Chapter 15

Cross Classified Models

One of the main uses of multilevel modelling is to account for the underlying structure in a dataset, whether it be pupils nested within schools or women nested within communities as seen in the examples so far. In accounting for the structure we are removing the independence assumption between level one units from the same level two units and instead partitioning the variance into variances between the units at the various levels in the dataset. The examples we have looked at so far have mainly concentrated on two-level structures but we have considered one three level structure (counties within regions within nations) in Chapter 11.

Historically most multilevel modelling has assumed a hierarchical or nested structure for two reasons. Firstly many applications naturally have a nested structure, for example pupils within classes within schools, or patients within wards within hospitals. Secondly the maximum likelihood based methods, for example IGLS, have been designed to work well for nested structures, as fast inversion routines are available for the block diagonal matrices that nested structures produce. However, as we will see in the next three chapters, often the structure of the dataset is not strictly nested. In this chapter we will consider cross-classified models before considering multiple membership models (Chapter 16) and spatial models (Chapter 17).

When cross-classified and multiple membership effects are combined we can produce multiple membership multiple classification (MMMC) models which are described in detail in Browne, Goldstein & Rasbash (2001*a*). Detailed descriptions of likelihood-based methods for both cross-classified models and multiple membership models are given in Rasbash & Goldstein (1994) and Rasbash & Browne (2001), while Rasbash & Browne (2002) compare the likelihood approaches with the MCMC approach that we use here. In this chapter we will describe what we mean by a classification and a cross-classified model before considering an education-based example from Fife, Scotland that is considered in Rasbash et al. (2008, chap. 18).

15.1 Classifications and levels

We have so far concentrated on different 'levels' in a dataset where the definition of a level has not been explicitly given, but we have been assuming a nested relationship between levels. For example in education we may have a three 'level' dataset with our three levels being pupil, class and school. Here pupils are nested within classes and classes are nested within schools. This implies that all pupils in the same class are also in the same school due to the nesting of the levels. The response variable will be at the lowest level in the dataset although predictors may be at the higher levels, for example the effect of class size on individual pupil scores.

Note that if the response was at a higher level than some of the predictors then these predictors could only be fitted in the model as aggregates. For example we may have several previous tests scores for each pupil, which would imply a lower level of time/test below pupil. If our response was exam score at 16 then we would either fit each previous test as a separate predictor or fit an average previous test mark, and so for the model the lowest level is pupil and not test.

In this chapter we will consider the more general definition of a classification. Having defined our lowest level in the data as the level at which the response variable is collected then we can define a *classification* mathematically as a function, c, that maps from the set θ of N lowest level units to a set Φ of size M where $M \leq N$, and we define the resulting set Φ of M objects as the *classification units*. In this chapter we will only consider single membership classifications, $c(n_i) = m_j, \forall n_i \in \theta$ where $m_j \in \Phi$.

In words, if we consider the educational example earlier then our lowest level was **pupil** and the lowest level units are the individual pupils. Both **school** and **class** will then be classifications (functions) that given an individual pupil will return their respective school and class, and so the sets of all schools and all classes will be the classification units associated with the classifications **school** and **class** respectively. Note that as these classifications map directly from the lowest level there is no guarantee that the classifications will be nested, and in fact nested classifications are a special case of the general 'cross-classified' classifications that we consider in this chapter.

MCMC methods treat each set of classification units (residuals in the model) as an additive term in the model and hence it is no more complicated (once the classifications have been calculated) to fit a cross-classified model than a nested model using MCMC. However there is one restriction and that is that we need unique classification identifiers. For example if we truly have a three-level nested model with class 1 in school 1 and class 1 in school 2, then these two classes will need unique identifiers if this model is fitted as a cross-classified model to differentiate between the two class 1s.

15.2 Notation

Browne, Goldstein & Rasbash (2001*a*) introduce notation for fitting cross-classified and more complex models based upon the definition of a classification. Rather than trying to introduce more complex indices that take account of the crossings and nestings (as in Rasbash & Browne, 2001) they instead simply give the response variable subscript i to index lowest level units, and then use the classification names for the subscripts of random effects. For example consider the variance components model described first in Chapter 3. This was written there as:

$$\begin{aligned}\text{normexam}_{ij} &\sim \text{N}(XB, \Omega)\\ \text{normexam}_{ij} &= \beta_{0ij}\text{cons} + \beta_1\text{standlrt}_{ij}\\ \beta_{0ij} &= \beta_0 + u_{0j} + e_{0ij}\end{aligned}$$

In the classification notation we would rewrite this as:

$$\begin{aligned}\text{normexam}_{i} &\sim \text{N}(XB, \Omega)\\ \text{normexam}_{i} &= \beta_{0i}\text{cons}_i + \beta_1\text{standlrt}_{i}\\ \beta_{0i} &= \beta_0 + u^{(2)}_{0,\text{school}(i)} + e_{0i}\end{aligned}$$

As there may be many classifications, rather than using different letters for each, we give a superscript to represent the classification number (note this starts at 2 as we consider the lowest level as classification 1). To change between notations we can use the **Notation** button on the **Equations** window that we earlier used for the alternative complex level 1 notation. We will now consider a cross-classified example from the educational literature.

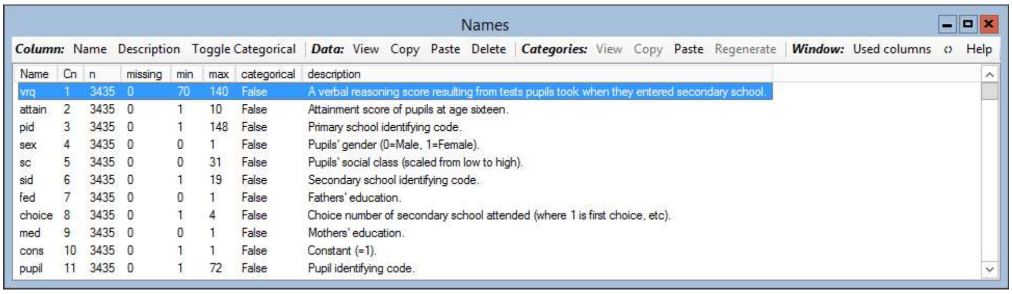
15.3 The Fife educational dataset

We will consider here an educational example from Fife in Scotland that is also considered in the User's Guide to MLwiN (Chapter 18). The data consist of pupils' overall exam attainment at age 16 (as with the **tutorial.ws** dataset studied earlier) and several predictor variables, including a verbal reasoning test taken at age 11 and information on social class and parent's occupation. The added complexity in the dataset is that we have information on both the secondary school (ages 12 through to 16) in which the children studied and the primary school (ages 5 through to 12) they attended prior to secondary school. Not all the children from a particular primary school will attend the same secondary school so we have two classifications that are crossed rather than nested. The data consists of records for 3,435 children from 148 primary schools and 19 secondary schools.

First we will load the dataset and look at the variable names:

- Select **Open Sample Worksheet** from the **File** menu.
- Select **xc1.ws** from the list and click on the **Open** button.

The **Names** window will appear as follows:

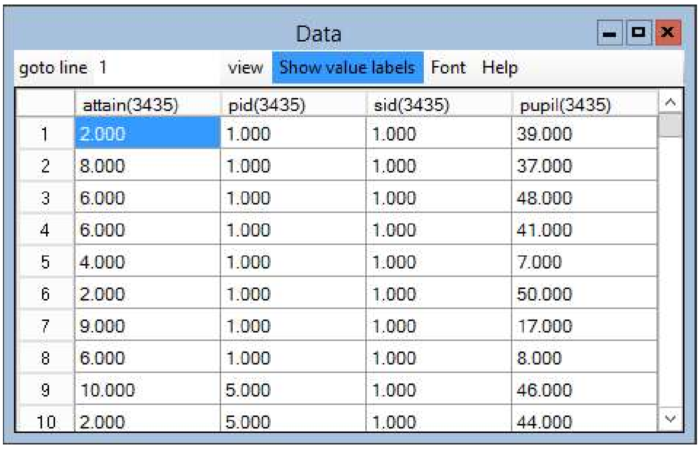


Name	Cn	n	missing	min	max	categorical	description
vrq	1	3435	0	70	140	False	A verbal reasoning score resulting from tests pupils took when they entered secondary school.
attain	2	3435	0	1	10	False	Attainment score of pupils at age sixteen.
pid	3	3435	0	1	148	False	Primary school identifying code.
sex	4	3435	0	0	1	False	Pupils' gender (0=Male, 1=Female).
sc	5	3435	0	0	31	False	Pupils' social class (scaled from low to high).
sid	6	3435	0	1	19	False	Secondary school identifying code.
fed	7	3435	0	0	1	False	Fathers' education.
choice	8	3435	0	1	4	False	Choice number of secondary school attended (where 1 is first choice, etc).
med	9	3435	0	0	1	False	Mothers' education.
cons	10	3435	0	1	1	False	Constant (=1).
pupil	11	3435	0	1	72	False	Pupil identifying code.

We here see that our response variable (**attain**) is a score from 1 to 10 that represents the pupils score on a school leaving exam. The intake ability is measured by a score in a verbal reasoning test, (**vrq**) and we also have predictors that represent gender (**sex**), social class (**sc**), father's education (**fed**), mother's education (**med**) and the choice of secondary school that they attend (**choice** where 1 is first choice and so on).

We can look at the dataset more closely by:

- Select **View or Edit Data** from the **Data Manipulation** menu.
- Select to view columns **attain**, **pid**, **sid** and **pupil**.



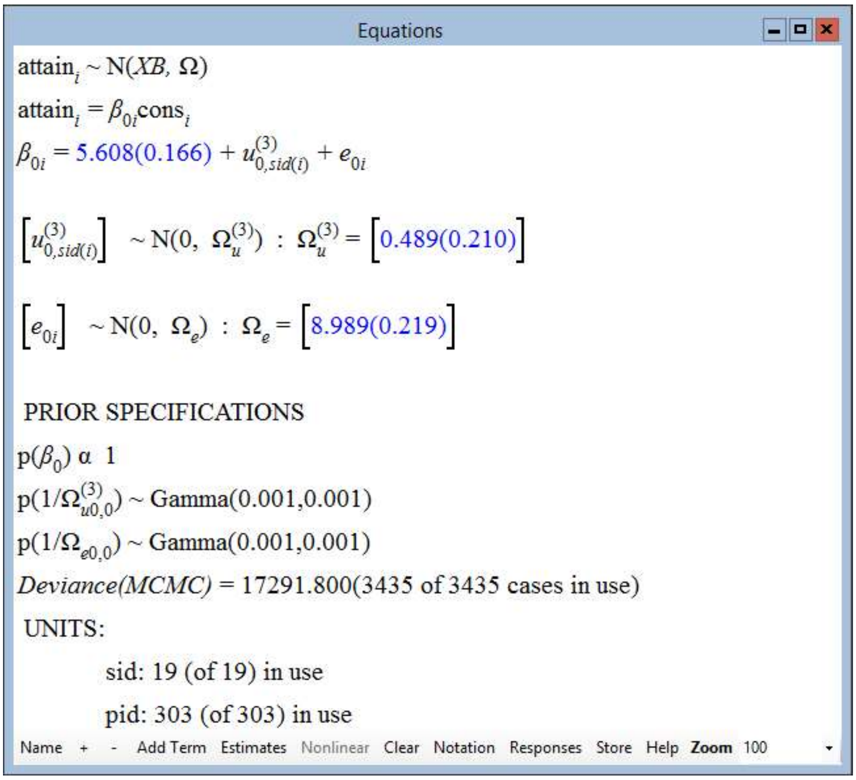
	attain(3435)	pid(3435)	sid(3435)	pupil(3435)
1	2.000	1.000	1.000	39.000
2	8.000	1.000	1.000	37.000
3	6.000	1.000	1.000	48.000
4	6.000	1.000	1.000	41.000
5	4.000	1.000	1.000	7.000
6	2.000	1.000	1.000	50.000
7	9.000	1.000	1.000	17.000
8	6.000	1.000	1.000	8.000
9	10.000	5.000	1.000	46.000
10	2.000	5.000	1.000	44.000

The data have been sorted on primary school within secondary school. We can see here that 8 of the pupils who attended primary school 1 then attended secondary school 1. If we were to scan down the columns we would find that the rest of primary school 1 went to two other secondary schools, 45 to secondary school 9 and 1 to secondary school 18 (to see this quickly type 1355 or 3068 into the **goto line** box and this will take you to these groups of pupils). So we can see that school 9 is the 'main' secondary school for primary school 1 with 83% of pupils attending it. In the entire dataset 59 of the 148 primary schools had all their pupils attend the same secondary school after leaving primary school and only 288 pupils did not attend their 'main' secondary school. So although the dataset structure is not nested it is close to nested and this helps the likelihood-based methods in the User's Guide to MLwiN (see Rasbash & Goldstein, 1994, for details). The degree of 'nestedness' does not matter so much to the MCMC methods and in fact it is probably easier to distinguish between two classifications if they are less nested!

As the data are sorted on secondary schools and their effects will have happened closer (in time) to the exam response of interest we will first consider fitting a two-level model of children within secondary school. We will however use the classification notation from the start and define the three-classification structure of the data.

- Select **Equations** from the **Model** menu.
- Click on the **Notation** button and remove the tick for multiple subscripts and an i subscript will appear on the red y.
- Click on the **Done** button.
- Click on the red y and select **ATTAIN** from the y pull down list.
- Select 3 from the **N classifications** box.
- Select **sid** as classification 3, **PID** as classification 2 and **PUPIL** as classification 1.
- Click on the **Done** button.
- Click on the red x_0 and select **CONS** from the pull down list.
- Select **cons** as a fixed effect and random at classifications **pupil(1)** and **sid(3)**.
- Click on the **Done** button.
- Click on the **Start** button.
- Change **Estimation method** to **MCMC**.
- Click on the **Start** button.

This will have set up the 2 level variance components model and run it using MCMC. The estimates in the **Equations** window will be as follows:



Here we see that there is significant variation between the secondary schools and this accounts for 0.489/(0.489+8.989)×100% = 5.1% of the total variation in exam marks.

We can compare the DIC for this model with a simpler model with no school effects, and we see a reduction in DIC of 120 showing this is a much better model. Also the 19 secondary school effects account for $18.2 - 2 = 16.2$ effective parameters so there are distinct differences between secondary schools.

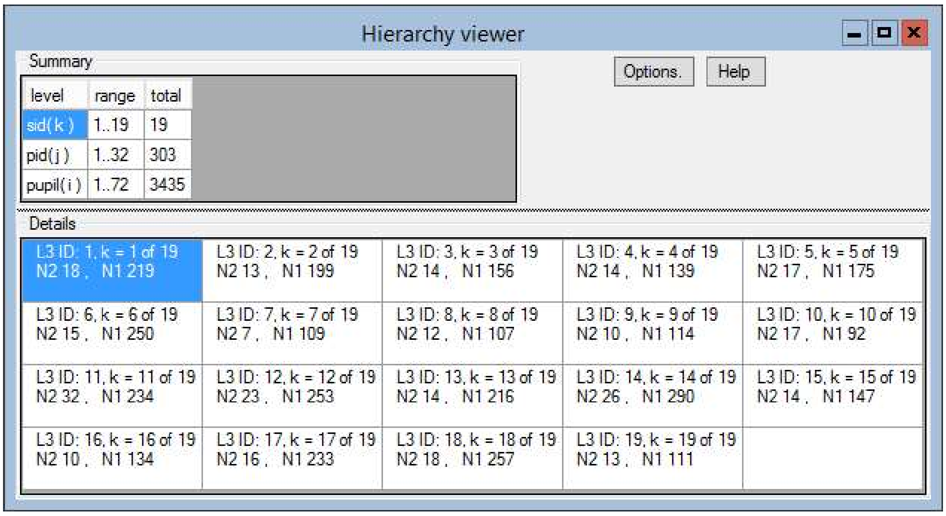
Dbar	D(thetabar)	pD	DIC	
17291.80	17273.61	18.19	17309.99	
17429.27	17427.26	2.01	17431.28	(no school effect)

15.4 A Cross-classified model

If we now consider adding in the effects of primary schools this can be done simply via the **Equations** window.

- Change **Estimation method** to **IGLS**.
- Click on x_0 (**cons**) and tick the **PID(2)** box.
- Click on the **Start** button.

What you have actually just done is fitted a ‘nested’ model of primary school nested within secondary school using IGLS. This can be confirmed by looking at the **Hierarchy viewer** available via the **Model** menu.

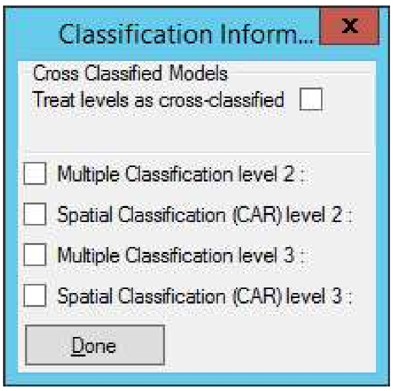


Here you can see that MLwiN has treated the individual groups of pupils that are from the same primary school and secondary school as separate primary schools, for example the pupils in primary school 1 are treated as three separate primary schools nested within secondary schools 1, 9 and 18 respectively. This results in 303 rather than 148 primary schools. To fit a cross-classified model in IGLS instead involves following the procedures given in Chapter 18 of the User’s Guide to MLwiN.

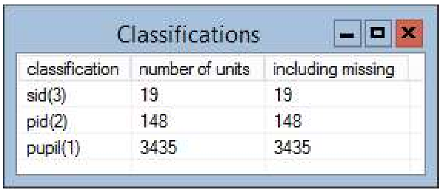
To fit the model (as cross-classified) using MCMC is however fairly simple.

- Change **Estimation method** to **MCMC**.
- Select **MCMC/Classifications** from the **Model** menu.

The window will appear as follows:

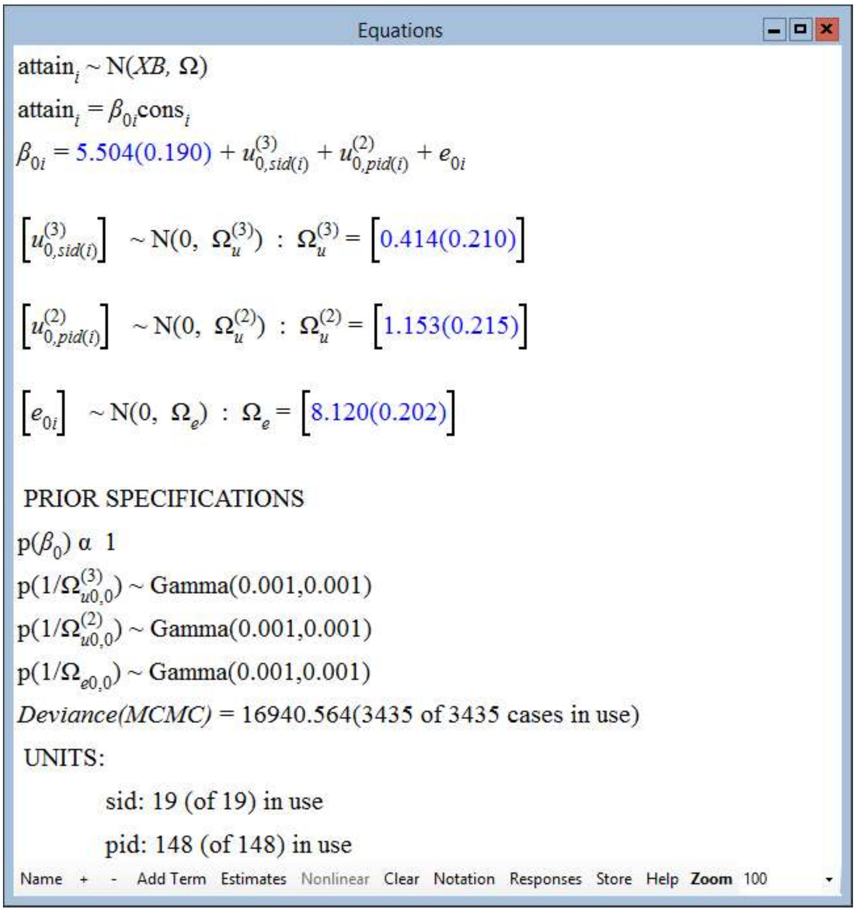


Here we now simply have to click in the **Treat levels as cross-classified** box and click on the **Done** button. If we now select the **Hierarchy Viewer** from the **Model** menu we get the alternative classifications viewer as shown below.

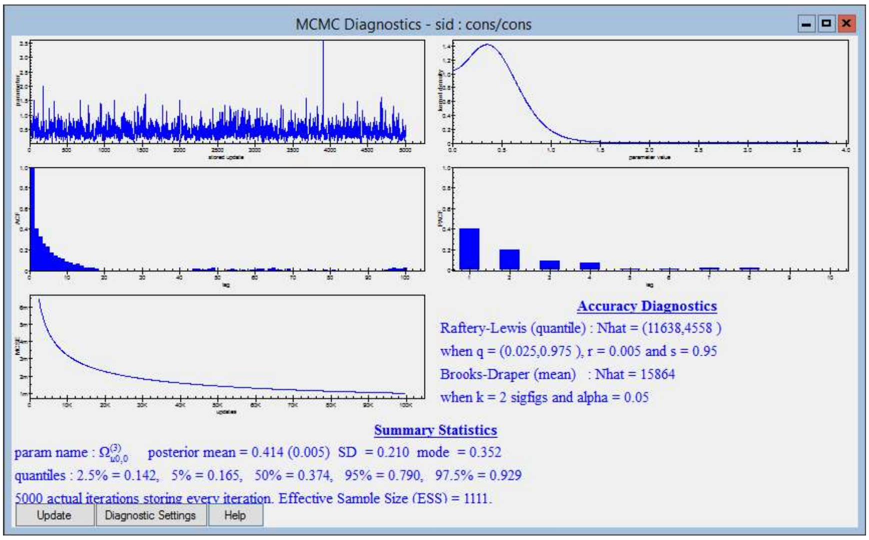


classification	number of units	including missing
sid(3)	19	19
pid(2)	148	148
pupil(1)	3435	3435

Here we see that this viewer shows we have only 148 primary schools as we are now taking account of the cross-classifications. After running the model by clicking on the **Start** button we will get the following estimates:



The estimates are fairly similar to those achieved using IGLS in the User's Guide to MLwiN although the variances for primary school (1.15 versus 1.12) and particularly secondary school (0.41 versus 0.35) are higher. This is due to the difference between mean estimates and mode (ML) estimates for the skewed variance parameter posterior distributions. The trajectory plots confirm this for the secondary school variance:



We can also see that primary school is actually more important in predicting the attainment score than secondary school. One possible reason for this is that secondary schools are generally larger (see Goldstein, 2003). Here primary school explains $1.15/(0.41+1.15+8.12)\times100\% = 11.9\%$ of variation while secondary school only explains $0.41/(0.41+1.15+8.12)\times100\% = 4.2\%$. The DIC diagnostic again shows that this model is an improvement with a reduction in DIC of over 250.

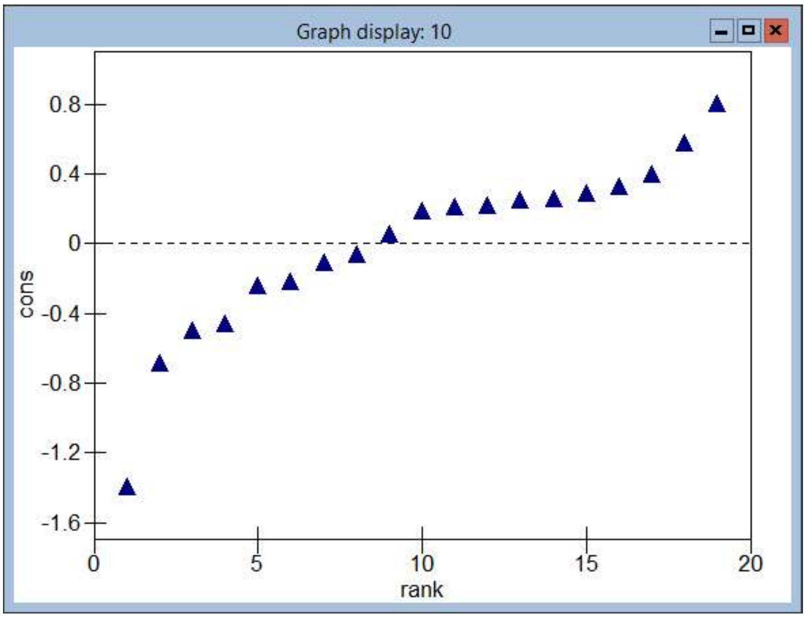
Dbar	D(thetabar)	pD	DIC	
16940.56	16833.40	107.16	17047.73	(with primary school)
17291.80	17273.61	18.19	17309.99	(without primary school)

15.5 Residuals

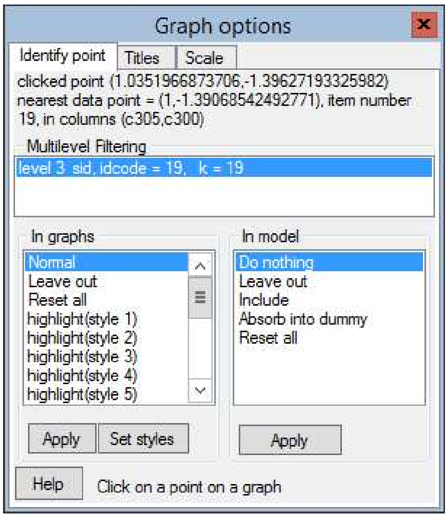
As with nested models we can work out residuals for the various levels of our model. This may be done via the **Residuals** window available from the **Model** menu. We will look firstly at secondary school residuals:

- On the **Residuals** window, change the **level** box to **3:SID**.
- Click on the **Calc** button.
- Click on the **Plots** tab, and if not selected, select **residual x rank**.
- Click on the **Apply** button.

The plot will then appear as follows:

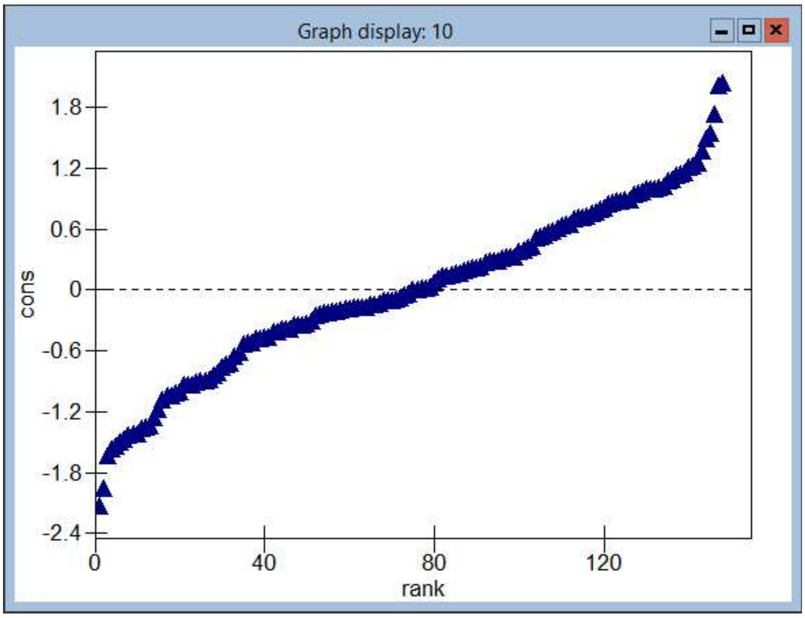


Here we see the lowest ranked secondary school has a very low residual and may be an outlier. Clicking on the graph on this point we get:

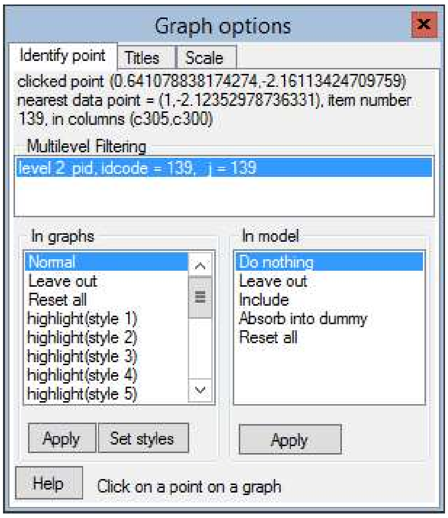


showing that this is secondary school 19. We will revisit this plot after adding in other variables. If we now look instead at the primary schools:

- On the **Residuals** window, click on the **Settings** tab.
- Change the **level** box to **2:PID**.
- Click on the **Calc** button.
- Click on the **Plots** tab, and if not selected select **residual x rank**.
- Click on the **Apply** button.



Here we see the 148 primary school residuals. Here there is no evidence of outliers. If we click on the lowest residual (rank 1) we get the following:



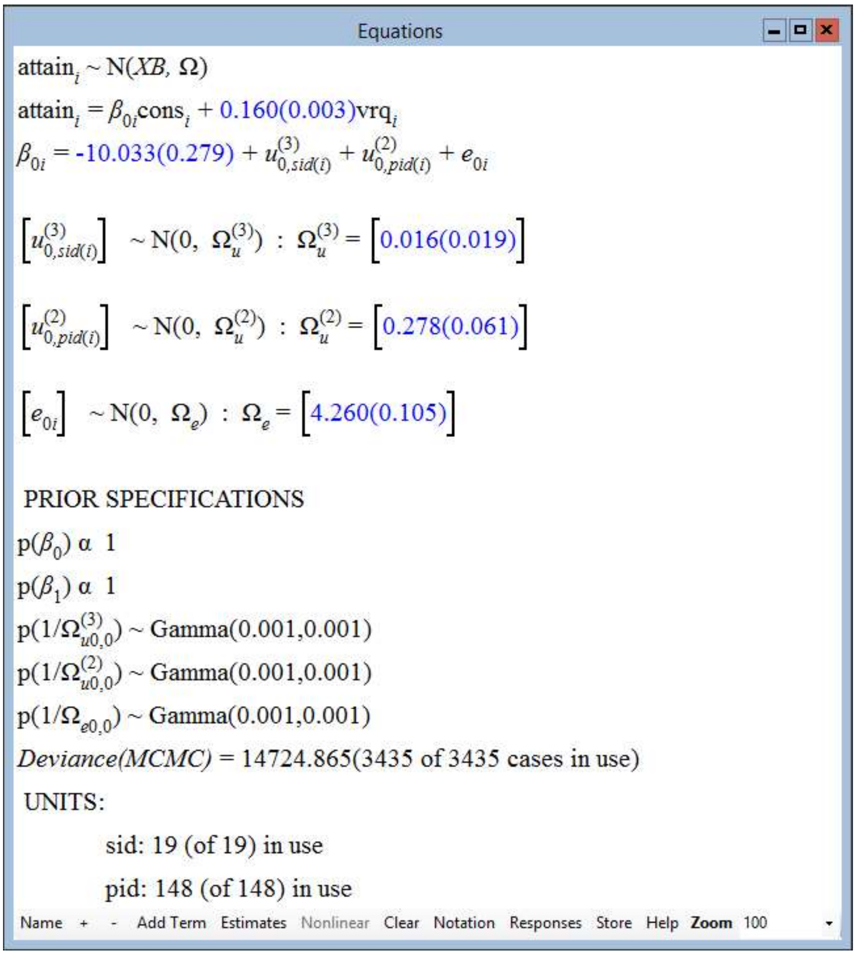
So we see here that the lowest ranked primary school is school number 139 and that even though the data are not nested the residuals screen can identify correctly the primary school. Note however that unlike nested models we do not get a level 3 identifier as primary school is not nested within secondary school.

15.6 Adding predictors to the model

We have so far not considered any of the available predictors in our model. We will firstly consider the effect of intake score (**VRQ**) in our model.

- Change **Estimation method** to **IGLS**.
- Click on the **Add Term** button and select **VRQ** from the **variable** list.
- Click on the **Done** button.
- Click on the **Start** button.
- Change **Estimation method** to **MCMC**.
- Click on the **Start** button.

The estimates produced are as follows:



The predictor, **vrq**, explains not only a large amount of the residual variation but also a large amount of the differences between secondary schools and between primary schools. Of the remaining variation, 6% is explained by primary schools and less than 0.4% by secondary schools. The DIC diagnostic gives:

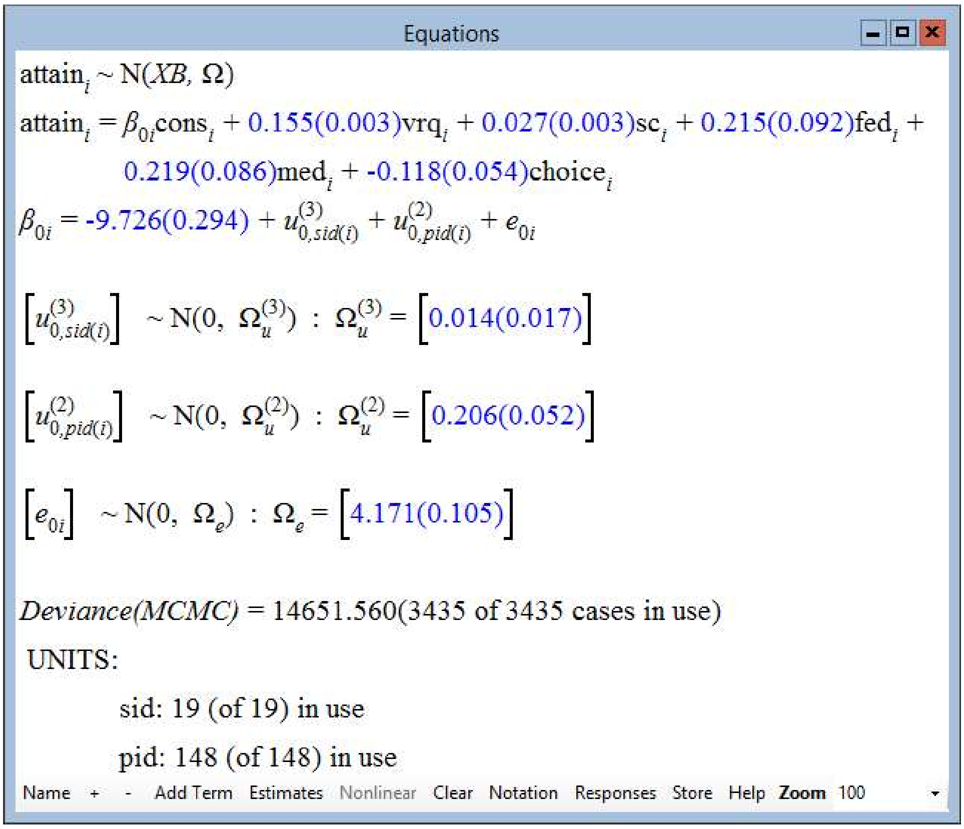
Dbar	D(thetabar)	pD	DIC	
14724.86	14644.21	80.66	14805.52	(with vrq)
16940.56	16833.40	107.16	17047.73	(without vrq)

which shows a reduction in DIC of over 2000! It is also interesting that the effective number of parameters is reduced and this is clearly because VRQ is explaining many of the differences between secondary schools and between primary schools.

We can continue adding in the other predictor variables and retaining significant predictors. In this case all predictors tested apart from gender (**SEX**) are significant. The model with all significant predictors can be obtained by:

- Change **Estimation method** to **IGLS**.
- Add the variables **SC**, **FED**, **MED** and **CHOICE** as fixed effects to the model.
- Click on the **Start** button.
- Change **Estimation method** to **MCMC**.
- Click on the **Start** button.

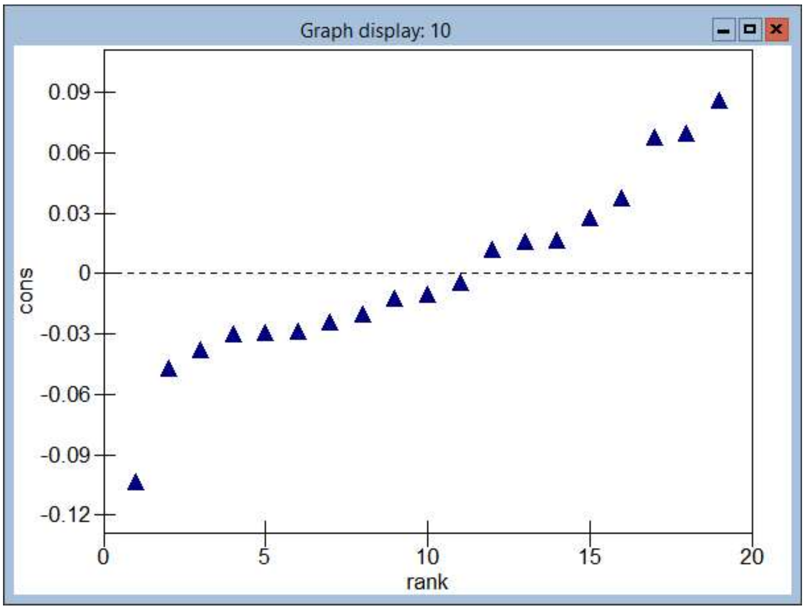
When the 5,000 iterations have been run we get the following estimates:



Here we see that on average a pupil's attainment is higher if they come from a higher social class, if their parents are better educated or if the school they attend is their first choice. Adding the additional predictors has the effect of reducing the DIC diagnostic by 80 and again reducing the effective number of parameters slightly, suggesting more of the differences between schools have been explained by the additional predictors.

Dbar	D(thetabar)	pD	DIC	
14651.56	14575.02	76.54	14728.10	
14724.86	14644.21	80.66	14805.52	(without additional predictors)

The secondary school variance is very small and if we now look at the residuals plot of the school residuals against rank (see instructions earlier on how to produce this) we see that the residual for school 19 is still lowest and looks like an outlier. (Note that a number of error messages may crop up during the estimation here. It is safe to ignore them by clicking the **OK** button.)



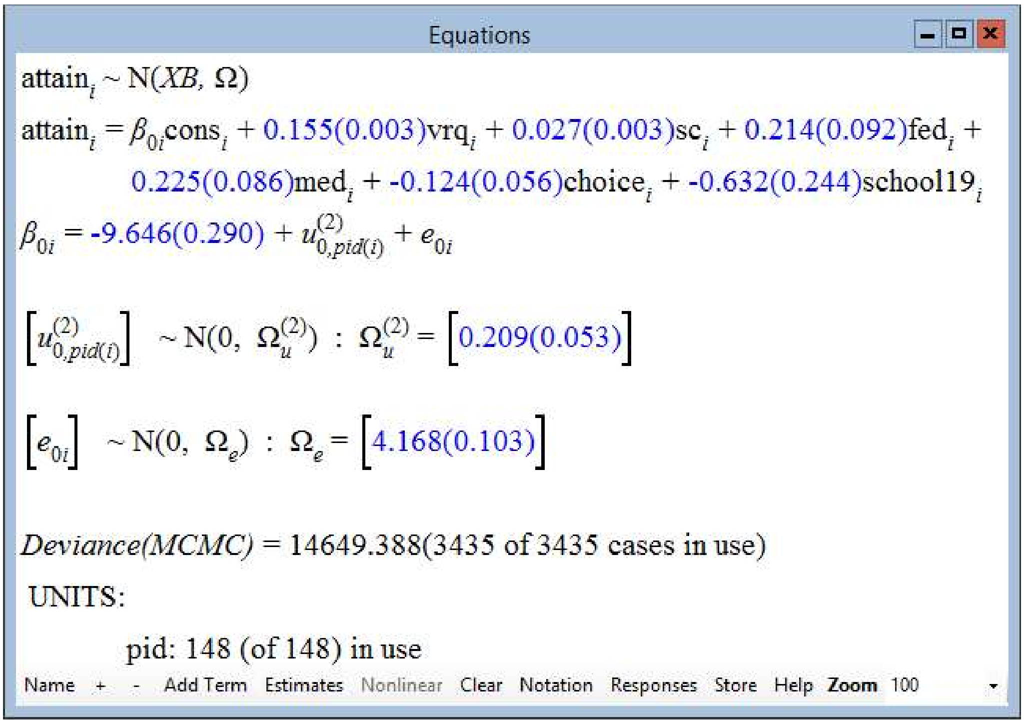
We will therefore consider fitting a dummy variable for school 19 and removing secondary school from the model.

- Select **Command Interface** from the **Data Manipulation** menu and enter the following commands:

```
▶ calc c12 = 'sid' == 19
▶ name c12 'school19'
```

- Change **Estimation method** to **IGLS**.
- Click on the β_0 (**cons**) and remove the tick for **sid(3)**.
- Click on the **Add Term** button and select **school19** from the list.
- Click on the **Done** button.
- Click on the **Start** button.
- Change **Estimation method** to **MCMC**.
- Click on the **Start** button.

After the 5,000 iterations have completed our estimates are as follows:



We can see that school 19 has a significant negative effect on attainment and if we look at the DIC diagnostic we see an improvement in DIC diagnostic of 3.4.

Dbar	D(thetabar)	pD	DIC	
14649.39	14574.03	75.36	14724.74	(with secondary school 19 only)
14651.56	14575.02	76.54	14728.10	(with all secondary school effects)

So in adding the predictors to our model we have explained all the secondary school variation down to a difference between school 19 and the rest of the secondary schools. This of course means that, for the Fife dataset, we now no longer need to fit a cross-classified model. Therefore if we were to re-sort the data on primary school we could have fitted the final model directly using IGLS or MCMC. Some people may think this is disappointing but with only 19 secondary schools to start with it is unlikely that we will find much variation and in fact we now have a more parsimonious model. It may be interesting for the researchers to now go and investigate why school 19 was a potential outlier.

15.7 Current restrictions for cross-classified models

As has been shown in this chapter it is now possible to quite easily fit cross-classified models in MLwiN using MCMC, although not all features have been updated to account for these models. For example currently the **Predictions** window does not account for cross-classified random effects and will therefore give error messages if it is used. It should also be noted that the starting values that MCMC gets for the residuals will be based on the values obtained from the nested model and so will often be meaningless. It is possible by running the MCMC and other commands in the **Command interface** window to fit the separate IGLS two-level models and store these residuals in columns to be used as starting values, but generally the MCMC routines are robust to the nested model starting values. Currently cross-classified models can be fitted using IGLS, but only via additional commands that transform the cross-classified model into a constrained nested model.

Chapter learning outcomes

- ⋆ What is meant by a classification and a cross-classified model.
- ⋆ How to fit cross-classified models in MLwiN using MCMC.
- ⋆ How to look at residuals in a cross-classified model.
- ⋆ Some of the current restrictions in fitting cross-classified models in MLwiN.