFILEG

Introduction

The program PROFILEG needs quite a bit of information to run properly. All this information is stored in five different files, which have to be prepared by the user. All files required are text files. The name of the text files is arbitrary, but in the context of the present manual they are given a generic name. The generic name of the text files will be underlined in this manual.

The five files needed are characterized briefly in this introduction section, and are fully described in separate sections.

1. The file Profile-JDF is the so-called Job Definition File. It contains some general information on the tasks to be carried out. It is the first file that is read by the program.
2. The file Label file contains the labels (names) of the groups, the categorizations and the categories.
3. The file Item file contains for each item the maximum score, the discrimination parameter, the difficulty parameter(s), and for each categorization the category number of the item.
4. The file Booklet file contains for each test form (booklet) the sequence number of the items which appear in the booklet. In this file the data collection design (either complete or incomplete) is described.
5. The file Data file contains for each respondent an identification of the test form and of the group the respondent belongs to, and it contains all item responses.

The Job Definition File (Profile-JDF)

This text file contains 11 lines (records), and each record contains either a text string or one or more numbers.

If several numbers are read from a single record, they are read in so-called ‘free format’. This means that the numbers need not to be placed at fixed positions, but that their place is free. Numbers must be separated by one or more blanks or a comma[6]. The last number may be followed by arbitrary text (comment), but there must be a blank after the last number.

If a text string (like a file name) is read, it may be placed freely in the record. Leading blanks are ignored by the program. However, the string length must not exceed 255 characters, including leading blanks. Characters from position 256 on are not read. The first 255 positions must not contain comments.

1. Record 1 contains three numbers: the number of test forms (or booklets), the total number of items and the number of categorizations, in that order.
2. Record 2 contains the number of categories for each categorization. The sequence of the number of categories must correspond to the category identifications as indicated in the Item file (see below).
3. The program provides the possibility to exclude some item responses from the analysis, e.g., if a respondent has not reached the end of the test. Such excluded responses must be coded especially in the Data file. Record 3 contains one or several numbers:
    a. If there are no such codes, record 3 contains the number zero;

[6] Be careful not to use several commas in sequence. They will be accepted by the program, but the results of the computations will in general not be correct. It may also cause the program to crash.

    b. If there is such a code or several such codes, then record 3 contains the number of exclusion codes and the codes themselves. These codes must be numeric. Example: if record 3 contains 1 6, this means that all codes '6' in the data file will be considered as not-administered items. If it contains 2 6 8, then all codes 6 or 8 will be considered as referring to not administered items.
    c. The maximum number of exclusion codes is 5.
    d. The precise way on how the coded data are treated in the program is discussed in the section on the Data file.
4. Record 4 gives the user the possibility of excluding respondents from the analysis. It contains two numbers, referring to the variables MAXECL and MINSC (in that order).
    a. MAXEXCL is the maximal number of exclusion codes that is allowed. Example: if this value is 4, then all respondents with up to 4 exclusion codes will be included in the analysis, and the items which have an exclusion code will be treated as not administered. If there are more than 4 exclusion codes, however, the respondent is dropped altogether from the analysis.
    b. MINSC is the minimal test score for a respondent to be included in the analysis; it automatically also puts a restriction on the maximum score in the following way: suppose MINSC = 3, then every respondent with a score less than 3 or a score greater than the maximum possible score (in the test form of the respondent) minus 3 is dropped from the analysis. The minimal value of MINSC is 1, meaning that respondents with zero or perfect test scores are always excluded from the analysis.
5. Record 5 contains one number, the number of groups. Further down in this manual this number will be referred to symbolically as NG. (See the section on the Label file.)
6. Record 6 contains the name of the Label file, preceded by the path. For example, the string to be read from this record might be: '*D:\myproject\profile\mylabels.txt*' (without the quotes). This means that the file containing the labels has the name '*mylabels.txt*', and it is stored on disk *D:* in the folder '*profile*' which is a subfolder of the folder '*myproject*'.
7. Record 7 contains the name of the Item file, preceded by the path.
8. Record 8 contains the name of the Booklet file, preceded by the path.
9. Record 9 contains the name of the Data file, preceded by the path.
10. Record 10 contains information on **how** to read the Data file. This information may be conveyed in two different ways:
    a. By a legal FORTRAN format. This is a text string starting with a left parenthesis '(' and ending with a right parenthesis ')'.
    b. By a sequence of six numbers. Notice that, if this option is chosen, no comments must appear in this record in the first 255 positions.

    Details on how this information must be conveyed will be given in the section on the Data file.
11. Record 11 contains the name of the Output file, preceded by the path. The name of the Output file is arbitrary. This file is constructed by the program; it is a text file. If a file of that name (and at the same location as indicated by the path) exists before the program is run, it will be overwritten by the program.

The Label file

This file contains the labels for the groups, the labels for the categorizations and for the categories. Each label is placed on a separate record. The first **16 positions** of each record are read, and their contents will define the labels as they appear in the output. All characters beyond the 16^{th} position are ignored and leading blanks are part of the label.

The first NG (number of groups) labels are the labels of the groups. The order in which they appear here must correspond to the group identification in the Data file. Example: suppose there are two groups corresponding to boys and girls, and suppose the labels given are 'boys' and 'girls'. If the label 'boys' is the first label in the Label file, then the identification of the group membership in the Data file must be '1' for boys and '2' for girls.

Next to the group labels, labels must be provided for the categorization and for the categories of that categorization, following these rules:

- The first label is the name of the categorization.
- This label is followed (immediately) by as many labels as there are categories in this categorization. The number of category labels per categorization is specified in record 2 of the Job Definition File.

In Table 10, an example is given. The left-hand part is the Label file; the right-hand part contains some comments.

Table 10. Example of a Labelfile

Labelfile	Comments
Boys	There are two groups (see record 5 of JDF)
Girls	The group id. of the boys in the Data file is 1.
ITEM FORMAT	This is the label of the first categorization
Multiple choice	Two different item formats are distinguished.
Open Ended	MC items will be identified as category 1.
DOMAIN	This the label of the second categorization.
Personal	The order of the category labels must correspond to
Educational	the category identification in the Item file.
Professional	Record 2 of the Job Definition File contains the
	numbers 2 and 3 (in that order).

Here are some more comments:

- The use of lower case or upper case is arbitrary. Labels will be printed in the Output file as found in the Label file, up to the first 16 characters.
- Labels are read using the information in the Job Definition File (records 1, 2 and 5). If too many labels are provided, the program will read the number of labels it needs and will continue with the analysis. If too few labels are found, this will result in a run time error and the program will crash.
- If the labels are not in the correct order, this will not lead to an error message, but the output may be very confusing.
- Do not insert blank (empty) records in the Label file, as they will be read as empty labels.

The Item file

- This file contains as many records as the total number of items, one record per item.
- All information in the records is numeric, and the records are read with free format.
- The following information must be provided in each record, and in the given order:
  - The item number;
  - The maximum score obtainable on the item;
  - The discrimination parameter of the item;
  - The difficulty parameter(s) of the item;
  - The category the item belongs to, for all categorizations requested.

- The records in this file may be sorted according to any criterion.
- The numbers in this file must not contain fractional numbers, except for the difficulty parameters. All other numbers are integer valued and must not contain a decimal dot.
- Decimals in fractional numbers are indicated by a dot, not by a comma.

The five kinds of information are discussed next.

Item Numbers

The items in the test (all booklets taken jointly) are identified by a number. The number must be in the range 1 to the total number of items, and the number must be unique, i.e. the same number must not appear twice.

These numbers are needed by the program PROFILEG but play also an important role in the construction of the Booklet file.

Maximum Score

If the item is binary, the maximum score equals 1. If the item is a partial credit item, the maximum score is the maximum of points (marks) that can be obtained on that item. Notice that this maximum score refers to an unweighted score: if one can earn either zero, one or two points on an item, the maximum score is 2, irrespective of the weight that is applied for this item in determining the weighted test score.

Discrimination Parameters

If the Rasch model or the Partial Credit Model is used, the discrimination parameter is 1 for all items.

If OPLM is used with different discriminations, the discrimination parameter to be used is the discrimination index used in the calibration. These indices are integer values.

Difficulty Parameters

If the item is binary, there is one difficulty parameter. The value to be filled out is the value found in the calibration.

For partial credit items, several different forms of representing the model are possible. Two of them are discussed here.

Let X_i be the response variable for item *i,* and suppose it can take the values 0, 1, …, m_i (m_i is the maximum score for item *i*). In the first parameterization (henceforth referred to as the β-parameterization), the response function for partial score *j* (i.e. the probability that the variable X_i takes the value *j*, conditional on the latent variable θ)[7] is given by

$$P(X_i = j \mid \theta) \propto \exp\left[j\theta - \sum_{g=1}^{j} \beta_{ig} \right], \quad (j = 1, \ldots, m_i). \qquad (2)$$

[7] In the two formulae to follow, the equality sign ('=') is not used, as the functions are only given up to a so-called proportionality constant (hence the proportionality sign '$\propto$'). The proportionality constant is one divided by the denominator one is used to in formulae about IRT. For the sake of the argument, this denominator is not needed, and is omitted here because it complicates the reading of the formulae.

In words: the probability that a score *j* is obtained depends on the difference between *j* times the latent ability on the one hand and the sum of the *j* parameters $\beta_{i1} + \beta_{i2} + \ldots + \beta_{ij}$, on the other hand. For a partial credit item, the number of β-parameters equals the maximum score m_i.

In the second parameterization (henceforth referred to as the δ-τ parameterization), the formula of the response functions is given as

$$P(X_i = j \mid \theta) \propto \exp\left[j\theta - \sum_{g=1}^{j} (\delta_i + \tau_{ig}) \right], \quad (j = 1, \ldots, m_i). \tag{3}$$

Where δ_i is usually called the difficulty parameter of the item, and the parameters τ_{ij}, (j=1,…,m_i) are referred to as the step parameters[8]. So, it seems that in this representation there are $m_i + 1$ parameters associated with item *i*. This is true, but these parameters are not all free: the sum of the τ-parameters (within a single item) is restricted to be zero, such that there are only m_i free parameters.

In the Item file the item parameters to be filled out are the estimates of the β-parameters. These are given directly if one uses the IRT software package OPLM for calibration. Other program packages (e.g., Conquest, WINsteps or FACETS) use the δ-τ parameterization, and the parameter estimates resulting from these packages have to be transformed to the ones of the β-parameterization. This transformation has to be done by the user. One can use the following rule:

$$\beta_{ij} = \delta_i + \tau_{ij} \tag{4}$$

In some applications, the so-called Rating Scale Model (RSM) is used as measurement model. This model is a special case of the Partial Credit Model, because it puts further restrictions on the PCM-parameters (in the δ-τ parameterization): it assumes that the step parameters of any step *j* have the same value for all items. In formula form, this means that in the RSM it holds that

$$\tau_{ij} = \tau_j \text{ (for all items } i \text{ and all steps } j\text{).} \tag{5}$$

Of course, formula (5) only makes sense if the maximum score m_i has the same value for all items. If this model is used as measurement model, formula (4) has to be applied also, with τ_{ij} replaced by τ_j.

Category Identification

Next to the item parameters the category identification of the item has to be indicated for each categorization. This identification is numeric: 1 for the first category, 2 for the second category, etc. If there are five categorizations, the difficulty parameter(s) of the item must be followed by five numbers.

There is, however, a powerful feature in the program, that allows the user to exclude items from the profile analysis. This is done by the so-called zero category: an item with category identification equal to 0 does not enter the analysis. In particular this means that the test score is redefined: it is computed from the item responses which do take part in the analysis. Here are two conditions where the use of zero-category items is meaningful:

[8] It is possible that other terminology is used as well. The term step parameter is used in the manual of the software package Conquest.

- Suppose some item has been removed from the test (because it appears to behave oddly) during the calibration. Removing this item physically from the Data file (and decreasing the total number of items by one) is very error prone. To avoid such errors, one can leave the responses to this item in the Data file and use category zero for this item in all categorizations.
- Suppose one is interested in the contrast between multiple choice (MC) items and open ended (OE) items in two or more groups. Some items in the test, however, may not be clearly classified as belonging to either of these two categories, e.g., items which require a short response (a number or a single word). Classifying such items anyway as MC or OE may obscure the contrast (i.e. lower the power of the statistical tests). An elegant way out of this problem is to classify these doubtful cases as zero category items, and to leave them out of the profile analysis.

Example

In Table 11 an example of the first four records of an Item file are displayed. The number of categorizations is three. The table is followed by some comments.

Table 11. Example of an Item file

```
15 1 2  0.789             1 2 0
 2 2 1 -0.545 0.623       2 1 1
19 1 1  1.000             0 0 0
 5 3 1  0.765 0.552 1.201 1 1 2
. . .
```

- The records are not sorted by item number. This is allowed: any sorting will do, and the sorting used has no influence on the performance of the program.
- Items 15 and 19 have a maximum score of 1, i.e., they are binary items. Items 2 and 5 are partial credit items, with a maximum score of 2 and 3, respectively.
- Item 15 has a discrimination index (the weight used to compute the test score) of 2, the other weights are equal to 1.
- The number of β-parameters to appear after the weight equals the maximum score. Notice that these parameter values must be given in the correct order: the first number for item *i* is the estimate of β_{i1}, the second number is β_{i2}, and so on.
- Item 15 is excluded from the third categorization, but it plays a role in the other two.
- Item 19 is excluded from all categorizations. This may be an item that has been excluded at calibration time. It is included here to have consistent item numbering. Notice that for such an item, maximum score, weight and item parameter(s) have to be specified (i.e., they must be there). Their value, however, is arbitrary.
- In the example, spaces (blanks) have been used to give the file an orderly appearance. This is not necessary: blanks can be added or deleted arbitrarily on the condition that
    - Different numbers are separated by at least one blank (or a comma);
    - Digits belonging to the same number are not separated. Replacing '15' in the first record by '1 5' (a blank between the 1 and the 5) will cause the program to read two numbers (1 and 5), and this will eventually lead to a crash of the program.

The Booklet file

This file contains all information about the design of the data collection. If there are *B* booklets (test forms) the file contains *B* + 1 records

The first record contains *B* numbers (to be read in free format): the number of items in each booklet.

Then for each booklet, a record is read with the item identifications of the items in the booklet (in free format).

- The order of these records is important: the first one refers to booklet 1, the second to booklet 2, and so on. These booklet numbers also appear in the Data file.
- The item identifications are the item numbers used in the Item file.
- The order in which the item identifications are written in the record must correspond to the order of the item responses in the Data file. Usually this will be the order of administration of the items.

In Table 12, a small example with two booklets is given. The total number of items is six, but each booklet contains fewer items.

Table 12. An example of a Booklet file

```
4 5
1 2 3 4
3 1 5 6 2
```

- Booklet 1 contains 4 items and booklet 2 has 5 items.
- If in the Data file a record is found referring to booklet 1, four item responses will be read and they will interpreted as the answers to the items 1, 2, 3 and 4, in that order.
- If a record is found referring to booklet 2, five responses will be read, and they will be interpreted as the responses to the items 3, 1, 5, 6 and 2, in that order.
- Notice that it is convenient to include also the items which are removed (logically but not physically) from the item set during the calibration. Such items are removed from the profile analysis by assigning them category zero in all categorizations. (see the section on the Item file.)
- If the sequence of item numbers is increasing (by one), a range can be specified by using a dash: 1-4 is equivalent to 1 2 3 4.

The Data file

The Datafile is a text file that contains a single record for each respondent. Each record must contain at least:

- The booklet number of the booklet that has been administered to the respondent;
- The group identification of the group the respondent belongs to;
- The answers to all items of the booklet administered in some coded form.

Next some comments are given for each of these three kinds of information:

Booklet Number

If there are in total *B* booklets used, the booklet identification is a number in the range 1 to *B*. This number must correspond to the ordering of the records in the Booklet file.

If the booklet number found in the Data file is out of range (i.e., smaller than 1 or larger than *B*), the record is skipped. The number of records skipped for this reason is counted, and the count is reported in the Output file. The code for the booklet must be numeric. Other codes like '?' or 'x' will make the program crash.

Group Number

If there are *NG* groups, as specified in the 5th record of the Job Definition File, the group identification is a number in the range 1 to *NG*. Take care that the specific group number corresponds to the labelling of the groups in the Label file.

If the group number found in the Data file is out of range (i.e., smaller than 1 or larger than *NG*), the record is skipped. The number of records skipped for this reason is counted, and the count is reported in the Output file. If booklet number as well as group number are out of range, the record is counted only once (skipped for out-of-range booklet number).

The code used for the group identification must be numeric. Other codes like '?' or 'x' will make the program crash.

Item Answers

For the item answers the following rules apply:

- All codes used for the item answers must be numeric, with one possible exception: a blank is read as a zero. Any other coding (like a '?' or an 'x') will cause a crash of the program.
- The codes for partial credit items are equal to the score obtained on that item. If the maximum score is 3, codes '0', '1', '2' and '3' are interpreted as representing the score obtained on that item.
- For other codes, the following rule applies:
    - If the code corresponds to one of the exclusion codes (see record 3 of the Job Definition File), the item is considered as not administered to the respondent. This corresponds to defining implicitly a new booklet for this respondent.
    - If the code is not an exclusion code, it is transformed (internally in the program, not physically in the Data file) to a zero. This corresponds to the viewpoint that the respondent has seen the item, but did not give an answer, because he/she did not know the answer, and therefore the score is zero.
    - It should be clear that the ultimate control over this rule is the user's: its is his/her responsibility to decide how to treat missing observations, and the way to control it is by using the exclusion codes.

The Test Score

For each respondent the test score is defined as the (weighted or unweighted) sum of the item scores he/she has obtained on the collection of items that enters the profile analysis and that has been administered to him/her. This definition has some important implications:

- For each respondent, the sum of the subscores (on the categories) is always equal to the test score;
- Item scores on items with a category zero assignment are not part of the test score;
- Respondents can be included for some categorization but excluded for another one.
    - Suppose MAXEXCL has been set to 5 (by the user; see record 4 of the Job Definition File) and some respondent has 6 exclusion codes in his record. In a categorization where none of the items has category zero, this respondent will be

excluded from the analysis, but for a categorization where one or more of these items have category zero, the respondent will be included.
   - Much the same applies to the differential effects of the minimal required score MINSC.

The Format of the Data

The data in the Data file are read with a fixed format. The user has to communicate to the program how the data are to be read, i.e., it must be specified where the numerical data that the program needs are to be found in the record. This can be done in two different ways, which will be discussed in turn. The discussion will be illustrated by a sample data file, part of which is displayed in Table 13.

Table 13. Example of a Datafile

```
         1         2         3         4
1234567890123456789012345678901234567890
John       1 1      100110111010111
Anne       1 2      110101110001110
Mary-Ann   1 2      101000101100666
Neil       1 1      000110000101010
. . .     . . .        .  .  .
Neil-2    13 1      01111001010011101010
Monica    13 2      11111011101111110011
Kevin     13 1      00010001010001006666
```

- The top section of Table 13 indicates the position of the file content. This section does not belong to the file.
- The data are collected with a design that has more than 9 booklets. In the table four records from booklet 1 and three records from booklet 13 have been displayed. In each record the booklet number is at position 12 if it is smaller than 10, and at the positions 11 and 12 if it consists of two digits.
- The group number is in every record at position 14
- The item answers start in every record at position 21, but the records are not equally long: in booklet 1 there are only 15 items, while there are 20 items in booklet 13.
- An item answer at the same position in the record does not necessarily refer to the same item. The answer given at position 21 refers to the first item of that booklet in the Booklet file, and the first item in booklet one can be another item than the first item in booklet 13.
- The file may contain other information as well. In Table 13, the first 10 positions have been reserved for identification of the respondent. This information will not be read by the program PROFILEG

By a *Format* we mean a specification such that the data are read in the right order and at the right positions in the Data file.

The order in which the data have to be read is this:
- The booklet number has to be read as the first number;
- The group number has to be read as the second number;
- The item answers have to be read next in the order that corresponds to the item identifications in the Booklet file. The easiest and safest way to accomplish this is to make the program read the item answers from left to right, and to construct the Booklet file in

such a way that it contains for each booklet the correct item identifications (item numbers) in that order.

The *Format* specification (record 10 of the Job Definition File) may be conveyed to the program in two different ways: by a FORTRAN Format specification or by a sequence of six integer numbers. These are discussed in turn.

A FORTRAN Format

The FORTRAN Format that goes with the example in Table 13 is the following string:

(T11, I2, T14, I1, T21, 20I1)

- The string starts with a left parenthesis and closes with a right parenthesis. The string is parsed from left to right. The use of commas as indicated is compulsory, the blanks are optional.
- The string may contain two different letters: 'T' and 'I' (or 't' and 'i'):
    - A 'T' is followed by a number. T stands for Tab, and the number following the 'T' specifies the position in the record. Loosely speaking, one can read 'T11' as 'position the reading head at position 11'.
    - An 'I' indicates that an integer number has to be read. The I is followed and (optionally) preceded by a number:
        - The number following the 'I' indicates how many positions the number occupies in the record. I1 indicates that the number is contained in one position; I2 indicates that the number occupies 2 positions. After reading of the number, the 'reading head' automatically shifts that number of positions to the right.
        - The number preceding the 'I' is called the repetition factor. '20I1' means read 20 numbers of 1 position. If the repetition factor is omitted, it takes its default value of 1: 'I2' means the same thing as '1I2'.
        - The repetition factor '20' in the example is larger than the number of items in booklet 1, but this is not a problem: a repetition factor may be too large, but it must not be too small. If we change '20I1' in the Format string to '15I1', then the format would be suited for booklet 1, but not for booklet 13. What happens then is unpredictable: the program may or may not crash, but the results (if any) of the program will certainly be meaningless.
    - The Format string above can now be read in words as follows: put the reading head at position 11, read a number of two digits, put the reading head at position 14, read a number of 1 digit, put the reading head at position 21, and read consecutively (maximally) 20 numbers of 1 digit.
- From the Format one cannot deduce that the first number read is the booklet number. This is decided by the program. The user must take care that the first number read is the booklet number, the second the group number and all the others are item responses. As an example, suppose that in every record, the group number is placed at position 51 and the booklet number at positions 61-62. Then the FORTRAN Format string needed would be: (T61,I2,T51,I1,T21,20I1), meaning loosely that by using the 'T', one makes the reading head jump forwards and backwards to any position in the record.

The 6 Number Format

Instead of using a FORTRAN Format, one can also give the specification on how to read the Data file by a sequence of six numbers. These six numbers form three consecutive pairs of numbers. The first member of each pair indicates where the number is positioned in the record, the second member indicates how many positions the number occupies. For the example of Table 13, these six numbers are

11 2 14 1 21 1

- The first pair (11 2) says that the booklet number starts at position 11 and that it is to be read as a number of two digits (occupying jointly the positions 11 and 12).
- The second pair (14 1) says that the group number starts at position 14 and that it is a one digit number.
- The third pair (21 1) says that the **first** item answer starts at position 21, and that it is a number of one digit. The program will automatically assume then that all item answers are numbers of 1 digit (just like the first item answer), and that all item answers are contiguous in the record. If this is not the case, then one has to use the FORTRAN Format.
- The starting positions of booklet number, group number and first item may or may not be in increasing order. As an example, suppose that in every record, the group number is placed at position 51 and the booklet number at positions 61-62. The six numbers for the format specification would then be 61 2 51 1 21 1.
- If this form is used (as record 10 of the Job Definition File), make sure **not** to include the six numbers in parentheses, as the program will decide on the presence of parentheses whether the FORTRAN Format or the 6 number format has been used.

Restrictions on the data file

The following restrictions hold for the data file:

- The file is a text file having one byte per character, i.e. files saved in unicode format will not be read correctly. Moreover, the program has been developed and tested on a PC using Windows. Probably files saved on a Unix or Mac system will not be read correctly.
- The length of the records in the data file is arbitrary (up to 64K), but all data needed for the program PROFILEG (booklet number, group number and item answers) must be stored in the first 2047 positions.

The <u>Output file</u>

The name of the <u>Output file</u> is specified in record 11 of the Job Definition File. It is a simple text file, which consists of three main parts:

- The first part is an echo of the specifications, and besides this it contains a simple useful table. In this table a summary is given of the total number of data records in the <u>Datafile</u> and a count of the records that have a booklet number or a group number out of range. This part is not discussed here any further.
- Per categorization the results are displayed in four parts. Each of these is discussed further down.
- At the very end of the file, the total computation time and the file creation date and time are displayed.

For the discussion of the output an example will be used with two groups, a categorization with two categories (labelled as ‘cat A’ and ‘cat B’, respectively), and a categorization with three categories (labelled as ‘cat C’, ‘cat D’ and ‘cat E’, respectively). The groups will be indicated as ‘group 1’ and ‘group 2’, respectively.

Part 1: excluded data records. An example is shown in Table 14

Table 14. Example of the first part of the output

```
Number of categories = 2
Category labels: cat A    cat B

    Group    In the analyses  Extreme score  Excl. code   Not applicable   Total
    group 1         459              1           0               0           460
    group 2        2379             22           0               0          2401
    Total          2838             23           0               0          2861
```

In the table, the column labelled 'In the analyses' lists the number of observations which are actually used in the profile analyses. The next three columns list the number of respondents excluded from the analysis for the reasons mentioned in column head. In the example, the variable MINSC has been set to one. This means that in total 23 respondents were excluded from the analysis because they had a zero or a perfect score. The column 'Excl. code' mentions the number of respondents excluded because they had too many exclusion codes (as indicated by the variable MAXEXCL). The column 'Not applicable' is discussed in part 2 of the output.

Notice that the great majority of the respondents belongs to group 2. This will have consequences which will discussed in part 4 of the output

Part 2. Distribution of items across categories. An example is given in Table 15.

Table 15. Distribution of items across categories

```
booklet   cat A   cat B  code zero    Total
    1      15      31        0          46
    2      16      30        0          46
    3      12      34        0          46
    4      19      27        0          46
all items  42      91        0         133
```

The test data have been collected in an incomplete design with four test forms (booklets), each having 46 items. In total there are 133 different items. None of these items has a code zero, meaning that each item has been assigned to one of the two categories. As long as each test booklet has at least one item in each category, the analysis will do fine. But if a test booklet has no items in one or more categories, the profile analysis will either be trivial (2 categories) or the variances will be seriously underestimated. Therefore, the following rule is automatically applied in PROFILEG: **if for a respondent the number of items is zero for one or more categories, this respondent is excluded from the analysis.** The total count of such respondents per group is displayed in part 1 of the output in the column 'Not applicable'.

Profile Analysis and DIF analysis

Profile analysis can be considered as a generalization of DIF analysis, such that DIF analysis is just a special case of Profile Analysis, and, consequently, can be carried out by the program PROFILEG. Suppose one is interested in possible DIF for a particular item, number 10, say. Then one can use Profile Analysis to do a DIF analysis by using two categories and assigning item 10 to one category and all the other items to the other category.
Now suppose that in the example given, item 10 appears only in test forms (booklets) 1 and 4. In this case the distribution table of items across catgeories will look as in Table 16. The two booklets with a zero count in category 'item 10' are marked with a '*', and the number of

records in these booklets are counted as 'Not applicable' and this count will be displayed in the table of part 1.

Table 16. Distribution of items across categories (DIF)

booklet	item 10	others	code zero	Total
1	1	45	0	46
2*	0	46	0	46
3*	0	46	0	46
4	1	45	0	46
all items	1	132	0	133

Counting the excluded respondents

A record with a legal booklet number and a legal group number may be excluded from the analysis for three different reasons: there are too many exclusion codes (controlled by the variable MAXEXCL), the test sore is too extreme (controlled by the variable MINSC) or the number of items in one of the categories is zero (which may happen if all responses to some category are exclusion codes), in which case the reason for exclusion is 'Not applicable'. But it may happen that all three reasons or two of them apply to a particular record in the data file. **In such a case the record is counted only once**: in the program the criteria for exclusion are tested sequentially, and as soon as a criterion (reason) for exclusion is found, the count for that criterion is updated and the remaining criteria are not checked any further. In the checking sequence, applicability is checked first, then the number of exclusion codes and last the extremity of the test score.

Part 3. Contingency tables. These tables are only computed and displayed if the number of categories for the current categorization is two or three. Table 17 gives an example for two categories. For each group 4 cells are displayed, and each cell contains two numbers: the observed frequency (an integer number) and the expected frequency (a fractional number with two decimals). For the definition of trivial and non-trivial deviation profiles, see Section 1. Notice that a label like 'cat A+' denotes a deviation profile with a better than expected performance on 'cat A' items.

Table 17. Example of a contingency table (two categories)

Group	Non-trivial cat A+	Non-trivial cat B+	Trivial cat A+	Trivial cat B+
group 1	129	178	60	92
	144.65	148.63	82.25	83.48
group 2	751	709	455	464
	735.52	734.07	450.54	458.88

If there are three categories, the contingency table has 12 columns, six for trivial profiles and six for non-trivial profiles. In Table 18, part of such a table is shown for a categorization with three categories C, D and E. Notice that the profile type 'cat E–' denotes a deviation profile where the performance is below expectation on 'cat E' and above expectation on the other two categories. More detail can be found in Section 1.

Table 18. Example of a contingency table (three categories)

Group	. . .	Non-trivial cat E-	Trivial cat C+	Trivial cat D+	. . .
group 1	. . .	42	24	49	. . .
	. . .	43.92	27.25	42.51	. . .
group 2	. . .	209	155	196	. . .
	. . .	222.57	157.29	214.31	. . .

Part 4. Mean deviation profiles, standard errors and statistical tests.

If there are only two categories, the results are displayed in a different way than in the case of three or four categories. We discuss both cases in turn.

Case 1. Two categories

An example with three groups of repondents is displayed in Table 19.

Table 19. Example 1 of mean deviation profiles with two categories (more than two groups)

Group	cat A	cat B	St. error	z
group 1	-0.2194	0.2194	0.0652	-3.3626
group 2	0.0209	-0.0209	0.0334	0.6247
group 3	0.1013	-0.1013	0.0546	1.8569

Here are some comments going with Table 19:

- Notice that in this table the distinction between trivial and non-trivial profiles is not made.
- For each group it holds that the mean deviations in the two categories are equal in absolute value and have opposite algebraic signs, because the sum of the deviation averages (across categories) is zero by definition.
- Therefore the two averages have the same standard error. It is desplayed in the table[9].
- The null hypothesis is that the mean deviation is zero in each group. The ratio of the observed deviation divided by its standard error is (by virtue of the central limit theorem) approximately standard normally distributed. If this ratio is larger (in absloute value) than 1.96, the null hypothesis can be rejected at the 5% level of significance. The ‘z’-value in Table 19 is the mean deviation of the first category (cat A) divided by the standard error.

If there are only two groups involved in the analysis, the program also computes the difference of the means of the two groups, reports the standard error of this difference and the z-statistic. An example is given in Table 20.

Table 20. Example 2 of mean deviation profiles with two categories (two groups)

Group	cat A	cat B	St. error	z
group 1	-0.1068	0.1068	0.0434	-2.4623
group 2	0.0220	-0.0220	0.0206	1.0707
Difference	-0.1289	0.1289	0.0480	-2.6836

[9] The variance of each mean deviation is just the square of the standard error, and the covariance is minus the variance. See the first part of this report.

- For both categories the difference of the two means (group 1 minus group 2) is reported. The standard error of the mean difference is the square root of the sum of the two squared standard errors, i.e. $0.0480 = \sqrt{(0.0434^2 + 0.0206^2)}$.
- The same procedure can be applied if there are more than two groups. The program does not report all pairwise differences to avoid large tables in the output. Here is an example taken from Table 19, where the comparison between groups1 and 2 is made. For category A the mean difference is -0.2194 – 0.0209 = -0.2403 and the standard error is $\sqrt{(0.0652^2 + 0.0334^2)} = 0.0733$, yielding a z-statistic of -0.2403/0.0733 = -3.2738
- It may seem remarkable that the mean deviations for group 2 (in Table 20) are smaller in absolute value than the deviations for group 1, and that group 2 is therefore the 'normal' group while group 1 is the 'deviant' group. Such an interpretation, however, is not justified. The reason for this phenomenon is that group 2 is far more numerous (more than four times) than group 1. Since the calibration of the items is done on the data of the two groups jointly, the outcome of the calibration is influenced far more by group 2 than by group 1, and in particular, the mean deviation profile will tend to be closer to zero. If the calibration had been done only on the data of group 2, the the mean deviation for this group would be exactly zero.
- Therefore it is in general better to base judgments on differential functioning of item categories on statistical tests of differences between means than on the means themselves.

Case 2. Three or four categories

An example with three categories is displayed in Table 21.. The number of observations on which these means are based is displayed in part 1 (see Table 14, the column labelled 'In the analyses'.)

Table 21. Example of mean deviation profiles with three categories

Group	cat C	cat D	Cat E
group 1	-0.0997	0.1145	-0.0148
group 2	0.0194	-0.0227	0.0033

Since there are three categories, there are three subscores per respondent and thus there are three means, and a three by three variance-covariance matrix of the mean deviation profile. There is such a matrix for each group, and the program PROFILEG computes them and puts them in the output file. An example, going with the means in Table 21, is given in Table 22.

Table 22. Example of the variance-covariance matrices

	1.9922E-03	-1.4481E-03	-5.4414E-04
group 1	-1.4481E-03	**4.2775E-03**	-2.8294E-03
	-5.4414E-04	-2.8294E-03	**3.3735E-03**
	4.5036E-04	-3.1848E-04	-1.3188E-04
group 2	-3.1848E-04	**8.3036E-04**	-5.1188E-04
	-1.3188E-04	-5.1188E-04	**6.4376E-04**

Notice that the numbers are displayed in scientific notation, to have an equal number (5) of significant digits (i.e., without leading zeros), independent of the sample sizes.

In both matrices, the variances are on the main diagonal (printed in bold face), the off-diagonal elements are the covariances and will in general be negative. Notice that the numbers in the table for group 1 are about five times as large as the numbers in the table for group 2.

The main reason for this difference is the difference in sample size of the two groups[10], group 1 being about five times smaller than group 2.

For all matrices it holds that the sums of rows and columns equals zero. This zero-sum feature reflects the fact that all deviation profiles (also mean profiles) are ipsative: the sum of their elements is zero. This zero-sum feature implies that in case of two categories, all four elements in a matrix have the same absolute value.

The theoretical matrix is computed using formula (1) (see Section 1). It is the exact matrix under the null hypothesis, i.e., the measurement model used in the calibration. It is this matrix that should be used in building statistical tests to judge on differences within and between groups.

This section is concluded with a simple example on how such a test may be constructed. Suppose one is interested in the difference between the two groups with respect to the two categories C and E jointly[11]. The result is given in Table 23.

Table 23. Results derived from Tables 21 and 22

Group	cat C or E	cat D	St. error	z
group 1	-0.1145	0.1145	0.0654	-1.7507
group 2	0.0227	-0.0227	0.0288	0.7878
Difference	-0.1372	0.1372	0.0715	-1.9197

The mean deviations for the joint category is just the sum of the means for categories C and E from Table 21. For the standard errors, we have to remember that the variance of a sum of two variables is the sum of the variances plus two times the covariance. Using the results in Table 22 we find that for group 1 $\sqrt{(1.9922E\text{-}03 + 3.3735E\text{-}03 + 2 * -5.4414E\text{-}04)} = 0.0654$. The standard error of the difference is computed in the same way as was explained in connection with Table 20.

How to install and run the program

The program can be run under two modes: either it is run directly as a command line application or one might use the program WPROFILEG. This latter program is a Windows GUI which gives some help in preparing two of the five files needed for the profile analysis and enables the user to run the program PROFILEG directly from the interface. Both modes are discussed in turn.

PROFILEG as a mere command line application

Installation of the program is trivial: just copy or download the file PROFILEG.exe to some folder on the hard disk. For reference in this manual, it will be assumed that the program file is stored in C:\program files\profile.

---

[10] There are other reasons as well: the distribution of the total scores in each group also has an influence on the variances and covariances.

[11] The results reported in Table 23 are exactly the same as the results from a two category analysis run with PROFILEG. But of course more complicated comparisons are possible which are not readily reproduced by a simple analysis.

To run the program successfully, the user must prepare the five files discussed in this section. For ease of reference, the Job Definition File will be referred to henceforth as JDF.txt (but remember that its actual name is arbitrary). To run the program, the easiest way is to open the command prompt box and to make the folder where the file JDF.txt is stored the default directory. Henceforth it will be assumed that this is the case.

To run the program, there are two possibilities. The first is to type[12]

C:\program files\profile\PROFILEG JDF.txt [CR][13]

and the program will run immediately. The second possibility is to type

C:\program files\profile\PROFILEG [CR]

where upon the program will prompt for the name of the Job Definition File.

If one does not want to open a command prompt box, one can activate the program by double clicking on its name (in Windows Explorer, for example). A command prompt box will be opened automatically and the program will prompt for the name of the Job Definition File.

Using WPROFILEG

This part is not written yet, as the interface is still under construction.

Limitations and Performance Control

The program PROFILEG can be used in a broad number of contexts, and there are virtually no limitations except for the number of categories per categorization and for the number of exclusion codes.

The minimum number of categories is two. With one category the results are trivial: all deviation profiles are zero.

The maximum number of categories that can be specified by the user is unlimited, but one obtains useful results only if the number categories does not exceed four:

- In case of two or three categories, contingency tables as well as mean deviation profiles and their variance-covariance matrices are computed.
- In case of four categories only mean deviation profiles and their variance-covariance matrices are computed.
- In all other cases, the program gives a message that no analysis is carried out.

The maximal number of different exclusion codes that can be specified is five. This will hardly be a practical limitation to the use of the program.

All other variables are for practical purposes unlimited: the number of categorizations, the number of groups, the number of items (in total or per booklet) and the number of observations, be it per group or in total can have any value[14].

---

[12] The full path is required if the system does not find the program automatically via the *path;* otherwise the path may be omitted. The path is not needed if a copy of the program is stored in the default directory.

[13] The string [CR] (carriage return) means that one has to type the Enter key.

[14] Technically, there are limits: for most variables the technical limit is $2^{15} - 1 = 32{,}767$, except for the number of observations, where the technical limit is $2^{31} - 1 = 2{,}147{,}483{,}647$.

As to the performance of the program, i.e., the time needed to do the analyses requested, the user has an important influence. The performance is optimal when the following two conditions are fulfilled:

- Records with the same booklet number are contiguous in the Data file. This can be accomplished simply by sorting this file on booklet number (in ascending or descending order).
- Records with the same pattern of exclusion codes and the same booklet number are contiguous in the Data file. This may be more difficult to accomplish, but clever sorting may give a good approximation to this criterion.

Notice that the program will run correctly, irrespective of the sorting of the data file, but a random order of the records in the data file may result in a computation time up to 100 or more times the time needed with an optimal sorting.