# Modelling ethnic segregation using MLwiN

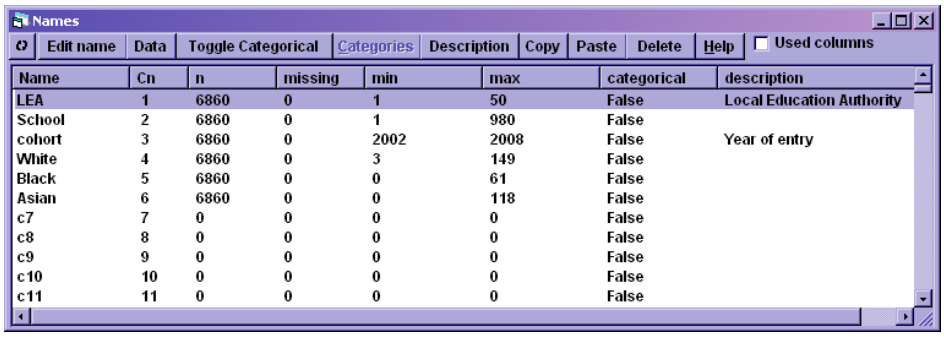
Rebecca Pillinger Centre for Multilevel Modelling

*Note that this practical is accompanied by worksheets* ***segregation.wsz*** *and* ***segregation2.wsz****. These worksheets will only work with MLwiN version 2.10 (not version 2.02). They will work with the free training version of MLwiN 2.10.*

## 1 Examining the data

This practical session uses simulated data. Obviously it would be most interesting to use real data, but there are several reasons why this is not practical. To use real data on school composition, such as the Pupil Level Annual School Census (PLASC) produced by the DCSF, it is generally necessary to complete a registration process and give details of the research that the data will be used for, and to promise not to pass the data on to anyone else. Real datasets also tend to be very large, which makes estimation slow (and it is not desirable to spend the whole practical session waiting for the first model to run). Finally, the levels of segregation used to create the simulated datasets have been chosen to be less extreme than the values estimated from real data. This also improves the running of the models, for example because it is not necessary to run for so many iterations.

- Open the worksheet **segregation.wsz**




| Name | Cn | n | missing | min | max | categorical | description |
|---|---|---|---|---|---|---|---|
| LEA | 1 | 6860 | 0 | 1 | 50 | False | Local Education Authority |
| School | 2 | 6860 | 0 | 1 | 980 | False | |
| cohort | 3 | 6860 | 0 | 2002 | 2008 | False | Year of entry |
| White | 4 | 6860 | 0 | 3 | 149 | False | |
| Black | 5 | 6860 | 0 | 0 | 61 | False | |
| Asian | 6 | 6860 | 0 | 0 | 118 | False | |
| c7 | 7 | 0 | 0 | 0 | 0 | False | |
| c8 | 8 | 0 | 0 | 0 | 0 | False | |
| c9 | 9 | 0 | 0 | 0 | 0 | False | |
| c10 | 10 | 0 | 0 | 0 | 0 | False | |
| c11 | 11 | 0 | 0 | 0 | 0 | False | |

The **Names** window appears showing a list of the variables in the worksheet. An MLwiN worksheet consists of a number of columns, 1500 by default. In this worksheet, the first 6 columns are our

variables and contain data, while the remaining 1494 columns are empty. Note that it is perfectly acceptable to have columns of differing lengths in an MLwiN worksheet, and to have empty columns in between columns containing data.

Our dataset has one row per cohort per school. The six variables consist of three unit identifiers, **Cohort**, **School** and **LEA**, which specify which cohort, school and Local Education Authority each row refers to, and three columns **White**, **Black** and **Asian** giving the number of students belonging to each ethnic group in each cohort in each school. The **Names** window displays some information about our variables: from left to right, the name of the variable, the column of the worksheet that it occupies, the number of observations (the length of the variable), the number of missing values, the minimum value, the maximum value, whether the variable is categorical (True) or continuous (False), and in some cases some written information about the variable. We can see that we have 6860 observations in each of our variables (6860 rows in our dataset) with no missing values, and that all our variables are defined as continuous. Looking at the maximum and minimum values, we can see that we have information for years (cohorts) between 2002 and 2008, and some cohorts have no Black students, some have no Asian students, but there are no cohorts without any White students.

We can have a look at the actual data:

- In the **Names** window, highlight the first six columns (**LEA** to **Asian**) by clicking and dragging, using shift + click, or using ctrl + click
- Press the **Data** button at the top of the **Names** window

Data

goto line 1 | view | Help | Font | Show value labels

| | LEA( 6860) | School( 6860) | cohort( 6860) | White( 6860) | Black( 6860) | Asian( 6860) |
|---|---|---|---|---|---|---|
| 1 | 1.000 | 1.000 | 2002.000 | 62.000 | 25.000 | 42.000 |
| 2 | 1.000 | 1.000 | 2003.000 | 80.000 | 22.000 | 41.000 |
| 3 | 1.000 | 1.000 | 2004.000 | 80.000 | 14.000 | 34.000 |
| 4 | 1.000 | 1.000 | 2005.000 | 94.000 | 12.000 | 29.000 |
| 5 | 1.000 | 1.000 | 2006.000 | 96.000 | 17.000 | 28.000 |
| 6 | 1.000 | 1.000 | 2007.000 | 98.000 | 19.000 | 26.000 |
| 7 | 1.000 | 1.000 | 2008.000 | 94.000 | 12.000 | 28.000 |
| 8 | 1.000 | 2.000 | 2002.000 | 50.000 | 16.000 | 27.000 |
| 9 | 1.000 | 2.000 | 2003.000 | 60.000 | 15.000 | 29.000 |
| 10 | 1.000 | 2.000 | 2004.000 | 78.000 | 27.000 | 40.000 |
| 11 | 1.000 | 2.000 | 2005.000 | 70.000 | 10.000 | 25.000 |
| 12 | 1.000 | 2.000 | 2006.000 | 75.000 | 23.000 | 34.000 |
| 13 | 1.000 | 2.000 | 2007.000 | 67.000 | 13.000 | 30.000 |

We can see that the first row of the dataset contains information for the cohort entering School 1 in LEA 1 in 2002. In this cohort there are 62 White students, 25 Black students and 42 Asian students.

### Exercise 1

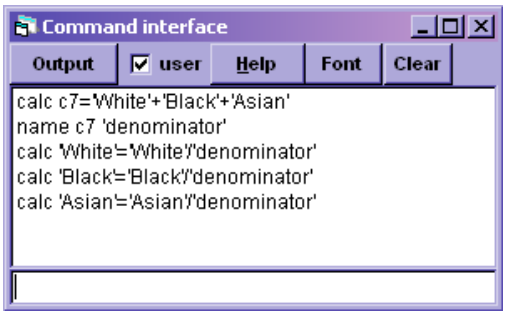
How many White, Black and Asian students are there in the 2005 cohort in School 29 in LEA 2?

## 2 Preparing the data

To specify our model correctly, we will need to use the proportions in each category in each cohort, not the actual numbers. We will also need a variable which gives the total number of students in each cohort, which we will call **denominator**. We will create these now.

- From the **Data manipulation** menu, select **Command interface**
- In the box at the bottom of the window that appears, type

```
► calc c7 = 'White'+'Black'+'Asian'
► name c7 'denominator'
► calc 'White' = 'White' / 'denominator'
► calc 'Black' = 'Black' / 'denominator'
► calc 'Asian' = 'Asian' / 'denominator'
```




If we look at the **Names** window again, we see that we now have a new variable, **denominator**. We also see that our variables **White**, **Black** and **Asian** have changed. These variables now take values between 0 and 1, since they are now the proportions in each category instead of the actual numbers. We can see that the maximum value of **White** is 1, so there is at least one cohort that has only White students, but the maximum values for **Black** and **Asian** are 0.59 and 0.94, so there are no cohorts which have only Black or only Asian students.

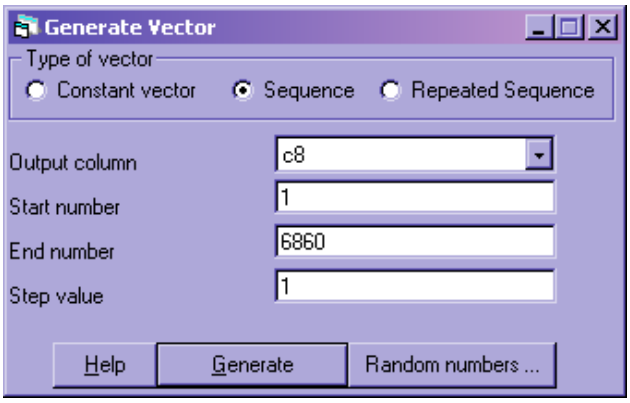
### Exercise 2

We saw in the previous section that there are no cohorts without any White students. What is the smallest proportion of White students in a cohort that occurs in the dataset?

Our data is now in the correct format, but we will need three more variables to properly specify the model. One variable which we will call **cell** will simply be a sequence from 1 to 6860 and will give each row of the dataset a unique ID- we will use this as our level 1 ID instead of **Cohort** since

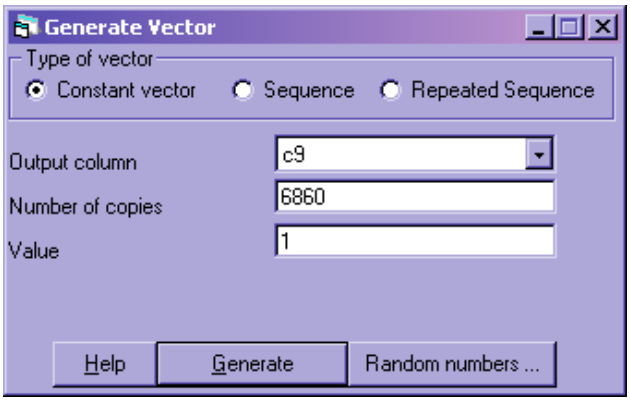
**Cohort** does not have a different value for every row of the dataset[1]. The second variable takes the value 1 for every row of the dataset. We need this variable to specify an intercept because MLwiN requires every coefficient to have an associated variable. (Using a variable which always has the value 1 is mathematically equivalent to including a coefficient with no associated variable.) We will call this variable **cons**. Finally, because later we will be fitting a multinomial model to proportions data, we are going to need a variable which will stand in for the response while we trick MLwiN into expanding the dataset, and which we can then overwrite with our proportions. (We will go into more detail as to why we need to do this in the section where we set up the multinomial model). This variable needs to be a categorical variable with the same number of categories that we have proportions for, and with category names that match those categories. We will call this variable **dummyresp**

- From the **Data manipulation** menu select **Generate vector**
- Under **Type of vector**, select **Sequence**
- Next to **Output column**, select **c8**
- Next to **Start number**, type **1**
- Next to **End number**, type **6860**
- Next to **Step value**, type **1**
- Click **Generate**

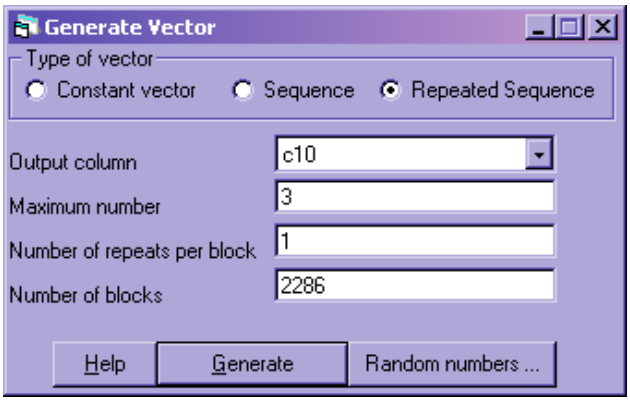



- Under **Type of vector**, select **Constant vector**
- Next to **Output column**, select **c9**
- Next to **Number of copies**, type **6860**
- Next to **Value**, type **1**
- Click **Generate**

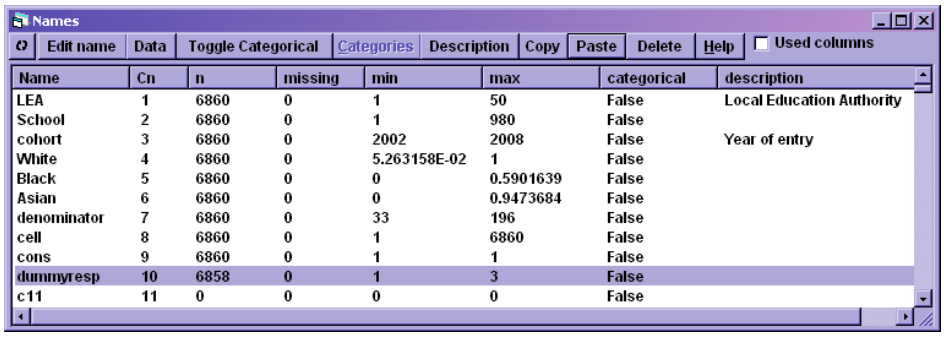
[1]This is because we will be using a multinomial model later on– if we had a continuous response model it would be fine to use **Cohort** as the level 1 ID




- Under **Type of vector**, select **Repeated Sequence**
- Next to **Output column**, select **c10**
- Next to **Maximum number**, type **3**
- Next to **Number of repeats per block**, type **1**
- Next to **Number of blocks**, type **2286**
- Click **Generate**



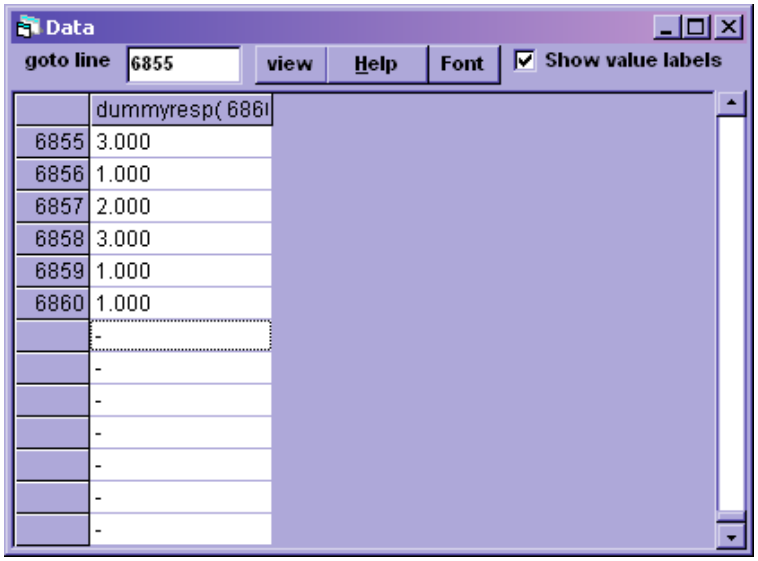

- In the **Names** window, name c8 **cell**, c9 **cons** and c10 **dummyresp** (highlight the variable and click the **Edit name** button at the top of the window, or alternatively double click on the variable)




| Name | Cn | n | missing | min | max | categorical | description |
|---|---|---|---|---|---|---|---|
| LEA | 1 | 6860 | 0 | 1 | 50 | False | Local Education Authority |
| School | 2 | 6860 | 0 | 1 | 980 | False | |
| cohort | 3 | 6860 | 0 | 2002 | 2008 | False | Year of entry |
| White | 4 | 6860 | 0 | 5.263158E-02 | 1 | False | |
| Black | 5 | 6860 | 0 | 0 | 0.5901639 | False | |
| Asian | 6 | 6860 | 0 | 0 | 0.9473684 | False | |
| denominator | 7 | 6860 | 0 | 33 | 196 | False | |
| cell | 8 | 6860 | 0 | 1 | 6860 | False | |
| cons | 9 | 6860 | 0 | 1 | 1 | False | |
| dummyresp | 10 | 6858 | 0 | 1 | 3 | False | |
| c11 | 11 | 0 | 0 | 0 | 0 | False | |

We have now created our variables **cell** and **cons** but we still have a little more to do on **dummyresp**. **dummyresp** is currently $1, 2, 3, 1, 2, 3, 1, 2, 3, \ldots$, which is what we wanted: it has 3 values and we have 3 ethnic categories making up our data, **White**, **Black**, and **Asian**. However, looking in the names window we see that it is not a categorical variable, and it is only 6858 rows long whereas our dataset is 6860 rows long. This is because we constructed it to be 2286 (= 6858/3) lots of $1, 2, 3$ (we couldn't construct it to be 6860/3 lots of $1, 2, 3$ because 6860 does not divide by 3)[2]. We can lengthen **dummyresp** to 6860 observations:

- In the **Names** window, highlight **dummyresp**
- Click the **Data** button
- In the **Data** window, type **6855** in the **goto line** box at the top
- Click on the box next to **6859**, type **1** and press return
- Repeat for the box next to **6860**

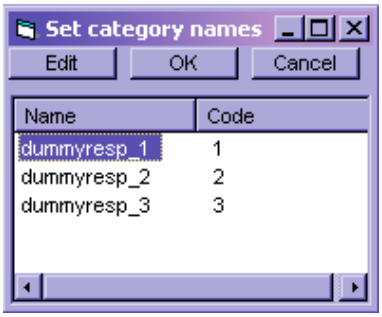



| | dummyresp( 6861 |
|---|---|
| 6855 | 3.000 |
| 6856 | 1.000 |
| 6857 | 2.000 |
| 6858 | 3.000 |
| 6859 | 1.000 |
| 6860 | 1.000 |
| | - |
| | - |
| | - |
| | - |
| | - |
| | - |
| | - |

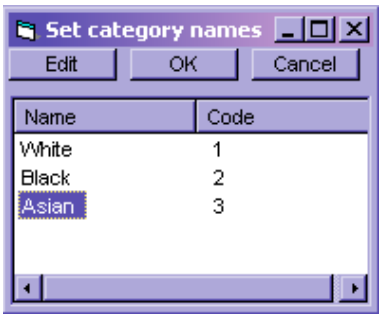
[2] **dummyresp** needs to have at least one 1, at least one 2 and at least one 3, and it must have no other numbers. It doesn't matter how many 1s, 2s and 3s or in what order, but this repeated sequence is the quickest way to generate a variable satisfying these conditions.

If we return to the **Names** window we see that **dummyresp** now has the correct number of observations (you may need to press the refresh button at the top left of the **Names** window to see this). We now need to declare it as categorical and assign it appropriate category names:

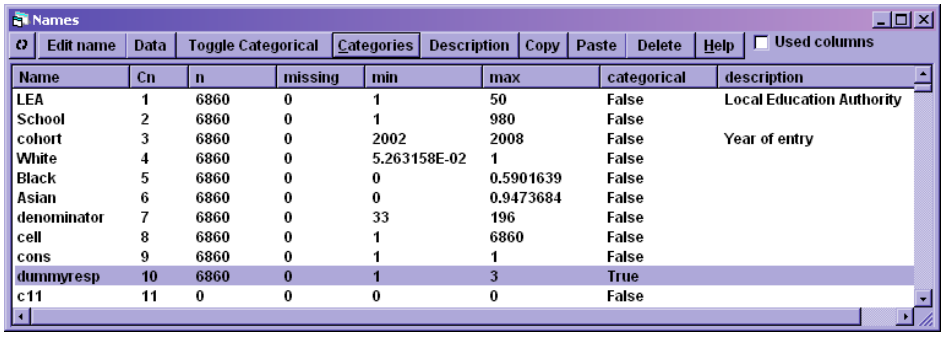
- With **dummyresp** highlighted, click the **Toggle Categorical** button
- Click the **Categories** button




- Using the **Edit** button at the top of the **Set category names** window or by clicking twice on each category name, change the name of category 1 to be **White**, the name of category 2 to be **Black**, and the name of category 3 to be **Asian**
- Click **OK**




The **Names** window now shows that **dummyresp** is of the correct length, and categorical as required.



# 3 Fitting a binomial model

*Note that the worksheet* ***segregation2*** *has had all the operations described in Section 2 performed, so you can carry out the analysis in this chapter by opening* ***segregation2*** *if you have not saved a copy of the work done in Section 2*

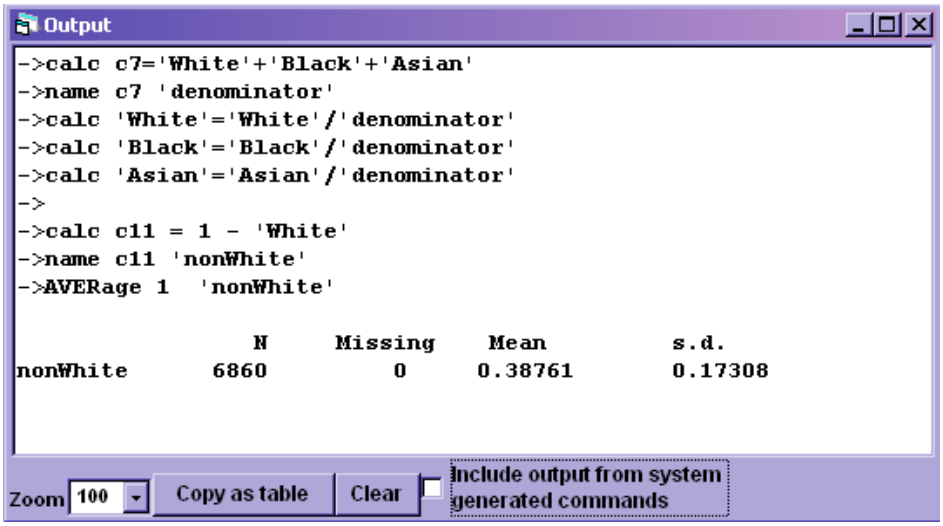
Later on, we will fit a multinomial model which allows us to model Black segregation from White and Asian segregation from White simultaneously. However, we will start by fitting a binomial model for which the response is the proportion of non-White students in the cohort. First we create this response:

- In the **Command interface** window, type

```
► calc c11 = 1 - 'White'
► name c11 'nonWhite'
```

We can find out what the average proportion of non-White students is:

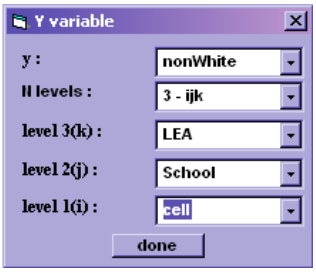
- From the **Basic Statistics** menu, select **Averages and Correlations**
- In the window that appears, in the box at the right select **nonWhite** and click **Calculate**




We see that the average proportion of non-White students is 0.39: less than half.

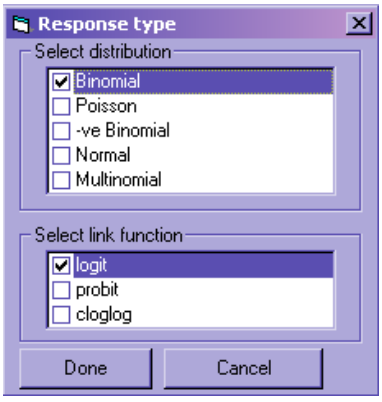
Now we can set up the model. In MLwiN, we fit models by building up their equations in the **Equations** window, and then pressing the **Start** button at the top of MLwiN to run the model and obtain the estimates.

- From the **Model** menu select **Equations**
- Click on either of the red $y$s
- In the window that appears, next to **y:** select **nonWhite**
- Next to **N levels:** select **3-ijk**
- Next to **level 3(k):** select **LEA**
- Next to **level 2(j):** select **School**
- Next to **level 1(i):** select **cell**
- Click **done**



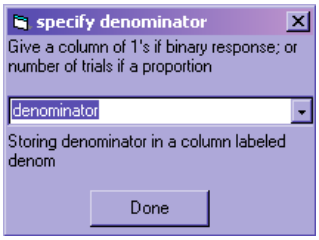

We have set our response variable to be **nonWhite**, and specified a model with 3 levels, **LEA**, **School** and **cell** (which identifies the rows– recall we are using it instead of **Cohort** because it has a different value for every row while **Cohort** takes the same value for example for the first cohort of School 1 in LEA 1 and the first cohort of School 2 in LEA 1). Next we need to specify that the model is binomial, instead of the default continuous response model:

- Click on the capital N in the first line of the **Equations** window
- Under **Select distribution**, tick **Binomial**
- Under **Select link function**, leave **logit** ticked
- Click **Done**

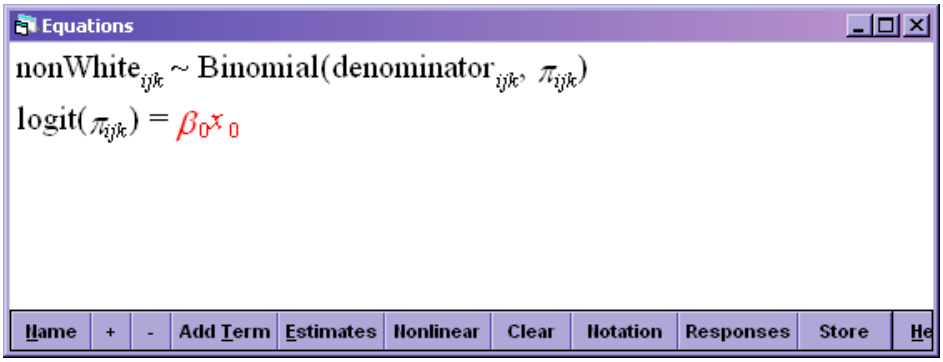


We now need to specify the denominator. This is a variable which says how many observations the proportions were based on. In our case, the proportions are the proportions of students in each cohort who are not White. The proportions are thus based on the number of students in each cohort. We have already created a variable **denominator** in the last section which consists of the number of students in each cohort.

- Click on the red $n_{ijk}$ in the first line of the **Equations** window
- Choose **denominator** from the drop-down box
- Click **Done**



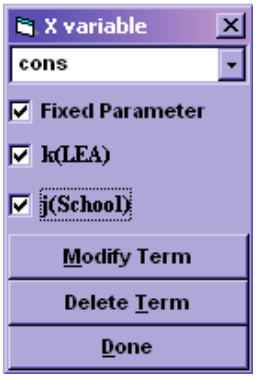

(Note that when we specify the denominator, MLwiN creates a new variable based on our denominator which it calls **denom**. Therefore the user should not use this name for the denominator variable which they create themselves and specify in the **Equations** window as this will cause problems when MLwiN tries to use the name again.)



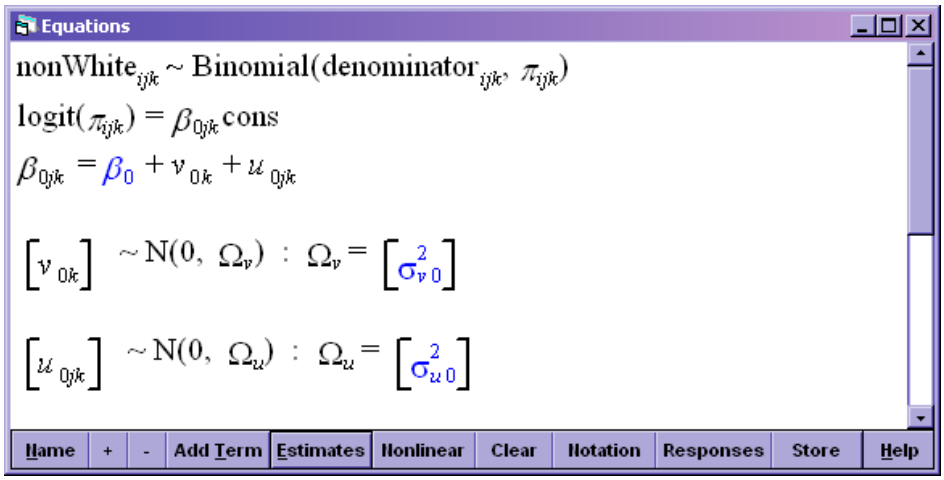

The **Equations** window now tells us that our response is assumed to have a binomial distribution, with the denominator (also known as 'number of trials') that we specified, and a probability that varies from row to row of the dataset. The logit of the probability is modelled by the equation in the second line of the window; so far we have not added any explanatory variables. We will do so now:

- Click on the red $\beta_0 x_0$ in the second line
- In the window that appears, choose **cons** from the drop-down box

- Leave the **Fixed Parameter** box ticked and tick the **k(LEA)** and **j(School)** boxes as well
- Click **Done**




- Press the **Estimates** button at the bottom of the **Equations** window

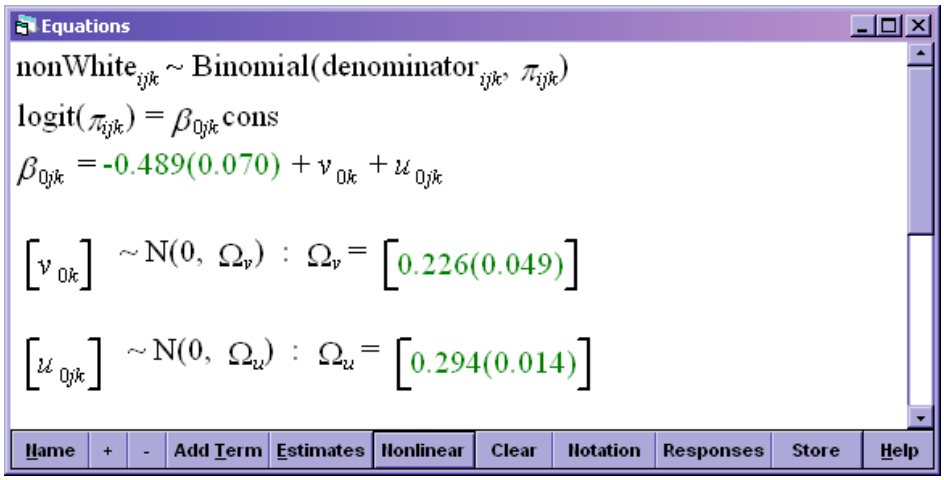



We have now set up a simple intercept-only model. Recall that **cons** takes a value of 1 for every row in the dataset. Therefore we are modelling the logit of the proportion by $\beta_{0jk} \times 1 = \beta_{0jk}$. The equation for $\beta_{0jk}$ is given in the next line. It is made up of the intercept $\beta_0$, which we will estimate, and two random effects: an LEA level effect $v_{0k}$ and a school level effect $u_{0jk}$. The next two lines give the distributions of these random effects and we see that each is normally distributed with a variance which we will estimate. These two variance parameters will be our measures of segregation. Thus we estimate 3 parameters in all, the intercept and 2 variances, but our main interest lies in just two of these, the variances.

If we press the **Estimates** button again, the parameters will be replaced by numbers. However we have not yet run the model so these numbers are not our estimates. We will run the model

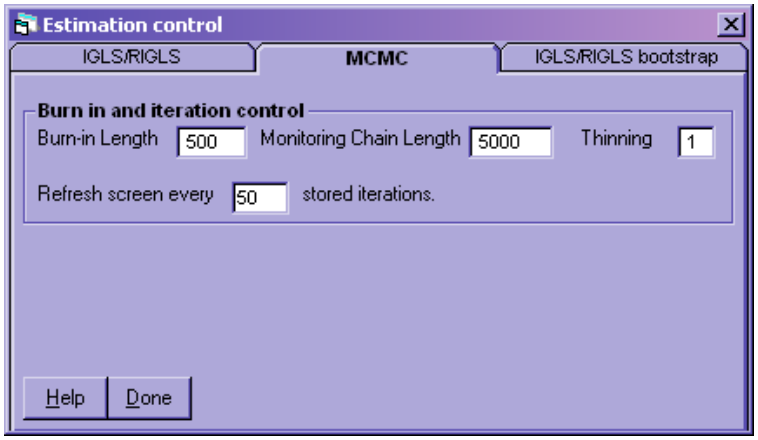
now. First we will set the estimation method to the default for binomial models, 1st order MQL (for more on the choice of estimation methods for binomial models see the *User's Guide to MLwiN* and the forthcoming modules of our online training materials on binomial models)

- Press the **Nonlinear** button at the bottom of the **Equations** window
- Click **Use Defaults** and **Done**
- Press the **Start** button at the top of MLwiN



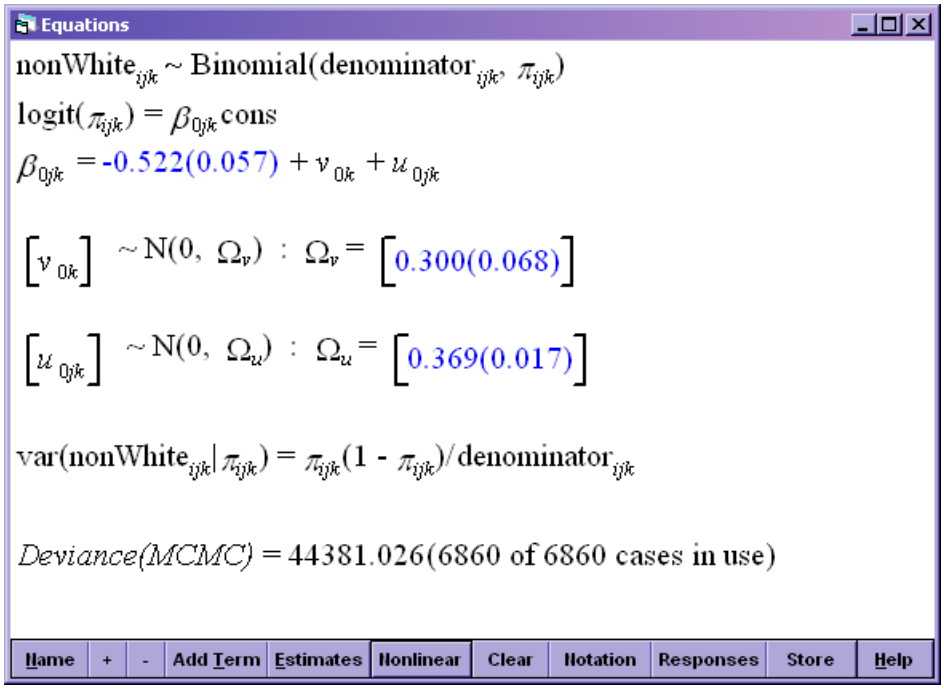

We will now switch to estimation by MCMC. We do this because quasi-likelihood methods, such as MQL which we have just used, can give biased estimates. (For more details, see the *User's Guide to MLwiN* and for more information about MCMC see *MCMC estimation in MLwiN* ). However it is important that we have run the model with MQL: in MLwiN estimates from likelihood or quasi-likelihood methods are used as the starting values in MCMC so MCMC cannot be used without first having run using one of these methods.

- Click on the **Estimation control** button at the top of MLwiN
- Click on the **MCMC** tab and click **Done**



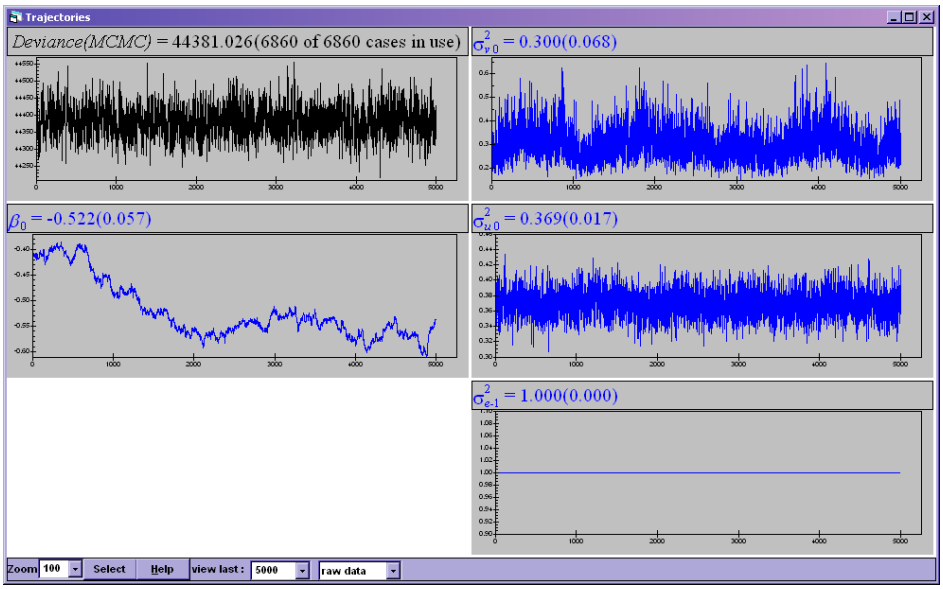

- Press **Start**

When estimation has finished we get these results:




Before we interpret them, we should check that the parameter chains have converged and that the burn-in has been long enough:

- From the **Model** menu, select **Trajectories**
- In the **view last:** box at the bottom of the **Trajectories** window, select **5000**

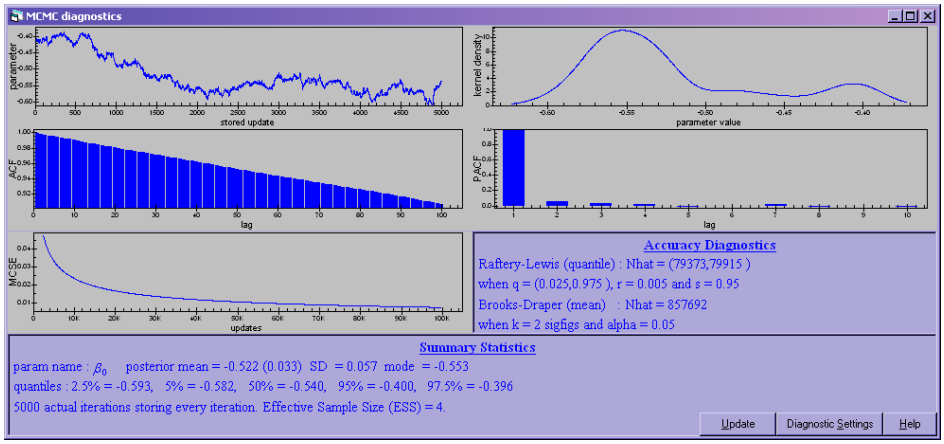



The **Trajectories** window shows a plot of the estimate of each parameter against the iteration number. We can see five graphs, but we are not interested in the first or last of these at the moment: the first is the deviance which is a measure for MCMC models used in a similar way to the likelihood (for more details see *MCMC estimation in MLwiN* ) and the last is a parameter involved in the variance at the cohort level which is constrained to be 1 because we are fitting a binomial model. The graphs we need to look at are the middle three, for $\sigma^2_{v0}$, $\beta_0$ and $\sigma^2_{u0}$.

The estimates that we see in the **Equations** window are arrived at by taking the mean of the estimates across all the iterations. This is why the burn-in period is important: estimates during the burn-in are discarded and so do not affect the estimate we see in the **Equations** window. Often the starting values we supply are quite wrong and the estimates need to move away from this wrong value over a number of iterations before settling down to vary around the correct value. If we look at the graph for $\beta_0$ we see that the estimates exhibit this behaviour: in early iterations the estimates are quite high, around -0.4, but they gradually move down to around -0.55 or -0.6. When we take the mean to arrive at our estimate for $\beta_0$ we do not want to include the iterations where the estimate varies around the higher value or where the estimate is moving from the higher to the lower value: we only want to include the iterations where the estimate fluctuates round the final value. So the graph tells us that we should have used a longer burn-in. Looking at the graph it is also not certain whether the estimates for $\beta_0$ have finished moving and will now continue to stay between around -0.55 and -0.6, or whether they will drop still further (or indeed rise again). We really need to run MCMC for a bit longer to be sure.

The graph for $\sigma^2_{u0}$ looks much better: this is an example of what a good MCMC chain should look like. However we should bear in mind that the estimates of all three of our parameters are related, and if $\beta_0$ has not in fact converged then we may find if we run for longer that as the value of $\beta_0$ changes the estimates for $\sigma^2_{u0}$ may suddenly jump to a different value. Thus it is important that we run until all parameters have converged, not just the parameters that we are interested in.

We can also get some information about whether we have run MCMC for long enough by clicking on any of the graphs, for example the graph for $\beta_0$, and clicking **Yes** to the question that appears. This brings up a window with some diagnostic information on the chain for that parameter:



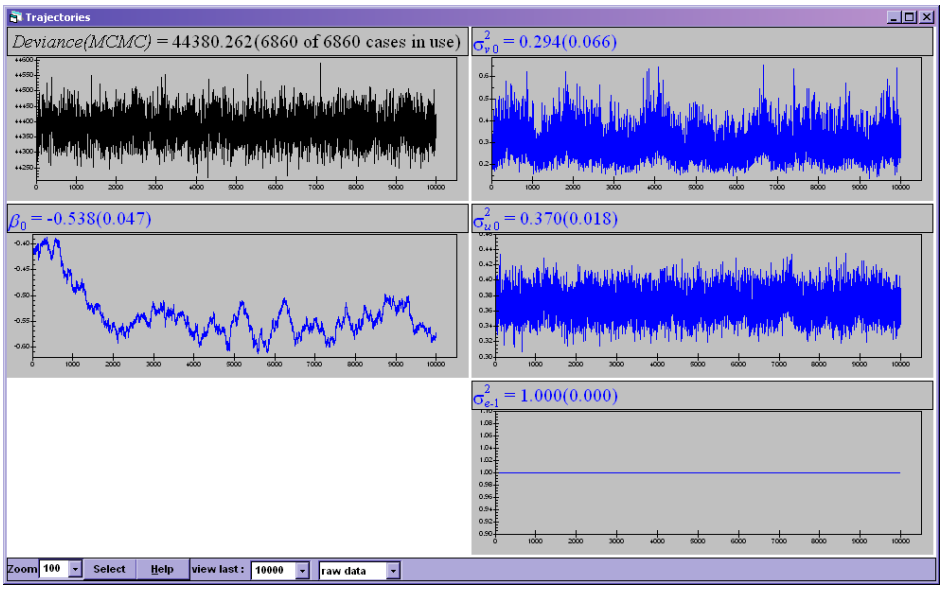

For details of what all this information means and how to use it, see *MCMC estimation in MLwiN* , but for now we will note that the effective sample size given at the bottom is only 4. We really want an effective sample size of several hundred at least, and preferably several thousand. By contrast, the effective sample size for $\sigma^2_{u0}$ is 3215; if all parameters had an effective sample size this large we would be quite happy.

We will run MCMC for another 5,000 iterations. You can leave the **Trajectories** window open if you want, but the model will run more slowly if you do since this window will be updated every 50 iterations.

- Press the **Estimation control** button
- Change **Monitoring Chain Length** to 10000
- Click **Done**
- Press **More** (*not* **Start**) at the top of MLwiN

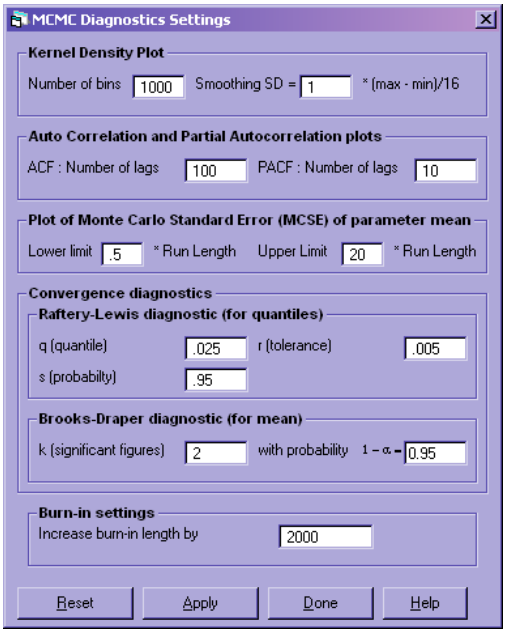
We have asked for a total of 10,000 iterations so MLwiN will perform another 5,000 to bring us up to this.

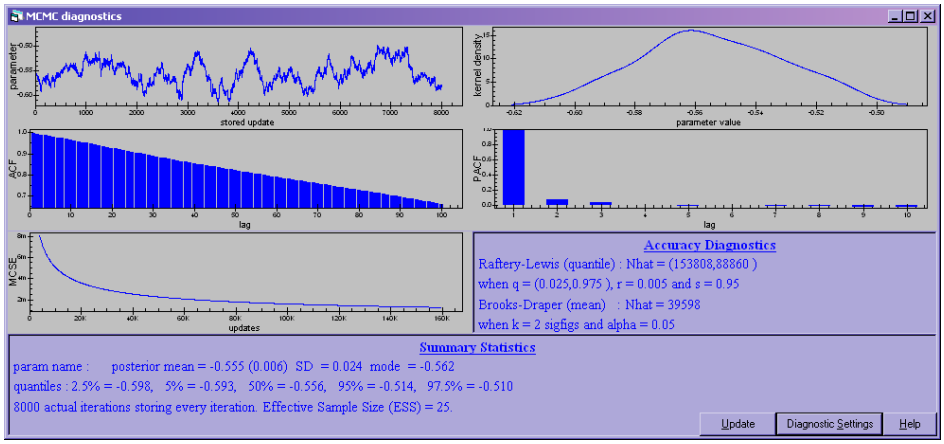
- In the **Trajectories** window, select **10000** in the **view last:** box




We can now be reasonably confident that estimates for all three of the parameters have converged, but the burn in is still not long enough as we can see looking at the graph for $\beta_0$. It looks as though we need to include the first 2,000 iterations that we can see here in the burn-in period.

- Click on the graph for any of the parameters (e.g. $\beta_0$)
- Click **Yes** to the question that appears
- Click the **Diagnostic Settings** button at the bottom of the **MCMC diagnostics** window
- In the **MCMC Diagnostics Settings** window that appears, under **Burn-in settings** at the bottom, in the box next to **Increase burn-in length by** type 2000
- Click **Apply** and **Done**




The **MCMC diagnostics** window for $\beta_0$ has been updated and we can see that now the chain we are basing our estimate of $\beta_0$ on no longer includes the 2,000 iterations where the estimates were moving away from the incorrect starting value. If we look in the section headed **Summary Statistics** we see that the mean is now calculated to be -0.555 and the standard error of our estimate is 0.024 – compare to the values of -0.538 and 0.047 before we increased the burn-in. We also see that the effective sample size has increased, to 25 (it was 9 after we ran for the second 5,000 iterations but before we increased the burn-in).

## Exercise 3

What are the effective sample sizes for $\sigma^2_{v0}$ and $\sigma^2_{u0}$ now?

We also need to find out what the estimates are for the other two parameters with the increased

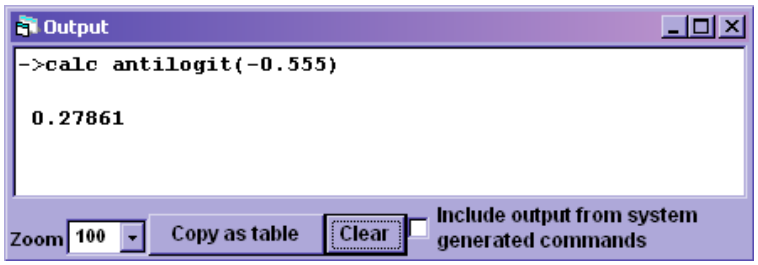
burn in (though we do not expect them to have changed much since these parameters appeared to have converged even with the shorter burn in). We do this by clicking on their graphs in the **Trajectories** window – the burn-in is still set to 2,000 so we do not need to specify this again. (Note however that this burn-in is not applied to the estimates displayed in the **Equations** window or above each graph in the **Trajectories** window – it only applies to the **MCMC diagnostics** window.)

| | | | 95% CI | |
|---|---|---|---|---|
| Parameter | Estimate | Standard error | Lower | Upper |
| $\beta_0$ | −0.555 | (0.006) | −0.598 | −0.510 |
| $\sigma^2_{v0}$ | 0.290 | (0.064) | 0.189 | 0.439 |
| $\sigma^2_{u0}$ | 0.370 | (0.018) | 0.337 | 0.406 |

Our estimates tell us that (as we expect from our earlier calculation of the average of **nonWhite**) the average (or predicted) proportion of nonWhite students across cohorts, schools and LEAs is less than 0.5 (since $\beta_0$ is negative). We can calculate the exact median predicted probability by taking the antilogit of $\beta_0$: in the **Command interface** window type

```
► calc antilogit(-0.555)
```

We get:



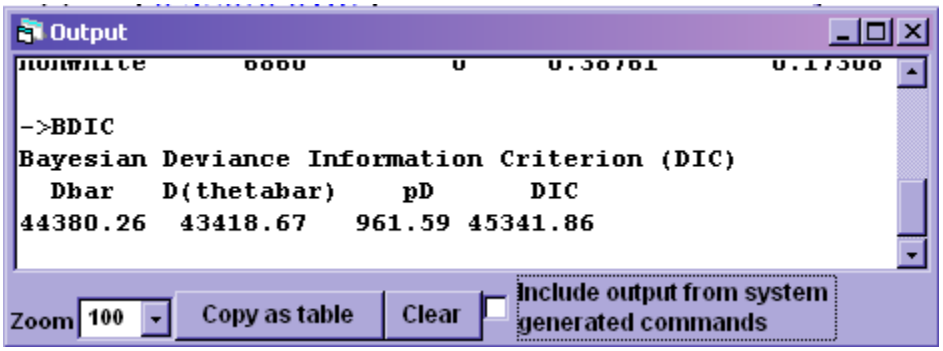

(press the **Output** button on the **Command interface** window to get the **Output** window if it is not already shown). The difference from the average value of **nonWhite** which we calculated earlier is partly because we are now taking the median rather than the mean (the median value of **nonWhite** is 0.372), and partly because our model allows for the dependency between cohorts in the same school and schools in the same LEA (when we took the average of **nonWhite** we were tacitly assuming each row of the dataset was an independent observation).

Our real interest, of course, is in the variance parameters since these are our measures of segregation. Our estimate for the LEA level segregation is 0.290, and our estimate for the school level segregation is 0.370. Both these estimates are significant as we can see by looking at the 95% confidence intervals (we get these from the quantiles shown in the **MCMC diagnostics** window). Thus we have found that, for our simulated dataset, there is significant variation between LEAs in their proportions

of non-White students, i.e. there is significant segregation, and that even after these differences between LEAs have been taken into account, there is still school segregation. We could if we wanted calculate indexes such as D from our estimates.

We will also calculate the DIC. This will not tell us anything useful right now, but we will need it later when we want to compare this model with the next model we go on to fit.

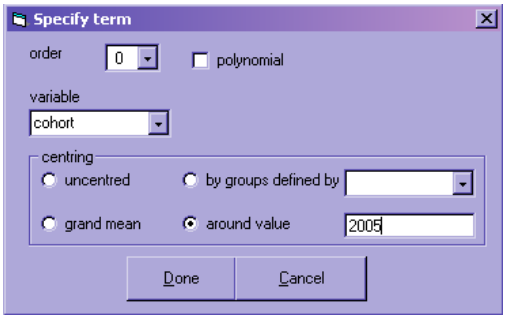
- From the **model** menu select **MCMC → DIC diagnostic**

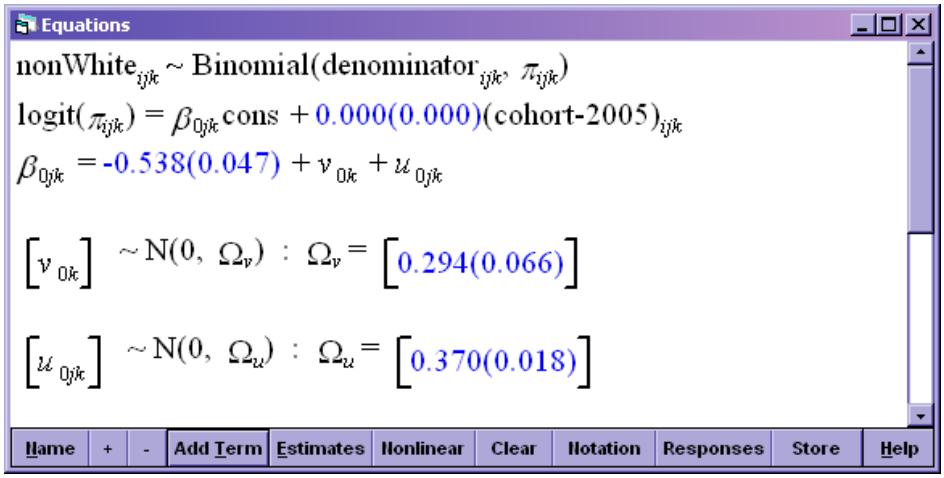



The DIC is shown in the fourth column and we can see for this model its value is 45341.86.

We might wish to know how segregation is changing over time: is it rising, falling, or remaining roughly the same? To address this question, we will need to include the year (i.e. **Cohort**) as an explanatory variable in our model. To make changes to our model, we need to change the estimation method back to IGLS, make the changes to the set-up, run the model again using MQL, and then run again in MCMC.

- Press the **Estimation control** button
- Click the **IGLS/RIGLS** tab
- Click **Done**
- At the bottom of the **Equations** window, click the **Add Term** button
- In the **Specify term** window that appears, select **Cohort** from the **variable** drop-down box
- Under **centring**, select **around value** and type **2005**
- Click **Done**

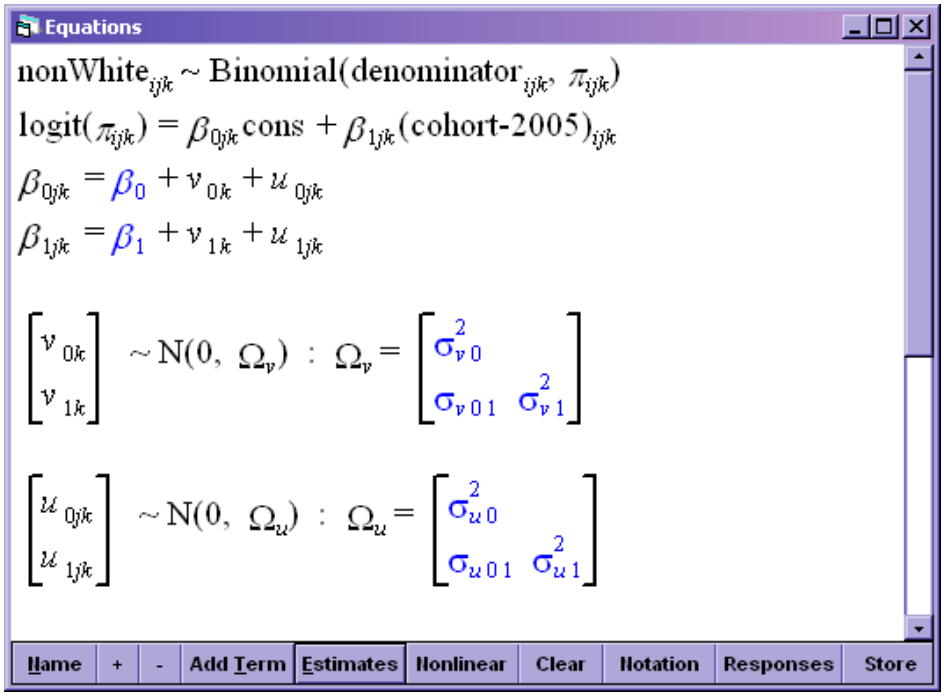



We have centred **cohort** around the middle of its possible values, 2005, which corresponds to subtracting 2005 from every value- this will help with the estimation and interpretation. MLwiN has automatically created the centred variable for us as we can see if we look in the Names window.

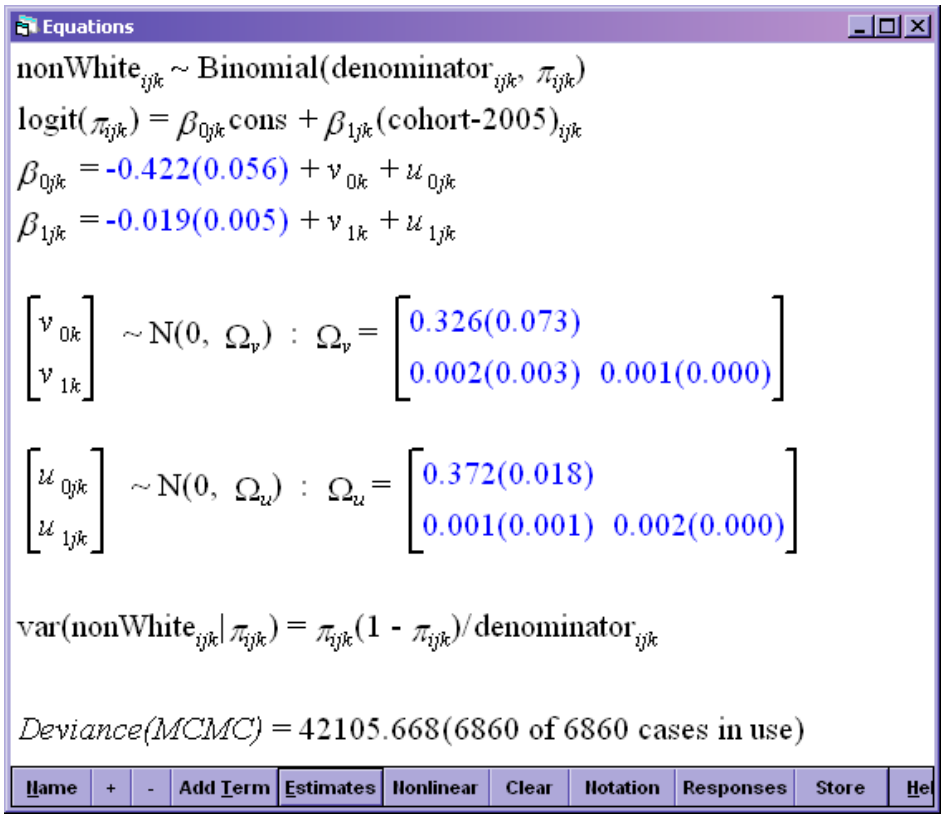
So far, we have allowed the overall proportion non-White to change linearly from year to year, but we have not allowed the segregation to change. To do this, we need to add random effects on **Cohort**. These will allow the school and LEA level variance to change with time.

- Click on the **(cohort-2005)** term in the **Equations** window
- In the **X variable** window which appears, leave **Fixed Parameter** ticked and tick **k(LEA)** and **j(School)**
- Click **Done**
- In the **Equations** window, click the **Estimates** button twice




We have 2 new parameters at the LEA level, $\sigma_{v01}$ and $\sigma^2_{v1}$, and 2 new parameters at the school level, $\sigma_{u01}$ and $\sigma^2_{u1}$. $\sigma^2_{v1}$ is the variance of the random effects on **Cohort**, and $\sigma_{v01}$ is the covariance between the random intercepts and the random effects on **Cohort**; and similarly for the school level random parameters. However as we will see when we have obtained our estimates, we will not interpret each parameter separately but combine them to get a single function for the variance at each level.

- Click the **Estimates** button again
- Click the **Start** button
- Change the estimation method to MCMC and keep the burn-in at 500 and the monitoring chain length at 10,000
- Click the **Start** button again




If we were actually using this analysis as part of our research, we should really check the convergence and the adequacy of the burn-in again by looking at the **Trajectories** window and the **MCMC diagnostics** window. However we will pretend that we are satisfied on both counts and use our estimates as they are, leaving the question of whether this is reasonable as an exercise:

### Exercise 4

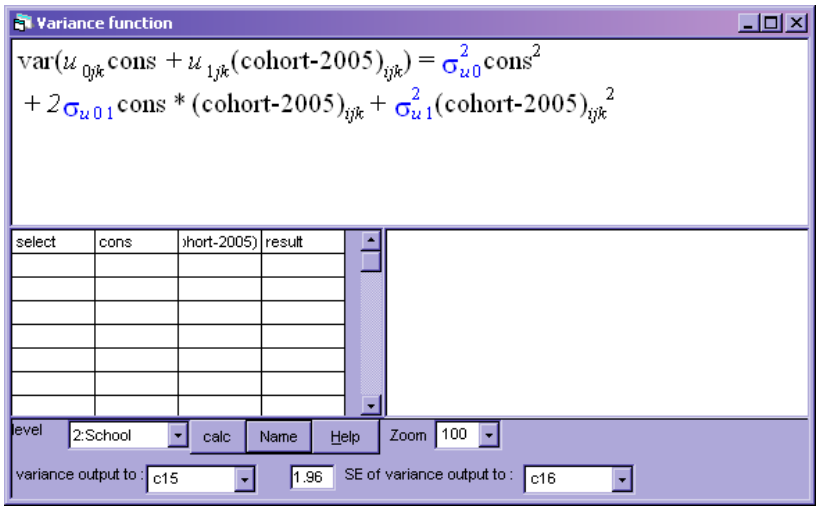
Check whether the burn-in and/or the total number of iterations should be increased, and if so whether this makes a difference to the estimates. (Note that the results displayed in the **MCMC diagnostics** window still have the extra burn-in of 2,000 that we specified for the previous model applied so you will need to change this back to 0 to see diagnostics for the estimates we use below).

---

Both our fixed part coefficients, $\beta_0$ and $\beta_1$, are negative (and significant), so the average proportion non-White in 2005 is less than 0.5, and the proportion non-White is decreasing each year.

As we stated above, we will not interpret our variance parameters individually, but will calculate a *variance function* for each level. This uses the random parameter estimates to calculate the total variance at each level for each year, and we can then plot this against the year (**Cohort**) to see how the segregation at each level changes with time.

- From the **Model** menu, select **Variance function**
- Click the **Name** button at the bottom of the **Variance function** window
- Next to **level** select **2:School**
- Next to **variance output to:** select **c15**
- Next to **SE of variance output to:** select **c16**
- Change the **1.0** in the box in front of **SE of variance output to** to **1.96** (so that we can plot 95% confidence intervals)
- Click **calc**
- Name c15 'schoolvar' and c16 'schoolvar_SE' (in the **Names** window or **Command interface** window)




The top pane of the **Variance function** window shows the formula used to calculate the variance function. Since **cons** is always 1, it simplifies to

$$\sigma^2_{u0} + 2\sigma_{u01}(\mathbf{Cohort}_{ijk} - 2005) + \sigma^2_{u1}(\mathbf{Cohort}_{ijk} - 2005)^2$$

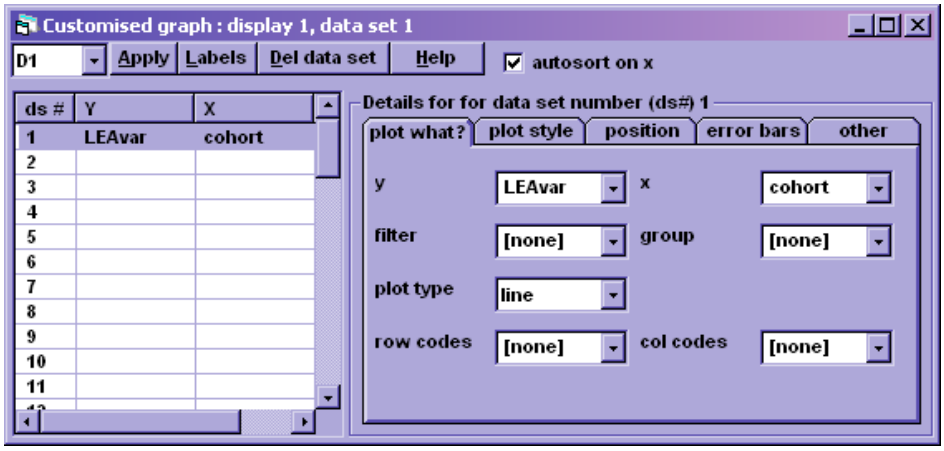
So we are modelling the variance (the segregation) as a quadratic function of the year.

- Repeat changing **level** to **3:LEA** and putting the variance into **c17** (name this **LEAvar**) and the standard error into **c18** (name this **LEAvar_SE**)

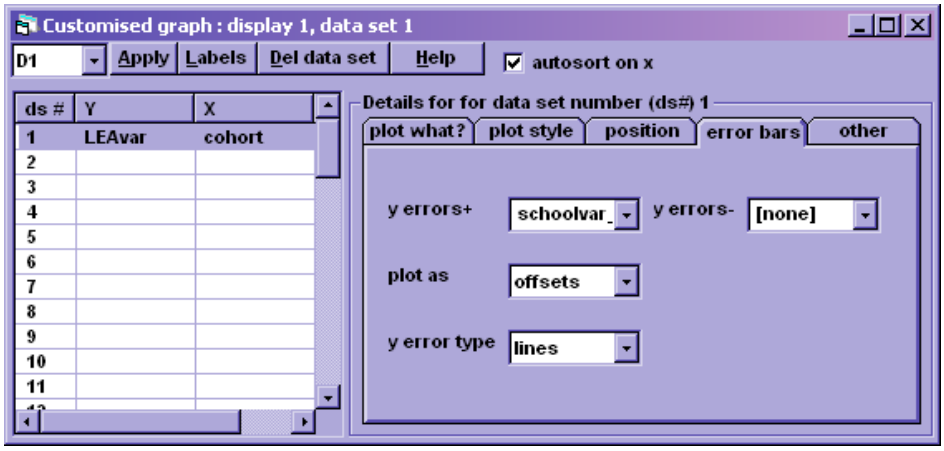
We can now plot the segregation as a graph

- From the **Graphs** menu select **Customised Graph(s)**

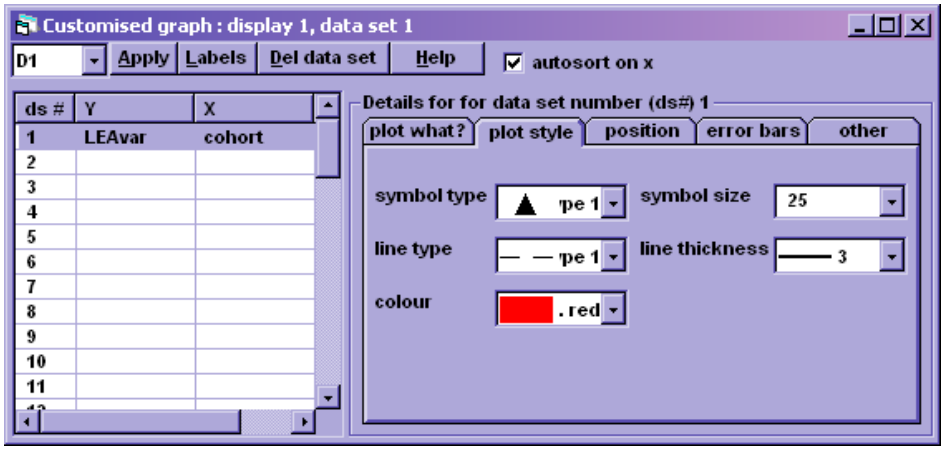
- In the **Customised graph** window, next to **y** select **LEAvar** and next to **x** select **cohort**
- Next to **plot type** select **line**



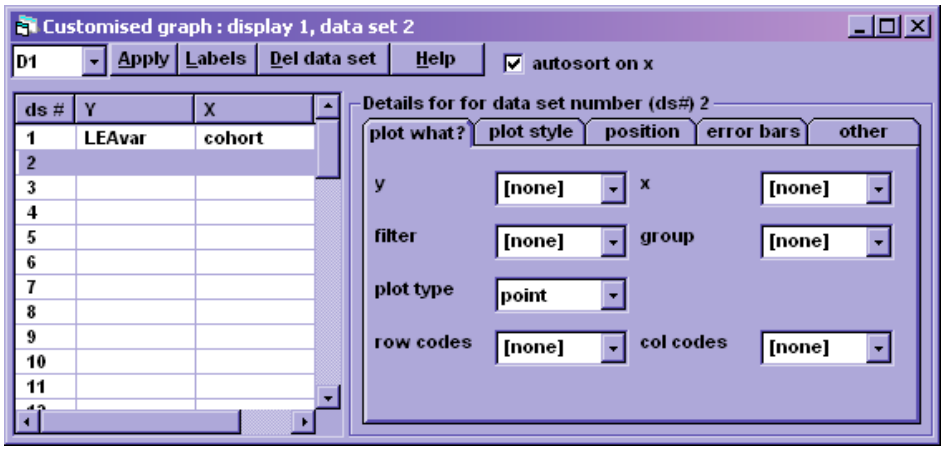

- Select the **error bars** tab
- Next to **y errors+** select **schoolvar_SE** and next to **y errors-** again select **schoolvar_SE**
- Next to **y error type** select **lines**



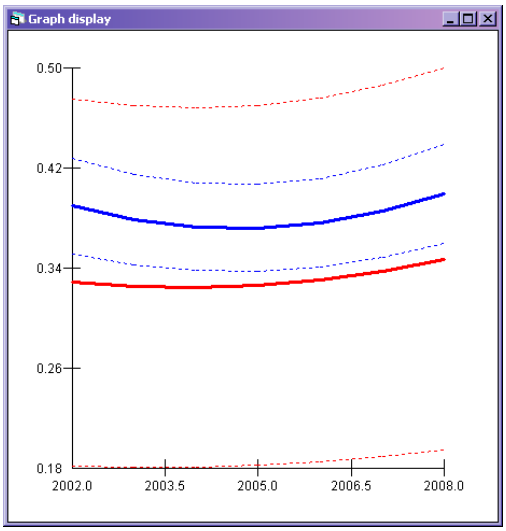

- Select the **plot style** tab
- Next to **line thickness** select **3**
- Next to **colour** select **12 l. red**




- Click the **plot what?** tab
- In the grid at the left of the window, click on the second row

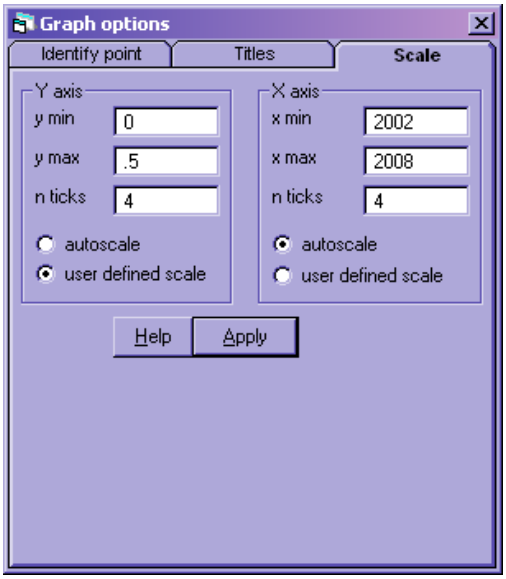


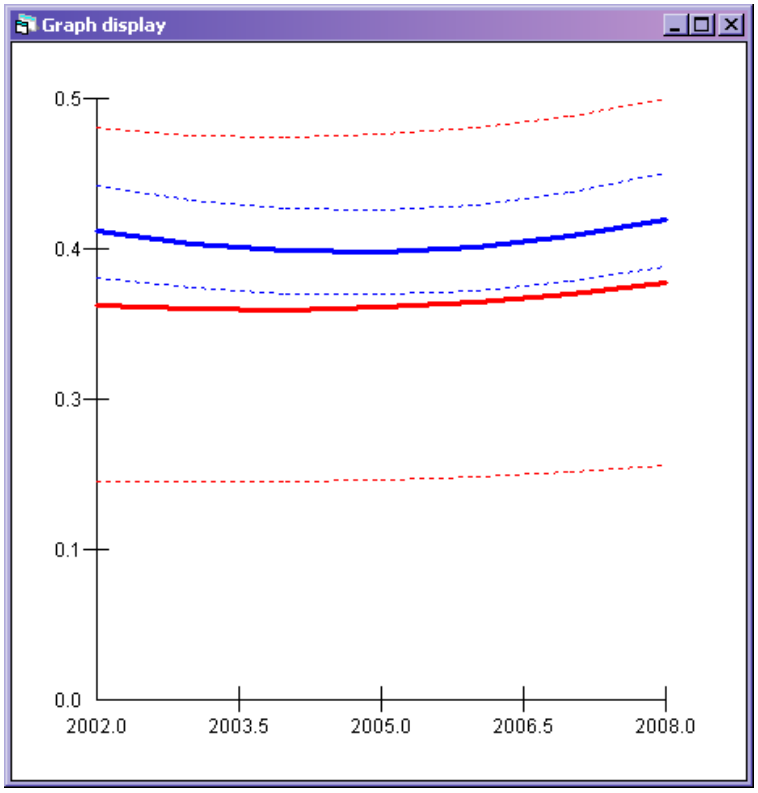

- Fill in the **plot what?**, **error bars** and **plot style** tabs to plot the school level variance and confidence intervals in colour **9 l. blue**
- Click **Apply** at the top left of the window




We can rescale so that the $y$-axis runs from 0 to 0.5:

- Click on the graph
- In the **Graph options** window, select the **Scale** tab
- Under **Y axis**, select **user defined scale**
- Change **y min** to 0
- Click **Apply**






We can see that both the school level segregation (in blue) and the LEA level segregation (in red) are significant in all years (the confidence intervals do not cross 0).

The confidence intervals we have plotted here are derived from the standard errors of the variance parameters. It would be better to use the stored chains of parameter estimates from MCMC to calculate 95% CIs (since these would be nonparametric; the confidence intervals we are using now are based on the assumption that the sampling distribution of the random parameters is Normal which is probably not true). We would need to unstack the chains of parameter values as described in *MCMC estimation in MLwiN* , then for each iteration calculate the variance function, then for each year take the 2.5th and 97.5th percentiles. We will not do this now (but you may want to try this in your own time).

### Exercise 5

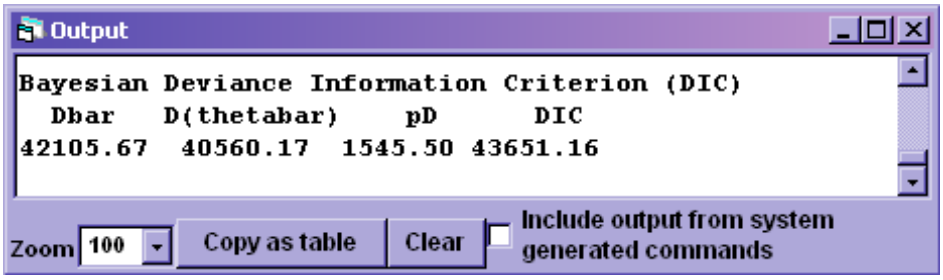
Were the confidence intervals for our measures of segregation in the simpler model (without **cohort**) obtained in this way?

### Exercise 6

If you run MCMC for more iterations, do the standard errors of the estimates reduce and hence the confidence intervals become narrower?

In order to assess whether the change in segregation over time is significant we will compare this model to our model without the time trend using the DIC (see *MCMC estimation in MLwiN* for more information about the DIC). If we look back to page 15, we can see (in the top left of the **Trajectories** window) that the deviance for the model without the time trend was 44380.262, and if we look at the **Equations** window we can see that the deviance for the model with the time trend is 42105.668. However we do not compare the deviances directly: we compare the DICs, which combine the deviance with information about the number of parameters in the model so that more complicated models are penalised. Recall that we have already calculated the DIC for the model without the time trend (p19) and we found it was 45341.86. To obtain the DIC for the current model, with the time trend, we do the same thing:

- From the **model** menu select **MCMC** → **DIC diagnostic**




A lower DIC implies a better model, and the difference in DIC in this case of around 1500 is substantial, so we conclude that the model with the time trend is definitely better than the model without and hence that (at least one of) the changes in segregation we observed is significant (we would need to fit the model with just the LEA level time trend and the model with just the school level time trend in order to establish whether just one or both are significant, but we will not do this now for time reasons).

# 4 Fitting a multinomial model

## 4.1 Multinomial models in MLwiN

To explain how the setting up of multinomial models works in MLwiN, let's first consider a simpler example. Instead of the situation we have, with the proportion of students of each ethnicity in each school cohort, suppose we have data on voting intentions, with one row of the dataset per person and people grouped by area. There are 3 possible parties to choose from, and each person chooses just one. So our response variable **vote** might look like this: $1, 1, 2, 1, 3, 3, 2, 1, \ldots$. We specify that **vote** is our response, and that we have two levels, **area** (level 2) and **individual** (level 1). We then specify that the model is multinomial and specify a reference category (let's choose category

1). When we do this, MLwiN creates some new expanded versions of **vote**, **area** and **individual**, which it calls **resp**, **area_long** and **individual_long**. This is because in order to estimate the multinomial model, MLwiN needs to create an extra level corresponding to the response category. In the new dataset, there is one row per response category per person. The reference category does not have a row, so in our example where we start with 3 response categories, we end up with 2 rows per person corresponding to categories 2 and 3. Another new variable, **resp_indicator**, is created to provide the ID codes for this new level.

In the expanded dataset, the response **resp** no longer consists of 1s, 2s and 3s. It now takes values 0 or 1. It takes the value 1 when in our original response **vote**, the person's value was the category which the current row corresponds to, and 0 when the person's value was a different category. So with 3 categories, each person's 2 rows can be $1, 0$ (if the person's value of **vote** was 2), $0, 1$ (if the person's value of **vote** was 3), or $0, 0$ (if the person's value of **vote** was 1). In our example, where the first 3 people had values of **vote** $1, 1, 2$, **resp** will have $0, 0, 0, 0, 1, 0$ for its first 6 values.

We next need to specify a denominator. This is a variable which says how many observations each row of our original dataset was based on. In this example we had one row per person, so each row referred to just one person and the denominator will have a value of 1 for every row. In the case of our segregation data, by contrast, every row corresponds to a school cohort, and the proportions are proportions of the total number of students in the cohort, so the denominator will consist of the total number of students in each school cohort. (In fact we have already created this variable and called it **denominator**). Note that as we continue to set up our model, specifying the denominator and adding explanatory variables, we specify variables from our original dataset, and MLwiN expands them properly for us in the background– we do not have to create expanded versions ourself to pass to MLwiN.

We can now add our explanatory variables. MLwiN displays a different equation for each response category, so explanatory variables can have different coefficients for different response categories. We can add variables to some equations and not others, and we can specify that an explanatory variable should have the same coefficient for all response categories if we want. We will say more about this later.

Returning to our segregation data, we now have data on the proportion in each response category, instead of having a particular category as the response for each row as in the voting example. The initial form of our dataset is thus quite different: instead of one categorical response variable we have three variables taking any value between 0 and 1. However if we consider the voting example after it was expanded by MLwiN, we see that one is just the generalisation of the other. We can think of the voting example as being proportions each based on just 1 person. Since there is only 1 person, the proportion will be 1 for the party they intend to vote for and 0 for the other parties. This is what we see in the new response variable that MLwiN created, **resp**. In our segregation example, we will need MLwiN to create an expanded dataset just as it did in the voting example, but this time instead of a response variable consisting of 0s and 1s, we want the response variable to contain our proportions (with the proportions for the reference category missed out). So (if we choose **White** as the reference) row 1 of the expanded dataset corresponds to response category **Black** in the first cohort of School 1 in LEA 1, and the response will be 0.194 since this is the

proportion of Black students in the first cohort of School 1 in LEA 1. Row 2 of the expanded dataset corresponds to response category **Asian** in the first cohort of School 1 in LEA 1, and the response will be 0.326 since this is the proportion of Asian students in the first cohort of School 1 in LEA 1. There is no row which has response 0.481, the proportion of White students in the first cohort of School 1 in LEA 1: we have omitted White because it is the reference category. But the information is not missing because the first row of the denominator we specify (which becomes the first and second rows of the expanded denominator created by MLwiN) is 129, the total number of students in the first cohort of School 1 in LEA 1.

So we know what the expanded dataset should look like. Creating it however is a little complicated. Since the multinomial facility in MLwiN was originally designed with individual rather than proportion data in mind (i.e. intended for data like the voting example rather than like our segregation data), MLwiN will not allow a multinomial model to be specified unless the response is a categorical variable (so we cannot specify a multinomial model if our response instead takes any value between 0 and 1). MLwiN then expands the dataset and creates a new variable consisting of 0s and 1s to be the response as described above. What we need to do therefore is to use the variable we created in the last section, **dummyresp**, to get MLwiN to allow us to specify a multinomial model and to expand the dataset properly. We will then overwrite the new response variable created by MLwiN with our proportions data, arranged in the form we described in the previous paragraph (proportion Black in first cohort in School 1 in LEA 1, proportion Asian in first cohort in School 1 in LEA 1, proportion Black in second cohort in School 1 in LEA 1, proportion Asian in second cohort in School 1 in LEA 1,...). In order for this to work properly, **dummyresp** needs to be the same length as our original dataset (6860), and it needs to be a categorical variable with 3 categories, and category names corresponding to the categories of our proportions data (because MLwiN will use the category names of **dummyresp** to refer to the response categories of the multinomial model). The variable **dummyresp** which we created in the last section meets these conditions so we are ready to set up the model.

**Exercise 7**

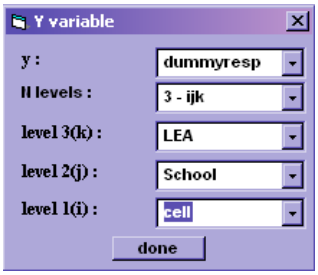
What is the mean proportion of Black students across school cohorts? What is the mean proportion of Asian students across school cohorts? What is the mean proportion of White students across school cohorts?

## 4.2 Setting up the multinomial model

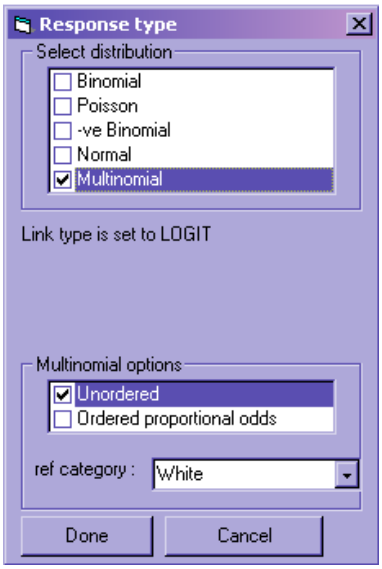
*Note that you can use the worksheet **segregation2** for the following analysis, if you have not still got the worksheet you were using in the previous sections*

- Set the estimation method back to IGLS/RIGLS
- Click the **Clear** button at the bottom of the **Equations** window

- Click on either of the red $y$s
- Next to **y:** select **dummyresp**
- Next to **N levels:** select **3 - ijk**
- Next to **level 3(k):** select **LEA**
- Next to **level 2(j):** select **School**
- Next to **level 1(i):** select **cell**
- Click **done**

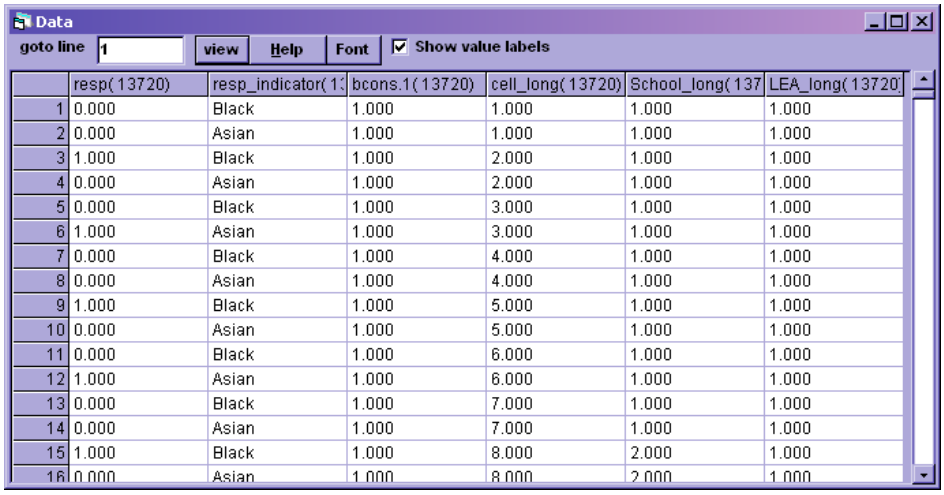



- Click on the capital N in the **Equations** window
- Under **Select distribution** tick **Multinomial**
- Under **Multinomial options**, make sure **Unordered** is ticked and next to **ref category:** make sure **White** is selected
- Click **Done**




MLwiN has now created a properly expanded dataset for us. We can see this if we got to the **Names** window. We have six new variables: **resp**, **resp_indicator**, **bcons.1**, **cell_long**, **School_long**,

and **LEA_long**. If we highlight these variables and click the **Data** button, we will see the following:

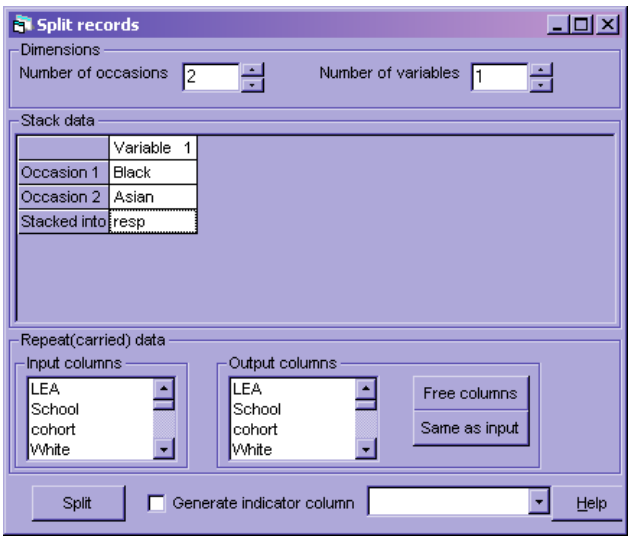



| | resp( 13720) | resp_indicator( 1 | bcons.1( 13720) | cell_long( 13720) | School_long( 137 | LEA_long( 13720) |
|---|---|---|---|---|---|---|
| 1 | 0.000 | Black | 1.000 | 1.000 | 1.000 | 1.000 |
| 2 | 0.000 | Asian | 1.000 | 1.000 | 1.000 | 1.000 |
| 3 | 1.000 | Black | 1.000 | 2.000 | 1.000 | 1.000 |
| 4 | 0.000 | Asian | 1.000 | 2.000 | 1.000 | 1.000 |
| 5 | 0.000 | Black | 1.000 | 3.000 | 1.000 | 1.000 |
| 6 | 1.000 | Asian | 1.000 | 3.000 | 1.000 | 1.000 |
| 7 | 0.000 | Black | 1.000 | 4.000 | 1.000 | 1.000 |
| 8 | 0.000 | Asian | 1.000 | 4.000 | 1.000 | 1.000 |
| 9 | 1.000 | Black | 1.000 | 5.000 | 1.000 | 1.000 |
| 10 | 0.000 | Asian | 1.000 | 5.000 | 1.000 | 1.000 |
| 11 | 0.000 | Black | 1.000 | 6.000 | 1.000 | 1.000 |
| 12 | 1.000 | Asian | 1.000 | 6.000 | 1.000 | 1.000 |
| 13 | 0.000 | Black | 1.000 | 7.000 | 1.000 | 1.000 |
| 14 | 0.000 | Asian | 1.000 | 7.000 | 1.000 | 1.000 |
| 15 | 1.000 | Black | 1.000 | 8.000 | 2.000 | 1.000 |
| 16 | 0.000 | Asian | 1.000 | 8.000 | 2.000 | 1.000 |

**resp** does not yet have the right values as it was created from **dummyresp**, but the other columns are as they should be. We can see from the values of **resp_indicator** that the first row refers to **Black**, the second row to **Asian**, the third row to **Black**, the fourth row to **Asian**, and so on. **cell**, our cohort identifier, has been expanded to **cell_long**, and where **cell** had values $1, 2, 3, \ldots$ **cell_long** has values $1, 1, 2, 2, 3, 3, \ldots$. **School_long** and **LEA_long** consist of similarly duplicated values. Thus the first and second rows both refer to the 2002 cohort in School 1 in LEA 1; when we put the correct values in the **resp** column, the first row will have the proportion Black in the 2002 cohort in School 1 and the second row will have the proportion Asian in the 2002 cohort in School 1. **bcons.1** is like an expanded version of our variable **cons**, which just takes the value 1 for every row. We will not see it appear in the **Equations** window, but MLwiN will use it when we add more explanatory variables so that it can expand them properly.

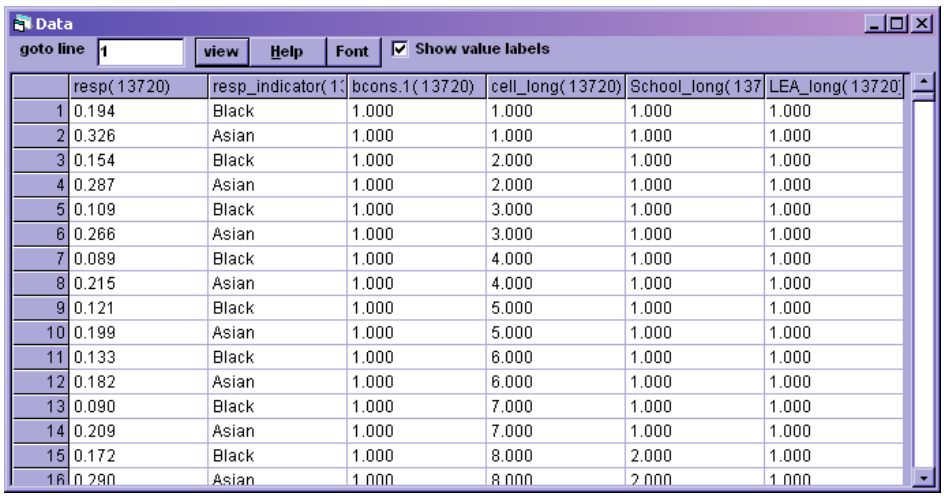
We will continue to set up the model by overwriting the current response variable **resp** with our proportions data:

- From the **Data manipulation** menu, select **Split Records**
- In the window that appears, increase the **Number of occasions** to 2
- In the **Stack data** area, next to **Occasion 1** select **Black**, next to **Occasion 2** select **Asian** and next to **Stacked into** select **resp**
- Click the **Split** button
- Click **No** on the message that appears asking whether you want to save the worksheet




(For more details on what this window is doing see the *User's Guide* chapter 13).

If we now look at the **Data** window again, we can see that the values in the **resp** column have changed:




| | resp(13720) | resp_indicator(1 | bcons.1(13720) | cell_long(13720) | School_long(137 | LEA_long(13720) |
|---|---|---|---|---|---|---|
| 1 | 0.194 | Black | 1.000 | 1.000 | 1.000 | 1.000 |
| 2 | 0.326 | Asian | 1.000 | 1.000 | 1.000 | 1.000 |
| 3 | 0.154 | Black | 1.000 | 2.000 | 1.000 | 1.000 |
| 4 | 0.287 | Asian | 1.000 | 2.000 | 1.000 | 1.000 |
| 5 | 0.109 | Black | 1.000 | 3.000 | 1.000 | 1.000 |
| 6 | 0.266 | Asian | 1.000 | 3.000 | 1.000 | 1.000 |
| 7 | 0.089 | Black | 1.000 | 4.000 | 1.000 | 1.000 |
| 8 | 0.215 | Asian | 1.000 | 4.000 | 1.000 | 1.000 |
| 9 | 0.121 | Black | 1.000 | 5.000 | 1.000 | 1.000 |
| 10 | 0.199 | Asian | 1.000 | 5.000 | 1.000 | 1.000 |
| 11 | 0.133 | Black | 1.000 | 6.000 | 1.000 | 1.000 |
| 12 | 0.182 | Asian | 1.000 | 6.000 | 1.000 | 1.000 |
| 13 | 0.090 | Black | 1.000 | 7.000 | 1.000 | 1.000 |
| 14 | 0.209 | Asian | 1.000 | 7.000 | 1.000 | 1.000 |
| 15 | 0.172 | Black | 1.000 | 8.000 | 2.000 | 1.000 |
| 16 | 0.290 | Asian | 1.000 | 8.000 | 2.000 | 1.000 |

It now contains the proportions that we required: for example its first value, 0.194, is the proportion of Black students in the 2002 cohort in School 1 in LEA 1.

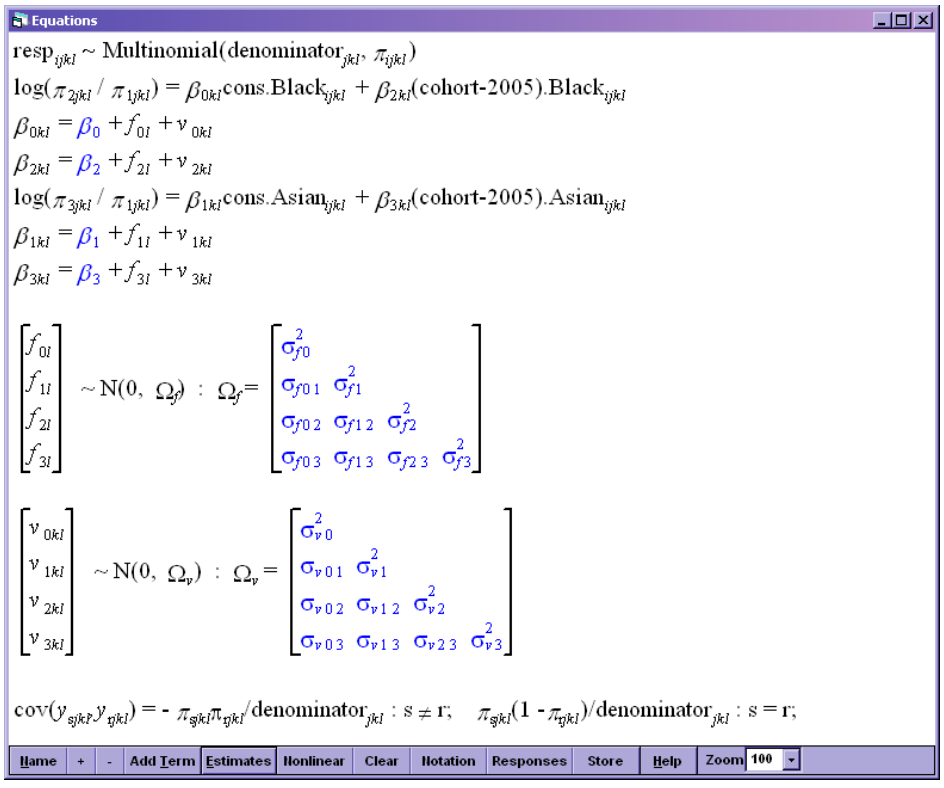
## Exercise 8

a. Look at the columns containing the original data to verify that this proportion is indeed 0.194

b. What is the value of **resp** on row 200? What does this value refer to (what is its interpretation)?

We now need to specify the denominator for the multinomial model (recall that this is the number which each proportion is based on, in our case the total number of students in each cohort in each school) and add some explanatory variables

- In the **Equations** window, click on the red $n_{jkl}$
- Select **denominator** from the drop-down box
- Click **Done**
- Click the **Add Term** button at the bottom of the **Equations** window
- From the variable, select **cons** and click **add Separate coefficients**
- Click the **Add Term** button again and select **Cohort** from the variable
- Under **centring** select **around value** and type **2005** in the box to the right
- Click **add Separate coefficients**
- Click on each of the four explanatory variables (**cons.Black**, **cons.Asian**, **(cohort-2005).Black** and **(cohort-2005).Asian**) in turn, and in the window that appears, leave **Fixed Parameter** ticked and tick **l(LEA_long)** and **k(School_long)** as well, then click **Done**
- Click the **Estimates** button




In the model we have set up, the log of the proportion Black versus the proportion White and

the log of the proportion Asian versus the proportion White are each modelled by a different equation. Both these equations have an intercept (the terms with **cons** in) and a time trend (the terms with **(cohort-2005)** in), but because we have a different equation for each category, we can have different coefficients for these terms for each category. For example, in the year 2005 (when **cohort** $- 2005 = 0$), there could be on average across schools a greater proportion of Asian than of Black students, which would mean a higher intercept for Asian than for Black students, i.e. we would estimate $\beta_1$ as larger than $\beta_0$. It could be that on average the proportion of Black students is increasing each year, which would give us a positive estimate for $\beta_2$, while the proportion of Asian students is decreasing, which would give us a negative estimate for $\beta_3$.

We have put LEA and school level random effects on all four of our explanatory variables. So in 2005, each LEA is allowed to have a different proportion of Black students (the LEA proportions vary around the overall mean), and within the LEAs, each school is allowed to have a different proportion of Black students (the school proportions vary around the LEA means); and the same is true for the proportions of Asian students. Each LEA is also allowed to have a different time trend in the proportion of Black students (the LEA time trends vary around the overall time trend), and each school is allowed a different time trend in the proportion of Black students too (the school trends vary around the LEA trends); again, the same is true for the proportions of Asian students. This has led to four variance terms at each level (on the diagonals of the two variance matrices), and six covariance terms. Once again when we have obtained our estimates we will plot a graph to examine the segregation instead of trying to interpret all the parameters separately (although we will have a look at some of them). In addition to these school and LEA level random effects, the last line of the Equations window specifies the cohort level variation– this allows each cohort's proportions to vary around the school trends. Its structure and the size of the variances and covariances at this level are fixed because we are estimating a multinomial model.

Note that when we added the explanatory variables, and when we specified the denominator, we used variables from our original dataset (**cons**, **Cohort** and **denominator**). MLwiN has created expanded versions of these as we can see if we look in the **Names** window.

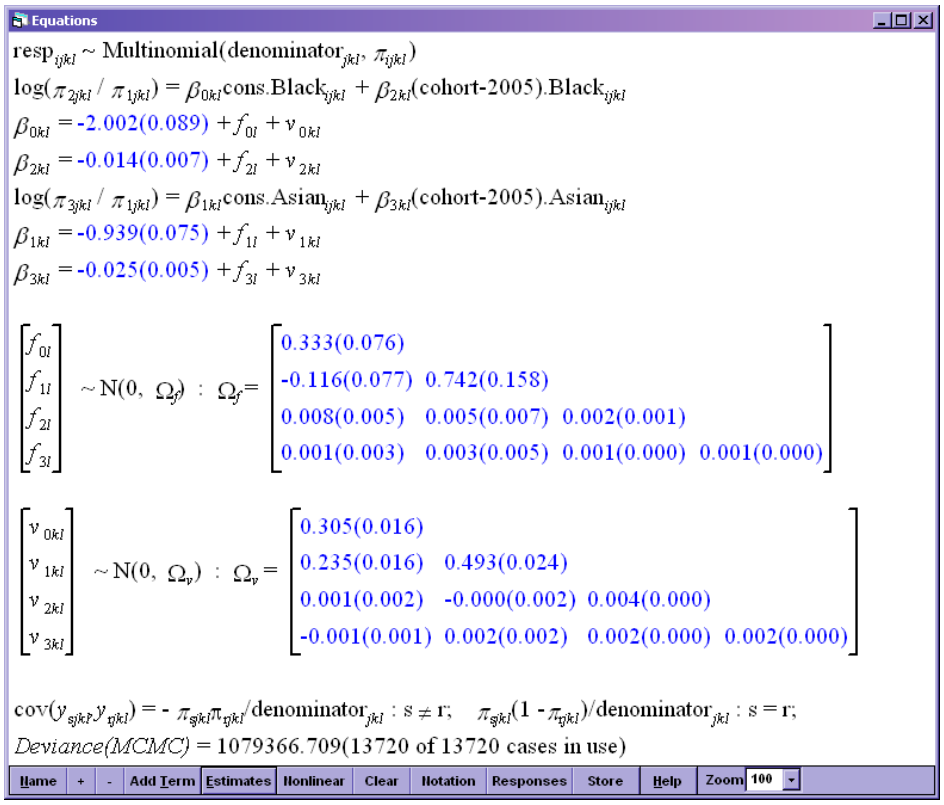
### Exercise 9

Identify these expanded variables in the **Names** window and view them in the **Data** window. Describe how the original variables have been altered to produce these new variables

We will now run the model:

- Click the **Nonlinear** button at the bottom of the **Equations** window
- Click **Use defaults** and **Done**
- Click the **Start** button
- Change the estimation method to MCMC and make sure the burn-in is set to 500 and the monitoring chain length to 5000

- Click **Start** again
- Click the **Estimates** button

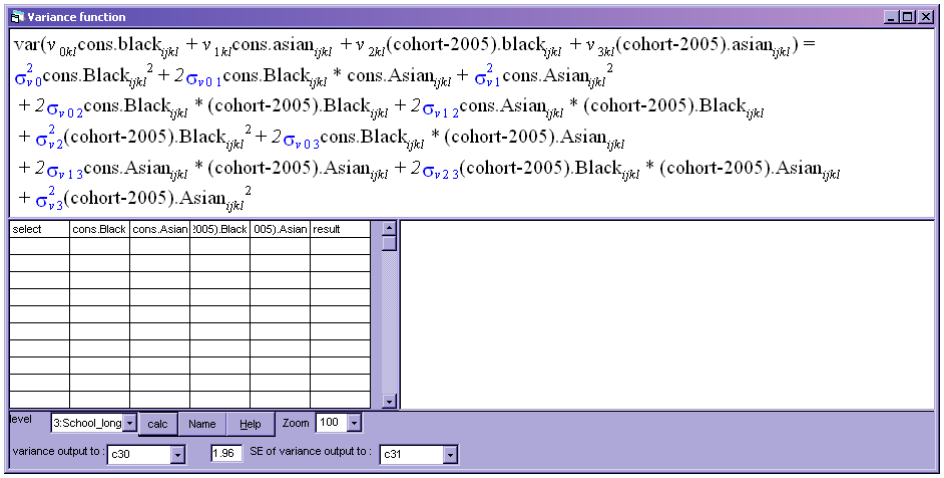



## Exercise 10

Are the overall proportions of Black and of Asian students relative to the overall proportion of White students increasing or decreasing over time?

Once again, our segregation is not a single number but a function of cohort (in fact, two functions of cohort: one for **Black** and one for **Asian**)

- From the **Model** menu select **Variance function**
- From the **level** drop-down box select **3:School_long**
- From the **variance output to:** drop-down box select **c30**
- From the **SE of variance output to:** drop-down box select **c31** and make sure the box to the left says **1.96**
- Click **calc**




## Exercise 11

Write down the simplified equation for the variance for the Black rows of the dataset and the equation for the variance for the Asian rows of the dataset. (Recall how **cons.Black** and the other explanatory variables are constructed: when they take on values of 0 and what values they take on when they are not 0). Are there any variance or covariance terms which do not feature in the equation for the Black rows or the equation for the Asian rows?

---

- Repeat for the LEA level, putting the variance in c32 and the standard errors of the variance in c33

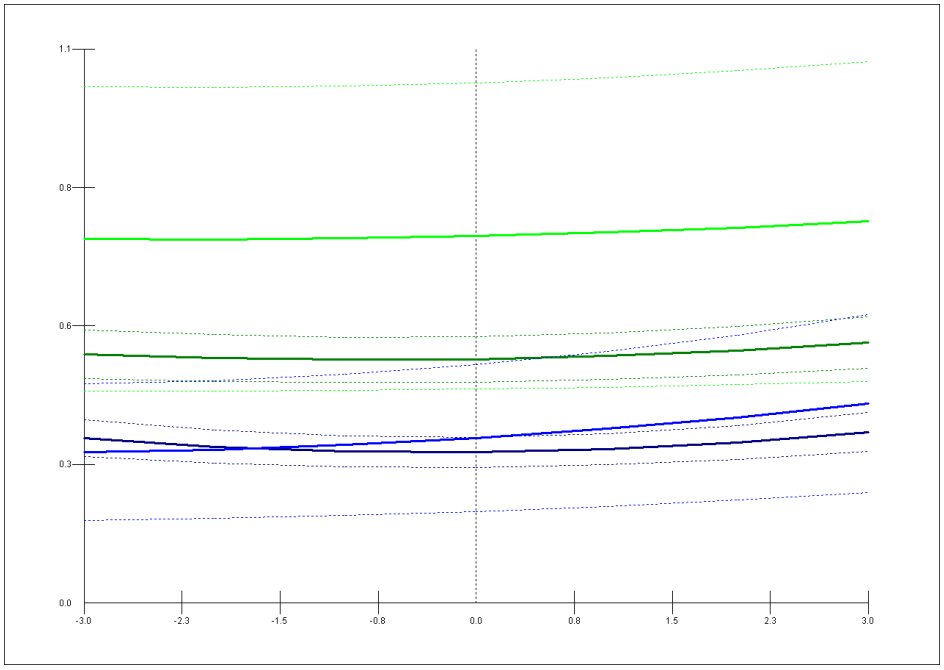
We'll plot these functions against cohort, but first we need to create two new variables **Black_filter** and **Asian_filter**. This is because we need to plot separate lines for Black and Asian, but the output of our variance function at each level is stored in one column, not one column for Black and one column for Asian. **Black_filter** takes the value 1 for Black rows and 0 for Asian rows, and we will use it to ask for only the Black values to be plotted. **Asian_filter** takes the value 1 for Asian rows and 0 for Black rows, and we will use it to ask for only the Asian values to be plotted.

- In the **Command interface** window type the following:

```
► calc c27 = 'resp_indicator' == 2
► calc c28 = 'resp_indicator' == 3
► name c27 'Black_filter'
► name c28 'Asian_filter'
```

- From the **Graphs** menu select **Customised Graph(s)**
- From the drop-down box in the top left corner select **D2**

- Fill out the **plot what?** tab with the following selections: **y**: c30, **x**: (cohort-2005).Black, **filter**: Black_filter, **plot type**: line
- Fill out the **plot style** tab with the following selections: **colour**: 1 blue, **line thickness**: 3
- Fill out the **error bars** tab with the following selections: **y errors+**: c31, **y errors-**: c31, **y error type**: lines
- In the grid in the left hand part of the window, click on the second row
- Fill out the tabs in the same way except using **(cohort-2005).Asian** in place of **(cohort-2005).Black**, **Asian_filter** in place of **Black_filter**, and **2 green** in place of **1 blue**
- Click on the third and fourth rows in the grid and fill out the tabs in the same way as for the first two rows, but using **c32** instead of **c30**, **c33** instead of **c31**, **9 l. blue** instead of **1 blue** and **10 l.green** instead of **2 green**
- Click **Apply**
- Rescale the $y$-axis to start at 0




The blue lines show Black segregation from White and the green lines show Asian segregation from White (the dotted lines give the 95% confidence intervals; note that just as in the binomial case we could obtain better, nonparametric, confidence intervals using the MCMC parameter chains but for time reasons we won't do this now). The lighter lines are the LEA level segregation and the darker lines are the school level segregation.

For both ethnicities, segregation from White is significant at both the LEA and the school level across the whole time period of our data. We can see this because none of the confidence intervals

overlap 0. The school level segregation of Black from White and the LEA level segregation of Black from White seem to be roughly equal. There is a bigger gap between the school and LEA level segregation for Asians but this may not be significant (to test the significance we would have to compare to a model constraining the random effects at school and LEA level to be equal for Asians, but it's not possible to fit this using MCMC in MLwiN). Black segregation from White is lower than Asian segregation from White at the school level, across the whole time period.

Some of the lines seem to show an upward trend while others seem roughly horizontal; the steepest trend appearst to be in Black LEA level segregation. We will test this in a moment by comparing the DIC for this model to the DIC for a model without the time trend, but before we do this we will finish looking at this model (because to calculate the DIC for the model without the time trend we will need to get rid of this model).

Let's now return to the **Equations** window and think a bit more about some of the random parameters. At each level, three random parameters went into the variance function for Black, and three random parameters went into the variance function for Asians (e.g. for Black LEA level segregation it was variance of Black intercepts, varaince of Black slopes and covariance between Black intercepts and Black slopes: $\sigma^2_{v0}$, $\sigma^2_{v1}$ and $\sigma_{v01}$). None of the covariances between Black terms and Asian terms were included in the variance functions. This is because in the variance functions the covariance terms are multiplied by both associated explanatory variables. For the Black rows of the expanded dataset the Asian explanatory variables are 0 and for the Asian rows of the expanded dataset the Black explanatory variables are 0. So for every row of the dataset, the covariances between Black and Asian terms were multiplied by 0 when calculating the variance functions.

These covariances therefore do not enter into our measure of Black segregation from White or of Asian segregation from White. This seems substantively reasonable. However these covariances are of interest– they tell us something about Black segregation from Asian (equivalently Asian segregation from Black).

For example, let's consider the second term in the LEA level covariance matrix, $-0.116$. This is the covariance between Black intercepts and Asian intercepts. Because it is negative, this covariance implies that LEAs with a greater proportion of Black students are likely to have a smaller proportion of Asian students, and LEAs with a greater proportion of Asian students are likely to have a smaller proportion of Black students. This covariance is not in fact significant; if it were it would suggest that as well as Blacks being segregated from Whites and Asians being segregated from Whites, Blacks are segregated from Asians. This would imply that many LEAs fall into one of these three patterns: comparatively many White students but comparatively few Black or Asian, comparatively many Black students but comparatively few White or Asian, or comparatively many Asian students but comparatively few Black or White students.

By constrast, the covariance between the Black and Asian intercepts at the school level is positive (and significant) with a value of 0.235. This means that schools which have a greater proportion of Black students also have a greater proportion of Asian students and vice versa. So Black students are segregated from White and Asian students are segregated from White, but Black students are not segregated from Asian. This suggests that schools vary quite a bit in their proportion nonWhite

but given this proportion there is not much variation in the ratio of Black to Asian students.

Finally, we will test whether the trends in segregation that we observed in our graph are significant. We will first calculate the DIC for the current model

**Exercise 12**

Calculate the DIC for the model

**Exercise 13**

Run the model without the time trend and calculate the DIC. What can you say about the changes in segregation over time that we observe?

## References

Rasbash, J., Steele, F., Browne, W. and Goldstein, H. (2008) *A User's Guide to MLwiN* University of Bristol

Browne, W, (2008) *MCMC estimation in MLwiN* University of Bristol

(both available to download for free from http://www.cmm.bris.ac.uk/MLwiN/download/manuals.shtml)

## Answers to exercises

1. The 2005 cohort of School 29 in LEA 2 can be found on row 200 of the dataset. In this cohort there are 66 White students, 23 Black students and 7 Asian students.

2. In the Names window, the minimum value of **White** is listed as 5.263158E-02. This means $5.263158 \times 10^{-2}$, which (rounded) is 0.053. So the smallest proportion of White students in a cohort is 0.053, or around 1 in 20.

3. Effective sample size for $\sigma^2_{v0}$ is 877; effective sample size for $\sigma^2_{u0}$ is 4485 (to get the effective sample size, click on the graph for the parameter in the **Trajectories** window to bring up the **MCMC diagnostics** window for that parameter- the effective sample size is given at the bottom of the window). These are both reasonable sizes so we can have confidence in the estimates for these parameters.

4. Looking in the **Trajectories** window, the estimates for all the parameters appear to have converged by 1,000 iterations ($\sigma^2_{u0}$ and $\sigma^2_{v0}$ for example look as though they may possibly not have converged within the burn-in period). However the effective sample size for $\beta_0$ for example is only 7 which suggests it might be beneficial to run for more iterations. If we run for a total of 50,000 iterations and increase the burn-in by 1,000 iterations, the value of $\beta_0$ decreases (to -0.531) and its standard error increases (to 0.107), but the other estimates do not change much. Looking at the trajectory for $\beta_0$ after 50,000 iterations we still might want to run for longer to be sure that it has converged and that we have enough iterations for a good estimate.

5. Yes, when we ran the model without **cohort**, we did indeed use confidence intervals obtained in this way. For that model our measure of segregation at each level was much simpler, since it consisted only of the variance of the random intercepts at that level, and so we did not need to calculate a variance function for each iteration but could take our confidence intervals directly as the 2.5th and 97.5th percentiles of the estimates for the variance parameter, which we obtained from the **MCMC diagnostics** window. We did not need to calculate percentiles separately for each value of **cohort** because the segregation measure was constant across cohorts.

6. No, if we run for example for 50,000 iterations the standard errors of the random parameters are about the same. This is as we expect: if we have not run MCMC for long enough we may find that the standard error increases when we run for longer (as we saw for $\beta_0$ in the previous question), but we do not expect that the standard error will decrease. (On the other hand if we have not used a long enough burn-in and then we increase the burn-in we may well find the standard error decreases).

7. Mean proportion of Black students: 0.10; mean proportion of Asian students: 0.29; mean proportion of White students: 0.61. To obtain these values, select **Averages and Correlations** from the **Basic Statistics** menu, check that **Averages** is selected under **Operation**, select **White**, **Black** and **Asian** in the box to the right and click **Calculate**.

8. (a) To do this, in the **Names** window highlight **LEA**, **School**, **Cohort** and **Black** and click **Data**. In the **Data** window, the 2002 cohort in School 1 in LEA 1 should be in the first row, and the value of **Black** in the first row souldd be 0.194.

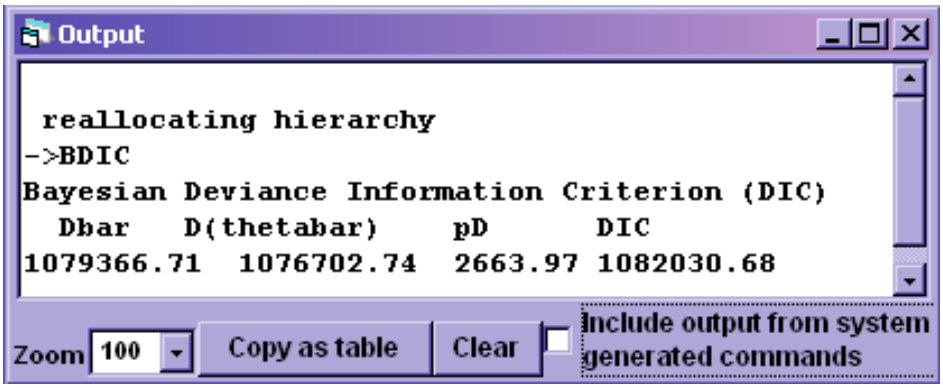
(b) The value of **resp** in row 200 is 0.346. The value of **resp_indicator** is **Asian**, the value of **School_long** is 15, the value of **LEA_long** is 1 and the value of **cell_long** is 100. So 0.346 is the proportion of Asian students in School 15 in LEA 1 in the cohort where **cell** = 100. To find out which cohort this is, bring up the data for **cell** and **cohort**, and scroll down till you find **cell** = 100 (because of the way this variable was created, this should be on row 100). The value of **cohort** on this row is 2003. So 0.346 is the proportion of Asian students in the 2003 cohort of School 15 in LEA 1.

9. The expanded variables are **denom**, **cons.Black**, **cons.Asian**, **(Cohort-2005).Black** and **(Cohort-2005).Asian**. **denom** has been created from **denominator** by simply duplicating every value. This makes sense: **denominator** referred to the original dataset with one row per cohort, whereas **denom** refers to the expanded dataset with two rows per cohort, so each cohort size needs to appear twice. **cons.Black** has been created by first duplicating **cons**, in the same way that **denominator** was duplicated to become **denom**, and then setting the value to 0 on every Asian row. **cons.Asian** was created in the same way but instead setting the value to 0 on every Black row. **(Cohort-2005).Black** was created by subtracting 2005 from **Cohort**, then duplicating the values, and then setting the value to 0 on every Asian row. **(Cohort-2005).Asian** was created in the same way but instead setting the value to 0 on every Black row. Multiplying these four explanatory variables by their coefficients and adding them up will now result in the two separate equations we see in the Equations window: for Black rows the values of **cons.Asian** and **(Cohort-2005).Asian** are always 0 so we get $\beta_{0kl}$**cons.Black** + $\beta_{2kl}$**(Cohort-2005).Black**, ans similarly for the Asian rows.

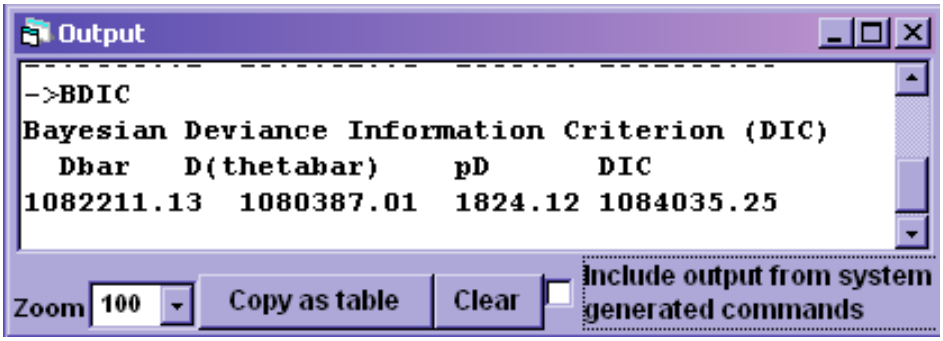
10. To answer this question we need to look at the fixed part coefficients of time, which is **(cohort-2005)**. So we need to look at the estimates of $\beta_2$ and $\beta_3$, which are $-0.014$ and $-0.025$ respectively. We are modelling the log of each proportion divided by the proportion in the reference category, so we would need to do some transformations to these values to provide an exact interpretation; however we can note that they are both negative (though $\beta_2$ is not significant) and so both the overall proportion of Black students compared to the overall proportion of White students and the overall proportion of Asians students compared to the overall proportion of White students are decreasing over time.

11. Equation for Black rows: variance $= \sigma^2_{v0} + 2\sigma_{v02}(\mathbf{Cohort}_{jkl} - 2005) + \sigma^2_{v2}(\mathbf{Cohort}_{jkl} - 2005)^2$. Equation for Asian rows: variance $= \sigma^2_{v1} + 2\sigma_{v13}(\mathbf{Cohort}_{jkl} - 2005) + \sigma^2_{v3}(\mathbf{Cohort}_{jkl} - 2005)^2$. To arrive at these equations, remember that in the Black rows, **cons.Asian** and **(Cohort-2005).Asian** are 0, so any terms containing these will be 0 for the Black rows and thus drop out. Similarly for the Asian rows any terms including **cons.Black** or **(Cohort-2005).Black** drop out. For the Black rows, **cons.Black** is just 1 and **(Cohort-2005).Black** is just **Cohort** $-$ 2005; for the Asian rows **cons.Asian** is just 1 and **(Cohort-2005).Asian** is just **Cohort** $-$2005. Covariance terms which don't appear in either equation are those multiplying a term involving Black and a term involving Asian (since this product will be 0 on all rows): $\sigma_{v01}$, $\sigma_{v12}$, $\sigma_{v03}$ and $\sigma_{v23}$

12. You should get:




13. For the model without the time trend, the output when we calculate the DIC is




Thus the model without the time trends shows a decrease of around 2000 compared to the model with the time trends, indicating that at least one of the trends is significant. To establish which trends are significant and which are not (or whether they are all significant) we would need to compare the DICs for models including different combinations of time trend.