# Chapter 12: Individual Participant Data Meta-Analysis

# Example IPD data and code

As a companion to Chapter 12 data and R code are provided to allow you to run some of the IPD meta-analysis examples from the chapter. The code assumes some general facility with R. You may need to use the R help system to understand how each function works if you are unfamiliar with them.

As the original PARIS data on aspirin to prevent pre-eclampsia are confidential and cannot be shared this example uses an artificial dataset called “BRUSSELS”.

The BRUSSELS data is constructed to mimic the PARIS IPD analysis as used in Chapter 12. The numbers of trials and the sample sizes are the same as in the original review. However, the outcome and other covariate data are synthetic. They are intended to give comparable, but not identical, results to the true PARIS data, as seen in Figure 12.2 and Tables 12.1 and 12.2.

The structure of the BRUSSELS data is designed to mimic a typical, if simplified, IPD meta-analysis data file.

## The BRUSSELS data

The data for this exercise are in the file “Brussels artificial data.csv”

The data can be viewed in any spreadsheet package (e.g. Excel), or opened in R using the **read.csv** command.

The BRUSSELS data contains the following columns: each row represents the data for one woman:

Trial: The trial name (using the same names as PARIS)

PatientID: A patient identifier

Treatment: The treatment received (Aspirin or Placebo)

Preeclampsia: Indicates whether the woman developed (1) or did not develop (0) pre-eclampsia during the trial

Diabetes: Indicates whether the woman had (1) or did not have (0) diabetes or hypertension at Hypertension: randomisation

GestatonalAge: Gestational age of the baby at randomisation (in weeks)

MaternalAge: Maternal age at randomisation (in years)

## Meta-analysis code for R

**General notes**

Example R code to analyse the BRUSSELS data using both one-stage and two-stage IPD meta-analysis methods is provided in the R code file “Analyse Brussels artificial data.R”.

If you wish to use this code to analyse the BRUSSELS data we recommend installing and using the RStudio software [see: https://posit.co/download/rstudio-desktop/], so the code can easily be read and run in R.

Performing meta-analysis with R requires installing and loading some additional code packages. Packages for standard (two-stage) meta-analysis include **meta** and **metafor**. We use **meta** here as it generally produces output that is easy to read and interpret. To perform one-stage meta-analyses generalised linear mixed models are needed: the **lme4** package contains the functions needed.

IPD meta-analyses commonly requires substantial data editing, summarising, and the facility to repeat analyses (such as performing an analysis in each trial as the first stage of a two-stage analysis). Although these are all possible in standard R, packages such as **dplyr** and **tidyr** (for data editing) and **purr** (for iteration), which are all part of the wider **tidyverse** family of packages, can aid IPD analysis. The example R code supplied makes use of all these packages.

**Getting started**

Throughout this exercise you can either try and create R code for yourself, or just follow though the supplied code in “Analyse Brussels artificial data.R”

*First install and load the libraries you will need (e.g.* ***meta*** *and* ***lme4****)*

*Read in the data (from the “Brussels artificial data.csv” file) using the* ***read.csv()*** *command.*

It will be useful to convert the Treatment variable from text (“Aspirin” / “Placebo”) to numbers:

*Create a new variable “Treatcode” which converts the Treatment data to 0s for placebo and 1s for aspirin.*

### The effect of aspirin on pre-eclampsia rates

***Two-stage meta-analysis to investigate the effect of aspirin.***

To perform a two-stage meta-analysis we first need to create summary data from our IPD. There are many ways to do this in R. The supplied code uses commands from the **dplyr** library, as this offers a clear and succinct method for summarising data.

*Create a new dataset from the IPD, with one row per trial and columns for:*

* *the number of patients in the placebo arm,*
* *number in the aspirin arm,*
* *number of pre-eclampsias in the placebo arm,*
* *number of pre-eclampsias in the aspirin arm.*

This summary data can be used to perform a meta-analysis. The two main functions for meta-analyses of binary data are **metabin** from the **meta** package, or **rma** from the **metafor** package.

*Perform a meta-analysis of the impact of aspirin on pre-eclampsia incidence using your summary data and either* ***metabin*** *or* ***rma****.*

*Produce a forest plot of your results by using the* ***forest*** *function.*

The results of this two-stage analysis should be similar to the results in Table 12.1 Of Chapter 12. The forest plot should be similar to the one in Figure 12.2 (but note that Figure 12.2 was produced in different software).

***One-stage meta-analysis***

One-stage meta-analysis does not require any editing of the data: you can work directly with the main IPD data set. We use generalised linear models to perform one-stage meta-analyses. For fixed effect analyses this is done using the **glm** function in R.

*Use* ***glm*** *to regress pre-eclampsia status against treatment and trial. Remember that pre-eclampsia is a binary variable, so you will need to declare this (using family=binomial) in your code.*

To add random effects, we switch to generalised linear mixed models, which are available in the **lme4** package. The function **glmer** is used for these models.

*Use the* ***glmer*** *function to extend your fixed effect model to add a random treatment effect across trials.*

Many one-stage IPD models are possible. A common choice is to use random effects for the trial intercepts, rather than separate parameters for each trial. This reduces the number of parameters in the model and usually makes the analysis more stable, particularly where trials are small. The random trial intercepts and the random treatment effects can be made to be correlated or uncorrelated with each other (although this rarely has much impact in practice).

*Modify your model code in* ***glmer*** *to include random trial intercepts.*

A disadvantage of **glm** and **glmer** is that the output printed on screen is limited, and results are given as log odds ratios and standard errors.

*Use the* ***summary*** *command to see better output from your one-stage models.*

*Convert the log odds ratios and standard errors for treatment effects into odds ratios and 95% confidence intervals.*

Your results should be similar to those in Table 12.1 of Chapter 12, but will not be identical.

*If you are familiar with writing functions in R, try creating a function to automatically extract odds ratios and 95% confidence intervals from* ***glm*** *or* ***glmer*** *output. An example function is provided in the supplied R code.*

It is possible to calculate relative risks in one-stage models by using the **family=binomial(link=”log”)** option in your **glm** or **glmer** code. However, this can lead to model convergence problems, particularly with more complex models.

### The impact of covariates on aspirin effectiveness

We will now investigate how the four covariates provided might alter the effectiveness of aspirin. Note that two covariates (diabetes, hypertension) are dichotomous covariates, and two (gestational and maternal age) are continuously distributed.

**One-stage meta-analyses of interactions**

We start with one-stage models, as these are easier to perform. Adding interactions to R models is done using an asterisk in the model formula. So, to add an interaction between treatment and maternal age for pre-eclampsia our model code is **Preeclampsia ~ Treatcode\*MaternalAge.** Otherwise, the **glm** or **glmer** code is unchanged.

*Adapt your random-effect one-stage models for treatment effect to include interaction terms for diabetes, hypertension, gestational age and maternal age (separate models for each covariate).*

*Use* ***summary*** *to view the results of each model, and extract the odds ratios and 95% confidence intervals for the interaction terms.*

For dichotomous and categorical covariates it is advisable to declare these as factors in your model (e.g. **Preeclampsia ~ Treatcode\*factor(Diabetes)**). This ensures R treats them as categories, rather than continuous variables.

For continuous variables it may help to “centre” the variables around their mean value IPD, so that the estimate of the treatment effect is not at the (usually implausible) zero value of the variable (such as a maternal age of zero). Examples are provided in the supplied R code.

***Two-stage analysis, subgroup analysis***

Subgroup analysis of IPD is more complex because of the need to summarise the data by trial, treatment arm and covariate subgroup first.

*Adapt the code you used for two-stage meta-analysis to calculate numbers of patients and numbers with pre-eclampsia by trial, treatment arm AND presence/absence of diabetes.*

*Separate this summarised data into a data set for women WITH diabetes and a dataset for women WITHOUT Diabetes.*

You should notice that the dataset with diabetes has some trials with no data, or no cases of pre-eclampsia, because not many women in the IPD data had diabetes. this complicates subgroup analysis, as treatment effects are inestimable in some trials. There are various ways around this.

*For this example, use the simple option of excluding from your summarised data any trial with no data, or with fewer than 2 cases of pre-eclampsia.*

*Use* ***metabin*** *to perform meta-analyses on the WITH diabetes and the WITHOUT diabetes data sets.*

*Repeat this whole process to perform a subgroup analysis for hypertension.*

Continuous covariates must be divided into subgroups first before performing the analysis. This is not recommended as a method of meta-analysis when IPD are available, as a model treating the data as continuous should be preferred. The analyses are presented here for information only.

*In your IPD data create two new variables. Create variable LowGest by dividing Gestational Age into women with a gestational age < 20 weeks and >= 20 weeks.*

*Similarly, create a variable MatAgeCategory to divide Maternal Age into 3 groups; under 20, 20-35 and over 30.*

*Repeat the process used for Diabetes to perform subgroup analyses for gestational age and maternal age.*

The **dplyr** package includes functions (**group\_map()** and **nest()**) which allows for performing meta-analyses directly on grouped data sets, which can streamline two-stage IPD analysis. An example is given in the supplied R code.

***Two-stage analysis, meta-analyses of interactions***

A further approach to two-stage IPD meta-analysis is to model treatment-covariate interactions separately in each trial in the first stage, and then meta-analyse the estimated interaction effects across trials in the second stage. This is discussed briefly in Section 12.4.1 of Chapter 12, but no results were presented.

The general process for this approach in R is:

1. For one trial, use **glm** to model the impact of treatment and covariate, with an interaction between them (as in the one-stage analysis of interactions)
2. Extract the estimated log odds ratio and its standard error for the interaction term from the model results.
3. Repeat steps 1 and 2 for all trials in the IPD.
4. Meta-analyse the extracted interaction estimates using **metagen** from the **meta** package.

Doing this requires experience in writing R functions and how to loop or iterate functions across trials. An example using the **map** function family from the **purrr** package is given in the supplied R code.