RECAP

"Research on European Children and Adults born Preterm"

Grant Agreement number: 733280

Deliverable 5.2

Workshop for project members on conceptual framework

Workpackage:	WP 5
Task:	T 5.2
Due Date:	30st September 2017 (M9)
Actual Submission Date:	30st September 2017 (M8)
Last Amendment deliverable:	15th January 2018
Project Dates:	Project Start Date: January 01, 2017
	Project Duration: 51 months
Responsible partner:	TNO
Responsible author:	Prof. dr. Stef van Buuren (TNO)
Email:	Stef.vanBuuren@tno.nl
Contributors:	Manon Grevinga (TNO), Sylvia van der Pal (TNO), Aurélie Piedvache (INSERM)

Project funded by the European Commission within H2020-SC1-2016-2017/H2020-SC1-2016-RTD		
Dissemination Level		
PU	Public	
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	CO

Document History:

Version	Date	Changes	From	Review
V0.1		Deliverable template	Maaike Beltman	
V1.0	August 1, 2017	Workshop slides	Manon Grevinga	Stef van Buuren
V1.1	September 4, 2017	Description of workshop	Manon Grevinga, Sylvia van der Pal, Stef van Buuren	
V1.2	September 15, 2017	Summary, lay-out and appendices	Sylvia van der Pal, Manon Grevinga	Stef van Buuren
V1.3	September 19, 2017	Content of workshop	Manon Grevinga, Aurélie Piedvache	Stef van Buuren, Sylvia van der Pal
V1.4	September 20-22, 2017	Discussion and final check	Stef van Buuren, Sylvia van der Pal	RECAP members (Dieter Wolke, Jennifer Zeitlin, Maaike Beltman, Erik Verrips, Juliane Dittrich)
V1.5	September 28, 2017	Adjust to review comments	Stef van Buuren, Manon Grevinga	

Open Issues

No:	Date	Issue	Resolved
1			

SUMMARY

This Deliverable 5.2 (D5.2) of the RECAP project describes the WP5 "Statistical methods for Individual Patient Data (IPD): Conceptual Framework" workshop that took place on the 4th and 5th of September, in Leiden, the Netherlands.

This deliverable reports on the 6 subjects discussed during the workshop and the discussion that followed.

1. Combining data sets & missing data
2. Multiple imputation
3. Creating comparable variables
4. Developmental milestones
5. Loss to follow-up
6. Multilevel analysis

The powerpoint slides of the lectures and the practical exercises were provided through a website: https://stefvanbuuren.github.io/RECAPworkshop/, and are inserted in appendices 6.1 & 6.2 of this report.

Table of contents

1 INTRODUCTION

1.1 Purpose and Scope

This document describes the "Statistical methods for Individual Patient Data (IPD): Conceptual Framework" workshop that was given on 4th and 5th of September 2017. The workshop gave more (practical) insight in the problems arising from the combined analysis of data from a collection of cohorts that track children who were born very preterm (VPT) or with a very low birth weight (VLBW) as brought together by the RECAP project. This report is deliverable 5.2 (D5.2) of the RECAP project.

Work package 5 of the RECAP project consists of activities to develop adequate statistical methodology needed to solve analytic problems arising from the other work packages. Work package 5 focusses on three problems associated with IPD: *harmonisation*, *loss to follow up*, and *individual prediction*. On the surface these problems appear to differ, but they can all be framed as 'missing data problems'. In each problem, only part of the needed information is observed, whereas other needed information is missing, and the analytic objective to find the missing information based on what we have. Benefits of framing the three IPD problems as a missing data problem include:

1. It may stimulate the use of a common and precise vocabulary for seemingly different problems;
2. As opposed to models, everybody understands data, so it is easier to communicate what exactly the problem is, and how we can attack the problem;
3. There is a general solution of missing data problems – multiple imputation – that nearly always works.

This report describes the workshop of WP5 on the conceptual framework as was described in the previous deliverable (D5.1). This previous deliverable describes several problems that need to be solved when combining data from different sources. Moreover, it outlines how a seven-step approach can be formulated from the missing-data perspective, illustrating how it can be applied to hypothetical questions of scientific interest in RECAP, and shows how a generic quantitative solution can be obtained by multiple imputation.

1.2 References to other RECAP Documents

This document gives an overview of the workshop of WP5 which is based on the conceptual framework as described in deliverable 5.1 (D5.1). The statistical concepts of these deliverables will be implemented in the RECAP statistical analysis platform (WP4), in close collaboration with the other work packages.

1.3 Definitions, Abbreviations and Acronyms

Table 1 List of Abbreviations and Acronyms

Abbreviation/ Acronym	DEFINITION
VPT	Very preterm
VLBW	Very Low birth weight
IPD	Individual Patient Data
MAR	Missing at random
	inverse probability weighting (IPW)
	missing completely at random (MCAR),

2 PREPARATION OF WORKSHOP

- Selection of topics:
    1. Combining data sets & missing data
    2. Multiple imputation
    3. Creating comparable variables
    4. Developmental milestones
    5. Loss to follow-up
    6. Multilevel analysis

For each subject: first theoretical background > then a partical in R (using R markdowns)
Used for background information: D5.1 (copies distributed during workshop & Book & article (2011) by Stef van Buuren about MI/MICE (http://www.stefvanbuuren.nl/publications.html)).
The workshop materials were made available through a website: https://stefvanbuuren.github.io/RECAPworkshop/, and are inserted in appendices 6.1 & 6.2 of this report. On this website the slides, markdowns of the practical exercises and other information could be downloaded.

3 WORKSHOP ON 4 & 5 SEPTEMBER

3.1 Schedule of workshop

Workshop Statistical Methods 4 & 5 September '17, Hotel Tulip Inn, Leiden the Netherlands.
Schedule Monday 4 September

Time	Topic	Remarks
>12:00	LUNCH	
13:00 – 14:30	Combining datasets & missing data	*Theory & practical*
14:30 – 15:00	*Break*	
15:00 – 16:00	Multiple Imputation	*Theory & practical*
16:00 – 16:30	*Break*	
16:30 – 18:00	Creating comparable variables	*Theory & practical*
19:00 - …	*Dinner*	*@ Scarlatti*

Schedule Tuesday 5 September

Time	Topic	Remarks
09:00 – 10:30	Developmental milestones	*Theory & practical*
10:30 – 11:00	*Break*	
11:00 – 12:30	Loss-to-follow-up	*Theory & practical*
12:30 – 14:00	*Lunch break*	
14:00 – 15:00	Multilevel analysis	*Theory & practical*
15:00 – 15:30	*Break*	
15:30 – 17:00	Discussion	

3.2 Participants of workshop

On Monday 4 September 30 participants we present and on Tuesday 5 September 31 RECAP members participated.

3.3 Content of workshop

The topics covered in the workshop of WP5 were:

1. Combining data sets & missing data
2. Multiple imputation
3. Creating comparable variables
4. Developmental milestones
5. Loss to follow-up
6. Multilevel analysis

More information about topics 1 – 4 can be found in Deliverable 5.1 (D5.1). This deliverable was also printed and handed out to the participants of the WP5 Workshop. An explanation of topic 5 (Loss to follow-up) and 6 (Multilevel analysis) is described below.

5/ Loss to follow-up

Loss to follow-up is a major challenge for cohorts of very preterm infants, especially when follow-up times are long or cohorts include a large number of children. Many reasons exist for loss to follow-up, including the death of the child, moving homes, lack of time due to other family obligations, work or not wanting to be reminded of the circumstances of child's birth. Although investigators do their best to minimize the number of non-responders, there are always at least some children that are lost to follow-up. Unfortunately, this loss can undermine the representativeness of estimates and introduce non-differential biases.

The most common approach to managing loss to follow-up has been to analyze data on the responders and to ignore non-responders. When possible, existing data are provided separately to compare characteristics of children lost to follow-up with children included in order to speculate on the potential for bias. However, other strategies can be used in the analyses when information on non-responders is available, based on the assumption that individual data can predict the probability of inclusion.

The first technique uses information from the study population eligible for the follow-up to generate a weight and inverse probability weighting (IPW) can be used for the analyses. Logistic regression model is used to estimate the probability of follow-up with covariates that are hypothesized to be associated with both follow-up and outcome. In this way, if some children with similar characteristics are less likely to respond, children with this co-variable profile who did respond would get a higher weight. Where response rates for a given co-variable profile are high, subjects receive a lower weight.

Another technique is to use multivariate imputation by chained equations (van Buuren & Groothuis-Oudshoorn, 2011). In this case, loss to follow-up is assumed to be missing at random (MAR), i.e. missing data does not depend on the outcome, but is related to some of the observed data (as in the IPW case). Or, they can be missing completely at random (MCAR), and in this last case, non-responders may be due to an external event – such as loss of the questionnaire by the postal service – which is not related to their characteristics. In this technique, factors associated with the probability of follow-up and those associated with outcome, as well as the outcome itself, are used to create multiple full datasets of the cohort with follow-up data. One benefit of this approach is that, if data are missing on the variables used to predict the probability of follow-up, these can be imputed, whereas in the IPW approach, these children would be excluded because a weight could not be calculated.

It should be noted that the two techniques can be merged if there is a lot of missing data on the individual data used to predict the probability of inclusion. In other words, MICE (Multivariate Imputation by Chained Equations in R[1]) can be used to generate a full dataset for use in predicting inverse probability

weights. Both techniques also rely on the assumption that the probability of loss to follow-up can be accurately described by the covariates included in the model.

6/ Multilevel Analysis

When combining data from different cohort studies (as done in the RECAP project) one could make use of three types of analysis:

- Separate model for each cohort study
- Dummy variable indicating the cohort study
- One general model by means of a multilevel model (mixed effects model)

In the first two types of analysis one assumes independence between the subjects, and the multilevel analysis assumes correlation between the cohort studies. In other words, multilevel analysis assumes that a child from one cohort is more similar to a child from the same cohort than a randomly drawn child from one of the other cohort studies.

The multilevel model consists of two parts: fixed effects (the same as in linear regression models) and random effects (allows for differences between the cohort studies). For example, a random intercept allows each cohort study to have its own intercept, while a random slope allows for a different effect of the predictor (e.g. gestational age) per cohort study in the general model. Figures 1, 2, and 3 visualize this concept.

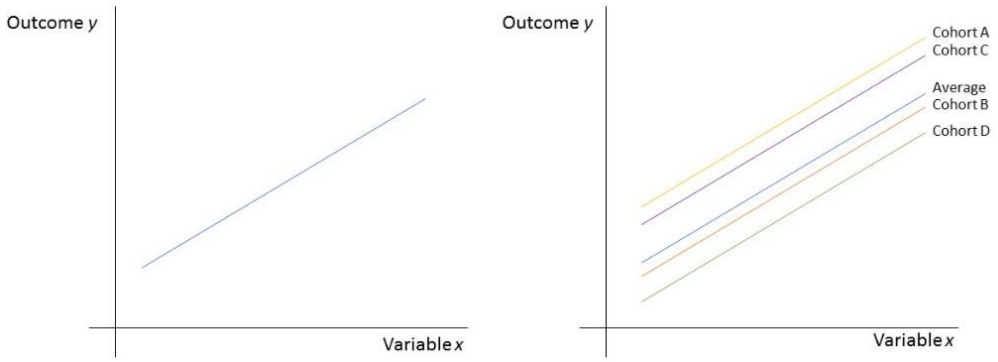


Figure 1 Random intercept model

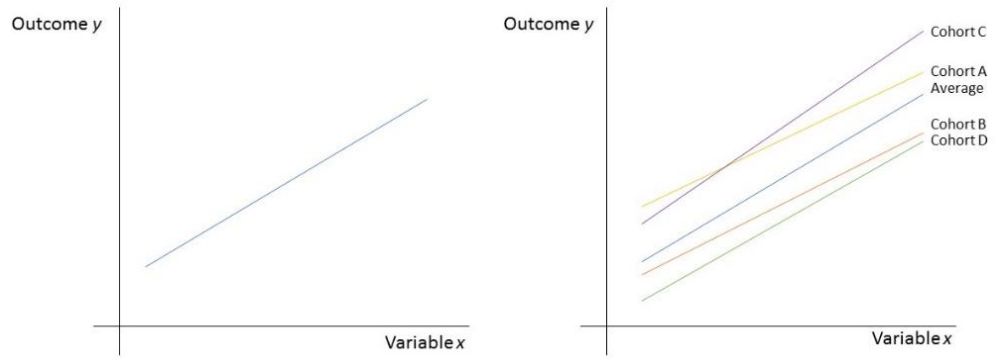


Figure 2 Random intercept and Random slope model

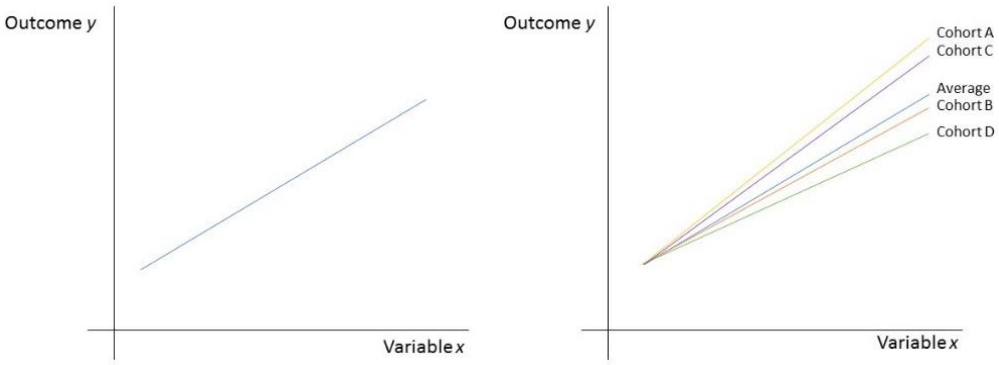


Figure 3 Random slope model (no random intercept)

Variables can be added on multiple levels, see figure 4. An higher level variable is always also allocated to the lower levels. A variable explaining variation within the country (a country-level variable) will also be allocated to the child. A relation we might want to investigate taking into account these different levels if the relation between gestational age and birthweight. Each country might have a different relation between the predictors and outcome, hence having its own model to explain the outcome. Using mixed effects model we can combine the models of all countries to one general model.

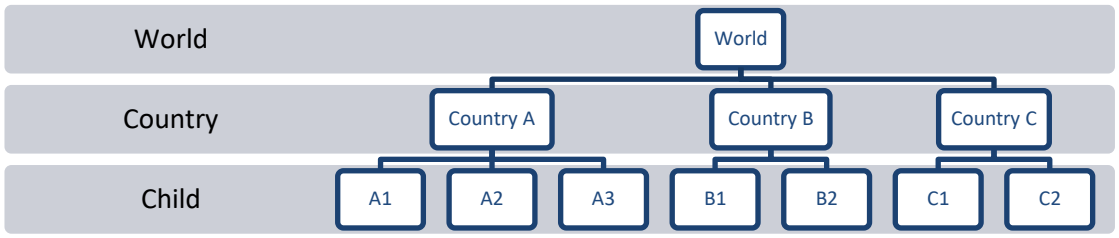


Figure 4 Level structure

3.4 Pictures of the Workshop

Below some pictures of the workshop of WP5 are inserted.

Figure 5. Welcome to the workshop

Figure 6. Explanation of multiple imputation I

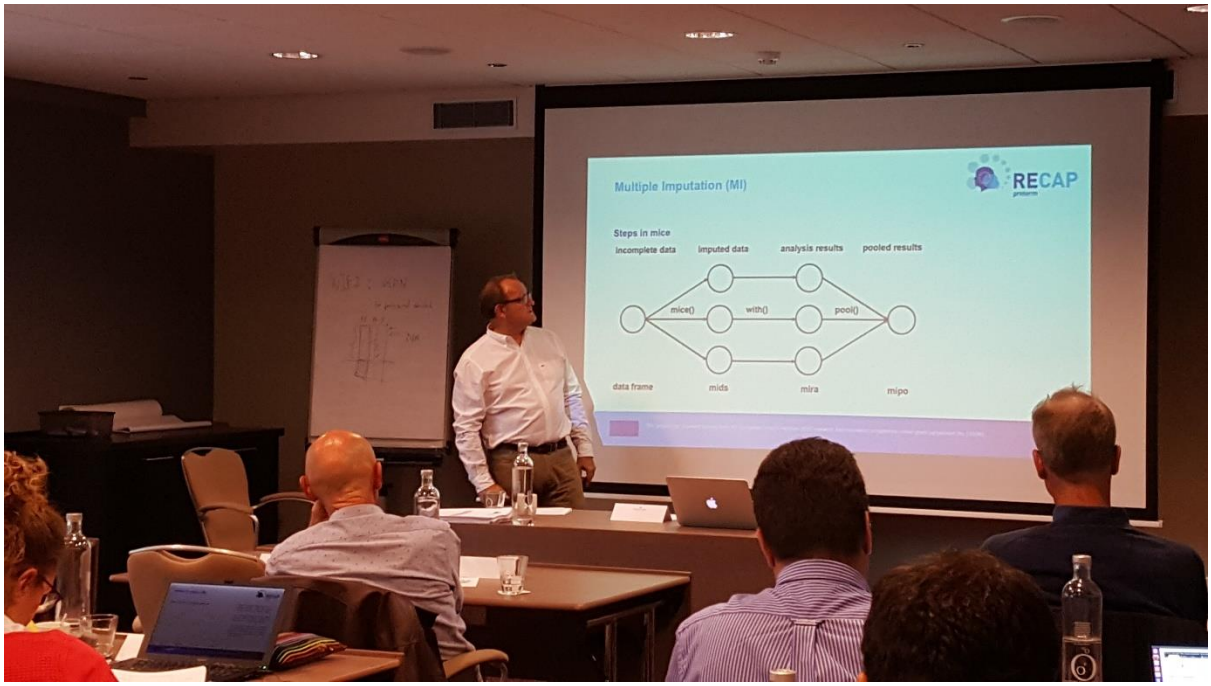


Figure 7. Explanation of multiple imputation II

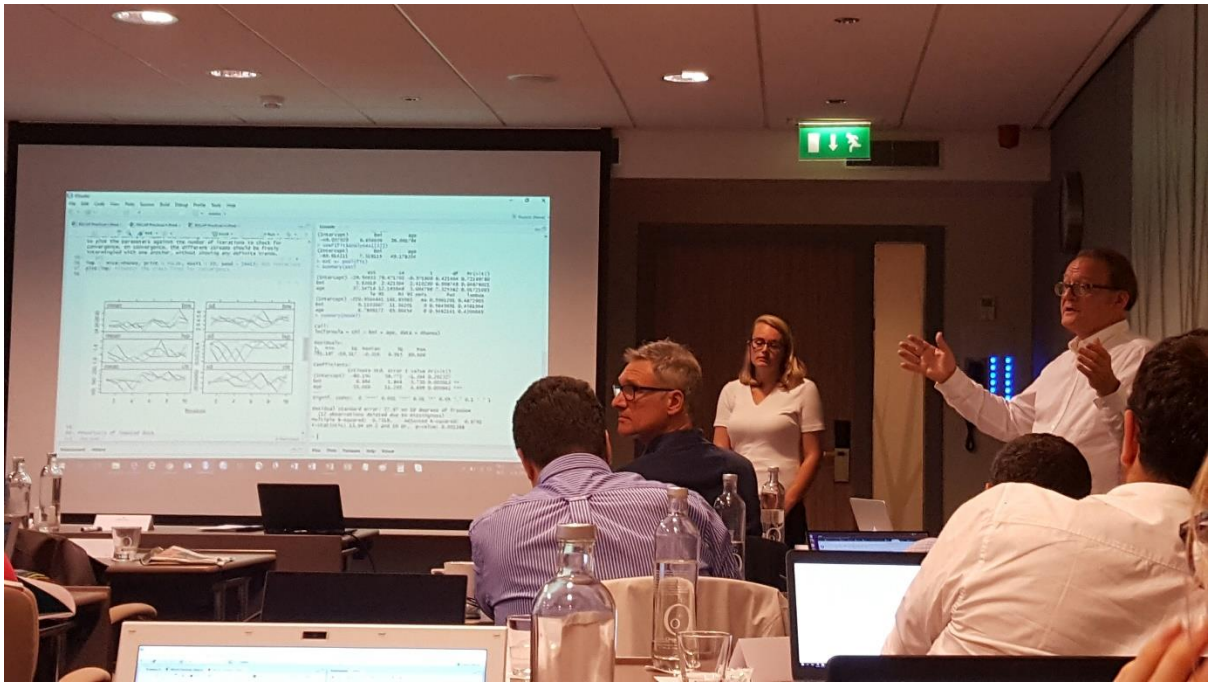


Figure 8. Doing the practical exercises together in R-studio

4 DISCUSSION

There were 31 participants, which is substantial for a technical workshop. The workshop was generally well received. Discussion focused on how to handle cultural/developmental differences between countries, and how longitudinal data with repeated measures at different ages can be compared/combined between cohorts. Below are some points that were raised during the workshop.

- How can longitudinal modelling be done within multiple cohorts that have gathered data at different timepoints?
  There are several ways to handle differential timing. One strategy is to fit time-based models that are relatively insensitive to the exact timing of the measurement, for example, by fitting multilevel or spline-based model. The same model can be fitted to different cohorts, and the parameter estimates can be compared across studies. Another strategy is to set of common time grid, and multiply impute plausible values at those times. This is more work, but allows for a far wider range of analysis options. One particular convenient model for this is the "broken stick" model.
- Can harmonization be done when variables were gathered at different time points?
  Yes, assuming that the interpretation of the measurements does not depend on age. For example, an item like "Can stack two blocks" remains the same irrespective of the age at which we administer it (though – of course – older children will do better). In any case, there is no need for equally advanced or equally old children to do successful harmonization. The more important thing is overlap in instruments.
- Can there be an additional WP5 workshop about how to solve this statistically?
  Longitudinal data analysis is a huge topic, and somewhat independent of harmonization, which I think is the key problem in RECAP. There is a relevant workshop in October in Rotterdam. See http://www.dohad2017.org/sunday-workshops/#strategies, and many universities offer summer courses. Within WP5 only one workshop was planned. We are happy to assist with longitudinal analysis in the other WP's on a case by case basis.
- Is there a thumb rule how many levels you can apply in multilevel analyses? For RECAP more specific: do we need to have a country-level as well as a cohort-level?
  Multilevel analysis becomes more powerful when there are many small groups, in particular if the scientific interest focusses on relations at the second (or higher) level. For example, measurements nested within children (longitudinal data), pupils nested within classes, patients clustered within caregivers, and so on. In this case, the analysis borrows strength across clusters. Multilevel analysis has less to offer if we have a few large studies where we can easily

estimate the effect of scientific interest from the separate studies. A rule of thumb? Well, let's say that you would probably not do multilevel with fewer than 25 clusters.

- Should imputation be done in the cohorts separately in advance, or after combining data? What is the solution when thinking about the separate nodes and aggregated data in the RECAP platform?

  There are pros and cons. "Separate" is more useful if the missing data appear mainly in well-measured and harmonized (core) variable. It has the advantage that it preserves differences between cohorts in the relations among the variables, which may turn up in the later statistical analysis as interaction effects. However, it does not work very well of studies become small, or when harmonization is suspect. A second scenario is to do multiple imputation as part of (not after) data combining. The workshop concentrated on this scenario, which lead to some novel harmonization/data combination tools. "After combining" borrows strength across the different studies. Suppose we have blood pressure as a predictor, but some studies did not measure it. We can still take this study into the model under the assumption that the relation of blood pressure with other covariates and the outcome is similar to that in the other studies.
- How does the aggregated data from the nodes in the RECAP platform influence the bridge harmonization analysis?

  Aggregating data is a bad strategy. Everything becomes more complicated and less precise when working from aggregated data. A primary problem is ecological fallacy, where we see relations in the aggregated data that do not exist in the individual level data. A more promising way is to estimate parameters from the individual level data, and combine the parameter estimates over sources. However, this becomes increasingly hard if we are fitting multivariate models, where we want to "control for" other variables. In general, we need access to the individual level data to do good harmonization.

Remarks:

- Effects of culture as result of difference between counties should be taken into account. E.g. Afro-American children have a faster motor development or cultural difference in perceived Quality of Life.

  Yes, agreed. Two children of the SAME ability but from DIFFERENT cultures should have the same probability of passing the item/test. There are ways to test for this.
- The workshop showed that it is important to think about and test the underlying assumptions, especially in these difficult analyses.

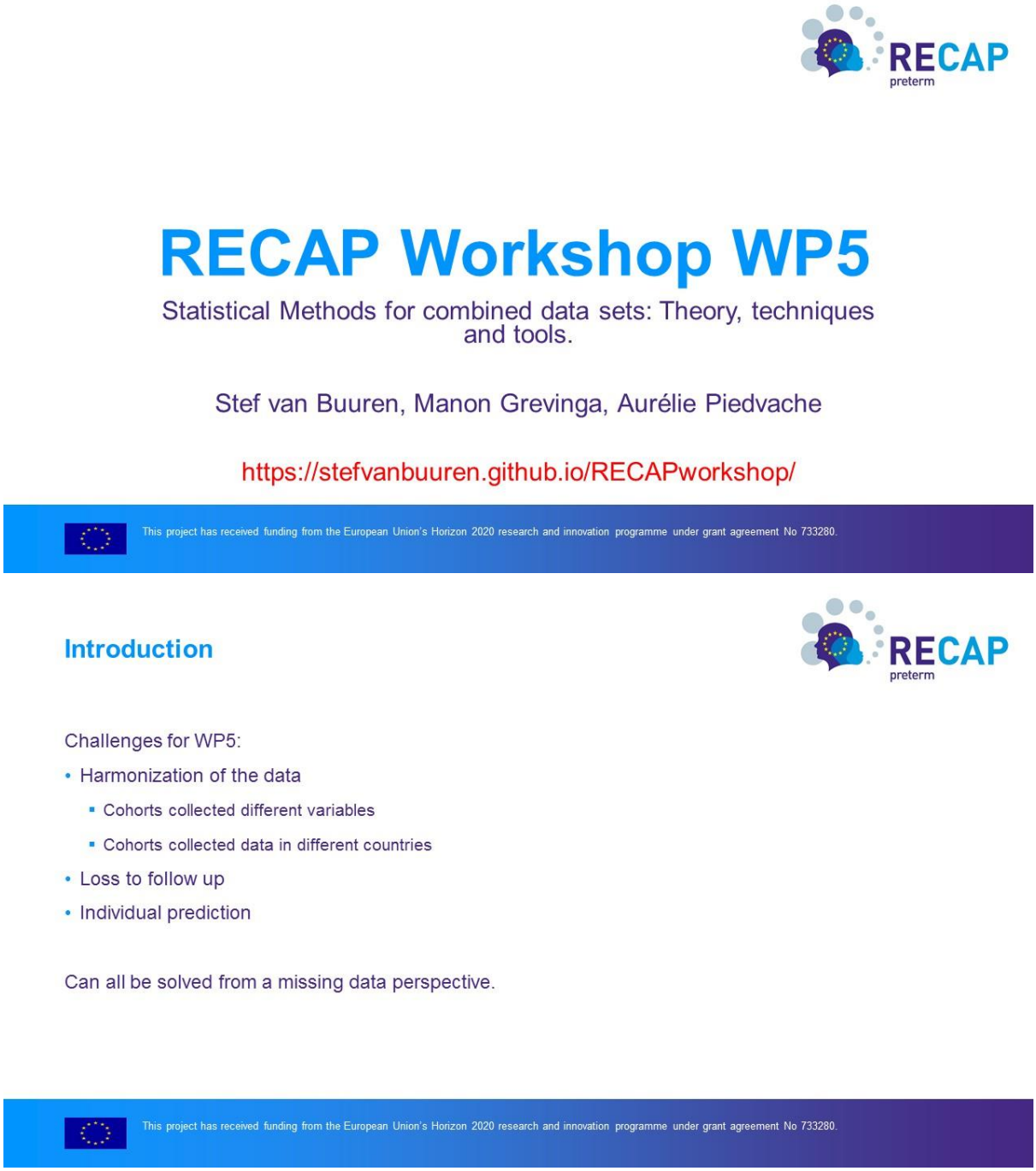
  Yes, everything we do rests on assumptions. Once we understand the assumptions, we may evaluate the relative merits of a particular approach.

5 LITERATURE

1. van Buuren, S., & Groothuis-Oudshoorn, K. (2011). MICE: Multivariate Imputation by Chained Equations in R. Journal of Statistical Software, 45(3), 1-67.

6 APPENDICES

6.1 Appendix A: Powerpoint slides of Workshop



Introduction

During this workshop we will address questions like:

- What are different ways of "combining data"?
  When are data linked?
- What can we do if response categories are different?
  How do our harmonization strategies affect our later analyses?
- What can we do if variables are entirely missing in some sources?
  What is the effect on later analyses?
- What statistical analyses can we do if our cohort shrinks due to loss-to-follow-up?
- Can we analyze combined data just as 'normal' data?
  When do we need a multilevel model, and how to do it?

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Table of Contents

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Table of Contents

Combining data sets and missing data

Datasets can be similar/be different on:

- The variables they collected
- The subjects included in the study

Based on these differences or similarities there are different ways to combine datasets.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Table of Contents

Multiple Imputation

Datasets can have missing values for multiple reasons.

Multiple imputation creates several complete datasets, where the missing values are replaced by plausible values.

In the analysis one takes the uncertainty about these plausible values into account.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Table of Contents

Creating comparable variables

Are item SIP01 and GARS9 measuring the same construct?

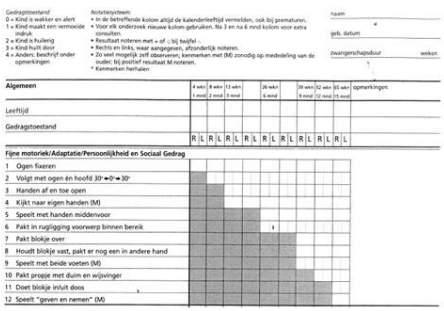
Can we recode SIP01 and/or GARS9 to make them comparable?

Item	Description	Response categories	Study	
			ERGOPLUS	EURIDISS
			n=306	*n=292*
SIP01	I walk shorter distances or often stop for a rest.	0 = No	276	
		1 = Yes	28	
GARS9	Can you, fully independently, walk outdoors (if necessary, with a cane)?	0 = Yes, no difficulty	145	
		1 = Yes, with some difficulty	110	
		2 = Yes, with much difficulty	29	
		3 = No, only with help from others	8	

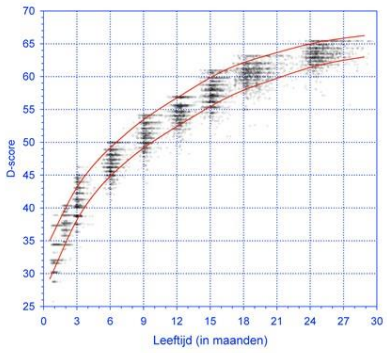
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Table of Contents

Developmental Milestones



D-score



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Table of Contents

Loss-to-follow-up

With longitudinal studies there is often loss-to-follow-up: no information is available for some of the subjects when they get older.

This results in missing values in the data.

Are the missing values/subjects independent of the outcome variable(s)?

Can we take this into account when analyzing the data?

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Table of Contents

Multilevel Analysis

Is it okay to assume that all observations are independent of each other, even though we combined multiple datasets from different cohort studies?

E.g., we expect that children from one mother are more similar than to a randomly chosen child from a different mother.

Does this also hold for cohort studies?

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Practicalities

SCHEDULE MONDAY

Time	Topic	Remarks
13:00 – 14:30	Combining datasets & missing data	*Theory*
14:30 – 15:00	*Break*	
15:00 – 16:00	Multiple imputation	*Theory & practical*
16:00 – 16:30	*Break*	
16:30 – 18:00	Creating comparable variables	*Theory & practical*
19:00 - …	Dinner	@ Scarlatti

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Practicalities

SCHEDULE TUESDAY

Time	Topic	Remarks
09:00 – 10:30	Developmental milestones	*Theory & practical*
10:30 – 11:00	*Break*	
11:00 – 12:30	Loss-to-follow-up	*Theory & practical*
12:30 – 14:00	*Lunch break*	
14:00 – 15:00	Multilevel analysis	*Theory & practical*
15:00 – 15:30	*Break*	
15:30 – 17:00	Discussion	

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Practicalities

Materials and setup

Software:

- R (http://cran.r-project.org/)
- R-studio might also be useful. (http://rstudio.com/)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Combining data sets & missing data

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Combining Data Sets: Why?

- Classic meta analysis
  - Increase the sample size → more certainty about the result(s)
- Individual patient data (IPD)
  - disentangle subject-level and study-level sources of heterogeneity in treatment effect;
  - study effect modification;
  - adjust for confounding variables;
  - improve data quality;
  - standardize definitions and analyses;
  - obtain complete follow-up data on all randomized participants;
  - combine studies with different follow-up times;
  - analyze multiple outcomes;
  - investigate long-term outcomes;
  - investigate rare exposures.

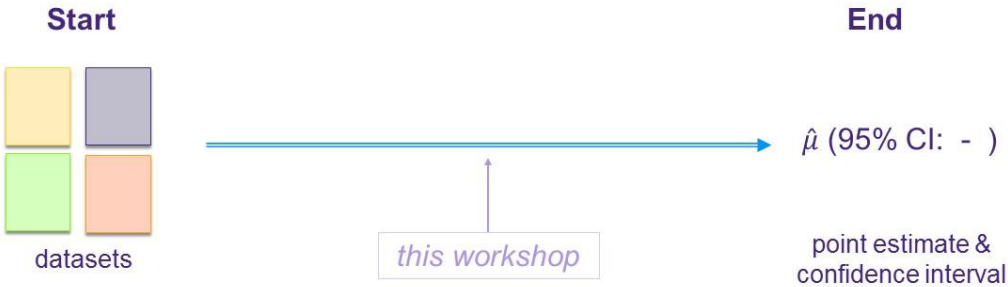
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Combining Data Sets: Why not?

- At least 10 reasons
  - Studies measure collect different sets of variables that measure the same construct;
  - Studies apply different measurement instruments;
  - The timing of the measurements varies widely between studies;
  - Study employ different designs to select units, or to allocate treatments;
  - Data are missing for different reasons, e.g. loss to follow up, not administered, skipped;
  - The key to link data from the same individual is imprecise, absent or contains duplicates;
  - The original data were collected for different analytic objectives;
  - Data may be sensitive, and at risk for de-identification after combining;
  - Definitions and classification may change over time;
  - Access to the original study sources is restricted.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Combining Data Sets



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Combining Data Sets

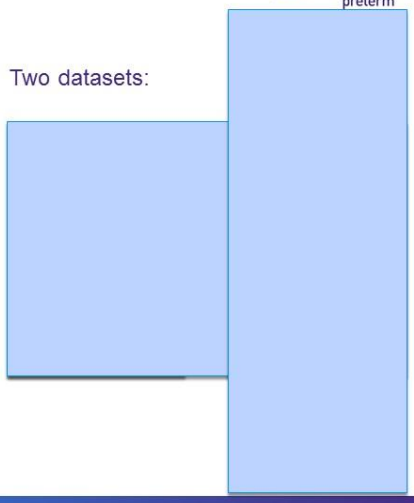
Questions:

- What are different ways of 'combining data'?
- When are data linked?

There are two ways to combine them:

- Join (next to each other)
- Add (among each other)

Two datasets:



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Combining Data Sets: Join

Requirements:

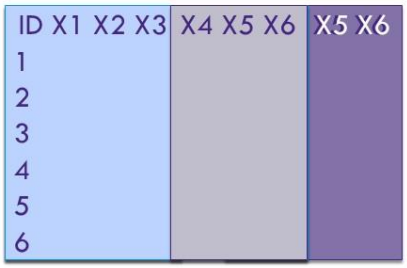
- Same subjects
- Different variables

Not every subject needs to be included in both datasets.

Four Join combining ways:

- Inner join
- Full outer join
- Master join
- Detail join

Two datasets:



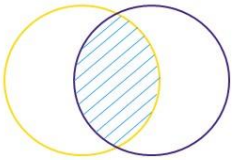
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Combining Data Sets: Inner Join

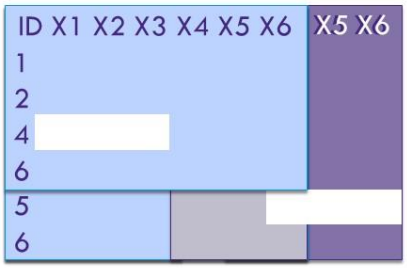
Requirements:

- Same subjects (not all have to be similar)
- Different variables

With Inner Join only keep the rows (subjects) that exist in both datasets.



Two datasets:



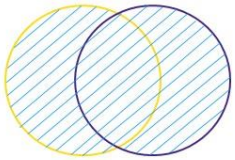
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Combining Data Sets: Full Outer Join

Requirements:

- Same subjects (not all have to be similar)
- Different variables

With Full Outer Join keep all the rows (subjects) with blanks.



Two datasets:

ID	X1	X2	X3	X4	X5	X6	X5	X6
1								
2								
3								
4								
5								
6								

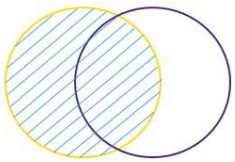
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Combining Data Sets: Master Join (Left Outer Join)

Requirements:

- Same subjects (not all have to be similar)
- Different variables

With Master Join keep all the rows (subjects) of one dataset and only the matching rows of the other.



Two datasets:

ID	X1	X2	X3	X4	X5	X6	X5	X6
1								
3								
4								
5								
6								
6								

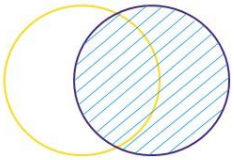
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Combining Data Sets: Detail Join (right outer join)

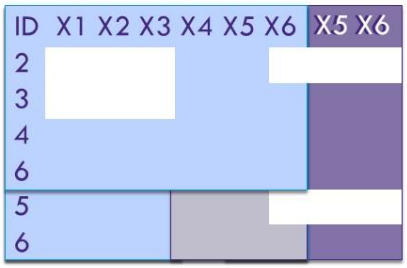
Requirements:

- Same subjects (not all have to be similar)
- Different variables

With Detail Join keep all the rows (subjects) of one dataset and only the matching rows of the other.



Two datasets:



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Combining Data Sets: Add

Requirements:

- Different subjects
- Same variables (not all have to be similar)

Two datasets:

ID	X1	X2	X3
1			
2			
3			
4			
5			
6			

ID	X1	X2	X3
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Combining Data Sets: Add

Requirements:

- Different subjects
- Same variables (not all have to be similar)

Not every variable needs to be included in both datasets.

Again two options:

- Keep X3
- Drop X3

Two datasets:

ID	X1	X2
1		
2		
3		
4		
5		
6		

ID	X1	X2	X3
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Combining Data Sets: Add

Requirements:

- Different subjects
- Some similar variables, some different variables

Two datasets:

ID	X1	X2	X3	X4
1				
2				
3				
4				
5				
6				

ID	X1	X2	X3	X4
1				
2				
3				4
4				
5				
6				
7				
8				
9				
10				
11				
12				

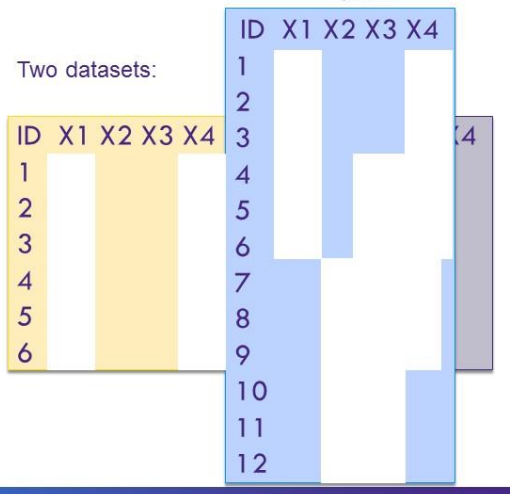
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Combining Data Sets: Add

Requirements:

- Different subjects
- No similar variables

Two datasets:



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Missing data perspective

- Ideal Data: envision what data we would like to have had to solve our problem given unlimited resources;
- Ideal Analysis: define what analysis we would perform if we had the ideal data;
- Available Data: evaluate which parts of the ideal data are available to us;
- Missing data: determine why some parts of the ideal data are missing;
- Replications: construct replications of the unseen ideal data;
- Calculate: our answer from each replication by the method of point 2;
- Summarize: the answer over the replications.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Multiple imputation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Missing data

Causes of missing data:

- Respondent skipped the item
- Data transmission/coding error
- Drop out in longitudinal research
- Refusal to cooperate
- Sample from population
- Question not asked, different forms
- Branching, routing
- Censoring
- Combining data

In general, missing data can severely complicate interpretation and analysis.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Missing data

When subjects are followed over time they can behave in three different ways:

- *Complete case*
  Subject has all measurements and hence has no missing response values.
- *Monotone missingness (drop-out)*
  Subject starts by having all measurements, but at a certain point in time the measurements are missing and after that no measurement is recorded.
- *Non-monotone missingness*
  The subject misses one (or more) measurements, however after a missed measurement there is at least one another measurement.

	T = 1	T = 2	T = 3	T = 4
Complete case	O	O	O	O
Monotone missingness (drop-out)	O	O	O	M
	O	O	M	M
	O	M	M	M
Non-monotone missingness	O	M	O	O
	O	O	M	O
	O	M	O	M
	M	O	O	O
	...	...	...	...

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Missing data

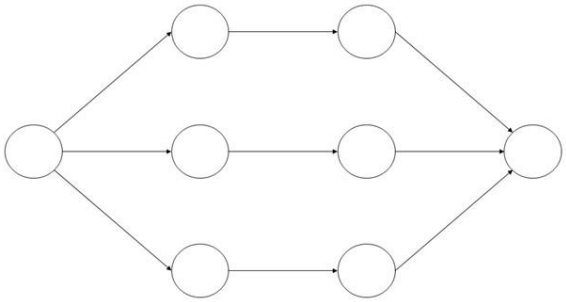
One can make an assumption about the missing mechanism, the missing is:

- *MCAR*: Missing Completely at Random
  Example: Mother filled in the questionnaire but it was not received because it was lost by the postal service
- *MAR*: Missing At Random
  Example: Low response from multiparous mothers
- *MNAR*: Missing Not At Random
  Example: If a child is unable to carry out a task, the mother is more likely to leave blank (missing)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Multiple Imputation (MI)

Main steps used in multiple imputation



Multiple imputation creates several $m > 1$ complete datasets, where the missing values are replaced by plausible values.

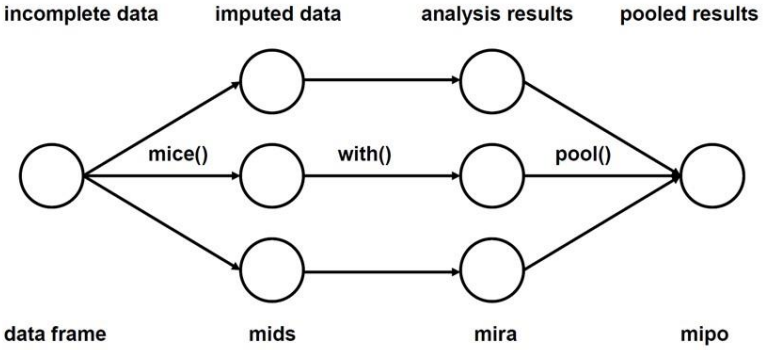
Each of these datasets is analyzed using standard software, and the m results are then pooled in to a final point estimate.

The magnitude of the difference between the imputed data points tells us something about the uncertainty about the imputed value: the bigger the difference the more uncertain we are.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Multiple Imputation (MI)

Steps in mice



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Multiple Imputation (MI)

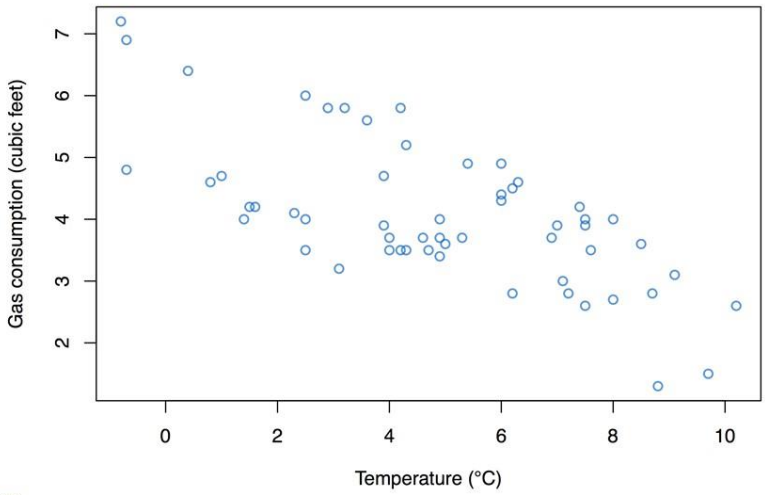
How large should *m* be?

- Use $m = 5$ or $m = 10$ if the fraction of missing information is low
- Develop your model with $m = 5$.
  Do final run with m equal to percentage of incomplete cases.
- Repeat the analysis with $m = 5$ with different seeds.
  If there are large difference for some parameters, this means that the data contain little information about them.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

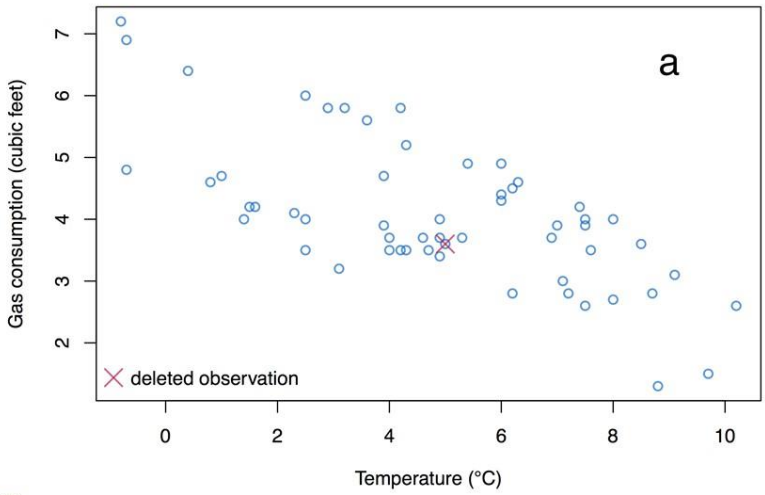
Multiple Imputation in Practice > C > Univariate imputation

Relation between temperature and gas consumption

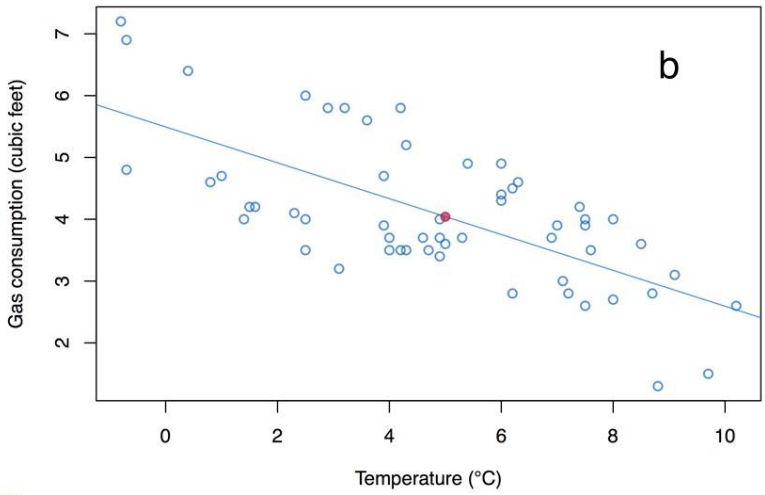


Universiteit Utrecht TNO SvB, GV

We delete gas consumption of observation 47

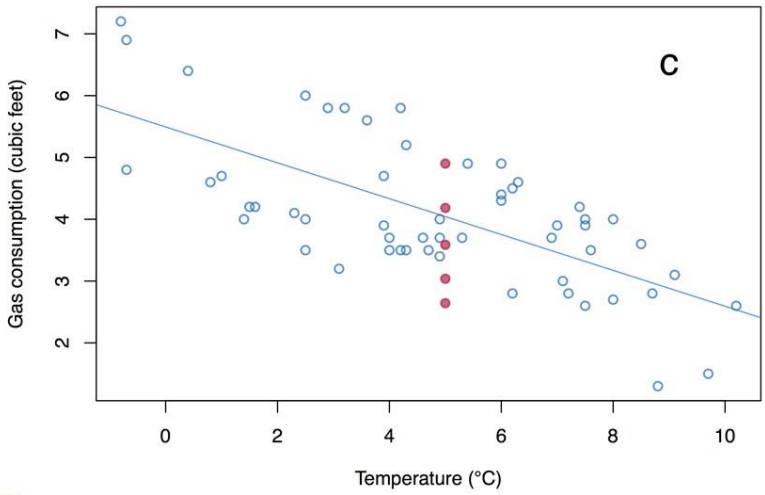


Predict imputed value from regression line



Multiple Imputation in Practice > C > Univariate imputation

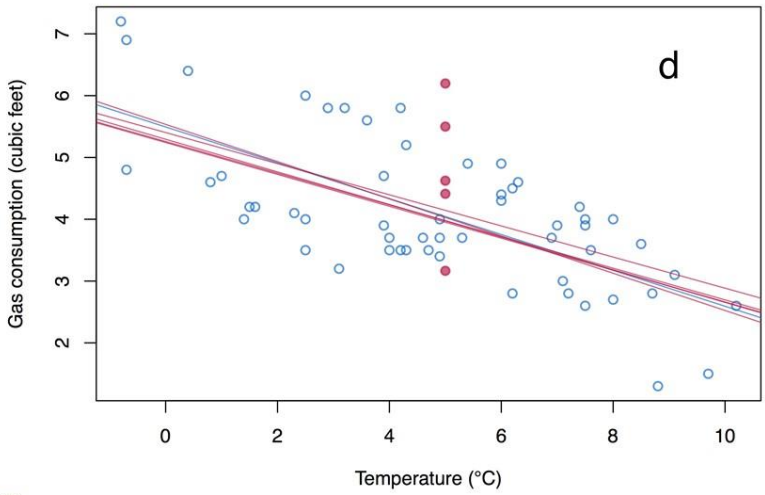
Predicted value + noise



Universiteit Utrecht TNO SvB, GV

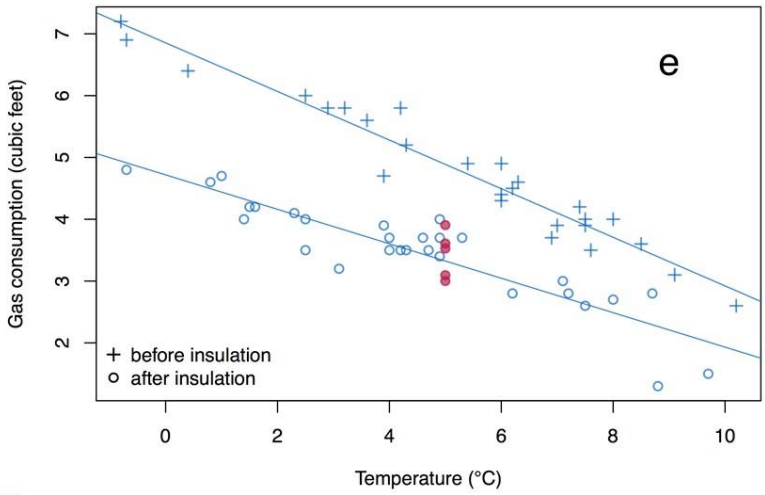
Multiple Imputation in Practice > C > Univariate imputation

Predicted value + noise + parameter uncertainty



Universiteit Utrecht TNO SvB, GV

Imputation based on two predictors



Multiple Imputation (MI)

Three sources of variation

- The variance cause by the fact that we are taking a sample rather than the entire population. (this is the conventional statistical measures of variability)
- The extra variance caused by the fact that there are missing values in the sample
- The extra simulation variance cause by the fact that the total variance itself is based on finite *m*.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Creating Comparable Variables

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Creating Comparable Variables

When combining data from cohort studies we want to *add* the datasets, since the cohorts study the same phenomena with different subjects.

However, the way these cohort studies investigated the phenomena might differ.

For example, all cohort studies might have the same survey question, however the categorical responses for this question differ.

To be able to *add* these datasets, we would like to harmonize the responses to these survey questions which can be done by e.g. **response conversion**.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Creating Comparable Variables

Levels of equivalence

5 **scalar equivalence**: same ratio scale across cohort

4 **unit equivalence**: same units but different anchors

3 **procedural equivalence**: common procedure to measure objects, but there is no underlying unit or ordering in the numbers

2 **construct equivalence**: same concept is measured, but scales differ

1 **construct inequivalence**: no equivalent concepts across cohorts

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Creating Comparable Variables // Recoding

Item	Description	Response categories	Study	
			ERGOPLUS	EURIDISS
			n=306	*n=292*
SIP01	I walk shorter distances or often stop for a rest.	0 = No	276	
		1 = Yes	28	
GARS9	Can you, fully independently, walk outdoors (if necessary, with a cane)?	0 = Yes, no difficulty		145
		1 = Yes, with some difficulty		110
		2 = Yes, with much difficulty		29
		3 = No, only with help from others		8

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Creating Comparable Variables // Recoding

Table 2.2 Example data with an additional bridge item.

Item	Description	Response categories	Study	
			ERGOPLUS	EURIDISS
			n=306	*n=292*
SIP01	I walk shorter distances or often stop for a rest.	0 = No	276	
		1 = Yes	28	
HAQ8	Able to walk outdoors on flat ground?	0 = Without any difficulty	242	178
		1 = With some difficulty	43	68
		2 = With much difficulty	15	42
		3 = Unable to do	0	2
GARS9	Can you, fully independently, walk outdoors (if necessary, with a cane)?	0 = Yes, no difficulty		145
		1 = Yes, with some difficulty		110
		2 = Yes, with much difficulty		29
		3 = No, only with help from others		8

The trait θ makes up a common scale for walking disability.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Creating Comparable Variables // Recoding

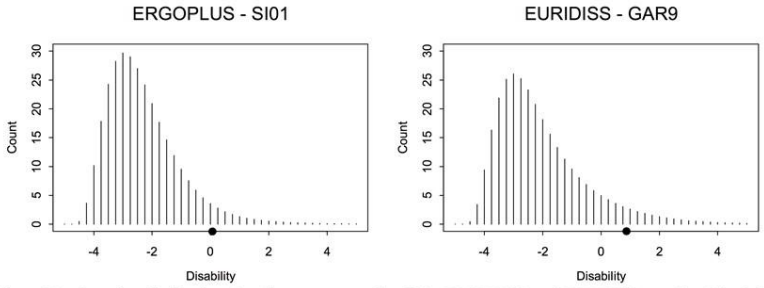


Figure 2.3 Posterior distributions (on the common scale) of the ERGOPLUS and EURIDISS samples. The left distribution is estimated from the SIP01, while the right panel is estimated from the GARS9 item. The dots on the horizontal axes indicate the position of the 95th percentiles.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

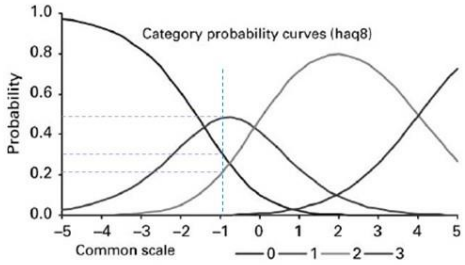
Creating Comparable Variables // Recoding

Horizontal axis: orders walking disability from no disability (left) to high disability(right)

Vertical axis: response probability

For someone with θ_i = -1 has a probability of

- 0.27 of responding in Category 0 of HAQ8
- 0.50 of responding in Category 1 of HAQ8
- 0.23 of responding in Category 2 of HAQ8
- 0.00 of responding in Category 3 of HAQ8



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

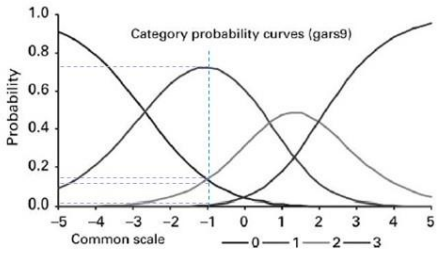
Creating Comparable Variables // Recoding

Horizontal axis: orders walking disability from no disability (left) to high disability(right)

Vertical axis: response probability

For someone with θ_i = -1 has a probability of

- 0.11 of responding in Category 0 of GAR9
- 0.72 of responding in Category 1 of GAR9
- 0.16 of responding in Category 2 of GAR9
- 0.01 of responding in Category 3 of GAR9



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Creating Comparable Variables // Recoding

The procedure of response conversion consists of two steps:

1. Construction of the conversion key
2. Using the conversion key

The **conversion key** models the relation between the common scale and the observed data.

It is important that the studies in need for harmonization need at least one identical item/question (**bridge item**) which can be used to develop the common scale.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Creating Comparable Variables // Recoding values

Table 2.4 Mean disability per category on the common scale for response patterns consisting of one item.

Item	Response category			
	0	1	2	3
SIP01	-2.44	-0.49		
HAQ8	-2.72	-1.71	0.06	2.68
GARS9	-2.89	-1.94	-0.22	2.00

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Creating Comparable Variables

Improving comparability can also be done by **Multiple Imputation**.

The multiple imputation approach is more flexible and more general than the recoding approach.

1. The MI approach does not require a common unidimensional latent scale, thereby increasing the range of applications.
2. MI approach takes uncertainty of recode into account

The MI approach will be explained by an example from the book 'Flexible Imputation of Missing Data' of Stef van Buuren (2012).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Creating Comparable Variables // Multiple Imputation

Take for example two bureaus, bureau *A* and *B*, that collect data about health in their own population.
The survey items that they used are similar, but not the same.
The survey used by bureau *A* contains the following question for measuring walking disability (item **A**):

Are you able to walk outdoors on flat ground?	Obs. frequency
0: Without any difficulty	242
1: With some difficulty	43
2: With much difficulty	15
3: Unable to do	0

And there are six missing values.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Creating Comparable Variables // Multiple Imputation

Bureau *A* produces a yearly report containing an estimate of the mean of the distribution of population A on item **A**.

When MCAR is assumed, we find

$$\hat{\theta}_{AA} = (242 * 0 + 43 * 1 + 15 * 2)/\,300 \;= 0.243$$

the disability estimate for population A using the method of bureau *A*.

Are you able to walk outdoors on flat ground?

0: Without any difficulty	242
1: With some difficulty	43
2: With much difficulty	15
3: Unable to do	0

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Creating Comparable Variables // Multiple Imputation

The survey of bureau *B* contains item **B**:

Can you, fully independent, walk outdoors (if necessary with cane)?	Obs. frequency
0: Yes, no difficulty	145
1: Yes, with some difficulty	110
2: Yes, with much difficulty	29
3: No, only with help from others	8

And there are no missing values.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Creating Comparable Variables // Multiple Imputation

Bureau *B* publishes the proportion of cases in category 0 as a yearly health measure.

Assuming a simple random sample, $P(Y_B = 0)$ is estimated by

$$\hat{\theta}_{BB} = 145 / 292 = 0.497$$

the health estimate for population B using the method of bureau *B*.

Can you, fully independent, walk outdoors (if necessary with cane)?

0: Yes, no difficulty	145
1: Yes, with some difficulty	110
2: Yes, with much difficulty	29
3: No, only with help from others	8

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Creating Comparable Variables // Multiple Imputation

Full dependence: simple equating

Note that $\hat{\theta}_{AA}$ and $\hat{\theta}_{BB}$ are different statistics calculated on different samples, and hence cannot be compared.

One solution, which is widely practiced, is just equating the four categories and apply the methods of bureau *A* and *B* and compare results.

Are you able to walk outdoors on flat ground?

0: Without any difficulty	242
1: With some difficulty	43
2: With much difficulty	15
3: Unable to do	0

$\hat{\theta}_{AA} = 0.243$

Can you, fully independent, walk outdoors (if necessary with cane)?

0: Yes, no difficulty	145
1: Yes, with some difficulty	110
2: Yes, with much difficulty	29
3: No, only with help from others	8

$\hat{\theta}_{BB} = 0.497$

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Creating Comparable Variables // Multiple Imputation

Full dependence: simple equating

$\hat{\theta}_{AA} = 0.243$

$\hat{\theta}_{BB} = 0.497$

To estimate walking disability in population B using the method of bureau *A* we obtain

$$\hat{\theta}_{BA} = (145 * 0 + 110 * 1 + 29 * 2 + 8 * 3)/\ 292\ = 0.658$$

The difference equals

$$\hat{\theta}_{BA} - \hat{\theta}_{AA} = 0.658 - 0.243 = 0.414$$

on a scale from 0 to 3.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Creating Comparable Variables // Multiple Imputation

Full dependence: simple equating

$\hat{\theta}_{AA} = 0.243$

$\hat{\theta}_{BB} = 0.497$

Likewise, we may estimate bureau's B health measure $\hat{\theta}_{AB}$ in population *A* as

$$\hat{\theta}_{AB} = 242/300\ = 0.807$$

Thus, over 80% of population A scores in category 0.

So by equating categories both bureaus conclude that population A is healthier, and by a fairly large margin.

As we will see, this result is however highly dependent on assumptions that may not be realistic for these data.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Creating Comparable Variables // Multiple Imputation

Full dependence: simple equating

$\hat{\theta}_{AA} = 0.243$

$\hat{\theta}_{BB} = 0.497$

Likewise, we may estimate bureau's B health measure $\hat{\theta}_{AB}$ in population *A* as

$$\hat{\theta}_{AB} = 242/300 = 0.807$$

Thus, over 80% of population A scores in category 0.

So by equating categories both bureaus conclude that population A is healthier, and by a fairly large margin.

This result is however highly dependent on assumptions that may not be realistic for these data → practical

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Creating Comparable Variables // Multiple Imputation

Table 7.8: Contingency table of responses on Y_A and Y_B in an external sample E ($n = 292$).

			Y_B		
Y_A	0	1	2	3	Total
0	128	45	3	2	178
1	13	45	10	0	68
2	3	20	14	5	42
3	0	0	1	1	2
NA	1	0	1	0	2
Total	145	110	29	8	292

Multiple imputation will fill-in the missing parts, using the relation observed study E

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Developmental Milestones

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Developmental Milestones

Gedragstoestand
0 = Kind is wakker en alert
1 = Kind maakt een vermoeide indruk
2 = Kind is huilerig
3 = Kind huilt door
4 = Anders: beschrijf onder opmerkingen

Notatiesysteem:
- In de betreffende kolom altijd de kalenderleeftijd vermelden, ook bij prematuren.
- Voor elk onderzoek nieuwe kolom gebruiken. Na 3 en na 6 mnd kolom voor extra consulten.
- Resultaat noteren met + of -; bij twijfel -.
- Rechts en links, waar aangegeven, afzonderlijk noteren.
- Zo veel mogelijk zelf observeren; kenmerken met (M) zonodig op mededeling van de ouder; bij positief resultaat M noteren.

* Kenmerken herhalen

naam

geb. datum

zwangerschapsduur weken

Algemeen	4 wkn / 1 mnd	8 wkn / 2 mnd	13 wkn / 3 mnd	26 wkn / 6 mnd	39 wkn / 9 mnd	52 wkn / 12 mnd	65 wkn / 15 mnd	opmerkingen
Leeftijd								
Gedragstoestand	R L	R L	R L	R L	R L	R L	R L	

Fijne motoriek/Adaptatie/Persoonlijkheid en Sociaal Gedrag												
1 Ogen fixeren												
2 Volgt met ogen én hoofd 30°←0°→30°												
3 Handen af en toe open												
4 Kijkt naar eigen handen (M)												
5 Speelt met handen middenvoor												
6 Pakt in rugligging voorwerp binnen bereik												
7 Pakt blokje over												
8 Houdt blokje vast, pakt er nog een in andere hand												
9 Speelt met beide voeten (M)												
10 Pakt propje met duim en wijsvinger												
11 Doet blokje in/uit doos												
12 Speelt "geven en nemen" (M)												

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Example items

Visit	Item
4 weeks	Fixates eyes
	Reacts to speech
	Moves both arms as much
	Moves both legs as much
	Lifts chin
8 weeks	Smiles in response
	Follows with eyes and head

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

The ideal situation

One unit for development: D-score

Three desirable properties:

1. Difference between two D-scores measured on the **same child** at **dif**ring the int
2. Difference between the D-score measured on two **different children** of the **same age** quantifies the difference in development between them
3. Co tifies ab

We need an interval scale

We need growth references

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

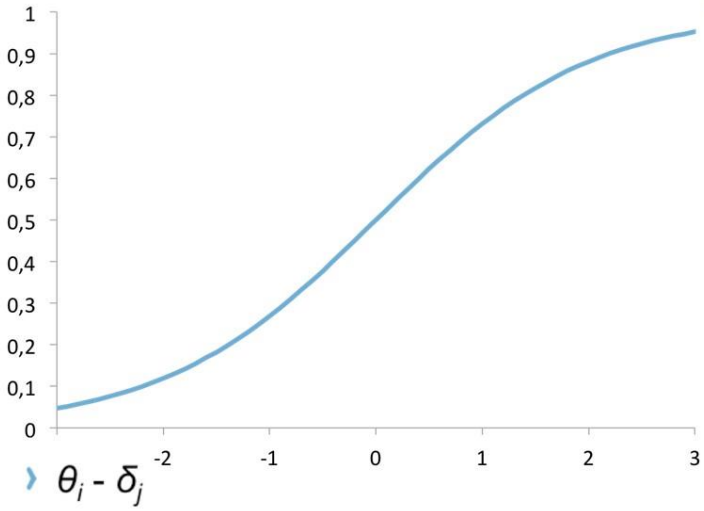
Rasch model


Georg Rasch (1901-1980)

- The probability of a "+" depends on
  - The ability θ_i of person *i*
  - The difficulty τ_j of item *j*
  - The difference θ_i - τ_j drives the probability of passing

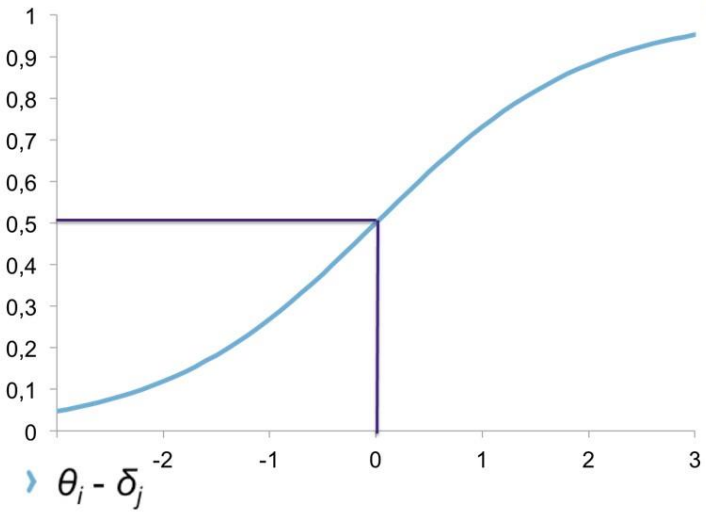
$$P(X_{ij} = + \mid \theta_i, \tau_j) = \frac{\exp(\theta_i - \tau_j)}{1 + \exp(\theta_i - \tau_j)}$$

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

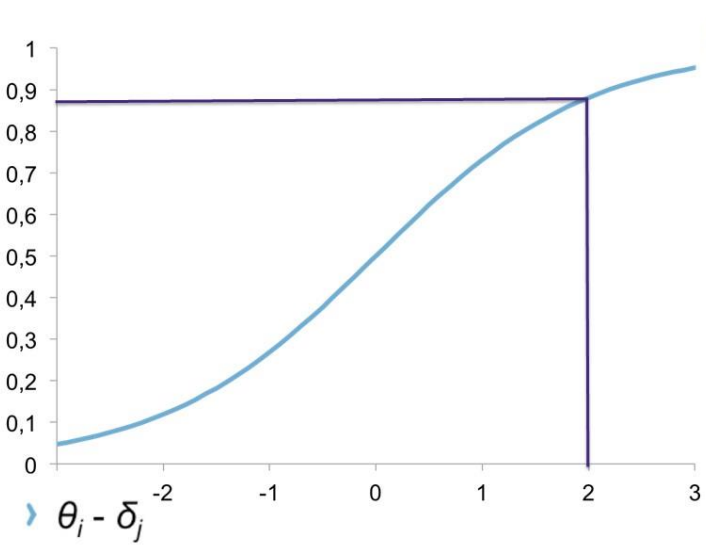


- θ_i - δ_j

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Key assumptions of the Rasch model

- **Unidimensionality**
  - One-dimensional latent scale θ that expresses differences in maturation
- **Parallel curves**
  - Probability of passing follows parallel logistic curves. Tests vary only in location (difficulty) on the θ axis
- **Local independence**
  - At a given θ the probabilities of passing two tests are independent

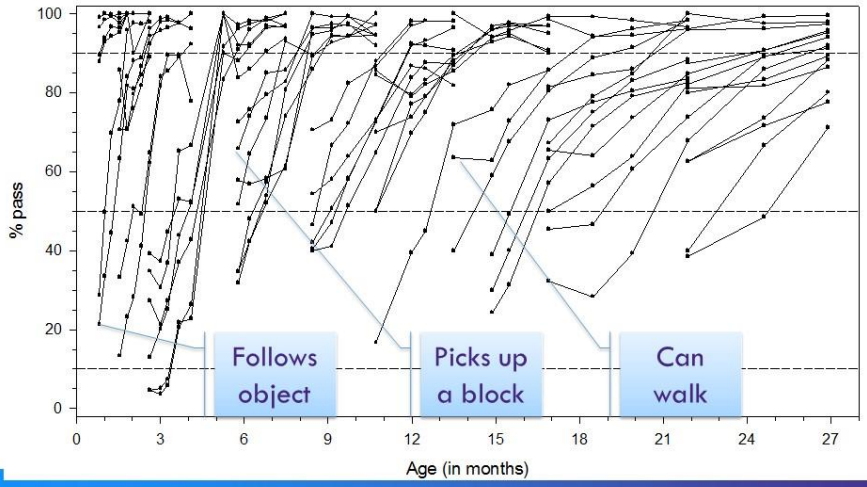
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

When do we have an interval scale?

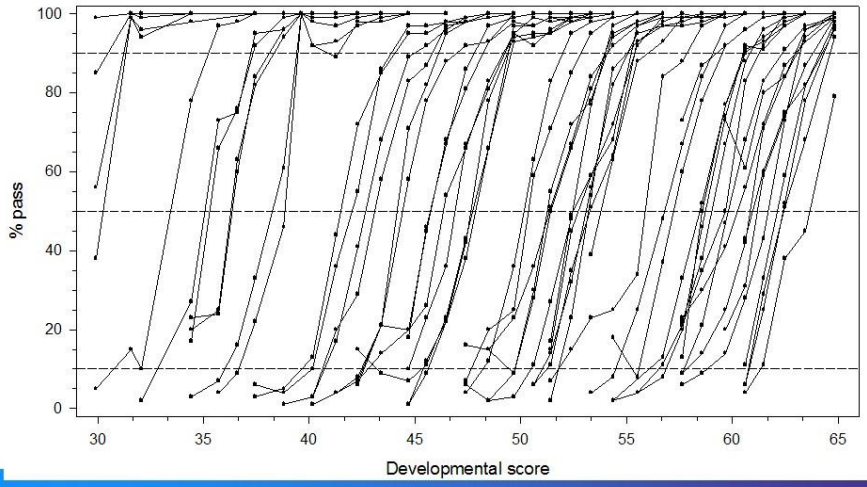
"The Rasch model, when it holds, yields measures of person abilities and item facilities on an interval scale with a common unit"

Perline R, Wright BD and Wainer H. The Rasch model as Additive Conjoint Measurement. www.rasch.org/memo24.htm

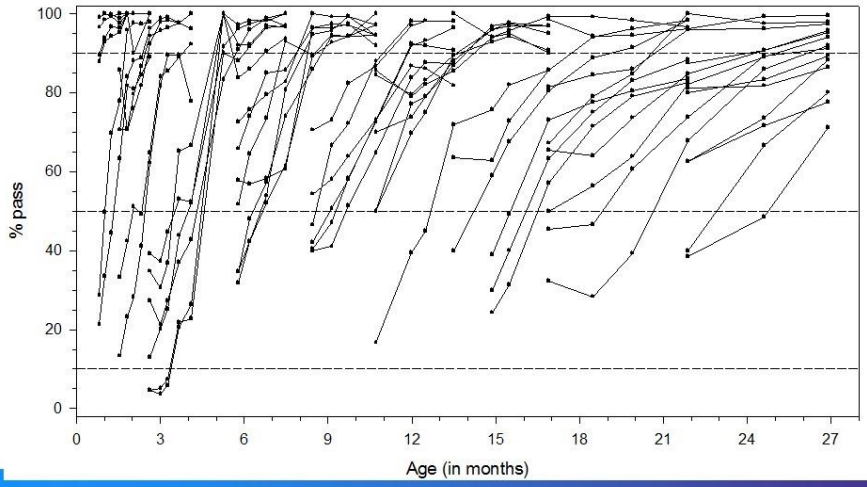
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.



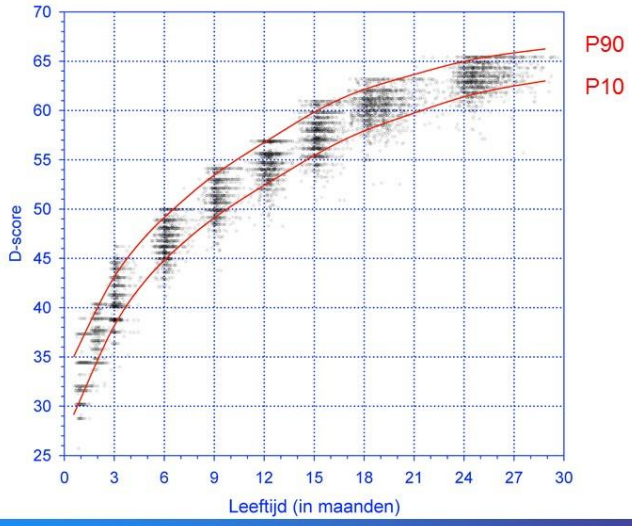
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.



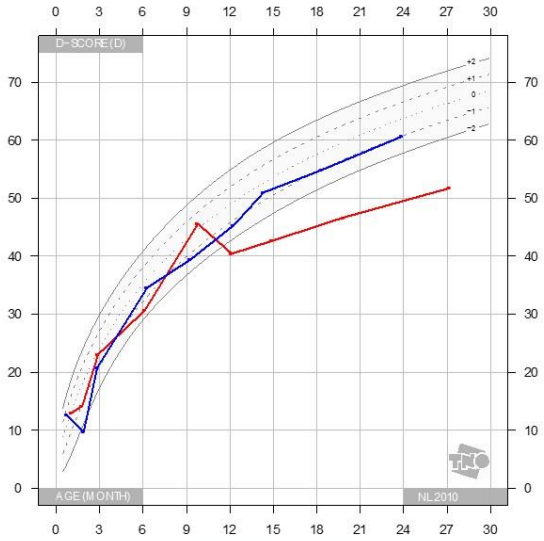
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Special properties of the D-score

- Measurement scale is independent of population
- Common metric, difference scores are meaningful
- Common scale, the same concept across age

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

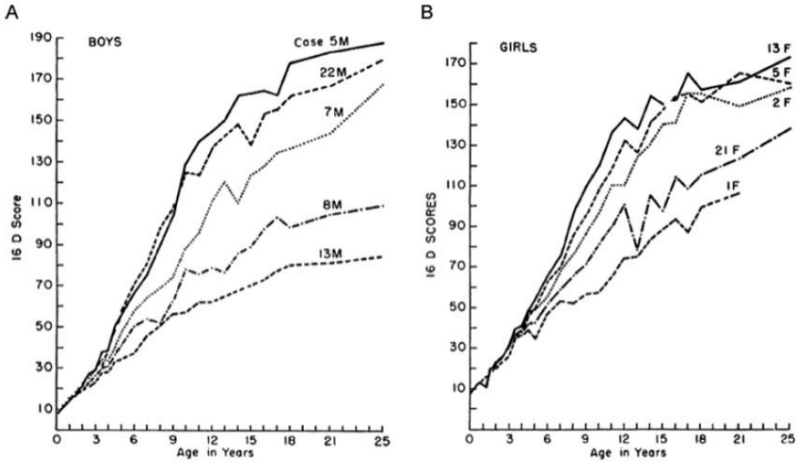


Figure 1. Growth curves of intellectual abilities from the Berkeley Growth Study of Bayley (1956; age 16 D scores). From "Individual Patterns of Development," by N. Bayley, 1956, *Child Development, 27,* p. 67. Copyright 1956 by Wiley-Blackwell.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Relation with later IQ

- Design: A selection of 619 children from SMOCC were measured again at 5 years
- Measures
  - Background characteristics at birth (gender, SES, age mother)
  - 0-2 years: Dutch developmental data
  - 5-6 years: UKKI, Dutch intelligence test

Hafkamp-de Groen E, Dusseldorp E, Boere-Boonekamp MM, Jacobusse GW, Oudesluijs-Murphy AM, Verkerk PH (2009). Relatie tussen het Van Wiechenonderzoek (D-score) op 2 jaar en het intelligentieniveau op 5 jaar. Tijdschr Jeugdgezondheidsz, 1, 10-4.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Relation with later IQ

- Pearson correlation D-score (2 yrs) with IQ (5 yrs): **0.34**
- Pearson correlation with background variables (0 yrs) with IQ (5 yrs)
  - R(SES) = **0.17**; R(age mother) = **0.12**
- Prediction of IQ:
  - D-score alone predicts 11% of variance
  - D-score + background 16% of variance

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

D-score screening?

- Can we use the D-score as a screening instrument to identify children with delayed cognitive development ($50 < IQ \leq 85$)?
- And if so, from what age?

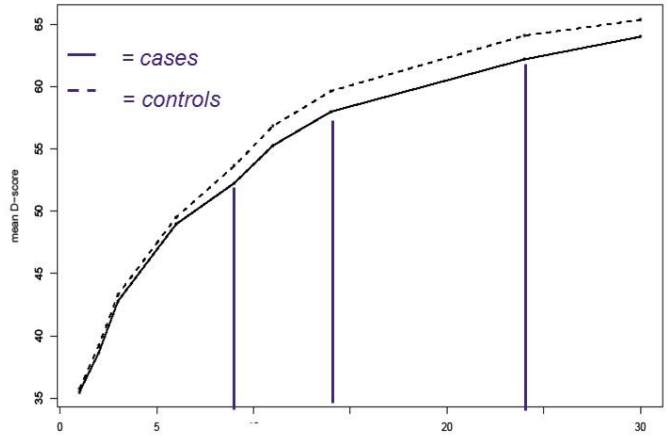
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

D-score screening?

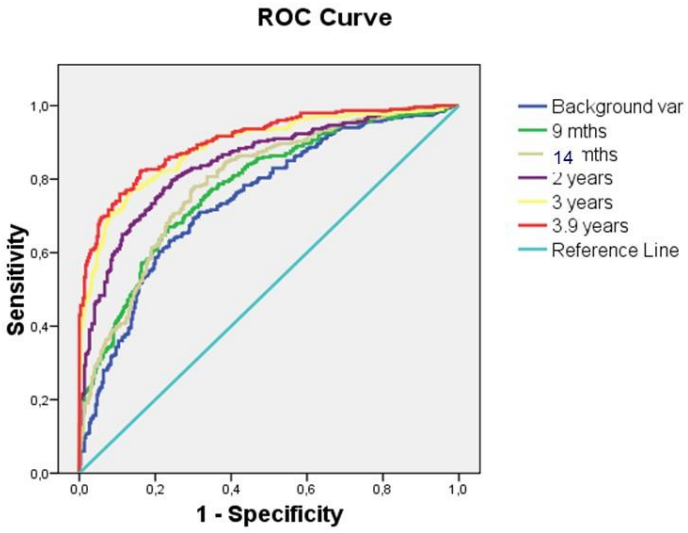
Case-control study

- 300 cases attending special education, 50 < IQ ≤ 85
- 300 controls: regular education; no developmental delay, did not repeat a class

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.



	9 mth	14 mth	24 mth
***P*-value:**	0.001	0.001	0.001
Cohen's *d*:	0.61	0.65	0.96
AUC:	0.59	0.64	0.70

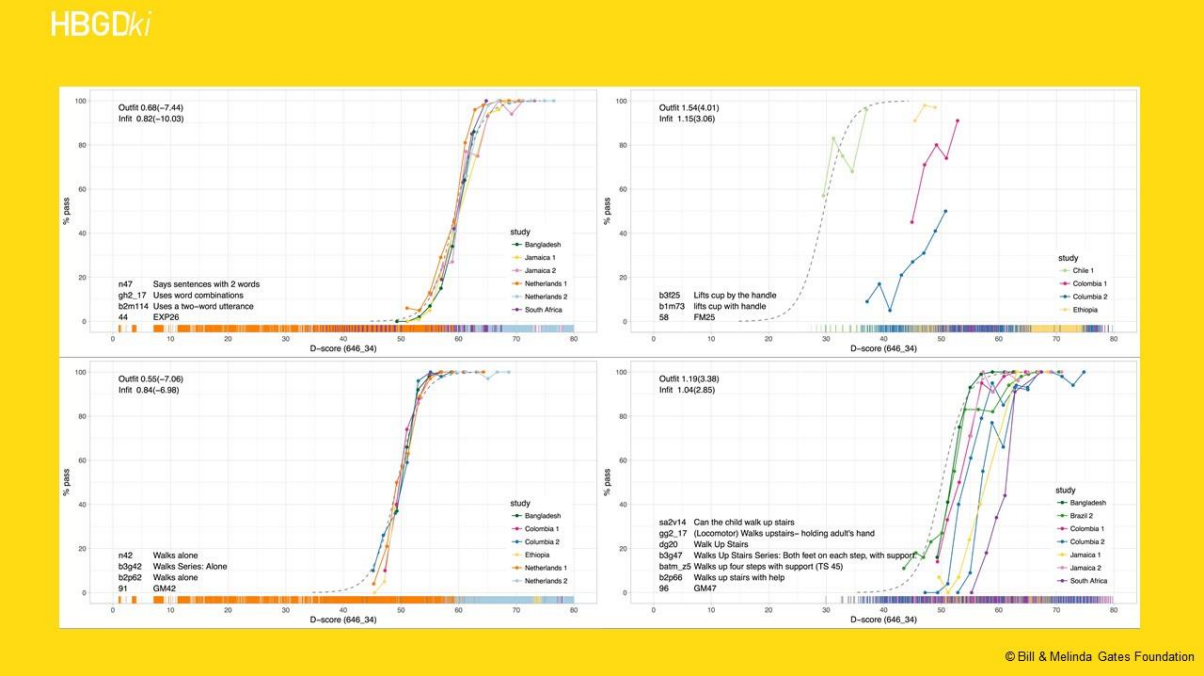


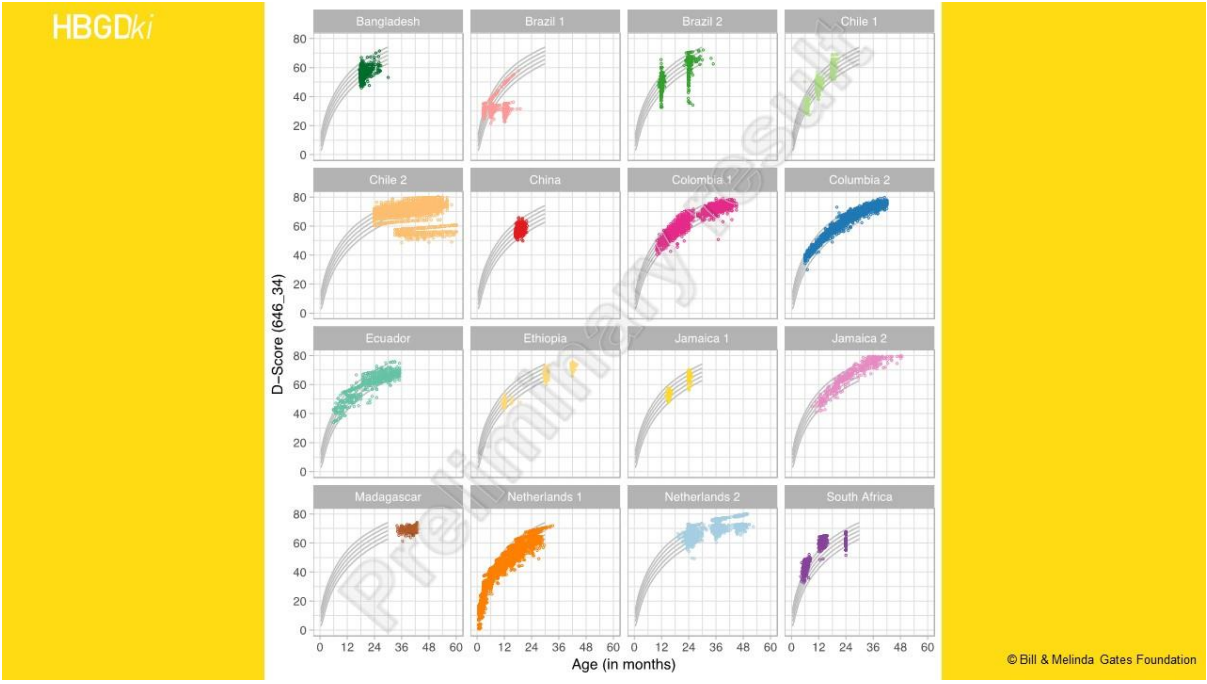
Does it work for other countries?

- Database
  - 16 cohorts
  - > 75000 records
  - > 1300 items
- Expert equality mapping at item level
- Which items form a scale?

Country	Investigators	Bayley	Denver	Griffiths	Battelle	Other
Bangladesh	Hamadani*, Tofail	X				X
Brazil (1993)	Menezes, Victora, Karam*		X			
Brazil (2004)	Barros, Victora, Karam*				X	
Chile	Lozoff*	X				X
Chile	Behrman, Bravo, Fernald*, Reynolds					X
China	Lozoff*	X				
Colombia (Bogota)	Attanasio*, McGregor*, Rubio-Codina*	X	X		X	X
Colombia	Attanasio*, McGregor*, Rubio-Codina*	X				X
Ecuador	Araujo*, Schady					X
Ethiopia	Hanlon*, Medhin	X				
Jamaica	Walker*, Chang*			X		
Jamaica	McGregor*, Powell			X		
Madagascar	Galasso, Fernald*, Ratsifandrihamanana*, Weber*					X
Netherlands	Verkerk, Schönbeck, Van Buuren*					X
South Africa	Richter*, Cameron	X		X		

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.


HBGDki
Outfit 0.68(−7.44)
Infit 0.82(−10.03)
n47 Says sentences with 2 words
gh2_17 Uses word combinations
b2m114 Uses a two−word utterance
44 EXP26
Outfit 1.54(4.01)
Infit 1.15(3.06)
b3f25 Lifts cup by the handle
b1m73 lifts cup with handle
58 FM25
Outfit 0.55(−7.06)
Infit 0.84(−6.98)
n42 Walks alone
b3g42 Walks Series: Alone
b2p62 Walks alone
91 GM42
Outfit 1.19(3.38)
Infit 1.04(2.85)
sa2v14 Can the child walk up stairs
gq2_17 (Locomotor) Walks upstairs− holding adult's hand
dg20 Walk Up Stairs
b3g47 Walks Up Stairs Series: Both feet on each step, with support
batm_z5 Walks up four steps with support (TS 45)
b2p66 Walks up stairs with help
96 GM47
% pass
D−score (646_34)
study
Bangladesh
Jamaica 1
Jamaica 2
Netherlands 1
Netherlands 2
South Africa
Chile 1
Colombia 1
Colombia 2
Ethiopia
Brazil 2
© Bill & Melinda Gates Foundation


HBGDki
Bangladesh
Brazil 1
Brazil 2
Chile 1
Chile 2
China
Colombia 1
Columbia 2
Ecuador
Ethiopia
Jamaica 1
Jamaica 2
Madagascar
Netherlands 1
Netherlands 2
South Africa
D−Score (646_34)
Age (in months)
Preliminary results
© Bill & Melinda Gates Foundation

Relevance D-score for RECAP

- Way to harmonize different scales of child development in **existing** data
- Way to compare child development to a **norm**

- Some scientific questions of interest:
  - Is gestational age inversely related to the risk of developmental delay?
  - Should we correct the D-score and DAZ for differences in gestational age, and if so, how?
  - Does the D-score predict later health, school succes, quality of life, and so on?
  - Can the D-score be used to identify (and treat) children before delay sets in?

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Loss to Follow Up

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Loss to Follow Up

Problem: Loss to follow-up can undermine the representativeness of estimates and introduce non-differential biases.

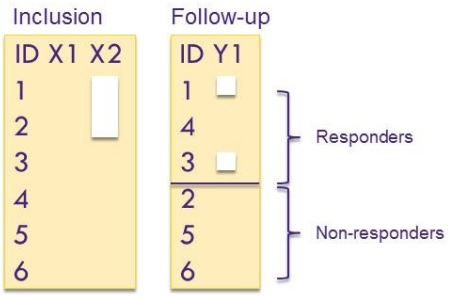
Usual approach: to ignore them

But, other approachs are possibles:

- Multiple imputation
- Inverse probability weighting (IPW)

Assumption with these approaches:

- For imputation: loss to follow up have to be MAR or MCAR



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Loss to Follow Up // Imputation

Recall: missing values are replaced by plausible values using chained equations method.

- Step 1: To impute data
  - What variables should be in the imputation model?
    - Factors associated with the probability of follow-up and outcome
    - Outcome
    - Follow-up
- Step 2: Estimate outcome from imputed datasets

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Loss to Follow Up // Inverse Probability Weighting

We give weights to responders to offset non-response and these weights are equivalent to the inverse probability of follow-up

- Step 0: If you have missing values on factors or the outcome(s), you can impute them beforehand
- Step 1: Estimate the probability of follow-up
  - **Dependent variable**: variable responder yes/no (Y)
  - **Independent variables**: factors associated with the probability of being followed-up and with the outcome ($X=(X_1, \ldots, X_k)$)

$$\text{logit}\,[P(Y = 1 \mid X)] = \alpha + \beta X + \gamma$$

$$\text{weight} = [\![\frac{1}{\text{logit}\,[P(Y = 1 \mid X)]}]\!]$$

- Step 2: Estimate outcome(s) with the weights

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Loss to Follow Up // Exercise

- **Objective**: Assess the potential impact of perinatal factors on the estimated prevalen neurodevelopmental delay with respect to non response.
- **Data**: Perinatal data were collected from medical records during the neonatal hospitalization and a parental questionnaire at 2 years of corrected age was sent to assess neurodevelopmental delay using a standardized parental report instrument.
- **Variables**:
  - follow: the child is followed up at 2 years – Yes/No
- Outcome
  - gmi_vi_hi_parca_asq_ten2: the child has a neurodevelopmental delay – Yes/No/Missing values
- Perinatal factors :
  - a4_weeks: gestational age of the child – (range: 23-31)
  - motherage: mother's age (range:14-53) / Missing values
  - native2: the mother is inborn in country – Yes/No/Missing values
  - f10: Was infant receiving human milk at discharge? - Yes/No/Missing values

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Multilevel Analysis

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Multilevel Analysis

Three types of analysis when combining data from different cohorts:

- Separate model for each cohort
- Dummy variable for each cohort
- One general model by means of a multilevel model

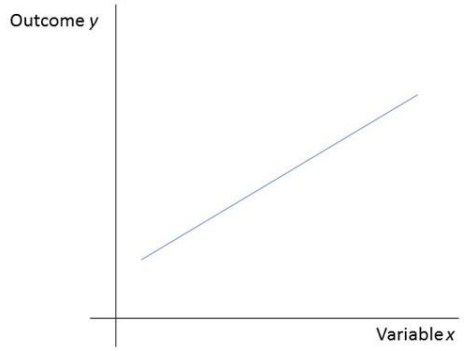
Multilevel model (mixed effects model) consists of two parts:

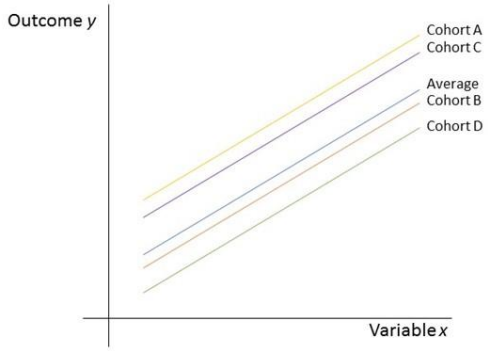
- Fixed effects
- Random effects (allows for different effects per country/cohort)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Multilevel Analysis // Mixed effects model

Random intercept





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Multilevel Analysis // Mixed effects model

Intraclass correlation (ICC)

$$ICC = \frac{\tau^2}{\sigma^2_{error} + \tau^2}$$

τ^2: the variance of the random intercept (*between variance*)
σ^2_{error}: the variance of patient-level error terms.

ICC can be interpreted as the proportion of total variance due to variation between clusters.

Rule of thumb:

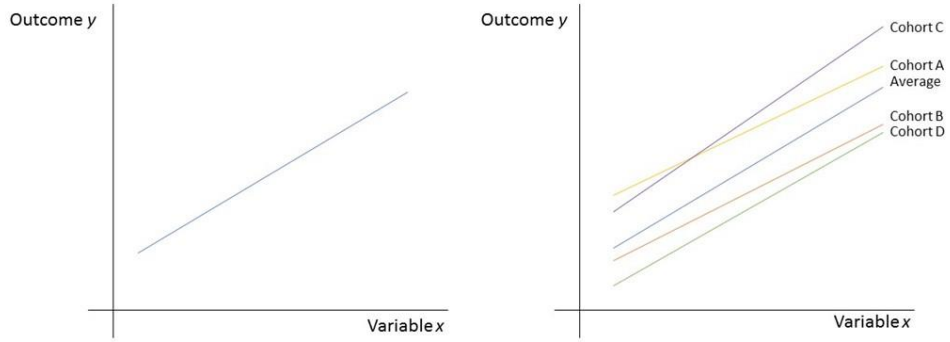
ICC > 5%
→ Model the correlation structure, e.g. by using a mixed effects model.

ICC < 5%
→ Add country as a categorical variable

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Multilevel Analysis // Mixed effects model

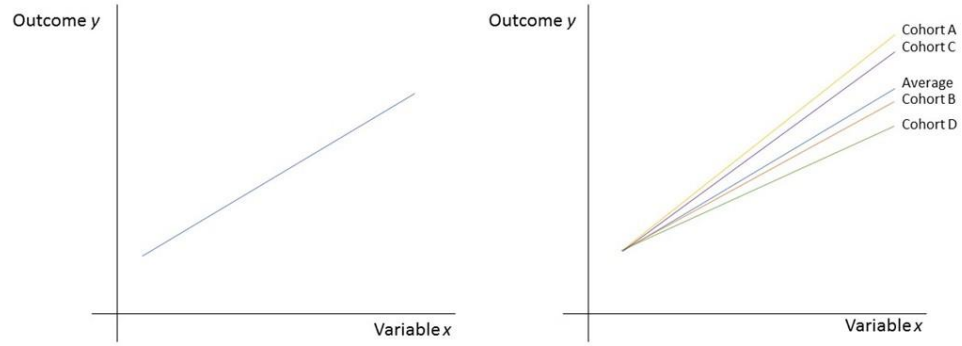
Random intercept and random slope



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Multilevel Analysis // Mixed effects model

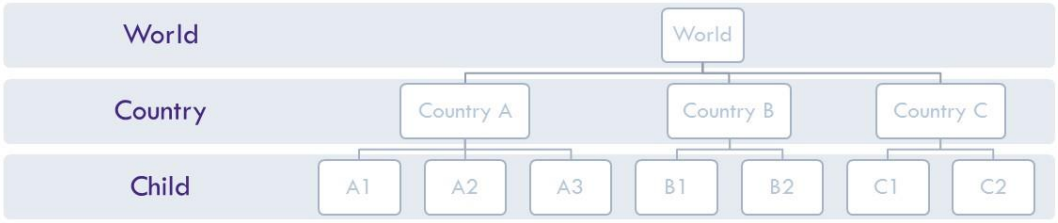
Random slope (no random intercept)



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Multilevel Analysis

Variables can be added to a model on multiple levels:



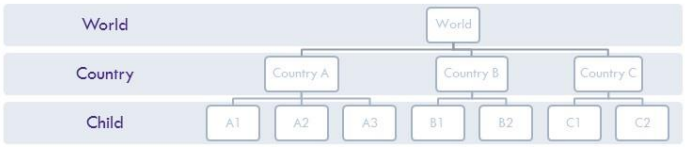
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Multilevel Analysis

An higher level variable is always also allocated to the lower levels.

A variable explaining variation within the country (a country-level variable) will also be allocated to the child

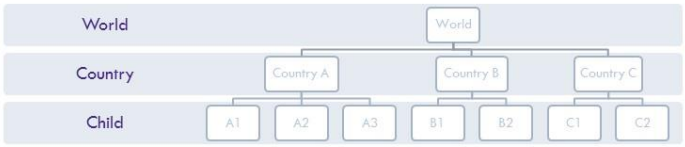
Example:
Relation between gestational age and birthweight.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Multilevel Analysis

- Each country might have different relation(s) between the predictor(s) and the outcome
- Hence, have its own model to explain the outcome.
- Using mixed effects model we can combine the models of the countries to one general model.

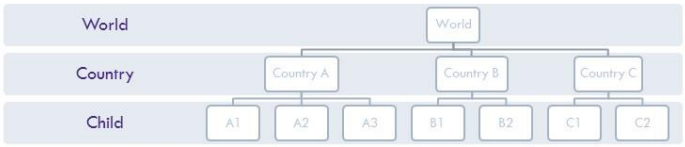


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Multilevel Analysis

For each level the total variability can be explained by two sources:

- Within cluster variability
- Between cluster variability



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

Multilevel Analysis // Conclusion

Problem: combining data of cohort studies

Usual approach: dummy variable for each cohort study

Assumption with this approach: it is assumed that the cohorts are independent of each other.

Solutions available: mixed effects model

How does it work?: allows for fixed and random effects. The random effects can capture the dependence of observations.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 733280.

6.2 Appendix B: Practical exercises workshop WP5

PRACTICAL I: Combining Datasets & Missing Data

Manon Grevinga, Stef van Buuren

Practical 1 of 6

This is an R Markdown document. When you click the Knit button an HTML document will be generated that includes both content as well as the output of any embedded R code chunks within the document. Moreover, clicking on the green triangles in the right upper corner of code chunks will run small parts of the code. This will be most convenient when we go through all the practicals step by step. Moreover, it is possible to following everything we do by means of the HTML document.

First, we need to install packages that we need during the workshop.

```
install.packages(c("mice", "lme4", "dplyr", "plyr", "mlmRev"))
```

In practical I, we are using only the plyr package.

```
library(plyr)
```

Combining Datasets & Missing Data

Lets assume we have two datsets, which we want to combine. This can be done in two ways: join and add. When we want to join two datasets they need to have some similar subjects (the variables may differ). When we want to add two datasets they need to contain similar variables, but may contain different subjects.

Join two datasets

First, we will generate two datasets A and B, which have some similar subjects and different variables:

```
#the randomly generated numbers will be same each time we use this set.seed

set.seed(40917)

df <- data.frame(subject = seq(1, 15, 1),

                 mean = seq(10, 24, 1),

                 sd = seq(2, 2.14, 0.01))


datasetA <- cbind(seq(1, 15,1),
```

```
                  data.frame(matrix(rnorm(15*3), 15, 3) * df$sd + df$mean))
datasetA[, 2:4] <- round(datasetA[, 2:4], 2)
names(datasetA) <- c("subjectID", "X1", "X2", "X3")


df <- data.frame(subject = seq(1, 15, 1),
                 mean = seq(110, 124, 1),
                 sd = seq(2, 2.14, 0.01))
datasetB <- cbind(seq(8, 22,1),
                  data.frame(matrix(rnorm(15*3), 15, 3) * df$sd + df$mean))
datasetB[, 2:4] <- round(datasetB[, 2:4], 2)
names(datasetB) <- c("subjectID", "X4", "X5", "X6")
```

This leads to the following summary statistics, where datasetA contains subjects 1 to 15, and datasetB contains subjects 8 to 22:

```
datasetA #subjectIDs from 1 to 15
##    subjectID    X1    X2    X3
## 1          1  7.93  8.53  8.76
## 2          2  6.92 11.91 11.28
## 3          3  8.80 13.02 13.50
## 4          4 12.60 11.15 12.29
## 5          5 13.23 13.08  9.33
## 6          6 14.39 16.65 14.55
## 7          7 12.24 14.87 13.65
## 8          8 10.79 18.58 15.09
## 9          9 18.26 15.21 16.57
## 10        10 16.66 17.40 19.75
## 11        11 20.67 19.79 20.33
## 12        12 20.05 21.01 21.83
## 13        13 22.31 19.55 20.96
```

```
## 14         14 22.94 24.52 28.03

## 15         15 26.90 24.45 22.41
```

```
datasetB #subjectIDs from 8 tot 22.
```

```
##    subjectID     X4     X5     X6

## 1          8 111.15 108.22 112.64

## 2          9 111.85 106.89 110.61

## 3         10 112.05 112.41 109.47

## 4         11 110.50 116.91 107.40

## 5         12 116.88 115.25 113.11

## 6         13 115.87 113.88 114.52

## 7         14 117.34 116.56 116.28

## 8         15 121.13 114.77 118.78

## 9         16 117.66 118.53 119.89

## 10        17 119.91 119.56 120.85

## 11        18 119.10 121.02 120.11

## 12        19 123.36 121.97 120.31

## 13        20 121.59 121.11 118.18

## 14        21 123.76 122.39 126.91

## 15        22 125.10 125.48 123.59
```

Inner Join

With Inner join only keep the ***subjects that exists in both datasets***:

```
AB.innerjoin <- join(datasetA, datasetB, by = "subjectID", type = "inner")

AB.innerjoin #keep subjects 8 to 15
```

```
##   subjectID    X1    X2    X3     X4     X5     X6

## 1         8 10.79 18.58 15.09 111.15 108.22 112.64

## 2         9 18.26 15.21 16.57 111.85 106.89 110.61

## 3        10 16.66 17.40 19.75 112.05 112.41 109.47

## 4        11 20.67 19.79 20.33 110.50 116.91 107.40
```

```
## 5         12 20.05 21.01 21.83 116.88 115.25 113.11
## 6         13 22.31 19.55 20.96 115.87 113.88 114.52
## 7         14 22.94 24.52 28.03 117.34 116.56 116.28
## 8         15 26.90 24.45 22.41 121.13 114.77 118.78
```

Note, we have 8 observations (for subjectID 8 untill 15) and that for each subject we have an observation for each variabele.

Full Outer Join

With Full outer join keep ***all subjects***:

```
AB.fullouterjoin <- join(datasetA, datasetB, by = "subjectID", type = "full
")

AB.fullouterjoin #keep all subjects
```

```
##    subjectID    X1    X2    X3     X4     X5     X6
## 1          1  7.93  8.53  8.76     NA     NA     NA
## 2          2  6.92 11.91 11.28     NA     NA     NA
## 3          3  8.80 13.02 13.50     NA     NA     NA
## 4          4 12.60 11.15 12.29     NA     NA     NA
## 5          5 13.23 13.08  9.33     NA     NA     NA
## 6          6 14.39 16.65 14.55     NA     NA     NA
## 7          7 12.24 14.87 13.65     NA     NA     NA
## 8          8 10.79 18.58 15.09 111.15 108.22 112.64
## 9          9 18.26 15.21 16.57 111.85 106.89 110.61
## 10        10 16.66 17.40 19.75 112.05 112.41 109.47
## 11        11 20.67 19.79 20.33 110.50 116.91 107.40
## 12        12 20.05 21.01 21.83 116.88 115.25 113.11
## 13        13 22.31 19.55 20.96 115.87 113.88 114.52
## 14        14 22.94 24.52 28.03 117.34 116.56 116.28
## 15        15 26.90 24.45 22.41 121.13 114.77 118.78
## 16        16    NA    NA    NA 117.66 118.53 119.89
```

```
## 17        17    NA    NA    NA 119.91 119.56 120.85
## 18        18    NA    NA    NA 119.10 121.02 120.11
## 19        19    NA    NA    NA 123.36 121.97 120.31
## 20        20    NA    NA    NA 121.59 121.11 118.18
## 21        21    NA    NA    NA 123.76 122.39 126.91
## 22        22    NA    NA    NA 125.10 125.48 123.59
```

Note, we have 22 observations and there are some non-availables (NA's) for each variabele. We have NA's for X1 till X3 for subjectID 16 till 22 and NA's for X4 till X6 for subjectID 1 till 7.

Master Join

With Master join (left outer join) keep ***all subjects of one dataset and only the matching rows of the other***:

```
AB.leftjoin <- join(datasetA, datasetB, by = "subjectID", type = "left")
AB.leftjoin #keep all subjects from datasets A and match rows from B
```

```
##    subjectID    X1    X2    X3     X4     X5     X6
## 1          1  7.93  8.53  8.76     NA     NA     NA
## 2          2  6.92 11.91 11.28     NA     NA     NA
## 3          3  8.80 13.02 13.50     NA     NA     NA
## 4          4 12.60 11.15 12.29     NA     NA     NA
## 5          5 13.23 13.08  9.33     NA     NA     NA
## 6          6 14.39 16.65 14.55     NA     NA     NA
## 7          7 12.24 14.87 13.65     NA     NA     NA
## 8          8 10.79 18.58 15.09 111.15 108.22 112.64
## 9          9 18.26 15.21 16.57 111.85 106.89 110.61
## 10        10 16.66 17.40 19.75 112.05 112.41 109.47
## 11        11 20.67 19.79 20.33 110.50 116.91 107.40
## 12        12 20.05 21.01 21.83 116.88 115.25 113.11
## 13        13 22.31 19.55 20.96 115.87 113.88 114.52
## 14        14 22.94 24.52 28.03 117.34 116.56 116.28
```

```
## 15        15 26.90 24.45 22.41 121.13 114.77 118.78
```

Note, that we have 15 observations and NA's for X4 till X6 for subjectIDs 1 till 7.

Detail Join

With Detail join (right outer join) keep ***all subjects of one dataset and only the matching rows of the other***:

```
AB.rightjoin <- join(datasetA, datasetB, by = "subjectID", type = "right")

AB.rightjoin #keep all subjects from datasets A and match rows from B
```

```
##    subjectID    X1    X2    X3     X4     X5     X6
## 1          8 10.79 18.58 15.09 111.15 108.22 112.64
## 2          9 18.26 15.21 16.57 111.85 106.89 110.61
## 3         10 16.66 17.40 19.75 112.05 112.41 109.47
## 4         11 20.67 19.79 20.33 110.50 116.91 107.40
## 5         12 20.05 21.01 21.83 116.88 115.25 113.11
## 6         13 22.31 19.55 20.96 115.87 113.88 114.52
## 7         14 22.94 24.52 28.03 117.34 116.56 116.28
## 8         15 26.90 24.45 22.41 121.13 114.77 118.78
## 9         16    NA    NA    NA 117.66 118.53 119.89
## 10        17    NA    NA    NA 119.91 119.56 120.85
## 11        18    NA    NA    NA 119.10 121.02 120.11
## 12        19    NA    NA    NA 123.36 121.97 120.31
## 13        20    NA    NA    NA 121.59 121.11 118.18
## 14        21    NA    NA    NA 123.76 122.39 126.91
## 15        22    NA    NA    NA 125.10 125.48 123.59
```

Note, that we have 15 observations and NA's for X1 till X3 for subjectIDs 16 till 22.

Add two datasets

Besides joining datasets, we can also add datasets. In this case we measured the same variables (not all have to be the same) on different subjects. First we will simulate two datasets C and D, with some variables similar and different subjects.

```
df <- data.frame(subject = seq(1, 15, 1),
                 mean = seq(10, 24, 1),
                 sd = seq(2, 2.14, 0.01))
datasetC <- cbind(seq(1, 15,1),
                  data.frame(matrix(rnorm(15*3), 15, 3) * df$sd + df$mean))
datasetC[, 2:4] <- round(datasetC[, 2:4], 2)
names(datasetC) <- c("subjectID", "X1", "X2", "X3")


df <- data.frame(subject = seq(1, 15, 1),
                 mean = seq(10, 24, 1),
                 sd = seq(2, 2.14, 0.01))
datasetD <- cbind(seq(16, 30,1),
                  data.frame(matrix(rnorm(15*3), 15, 3) * df$sd + df$mean))
datasetD[, 2:4] <- round(datasetD[, 2:4], 2)
names(datasetD) <- c("subjectID", "X4", "X2", "X3")
```

This leads to the following summary statistics, where datasetA contains subjectsIDs from 1 to 15 with variables X1, X2, and X3 and datasetB contains subjectIDs from 16 tot 30 with variables X4, X2, and X3:

```
datasetC
```

```
##    subjectID    X1    X2    X3
## 1          1 12.77 11.80 10.72
## 2          2 12.34  8.49 10.17
## 3          3 15.12 11.81 12.06
## 4          4 11.76 15.55 16.45
## 5          5 14.84  9.63 14.08
## 6          6 15.55 12.72 17.96
## 7          7 16.24 18.50 18.45
## 8          8 16.46 18.64 20.14
```

```
## 9          9 16.20 19.10 18.49
## 10        10 16.20 16.59 17.47
## 11        11 20.18 19.48 18.51
## 12        12 18.52 24.05 20.25
## 13        13 20.36 21.62 21.29
## 14        14 23.79 23.56 22.06
## 15        15 25.21 22.06 22.43
```

```
datasetD
```

```
##    subjectID    X4    X2    X3
## 1         16  9.76 10.29 10.18
## 2         17 10.17 11.91 10.36
## 3         18  9.08 13.56 14.37
## 4         19 13.75 11.21 11.29
## 5         20 13.90 11.99 14.75
## 6         21 16.34 18.64 14.43
## 7         22 18.44 14.91 16.33
## 8         23 15.44 18.09 14.31
## 9         24 14.35 18.63 19.76
## 10        25 22.58 22.20 19.36
## 11        26 17.62 17.02 18.80
## 12        27 22.59 19.09 26.22
## 13        28 23.22 23.63 17.00
## 14        29 26.57 24.15 19.24
## 15        30 23.21 20.21 24.71
```

When adding two dataframes that do not have all the same variables there are two options: 1. Drop the variables that are not similar 2. Keep the variables that are not similar and put them equal to NA for the other dataset.

Drop variables

This look as follows when we drop the variables that are not similar (in this case X1 in dataset C and X4 in dataset D):

```
datasetC.dropX1 <- subset(datasetC, select = c("subjectID", "X2", "X3"))

datasetD.dropX4 <- subset(datasetD, select = c("subjectID", "X2", "X3"))

datasetC.dropX1
```

```
##    subjectID    X2    X3
## 1          1 11.80 10.72
## 2          2  8.49 10.17
## 3          3 11.81 12.06
## 4          4 15.55 16.45
## 5          5  9.63 14.08
## 6          6 12.72 17.96
## 7          7 18.50 18.45
## 8          8 18.64 20.14
## 9          9 19.10 18.49
## 10        10 16.59 17.47
## 11        11 19.48 18.51
## 12        12 24.05 20.25
## 13        13 21.62 21.29
## 14        14 23.56 22.06
## 15        15 22.06 22.43
```

```
datasetD.dropX4
```

```
##    subjectID    X2    X3
## 1         16 10.29 10.18
## 2         17 11.91 10.36
## 3         18 13.56 14.37
## 4         19 11.21 11.29
```

```
## 5          20 11.99 14.75
## 6          21 18.64 14.43
## 7          22 14.91 16.33
## 8          23 18.09 14.31
## 9          24 18.63 19.76
## 10         25 22.20 19.36
## 11         26 17.02 18.80
## 12         27 19.09 26.22
## 13         28 23.63 17.00
## 14         29 24.15 19.24
## 15         30 20.21 24.71
```

Now that we dropped variables X1 and X4 we are left with two datasets that contain the same variables. Hence, we can add them.

```
add.CD.drop <- rbind(datasetC.dropX1, datasetD.dropX4)
add.CD.drop #subjectID are from 1 to 30.
```

```
##    subjectID    X2    X3
## 1          1 11.80 10.72
## 2          2  8.49 10.17
## 3          3 11.81 12.06
## 4          4 15.55 16.45
## 5          5  9.63 14.08
## 6          6 12.72 17.96
## 7          7 18.50 18.45
## 8          8 18.64 20.14
## 9          9 19.10 18.49
## 10        10 16.59 17.47
## 11        11 19.48 18.51
## 12        12 24.05 20.25
```

```
## 13         13 21.62 21.29
## 14         14 23.56 22.06
## 15         15 22.06 22.43
## 16         16 10.29 10.18
## 17         17 11.91 10.36
## 18         18 13.56 14.37
## 19         19 11.21 11.29
## 20         20 11.99 14.75
## 21         21 18.64 14.43
## 22         22 14.91 16.33
## 23         23 18.09 14.31
## 24         24 18.63 19.76
## 25         25 22.20 19.36
## 26         26 17.02 18.80
## 27         27 19.09 26.22
## 28         28 23.63 17.00
## 29         29 24.15 19.24
## 30         30 20.21 24.71
```

Keep Variables

However, normally we want to avoid dropping variables since they contain information. Hence, another way to add two datasets is to keep the variables that are not similar and make them NA for the other dataset:

```
datasetC.addX4 <- cbind(datasetC, rep("NA", 15))
names(datasetC.addX4) <- c("subjectID", "X1", "X2", "X3", "X4")


datasetD.addX1 <- as.data.frame(cbind(datasetD$subjectID, rep("NA", 15),
                                      datasetD$X2, datasetD$X3, datasetD$X4
))
```

```
names(datasetD.addX1) <- c("subjectID", "X1", "X2", "X3", "X4")
```

Now the datasets look as follows:

```
datasetC.addX4
```

```
##    subjectID    X1    X2    X3 X4
## 1          1 12.77 11.80 10.72 NA
## 2          2 12.34  8.49 10.17 NA
## 3          3 15.12 11.81 12.06 NA
## 4          4 11.76 15.55 16.45 NA
## 5          5 14.84  9.63 14.08 NA
## 6          6 15.55 12.72 17.96 NA
## 7          7 16.24 18.50 18.45 NA
## 8          8 16.46 18.64 20.14 NA
## 9          9 16.20 19.10 18.49 NA
## 10        10 16.20 16.59 17.47 NA
## 11        11 20.18 19.48 18.51 NA
## 12        12 18.52 24.05 20.25 NA
## 13        13 20.36 21.62 21.29 NA
## 14        14 23.79 23.56 22.06 NA
## 15        15 25.21 22.06 22.43 NA
```

```
datasetD.addX1
```

```
##    subjectID X1    X2    X3    X4
## 1         16 NA 10.29 10.18  9.76
## 2         17 NA 11.91 10.36 10.17
## 3         18 NA 13.56 14.37  9.08
## 4         19 NA 11.21 11.29 13.75
## 5         20 NA 11.99 14.75  13.9
## 6         21 NA 18.64 14.43 16.34
## 7         22 NA 14.91 16.33 18.44
```

```
## 8          23 NA 18.09 14.31 15.44
## 9          24 NA 18.63 19.76 14.35
## 10         25 NA  22.2 19.36 22.58
## 11         26 NA 17.02  18.8 17.62
## 12         27 NA 19.09 26.22 22.59
## 13         28 NA 23.63    17 23.22
## 14         29 NA 24.15 19.24 26.57
## 15         30 NA 20.21 24.71 23.21
```

Now, we can add the two datasets:

```
add.CD.keep <- rbind(datasetC.addX4, datasetD.addX1)
add.CD.keep
```

```
##    subjectID    X1    X2    X3    X4
## 1          1 12.77  11.8 10.72    NA
## 2          2 12.34  8.49 10.17    NA
## 3          3 15.12 11.81 12.06    NA
## 4          4 11.76 15.55 16.45    NA
## 5          5 14.84  9.63 14.08    NA
## 6          6 15.55 12.72 17.96    NA
## 7          7 16.24  18.5 18.45    NA
## 8          8 16.46 18.64 20.14    NA
## 9          9  16.2  19.1 18.49    NA
## 10        10  16.2 16.59 17.47    NA
## 11        11 20.18 19.48 18.51    NA
## 12        12 18.52 24.05 20.25    NA
## 13        13 20.36 21.62 21.29    NA
## 14        14 23.79 23.56 22.06    NA
## 15        15 25.21 22.06 22.43    NA
## 16        16    NA 10.29 10.18  9.76
```

```
## 17        17    NA 11.91 10.36 10.17
## 18        18    NA 13.56 14.37  9.08
## 19        19    NA 11.21 11.29 13.75
## 20        20    NA 11.99 14.75  13.9
## 21        21    NA 18.64 14.43 16.34
## 22        22    NA 14.91 16.33 18.44
## 23        23    NA 18.09 14.31 15.44
## 24        24    NA 18.63 19.76 14.35
## 25        25    NA  22.2 19.36 22.58
## 26        26    NA 17.02  18.8 17.62
## 27        27    NA 19.09 26.22 22.59
## 28        28    NA 23.63    17 23.22
## 29        29    NA 24.15 19.24 26.57
## 30        30    NA 20.21 24.71 23.21
```

So to conclude, we can join and add datasets. If we have observations from similar subjects on different variabiles we can join the datasets in four ways:

- Inner join
- Outer join
- Master join
- Detail join

When two datasets measures some similar variables on different subjects we can add theses datasets. To do this we have to decide on how to handle variables that were not included in both datasets:

- Drop these variables
- Keep these variables

PRACTICAL II: Multiple imputation using MICE

Manon Grevinga, Stef van Buuren

Practical 2 of 6

Multiple Imputation (using the package MICE)

For this practical we will use data from the package `mice`:

```
library(mice)
```

The dataset `nhanes` contains 25 observations on the following 4 variables:

- ***age***: Age group (1 = 20-39, 2 = 40-59, 3 = 60+)
- ***bmi***: Body mass index (kg/m^2)
- ***hyp***: Hypertensive (1 = no, 2 = yes)
- ***chl***: Total serum cholesterol (mg/dL)

In `R` the dataset looks as follows:

```
nhanes
```

```
##    age  bmi hyp chl
## 1    1   NA  NA  NA
## 2    2 22.7   1 187
## 3    1   NA   1 187
## 4    3   NA  NA  NA
## 5    1 20.4   1 113
## 6    3   NA  NA 184
## 7    1 22.5   1 118
## 8    1 30.1   1 187
## 9    2 22.0   1 238
## 10   2   NA  NA  NA
## 11   1   NA  NA  NA
## 12   2   NA  NA  NA
## 13   3 21.7   1 206
## 14   2 28.7   2 204
```

```
## 15   1 29.6   1  NA
## 16   1   NA  NA  NA
## 17   3 27.2   2 284
## 18   2 26.3   2 199
## 19   1 35.3   1 218
## 20   3 25.5   2  NA
## 21   1   NA  NA  NA
## 22   1 33.2   1 229
## 23   1 27.5   1 131
## 24   3 24.9   1  NA
## 25   2 27.4   1 186
```

Complete-case analysis

When we would model without taking the missing values into account, we will get the following model:

```
model <- lm(chl ~ bmi + age, data = nhanes)
summary(model)
```

```
##
## Call:
## lm(formula = chl ~ bmi + age, data = nhanes)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.187 -19.517  -0.310   6.915  60.606
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -80.194     58.772  -1.364 0.202327
## bmi            6.884      1.846   3.730 0.003913 **
```

```
## age           53.069     11.293   4.699 0.000842 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.67 on 10 degrees of freedom
##   (12 observations deleted due to missingness)
## Multiple R-squared:  0.7318, Adjusted R-squared:  0.6781
## F-statistic: 13.64 on 2 and 10 DF,  p-value: 0.001388
```

Note that almost half of the cases were not used in the analysis.

Missing data

With multiple imputation we want to provide plausible values for the missing values, while taking the uncertainty about these numbers into account. Hence, we will first inspect the missing data pattern:

```
md.pattern(nhanes)
```

```
##    age hyp bmi chl
## 13   1   1   1   1  0
##  1   1   1   0   1  1
##  3   1   1   1   0  1
##  1   1   0   0   1  2
##  7   1   0   0   0  3
##      0   8   9  10 27
```

Thus, for 13 subjects we have all variables. Moreover, for none of the subjects the variable age is missing. On the other hand, for 7 subjects we only have the age.

One useful feature of the `mice` package is the ability to specify which predictors can be used for each incomplete variable.

```
imp <- mice(nhanes, print = FALSE)
imp$predictorMatrix
```

```
##     age bmi hyp chl
```

```
## age   0   0   0   0
## bmi   1   0   1   1
## hyp   1   1   0   1
## chl   1   1   1   0
```

The rows identify which predictors can be used for the variable in the row name. Hence, to impute the variable `bmi` we can use the variables `age`, `hyp`, and `chl`. Note, that the diagonal is equal to zero, because a variable cannot predict itself. Moreover, there were no missing values for `age`, hence we do not need to predict its missing values and its row contains only zeroes.

Multiply impute the data

Now, we can multiply impute the missing values in our dataset. It is useful to plot the parameters against the number of iterations to check for convergence. On convergence, the different streams should be freely intermingled with one another, without showing any definite trends.

```
imp <- mice(nhanes, print = FALSE, maxit = 10, seed = 24415) #10 iterations

plot(imp) #inspect the trace lines for convergence
```

Analysis of imputed data

It is important to note that taking the average of the imputed datasets and analyze the averaged data is not the way to proceed. Doing this will yield incorrect standard errors, confidence intervals and p-values because it ignores the between-imputation variability. In other words, it does not take the uncertainty about the imputed variables into account.

The appropriate way to analyze multiply imputed data is to perform complete data analysis on each imputed dataset seperately. In the `mice`package we can use the `with()` command for this purpose. For example, we fit a regression model to each dataset and print out the estimate from the first and second completed datasets by:

```
fit <- with(imp, lm(chl ~ bmi + age))

coef(fit$analyses[[1]])

## (Intercept)         bmi         age
##  -49.037929    6.656636   36.061794

coef(fit$analyses[[2]])

## (Intercept)         bmi         age
##  -89.914211    7.318115   49.178204
```

Note, that the estimates for bmi and age are different from each other in the two completed datasets. This is due to the uncertainty created by the missing data. We can now apply the standard pooling rules by doing the following. In this way we get the final coefficient estimates for the model using imputed data:

```
est <- pool(fit)

summary(est)
```

```
##                   est        se         t       df   Pr(>|t|)        lo
95

## (Intercept) -29.54833 79.471793 -0.371809 6.421464 0.72199780 -220.95648
41

## bmi           5.83619  2.421364  2.410290 6.998748 0.04676021    0.11036
67

## age          37.34718 12.185849  3.064799 7.325582 0.01721003    8.78981
72

##                  hi 95 nmis       fmi    lambda

## (Intercept) 161.85983   NA 0.5961291 0.4872905

## bmi          11.56201    9 0.5649691 0.4561944

## age          65.90454    0 0.5482141 0.4396845
```

Comparison to complete-case analysis

The estimated model ignoring the missing values (complete-case analysis) was given by:

```
summary(model)
```

```
##

## Call:

## lm(formula = chl ~ bmi + age, data = nhanes)

##

## Residuals:

##     Min      1Q  Median      3Q     Max

## -31.187 -19.517  -0.310   6.915  60.606

##

## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -80.194     58.772  -1.364 0.202327
## bmi            6.884      1.846   3.730 0.003913 **
## age           53.069     11.293   4.699 0.000842 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.67 on 10 degrees of freedom
##   (12 observations deleted due to missingness)
## Multiple R-squared:  0.7318, Adjusted R-squared:  0.6781
## F-statistic: 13.64 on 2 and 10 DF,  p-value: 0.001388
```

When we compare this multiply imputed model model with complete-case analysis, we see that the coefficient estimates are quite different. The estimates for `bmi` and `age` are significant in both models. The standard errors of the coefficient estimates of complete-analysis are smaller here than the standard errors of the model were the missing values were imputed. This is not always the case. Because the multiply imputed model is based on 25 observations rather than 13, it could also have been the other way around.

In this case we assumed that the parameter estimates are normally distributed around the population value. Many types of estimates are approximately normally distributed: e.g., means, standard deviations, regression coefficients, proportions and linear predictors.

PRACTICAL III: Creating Comparable Variables

Manon Grevinga, Stef van Buuren

Practical 3 of 6

This document is based in section 7.4 of the book 'Flexible Imputation of Missing Data' by Stef van Buuren.

This practical needs the `mice` library:

```
library(mice)
```

Item YA

Are you able to walk outdoors on flat ground?

1. Without any difficulty
2. With some difficulty
3. With much difficulty
4. Unable to do

Item YB

Can you, fully independently, walk outdoors (if necessary with a cane)?

1. Yes, no difficulty
2. Yes, with some difficulty
3. Yes, with much difficulty
4. No, only with help from others

Equating categories

We have two studies, A and B. `YA` has been measured in Study A, and `YB` has been measured in Study B.

Would it be a good idea just to equate the four categories?

The equating assumption implicitly assumes that only combinations (0, 0), (1, 1), (2, 2) and (3, 3) can occur. Is that realistic?

Imputation under independence

Let `YA` be the item of Study `A`, and let `YB` be the item of Study `B`. The comparability problem is a missing data problem, where `YA` is missing for population `B` and `YB` is missing for population `A`. This formulation may help in using multiple imputation to solve the problem.

First, we create a small dataset with responses as follows:

```
fA <- c(242, 43, 15, 0, 6)          # frequencies of population A
```

```
fB <- c(145, 110, 29, 8)            # frequencies of population B

YA <- rep(ordered(c(0:3, NA)), fA) # outcome item A population A

YB <- rep(ordered(c(0:3)), fB)      # outcome item B population B
```

Combine both datasets with missing values for item `YB` for population `A`, and missing values for item `YA` for population `B`. The dataframe `Y`contains 604 rows and 2 columns: `YA` and `YB`.

```
Y <- rbind(data.frame(YA, YB = ordered(NA)),

           data.frame(YB, YA = ordered(NA)))

dim(Y)
```

```
## [1] 598   2
```

```
head(Y)
```

```
##   YA   YB

## 1  0 <NA>

## 2  0 <NA>

## 3  0 <NA>

## 4  0 <NA>

## 5  0 <NA>

## 6  0 <NA>
```

```
tail(Y)
```

```
##       YA YB

## 593 <NA>  3

## 594 <NA>  3

## 595 <NA>  3

## 596 <NA>  3

## 597 <NA>  3

## 598 <NA>  3
```

```
md.pattern(Y)
```

```
##      YA  YB

## 292   0   1   1
```

```
## 300   1   0   1
##   6   0   0   2
##     298 306 604
```

There no observations that link `YA` to `YB`, and so the missing data pattern is unconnected. Moreover, there are 6 records that contain no item data at all.

The following chunk is a bit of specialty code that defines two functions. The function `micemill()` calculates Kendall's τ (rank order correlation) between the imputed versions of `YA` and `YB` at each iteration. The function `ra` is a small helper function that puts the imputed data in proper shape.

```
micemill <- function(n){
  for (i in 1:n){
    imp <<- mice.mids(imp)
    cors <- with(imp, cor(as.numeric(YA),
                          as.numeric(YB), method = 'kendall'))
    tau <<- rbind(tau, ra(cors, s =T))
  }
}
ra <- function(x, simplify = FALSE) {
  if (!is.mira(x)) return(NULL)
  ra <- x$analyses
  if (simplify) ra <- unlist(ra)
  return(ra)
}
```

The following code imputes the missing data in `Y` under the (dubious) assumption that `YA` and `YB` are mutually independent.

```
tau <- NULL
imp <- mice(Y, max = 0, m = 10, print = FALSE,  seed = 32662)
micemill(25)
```

```
# define a function to plot tracelines of Kendall's tau
plotit <- function() matplot(x = 1:nrow(tau),
                             y = tau, ylab = expression(paste("Kendall's ",
tau)),
                             xlab = "Iteration", type = "l", lwd = 1,
                             lty = 1:10, col = "black")
plotit()
```

In the plot 25 iterations are plotted: the trace start near zero, but then freely wander off over a substantial range of the correlation. The MICE algorithm does not know where to go, and wander pointlessly through the parameter space. This occurs because the data contains no information that informs the relation between `YA` and `YB`, so $\tau\tau$ can be anything.

Why we cannot simply equate categories

Suppose that we have a third, external study `E` in which both `YA` and `YB` are measured.

```
##       0   1  2 3
## 0   128  45  3 2 178
## 1    13  45 10 0  68
## 2     3  20 14 5  42
## 3     0   0  1 1   2
## NA    1   0  1 0   2
##     145 110 29 8 292
```

The contingency table shows that there is a strong relation between `YA` and `YB`. However, it is far from perfect, so simply equating the four categories between `YA` and `YB` will distort their relationship. Note that the table is not symmetric, indicating that `YA` is more difficult than `YB`.

Simple equating assumes 100% concordance of the pairs. The contingency table clearly shows that this is not the case in study `E`. On surface, the four response categories of `YA` and `YB` may look similar, but the information from sample `E` suggests that the items work differently in a systematic way.

Imputation using a bridge study

Is there be a way to incorporate the relationship between `YA` and `YB` so that they will become comparable?

The answer is yes. We can redo the imputation, but now with sample `E` added to the data. In this way study `E` acts as a bridge study.

The relevant data are built-in in the `mice` under the name of `walking`.

```
head(walking)
```

```
##      sex age YA   YB src
## 1   Male  61  1 <NA>   A
## 2 Female  69  1 <NA>   A
## 3   Male  74  0 <NA>   A
## 4   Male  66  0 <NA>   A
## 5 Female  72  2 <NA>   A
## 6   Male  67  0 <NA>   A
```

```
table(walking$src)
```

```
##
##   A   B   E
## 306 292 292
```

```
with(walking, table(YA, YB, src, useNA = "always"))
```

```
## , , src = A
##
##       YB
## YA       0   1   2   3 <NA>
##   0      0   0   0   0  242
##   1      0   0   0   0   43
##   2      0   0   0   0   15
##   3      0   0   0   0    0
##   <NA>   0   0   0   0    6
##
```

```
## , , src = B
##
##       YB
## YA       0   1   2   3 <NA>
##   0      0   0   0   0    0
##   1      0   0   0   0    0
##   2      0   0   0   0    0
##   3      0   0   0   0    0
##   <NA> 145 110  29   8    0
##
## , , src = E
##
##       YB
## YA       0   1   2   3 <NA>
##   0    128  45   3   2    0
##   1     13  45  10   0    0
##   2      3  20  14   5    0
##   3      0   0   1   1    0
##   <NA>   1   0   1   0    0
##
## , , src = NA
##
##       YB
## YA       0   1   2   3 <NA>
##   0      0   0   0   0    0
##   1      0   0   0   0    0
##   2      0   0   0   0    0
##   3      0   0   0   0    0
```

```
##    <NA>    0    0    0    0     0
```

The missing data pattern of the combined dataset of populations A, B and E:

```
md.pattern(walking)
```

```
##     sex age src  YA  YB
## 290   1   1   1   1   1   0
## 294   1   1   1   0   1   1
## 300   1   1   1   1   0   1
##   6   1   1   1   0   0   2
##       0   0   0 300 306 606
```

Now, for 290 subjects we have scores on both `YA` and `YB` (from bridge study `E`).

Multiple imputation on the dataset `walking` can now be done as

```
tau <- NULL
imp <- mice(walking, max = 0, m = 10, seed = 92786)
pred <- imp$pred
pred[, c("src", "age", "sex")] <- 0
imp <- mice(walking, max = 0, m = 10, seed = 92786, pred = pred)
micemill(20)
plotit()
```

After five iterations the procedure seems to convergence. Speed of convergence is dependent on the size of the bridge study (now 1/3 of the total dataset). If the relative size of the bridge study was smaller, it might have taken more iterations to reach convergence.

Does the assumption matter?

We have made three different assumptions on the relation between `YA` and `YB`. Does the assumption matter for the conclusion we draw from the data?

Assumption	Mean	Mean	Perc(0)	Perc(0)
-	Study A	Study B	Study A	Study B
Equate	0.24	0.66	81	50
Independence	0.24	0.25	50	50
Bridge	0.24	0.45	58	50

We calculate two statistics of interest:

1. Mean: mean of the distribution, lower indicates a more healthy population
2. Perc(0): percentage zeroes in the distribution, higher indicates a more healthy population

From the table we see

- Under equate: Both according to Mean and Perc(0) persons from study `A` are healthier than persons from study `B`, and by a considerable margin (e.g. 81 versus 50 percent in the zero category).
- Under independence: Both according to Mean and Perc(0) persons from studies `A` and `B` are about equally healthy.

Thus, different assumption may lead to radically different conclusion. We find that

- Equate amplifies the relation between **`YA`** and **`YB`**
- Independence weakens the relation between **`YA`** and **`YB`**

Neither equate or independence is OK. The more reasonable assumption is here the bridge.

PRACTICAL IV: Developmental milestones

Stef van Buuren

Practical 4 of 6

PRELIMINARY NOTE

The `dscore` package is under development, and not yet publicly available. In order to run this document in RStudio, you need to install the `dscore` package from a private Github repository. If you want to do so, please drop a note to Stef van Buuren to getting a proper access key.

Overview

This vignettes shows how to estimate the D-score and the D-score SDS, a.k.a. DAZ in an excerpt from the POPS data. This vignettes covers some typical actions needed when estimating D-scores:

1. Rename item names in source data to item names used in itembank
2. Reorganize the source data into a long matrix
3. Calculate D-score and DAZ
4. Combine D-score and DAZ with source data

Rename item names

The `dscore` package has built-in example data from the POPS study, called `popsdemo`. The data set is of class `tbl_df` from the `dplyr` package.

```
library("dscore")


popsdemo
```

```
## # A tibble: 100 x 67
##    patid gender gestationalage moment   age   occ daycor  dead Fixatesey
es
##    <dbl>  <dbl>          <dbl>  <dbl> <dbl> <dbl>  <dbl> <dbl>       <db
l>
##  1     1      2       30.28571      2   161     1     95     0          0
##  2     1      2       30.28571      3   301     2    236     0          N
aN
##  3     1      2       30.28571      4   511     3    443     0          N
aN
##  4     1      2       30.28571      5  1008     4    940     0          N
aN
```

```
##  5     4      1       32.42857      2   140     1     93     0           0
##  6     4      1       32.42857      3   231     2    184     0         NaN
##  7     4      1       32.42857      4   420     3    368     0         NaN
##  8     4      1       32.42857      5   763     4    716     0         NaN
##  9     5      1       31.57143      2   147     1     94     0           0
## 10     5      1       31.57143      3   238     2    185     0         NaN
## # ... with 90 more rows, and 58 more variables: Reactstospeech <dbl>,
## #   Movesbotharmsequallyasmuch <dbl>, Movesbothlegsequallyasmuch <dbl>,
## #   Liftschin <dbl>, Smilesback <dbl>, Followswitheyesandhead <dbl>,
## #   Handsopennowandthen <dbl>, Looksatownhands <dbl>,
## #   Vocalizesresponsively <dbl>,
## #   Remainspositionedwhenliftedunderthearmpits <dbl>,
## #   Holdsheadupfortyfivedegreesinproneposition <dbl>,
## #   Handsplayinginmidline <dbl>, Graspstoywithinreach <dbl>,
## #   Noheadlagwhenpulledtosittingposition <dbl>, Turnsheadtosound <dbl>,
## #   Whenliftedverticallylegsbendedortrampling <dbl>,
## #   Holdsheadupninetydegreesinproneposition <dbl>,
## #   Transferstoyeasilyhandtohand <dbl>,
## #   Picksuponesmalltoythensecond <dbl>, Playswithbothfeet <dbl>,
## #   Rollsfrompronetosupineandback <dbl>,
## #   Holdsheadupinsittingposition <dbl>, Sitswithstretchedlegs <dbl>,
## #   Saysdadababaorgaga <dbl>, Sitswithoutsupport <dbl>,
## #   Picksupcrumbbetweenthumbandindexfinger <dbl>, Crawls <dbl>,
## #   Pullshimselftostandingposition <dbl>, Wavesbyebye <dbl>,
## #   Jabbering <dbl>, Getscubeintoandoutofbox <dbl>,
## #   Playsgiveandtake <dbl>, Crawlswithbellyliftedofheground <dbl>,
```

```
## #   Walkswhileholdingfurniture <dbl>, Understandssomesimplewords <dbl>,
## #   Usestwowords <dbl>, Makestoweroftwocubes <dbl>, Exploresroom <dbl>,
## #   Usesthreewords <dbl>, Identifiestwonamedobjects <dbl>,
## #   Walksonitsown <dbl>, Throwsballwithoutfalling <dbl>,
## #   Makestowerofthreecubes <dbl>, Imitateseverydayactivities <dbl>,
## #   Drinksfromcup <dbl>, Makestwowordsentences <dbl>,
## #   Putsballinboxwhenasked <dbl>, Squats <dbl>, Walkswell <dbl>,
## #   Makestowerofsixcubes <dbl>, Putsroundfigureintoplace <dbl>,
## #   Takesoffaclothshoesocktrousers <dbl>, Eatswithspoonwithouthelp <dbl>
,
## #   CallsitselfbynameorI <dbl>, Identifiespicturesinbook <dbl>,
## #   Kicksballaway <dbl>, dscore <dbl>, daz <dbl>
```

```
class(popsdemo)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

```
nrow(popsdemo)
```

```
## [1] 100
```

The are 25 children and 4 time points.

```
# 25 children, 4 time points per child
length(unique(popsdemo$patid))
```

```
## [1] 25
```

The item scores that form the test are located in columns 9-65.

```
test <- 9:65
```

These names of the columns need to be matched against one of the lexicons in the item bank. The built-in lexicons are:

```
names(itembank)[1:6]
```

```
## [1] "lex.dutch1996" "lex.dutch2005" "lex.dutch1983" "lex.SMOCC"
## [5] "lex.GHAP"      "lex.jam"
```

We first need to find out a proper lexicon for the data. For the POPS data, the closest lexicon is . Let us check the variable names in POPS with the item labels in the item bank.

```
itemset <- !is.na(itembank$lex.dutch1983)

cbind(names(popsdemo)[test], itembank[itemset, c("lex.dutch1983", "labelEN"
, "tau")])
```

```
##                          names(popsdemo)[test] lex.dutch1983
## 1                                  Fixateseyes            v1
## 2                                Reactstospeech            v2
## 3                    Movesbotharmsequallyasmuch            v3
## 6                    Movesbothlegsequallyasmuch            v4
## 9                                    Liftschin            v5
## 10                                   Smilesback            v6
## 11                      Followswitheyesandhead            v7
## 14                        Handsopennowandthen            v8
## 17                             Looksatownhands            v9
## 18                       Vocalizesresponsively           v10
## 19 Remainspositionedwhenliftedunderthearmpits           v11
## 20 Holdsheadupfortyfivedegreesinproneposition           v12
## 23                       Handsplayinginmidline           v13
## 24                        Graspstoywithinreach           v14
## 27       Noheadlagwhenpulledtosittingposition           v15
## 28                            Turnsheadtosound           v16
## 31  Whenliftedverticallylegsbendedortrampling           v17
## 34    Holdsheadupninetydegreesinproneposition           v18
## 35                Transferstoyeasilyhandtohand           v19
## 36                Picksuponesmalltoythensecond           v20
## 37                            Playswithbothfeet           v21
## 40              Rollsfrompronetosupineandback           v22
## 41                Holdsheadupinsittingposition           v23
```

```
## 42                      Sitswithstretchedlegs          v24
## 43                         Saysdadababaorgaga          v25
## 45                         Sitswithoutsupport          v26
## 46     Picksupcrumbbetweenthumbandindexfinger          v27
## 49                                    Crawls          v28
## 50             Pullshimselftostandingposition          v29
## 51                                Wavesbyebye          v30
## 52                                  Jabbering          v31
## 54                    Getscubeintoandoutofbox          v32
## 57                           Playsgiveandtake          v33
## 58            Crawlswithbellyliftedofheground          v34
## 59                 Walkswhileholdingfurniture          v35
## 60                 Understandssomesimplewords          v36
## 61                               Usestwowords          v37
## 63                      Makestoweroftwocubes          v38
## 66                               Exploresroom          v39
## 67                             Usesthreewords          v40
## 68                  Identifiestwonamedobjects          v41
## 69                               Walksonitsown          v42
## 70                   Throwsballwithoutfalling          v43
## 74                     Makestowerofthreecubes          v44
## 77                 Imitateseverydayactivities          v45
## 78                               Drinksfromcup          v46
## 79                     Makestwowordsentences          v47
## 80                     Putsballinboxwhenasked          v48
## 81                                     Squats          v49
## 82                                  Walkswell          v50
## 84                      Makestowerofsixcubes          v51
## 85                    Putsroundfigureintoplace          v52
```

```
## 86            Takesoffaclothshoesocktrousers          v53
## 87                  Eatswithspoonwithouthelp          v54
## 88                      CallsitselfbynameorI          v55
## 89                  Identifiespicturesinbook          v56
## 90                           Kicksballaway            v57
##                                                    labelEN  tau
## 1                                                Eyes Fixate  5.4
## 2                                      Reacts when spoken to  1.7
## 3                                    Moves arms equally well -2.2
## 6                                    Moves legs equally well -1.9
## 9                          Lifts chin off table for a moment  5.2
## 10                                       Smiles in response 11.3
## 11                                Follows with eyes and head 14.5
## 14                                   Hands occasionally open 16.5
## 17                                       Watches own hands 20.7
## 18                                    Vocalizes in response 14.5
## 19                     Stays suspended when lifted under armpits 15.8
## 20                   Lifts head to 45 degrees in prone position 20.0
## 23                                Plays with hands in midline 28.2
## 24                  Supine position: grasps object within reach 29.9
## 27                             Reactions if pulled to sitting 26.0
## 28                                       Turns head to sound 31.1
## 31                     Flexes or stomps legs while being swung 25.7
## 34         Looks around to side with angle face-table 90 degrees 27.8
## 35                               Passes cube from hand to hand 36.0
## 36               Holds cube, grasps another one with other hand 36.5
## 37                                    Plays with both feet 33.2
## 40                                Rolls over, back and forth 34.7
## 41                           Balances head well while sitting 32.5
```

```
## 42                          Sits on buttocks while legs stretched 34.9
## 43                                Says "dada", "baba", or "gaga" 36.0
## 45                        Sits in stable position, without support 40.0
## 46                 Picks up pellet between thumb and index finger 43.1
## 49                            Crawls forward, abdomen on the floor 43.1
## 50                                  Pulls up to standing position 44.3
## 51                                              Waves "bye bye"  43.1
## 52                    Jabbering while playing (M; can ask parents) 40.9
## 54                                  Puts cube in and out of a box 46.0
## 57                                           Plays "give and take" 46.5
## 58 Crawls, with belly lifted off the ground (M; can ask parents) 46.1
## 59                                                    Walks along 46.1
## 60             Understands some simple words (M; can ask parents) 45.7
## 61                         Says 2 "sound-words" with comprehension 50.1
## 63                                       Builds tower of two cubes 56.4
## 66                                           Explores environment 46.9
## 67                                                 Says 3 "words" 53.2
## 68                Identfies (point / graps) two mentioned objects 55.4
## 69                                           Walks on his/her own 51.9
## 70                               Throws ball without falling down 56.0
## 74                                    Builds tower of three cubes 59.2
## 77                                              Imitates others  52.3
## 78             Drink from cup by him/herself (M; can ask parents) 58.5
## 79                                    Says "sentences"of 2 words 60.2
## 80                                    Puts ball in box when asked 57.8
## 81                               Squats or bends to pick up things 55.3
## 82                                       Walks well without help 55.5
## 84                                       Builds tower of 6 cubes 62.6
## 85                                  Places round form in form-box 60.3
```

```
## 86                                                Undresses himself 60.6
## 87             Eats with spoon without help (M; can ask parents) 58.5
## 88                                Refers to self using "me" or "I" 61.7
## 89                                Points at 5 pictures in the book 62.2
## 90                                                       Kicks ball 64.2
```

In this case, we are lucky that all item names from the source data and the item bank match up exactly. In general, we will need to map carefully the names in the dataset to the names in the item bank. For POPS, we may take out the relevant parts of the item bank as

```
ib <- itembank[itemset,c("lex.dutch1983", "lex.GHAP", "labelEN", "tau")]
head(ib, 3)
```

```
##   lex.dutch1983 lex.GHAP                  labelEN  tau
## 1            v1 GSFIXEYE             Eyes Fixate  5.4
## 2            v2  GSRSPCH   Reacts when spoken to  1.7
## 3            v3   GSMARM Moves arms equally well -2.2
```

From here on, we will work in the GHAP lexicon. Renaming the source data is now done by

```
names(popsdemo)[test] <- as.character(ib$lex.GHAP)
```

The source data has now names that are recognized in the itembank. To check this, find the difficulties for each item by the `gettau()` function:

```
gettau(names(popsdemo)[test])
```

```
## GSFIXEYE  GSRSPCH   GSMARM   GSMLEG GSLFCHIN  GSSMILE   GSFEYE  GSHOPEN
##      5.4      1.7     -2.2     -1.9      5.2     11.3     14.5     16.5
##   GSLKHN  GSVOCAL     GSRP   GSHH45 GSHPLAYM    GSGRP GSNOHLAG  GSTHEAD
##     20.7     14.5     15.8     20.0     28.2     29.9     26.0     31.1
##  GSLBEND   GSHH90   GSTTOY   GSPTOY   GSPLFT  GSROLLS  GSHHSIT  GSSITST
##     25.7     27.8     36.0     36.5     33.2     34.7     32.5     34.9
##   GSSAYS  GSSITWS   GSPICK  GSCRAWL GSPULLST  GSWAVES GSJABBER   GSGETC
##     36.0     40.0     43.1     43.1     44.3     43.1     40.9     46.0
```

```
## GSPLAYGT  GSCRBLY  GSWALKS GSSIMPLE GSTWOWRD  GSMK2CB  GSEXPLR GSTHRWRD
##     46.5     46.1     46.1     45.7     50.1     56.4     46.9     53.2
##  GSIDOBJ GSWLKOWN  GSTBALL  GSMK3CB GSIMITAT GSDRNKCP GSTWOSEN GSPUTBAL
##     55.4     51.9     56.0     59.2     52.3     58.5     60.2     57.8
##   GSPKSQ  GSWLKWH  GSMKTW6 GSPUTFIG  GSTKCLO GSEATSPN  GSREFER GSID5OBJ
##     55.3     55.5     62.6     60.3     60.6     58.5     61.7     62.2
##    GSKIK
##     64.2
```

Reorganize the data into a long matrix

The `dscore()` function takes vectors of item scores, item names and ages. Rearringing the data makes it easy to extract the relevant vectors. We need to create a data set with the following variables: `patid`, `moment`, `age`, `daycor`, `item` and `score`, and select only the rows where we have an observed score.

```
library("tidyr")
library("dplyr")
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data <- popsdemo %>%
  select(patid, moment, age, daycor, GSFIXEYE:GSKIK) %>%
  gather(items, scores, GSFIXEYE:GSKIK, na.rm = TRUE) %>%
  mutate(scores = 1 - scores) %>%
  arrange(patid, moment)
```

```
## Warning: attributes are not identical across measure variables;
```

```
## they will be dropped
```

```
data
```

```
## # A tibble: 1,385 x 6
##    patid moment   age daycor    items scores
##    <dbl>  <dbl> <dbl>  <dbl>    <chr>  <dbl>
##  1     1      2   161     95 GSFIXEYE      1
##  2     1      2   161     95  GSRSPCH      1
##  3     1      2   161     95   GSMARM      1
##  4     1      2   161     95   GSMLEG      1
##  5     1      2   161     95 GSLFCHIN      1
##  6     1      2   161     95  GSSMILE      1
##  7     1      2   161     95   GSFEYE      1
##  8     1      2   161     95  GSHOPEN      1
##  9     1      2   161     95   GSLKHN      1
## 10     1      2   161     95  GSVOCAL      1
## # ... with 1,375 more rows
```

There are `nrow(data)` records with a nonmissing item score. Note also that the item scores have been reversed, as POPS uses a zero for a PASS, and a one for a FAIL.

Calculate D-score and DAZ

For illustration, let us first calculate the D-score of the first child. There are 75 scores for this child, spread over four time points. This is a preterm child, so we correct calener age for gestational age as in `daycor`:

```
child1 <- filter(data, patid == 1)


scores <- child1$scores
items <- as.character(child1$items)
ages <- round(child1$daycor/365.25, 4)


# calculate dscore and daz for each time point for given child
```

```
(d <- dscore(scores, items, ages))
## 0.2601 0.6461 1.2129 2.5736
##  25.25  42.73  55.42  70.97
daz(d)
## 0.2601 0.6461 1.2129 2.5736
##  0.159  0.920  0.788  0.595
```

If desired, one may also back-calculate the D-score from the standard deviation score by

```
zad(daz(d))
## 0.2601 0.6461 1.2129 2.5736
##  25.25  42.73  55.42  70.97
```

If we specify the child identifier as a by-group variable, we may calculate the D-score and DAZ for all children by

```
# use age corrected for gestational age
data <- data.frame(data)
data$ages <- round(data$daycor/365.25, 4)


# calculate D-score and DAZ
ds <- split(data, data$patid)
dl <- parallel::mclapply(ds, FUN = dscore)
dazl <- lapply(dl, FUN = daz)
df <- data.frame(
  patid = rep(as.numeric(names(dl)), times = unlist(lapply(dl, length))),
  ages = as.numeric(unlist(lapply(dl, names))),
  dscore = as.numeric(unlist(dl)),
  daz = as.numeric(unlist(dazl)))
head(df)
##   patid   ages dscore    daz
```

```
## 1     1 0.2601  25.25  0.159
## 2     1 0.6461  42.73  0.920
## 3     1 1.2129  55.42  0.788
## 4     1 2.5736  70.97  0.595
## 5     4 0.2546  23.15 -0.416
## 6     4 0.5038  31.75 -1.202
```

Combine D-score and DAZ with source data

Finally, in order to do further analyses, we need to put the estimated D-score and DAZ back into the source data.

```
# merge dscore and daz into popsdemo data
popsdemo$ages <- round(popsdemo$daycor/365.25, 4)
popsdemo <- merge(popsdemo, df, all.x = TRUE)
head(select(popsdemo, patid, moment, ages, dscore, daz))
```

```
##   patid moment   ages dscore    daz
## 1     1      2 0.2601  25.26  0.163
## 2     1      3 0.6461  42.72  0.916
## 3     1      4 1.2129  55.44  0.794
## 4     1      5 2.5736  70.82  0.541
## 5     4      2 0.2546  23.15 -0.416
## 6     4      3 0.5038  31.75 -1.202
```

PRACTICAL V: Loss-to-Follow-Up

Aurelie Piedvache, Manon Grevinga, Stef van Buuren

Practical 5 of 6

We use the following libraries:

```
library(mice)
```

First, we have to get the data. Make sure that the path is changed to the path you saved the datafile.

```
file <- path.expand("~/Project/060.19899 RECAP/Kluis/WP5 Statistical Methods/Workshop/Aurelie INSERM/data_July2017.txt")

mydata <- read.csv(file = file, na = "NA", stringsAsFactors=TRUE)

str(mydata, list.len = 999)
```

```
## 'data.frame':    5070 obs. of  6 variables:

##  $ a4_weeks                : int  24 31 30 29 31 31 29 28 31 29 ...

##  $ gmi_vi_hi_parca_asq_ten2: int  NA 0 0 NA 0 0 NA NA 0 0 ...

##  $ follow                  : int  0 1 1 0 1 1 0 0 1 1 ...

##  $ motherage               : int  25 29 41 40 41 40 37 35 29 30 ...

##  $ f10                     : int  1 0 1 0 1 1 0 0 1 0 ...

##  $ native2                 : int  1 1 1 1 0 1 1 1 NA 1 ...
```

Categorize alle variables except ***motherage*** by the following chunk of code:

```
varfactor <- c("a4_weeks","gmi_vi_hi_parca_asq_ten2","native2","f10","follow")

mydata[,varfactor] <- lapply(mydata[,varfactor] , factor)

str(mydata, list.len = 999)
```

```
## 'data.frame':    5070 obs. of  6 variables:

##  $ a4_weeks                : Factor w/ 9 levels "23","24","25",..: 2 9 8 7 9 9 7 6 9 7 ...

##  $ gmi_vi_hi_parca_asq_ten2: Factor w/ 2 levels "0","1": NA 1 1 NA 1 1 NA NA 1 1 ...
```

```
##  $ follow                  : Factor w/ 2 levels "0","1": 1 2 2 1 2 2 1 1
2 2 ...
##  $ motherage               : int  25 29 41 40 41 40 37 35 29 30 ...
##  $ f10                     : Factor w/ 2 levels "0","1": 2 1 2 1 2 2 1 1
2 1 ...
##  $ native2                 : Factor w/ 2 levels "0","1": 2 2 2 2 1 2 2 2
NA 2 ...
```

```
dim(mydata) #the number of observations and the number of variables
```

```
## [1] 5070    6
```

```
summary(mydata)
```

```
##     a4_weeks    gmi_vi_hi_parca_asq_ten2 follow     motherage
##  31     :1397   0   :2355                0:1757   Min.   :14.00
##  30     :1085   1   : 719                1:3313   1st Qu.:26.00
##  29     : 741   NA's:1996                         Median :31.00
##  28     : 620                                     Mean   :30.59
##  27     : 485                                     3rd Qu.:35.00
##  26     : 358                                     Max.   :53.00
##  (Other): 384                                     NA's   :18
##    f10       native2
##  0   :2058   0   : 995
##  1   :2880   1   :3748
##  NA's: 132   NA's: 327
##
##
##
##
```

To create the response indicator:

```
r <- mydata$follow == 1
```

Method 1: Get the crude prevalence

```
neuro <- as.numeric(mydata[r, "gmi_vi_hi_parca_asq_ten2"])-1


#number of responders

nb_responders <- length(which(mydata$follow == 1))

#number of responders without missing values on outcome

nb_responders_wo_miss <- length(which(neuro != "NA")) - sum(is.na(neuro))


1-(nb_responders_wo_miss/nb_responders) # 14% of missing values for the out
come - a lot
```

```
## [1] 0.1442801
```

```
mean_crude <- mean(neuro,na.rm = TRUE)*100

mean_crude
```

```
## [1] 23.38972
```

Method 2: Corrected the prevalence with taking into account non-responders - no correction on missing values.

```
## fit logistic regression model wihtout imputation

fit0 <- glm(follow == 1 ~ native2 + f10 + motherage + a4_weeks, family = bi
nomial(),na.action = na.exclude,data=mydata)


prop0 <- predict(fit0, type = "response")

weight0 <- 1/prop0


new_data <- cbind(mydata,weight0)

new_data <- na.omit(new_data)

new_data <- subset(new_data,follow == 1)

summary(new_data)
```

```
##     a4_weeks    gmi_vi_hi_parca_asq_ten2 follow     motherage     f10
```

```
##  31      :761   0:2201                    0:   0   Min.   :15.0   0:1053
##  30      :648   1: 679                    1:2880   1st Qu.:28.0   1:1827
##  29      :411                                      Median :31.0
##  28      :352                                      Mean   :31.4
##  27      :294                                      3rd Qu.:35.0
##  26      :210                                      Max.   :52.0
##  (Other):204
##  native2     weight0
##  0: 493   Min.   :1.086
##  1:2387   1st Qu.:1.313
##           Median :1.423
##           Mean   :1.499
##           3rd Qu.:1.609
##           Max.   :3.594
##
```

```
mean_weighted <- (weighted.mean(x = as.numeric(new_data[,"gmi_vi_hi_parca_asq_ten2"]), w = new_data[, "weight0"])-1)*100

mean_weighted
```

```
## [1] 23.98952
```

Method 3: Corrected the prevalence without taking into account non-responders - correction on missing values

```
## To get the number of missing values in your dataset

1-(sum(complete.cases(mydata))/dim(mydata)[1])
```

```
## [1] 0.4319527
```

```
# md.pattern(mydata)

# fluxplot(mydata)


ini <- mice(mydata, maxit = 0, m = 43, seed = 12345)
```

```
imp <- mice.mids(ini, maxit = 10, print = FALSE)


plot(imp)
```

```
#library("lattice")

#bwplot(imp, ~ motherage)



long <- complete(imp, "long", include = TRUE)

long$neuro <-  as.numeric(long$gmi_vi_hi_parca_asq_ten2) - 1


long2<-aggregate(long, by = list(long$.imp), FUN = mean, na.rm = TRUE)


mean_imp <- mean(subset(long2, Group.1 !="0")$neuro)*100

mean_imp
## [1] 24.03926
```

Method 4: Corrected the prevalence with taking into account non-responders - correction on missing values

```
## fit logistic regression model

fit <- with(imp, glm(follow == 1 ~ native2 + f10 + motherage + a4_weeks, fa
mily = binomial()))


prop <- matrix(NA, nrow = length(fit$analyses[[1]]$weights),

ncol = length(fit$analyses))


for (i in 1:length(fit$analyses)) {

prop[, i] <- predict(fit$analyses[[i]], type = "response")

}
```

```
propensity <- rowMeans(prop)

# construct inverse weights

weight_all <- 1/propensity

summary(weight_all)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.

##   1.096   1.360   1.496   1.579   1.723   3.636
```

```
hist(weight_all)
```

```
# select weight for followed-up respondents

weight <- weight_all[mydata$follow==1]

summary(weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.

##   1.096   1.340   1.452   1.531   1.650   3.636
```

```
hist(weight)
```

```
# two histograms

hist(weight_all, col = "grey")

hist(weight, col = "blue", add = TRUE)
```

```
neuro_imp <- as.numeric(imp$data[r, "gmi_vi_hi_parca_asq_ten2"]) - 1

mean_imp_weighted <- weighted.mean(x = neuro_imp, w = weight, na.rm = TRUE)
*100

#print results

cat("crude=",mean_crude," weigthed=",mean_weighted," imputed=",mean_imp," i
mputed and weighted=",mean_imp_weighted)
```

```
## crude= 23.38972  weigthed= 23.98952  imputed= 24.03926  imputed and weig
hted= 23.85348
```

PRACTICAL VI: Multilevel Analysis

Manon Grevinga, Stef van Buuren

Practical 6 of 6

For this example, we use a dataset from the package `mlmRev`.

```
library(mlmRev)
```

```
## Loading required package: lme4
```

```
## Loading required package: Matrix
```

The dataset is called `Exam`, and contains simulated data about examresults of children. However, since it is simulated data with an multilevel structure, we can rename the variables to something more related to RECAP. Which is what we will do, to show how such a multilevel structure works. In this case we assume that each cohort study collected the same variables (in the same units), there are no missings and the cohort studies started at the same time. Note, that this situation will practically never happen. However, to keep things simple in order to explain the multilevel analysis we will assume it holds.

In this example we want to explain birthweight by gestational age, gender and a cohort specific variable homebirth (the number of home births per 100 childbirths) to keep things simple. Moreover, for each child we know to which cohort study it belongs.

To get the data from the package `mlmRev` run the following code chunk

```
data(Exam) #get the data

child.data <- Exam[, c(1, 2, 4, 7, 8, 10)] #keep only a few variables

names(child.data) <- c("cohort", "birthweightnorm", "homebirth", "gestation
al.agenorm", "gender", "child") #rename the variables

head(child.data)
```

```
##   cohort birthweightnorm homebirth gestational.agenorm gender child
## 1      1           0.261     0.166               0.619      F   143
## 2      1           0.134     0.166               0.206      F   145
## 3      1          -1.724     0.166              -1.365      M   142
## 4      1           0.968     0.166               0.206      F   141
## 5      1           0.544     0.166               0.371      F   138
```

```
## 6      1           1.735     0.166               2.189      M   155
```

The simulated variables in this example where more or less standard normally distributed. This makes it easy to change the variables to gram (for birthweight) and weeks (for gestational age). We assume that the average birthweight is equal to 1325 gram with a standard deviation of 75. The average gestational age is 30 weeks with a standard deviation equal to 0.65. When running the following chunk of code the standard normally distributed variables are changed to variables in grams and weeks.

```
#change the standardized birthweight to birthweight in gram
child.data$birthweight <- (child.data$birthweightnorm)*75+1325


#Change the standardized gestational age to gestational age in weeks in two
decimals
child.data$gestational.age <- (child.data$gestational.agenorm)*0.65+30
child.data$gestational.age <- round(child.data$gestational.age, digits=2)


#keep only the relevant variables
child.data <- child.data[, c(1, 3, 5, 6, 7, 8)]


#Give each child it own specific childnumber (instead of per cohort study)
child.data$childnr <- seq.int(nrow(child.data))
```

The example contains 65 schools (we renamed them to cohort studies). To make this example more relatable to RECAP we will combine some similar schools/cohort studies by running the following chunk of code. We end up with cohorts A till T (20 cohort studies).

```
child.data$cohort <- as.numeric(child.data$cohort)
child.data[child.data$cohort %in%  c('1', '20', '11', '52'), 1] <- 'A'
child.data[child.data$cohort %in% c('2', '3', '55'), 1] <- 'B'
child.data[child.data$cohort %in% c('4', '29', '33', '49'), 1] <- 'C'
child.data[child.data$cohort %in% c('5', '7', '21'), 1] <- 'D'
child.data[child.data$cohort %in% c('6', '53', '63'), 1] <- 'E'
```

```
child.data[child.data$cohort %in% c('8', '15', '47', '48'), 1] <- 'F'

child.data[child.data$cohort %in% c('9', '26', '44', '54'), 1] <- 'G'

child.data[child.data$cohort %in% c('10', '16', '31', '40'), 1] <- 'H'

child.data[child.data$cohort %in% c('12', '61', '56'), 1] <- 'I'

child.data[child.data$cohort %in% c('13', '17', '36', '45'), 1] <- 'J'

child.data[child.data$cohort %in% c('14', '24', '62'), 1] <- 'K'

child.data[child.data$cohort %in% c('18', '42', '57'), 1] <- 'L'

child.data[child.data$cohort %in% c('19', '43', '60'), 1] <- 'M'

child.data[child.data$cohort %in% c('22', '46'), 1] <- 'N'

child.data[child.data$cohort %in% c('23', '25', '37'), 1] <- 'O'

child.data[child.data$cohort %in% c('27', '32', '34'), 1] <- 'P'

child.data[child.data$cohort %in% c('28', '59'), 1] <- 'Q'

child.data[child.data$cohort %in% c('30', '58', '64'), 1] <- 'R'

child.data[child.data$cohort %in% c('35', '39', '41'), 1] <- 'S'

child.data[child.data$cohort %in% c('38', '50', '51', '65'), 1] <- 'T'


#Sort the data by cohort

child.data$cohort <- sort(child.data$cohort, decreasing=FALSE)
```

Since we combined different cohort studies, we should redefine the cohort specific variable homebirth which should have the same value for each child in the same cohort. For the new (combined) cohorts we will take the average value of the cohort specific variable of the cohorts that were combined.

```
#Make one cohort variable

child.data[child.data$cohort == 'A', 2] <- mean(child.data[child.data$cohor
t == 'A', 2])

child.data[child.data$cohort == 'B', 2] <- mean(child.data[child.data$cohor
t == 'B', 2])

child.data[child.data$cohort == 'C', 2] <- mean(child.data[child.data$cohor
t == 'C', 2])
```

```
child.data[child.data$cohort == 'D', 2] <- mean(child.data[child.data$cohor
t == 'D', 2])

child.data[child.data$cohort == 'E', 2] <- mean(child.data[child.data$cohor
t == 'E', 2])

child.data[child.data$cohort == 'F', 2] <- mean(child.data[child.data$cohor
t == 'F', 2])

child.data[child.data$cohort == 'G', 2] <- mean(child.data[child.data$cohor
t == 'G', 2])

child.data[child.data$cohort == 'H', 2] <- mean(child.data[child.data$cohor
t == 'H', 2])

child.data[child.data$cohort == 'I', 2] <- mean(child.data[child.data$cohor
t == 'I', 2])

child.data[child.data$cohort == 'J', 2] <- mean(child.data[child.data$cohor
t == 'J', 2])

child.data[child.data$cohort == 'K', 2] <- mean(child.data[child.data$cohor
t == 'K', 2])

child.data[child.data$cohort == 'L', 2] <- mean(child.data[child.data$cohor
t == 'L', 2])

child.data[child.data$cohort == 'M', 2] <- mean(child.data[child.data$cohor
t == 'M', 2])

child.data[child.data$cohort == 'N', 2] <- mean(child.data[child.data$cohor
t == 'N', 2])

child.data[child.data$cohort == 'O', 2] <- mean(child.data[child.data$cohor
t == 'O', 2])

child.data[child.data$cohort == 'P', 2] <- mean(child.data[child.data$cohor
t == 'P', 2])

child.data[child.data$cohort == 'Q', 2] <- mean(child.data[child.data$cohor
t == 'Q', 2])

child.data[child.data$cohort == 'R', 2] <- mean(child.data[child.data$cohor
t == 'R', 2])

child.data[child.data$cohort == 'S', 2] <- mean(child.data[child.data$cohor
t == 'S', 2])

child.data[child.data$cohort == 'T', 2] <- mean(child.data[child.data$cohor
t == 'T', 2])
```

```
child.data$homebirth <- ((child.data$homebirth+0.8)/2)*100
child.data$homebirth <- round(child.data$homebirth, digits=0)
```

Now, the data is ready to be used for multilevel modelling. Note that the multilevel structure is as follows: level 1 contains the childeren and level 2 contains the cohort studies.

We can plot the birthweight (the outcome we want to explain) for each cohort studie included in the study by running the following chunk of code:

```
plot(as.factor(child.data$cohort), child.data$birthweight,
xlab="cohort study", ylab="birthweight", main= "Boxplot of the birthweights
")
```

From this plot, we can see that there is variation in birthweight between the different cohort studies: the median birthweight differs per cohort study. Moreover, the variability of birthweight within each cohort studies might also differ: the size of the white boxes (the first quantile - third quantile) differ per cohort study.

First, we will start with a linear regression model that does not take the multilevel structure into account.

```
#Normal linear regression model without cohort specific variable
LS.model <- lm(birthweight ~ gestational.age + gender, data = child.data)
summary(LS.model)
```

```
##
## Call:
## lm(formula = birthweight ~ gestational.age + gender, data = child.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -192.25  -38.97    1.34   40.25  218.08
##
## Coefficients:
```

```
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -714.68      43.97   -16.3  < 2e-16 ***
## gestational.age    68.16       1.46    46.6  < 2e-16 ***
## genderM           -12.73       1.93    -6.6  4.5e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60.1 on 4056 degrees of freedom
## Multiple R-squared:  0.357,  Adjusted R-squared:  0.357
## F-statistic: 1.13e+03 on 2 and 4056 DF,  p-value: <2e-16
```

```
#Normal linear regression model with cohort specific variable
LS.model1 <- lm(birthweight ~ gestational.age + gender + homebirth, data = child.data)
summary(LS.model1)
```

```
##
## Call:
## lm(formula = birthweight ~ gestational.age + gender + homebirth,
##     data = child.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -203.34  -38.09    1.23   41.01  208.87
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -663.0199    44.2425  -14.99  < 2e-16 ***
## gestational.age   65.5935     1.4952   43.87  < 2e-16 ***
## genderM          -12.2867     1.9167   -6.41  1.6e-10 ***
## homebirth          0.6234     0.0848    7.35  2.3e-13 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 59.7 on 4055 degrees of freedom
## Multiple R-squared:  0.365,  Adjusted R-squared:  0.365
## F-statistic:  778 on 3 and 4055 DF,  p-value: <2e-16
```

In both models all variables are statistically significant. Besides ignoring the multilevel structure, one commonly used method is to add a dummy variable for each cohort study. This means including 20 dummy variables.

```
#Normal linear regression model without cohort specific variable
LS.model2 <- lm(birthweight ~ gestational.age + gender + factor(cohort), da
ta = child.data)
summary(LS.model2)
```

```
##
## Call:
## lm(formula = birthweight ~ gestational.age + gender + factor(cohort),
##     data = child.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -227.70  -37.96    2.14   39.79  194.29
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -602.60      44.31  -13.60  < 2e-16 ***
## gestational.age    65.46       1.46   44.79  < 2e-16 ***
## genderM           -14.22       1.94   -7.32  3.1e-13 ***
## factor(cohort)B    -3.83       6.02   -0.64     0.52
## factor(cohort)C   -29.03       4.93   -5.89  4.2e-09 ***
```

```
## factor(cohort)D   -43.35       5.64   -7.68  2.0e-14 ***
## factor(cohort)E   -40.13       5.78   -6.94  4.4e-12 ***
## factor(cohort)F   -52.20       5.22  -10.00  < 2e-16 ***
## factor(cohort)G   -29.06       6.15   -4.73  2.4e-06 ***
## factor(cohort)H   -42.17       5.28   -7.99  1.7e-15 ***
## factor(cohort)I   -33.49       6.17   -5.43  6.1e-08 ***
## factor(cohort)J   -34.76       5.08   -6.85  8.7e-12 ***
## factor(cohort)K   -32.10       5.08   -6.32  2.9e-10 ***
## factor(cohort)L   -31.37       5.35   -5.86  5.0e-09 ***
## factor(cohort)M   -45.05       5.66   -7.95  2.3e-15 ***
## factor(cohort)N   -33.05       5.85   -5.65  1.8e-08 ***
## factor(cohort)O   -44.21       6.51   -6.79  1.3e-11 ***
## factor(cohort)P    10.93       6.81    1.61     0.11
## factor(cohort)Q     2.74       6.87    0.40     0.69
## factor(cohort)R   -30.72       6.27   -4.90  1.0e-06 ***
## factor(cohort)S   -34.69       6.18   -5.62  2.1e-08 ***
## factor(cohort)T   -27.15       5.23   -5.19  2.2e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 58.3 on 4037 degrees of freedom
## Multiple R-squared:  0.397,  Adjusted R-squared:  0.394
## F-statistic:  127 on 21 and 4037 DF,  p-value: <2e-16
```

```
#Normal linear regression model with cohort specific variable
LS.model3 <- lm(birthweight ~ gestational.age + gender + homebirth + factor
(cohort), data = child.data)
summary(LS.model3)
```

```
##
## Call:
```

```
## lm(formula = birthweight ~ gestational.age + gender + homebirth +
##     factor(cohort), data = child.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -227.70  -37.96    2.14   39.79  194.29
##
## Coefficients: (1 not defined because of singularities)
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -749.206     50.310  -14.89  < 2e-16 ***
## gestational.age   65.461      1.461   44.79  < 2e-16 ***
## genderM          -14.215      1.943   -7.32  3.1e-13 ***
## homebirth          2.715      0.523    5.19  2.2e-07 ***
## factor(cohort)B  -17.404      7.587   -2.29   0.0218 *
## factor(cohort)C   -1.886      4.761   -0.40   0.6920
## factor(cohort)D  -24.342      4.997   -4.87  1.1e-06 ***
## factor(cohort)E  -42.845      6.047   -7.09  1.6e-12 ***
## factor(cohort)F  -25.052      5.066   -4.94  7.9e-07 ***
## factor(cohort)G  -34.487      6.685   -5.16  2.6e-07 ***
## factor(cohort)H   14.838      9.351    1.59   0.1126
## factor(cohort)I   91.393     21.961    4.16  3.2e-05 ***
## factor(cohort)J   41.260     12.601    3.27   0.0011 **
## factor(cohort)K   30.343     10.174    2.98   0.0029 **
## factor(cohort)L    6.635      6.453    1.03   0.3039
## factor(cohort)M   14.676     10.019    1.46   0.1430
## factor(cohort)N   10.389      7.609    1.37   0.1722
## factor(cohort)O   18.232     10.959    1.66   0.0963 .
## factor(cohort)P   13.640      6.611    2.06   0.0392 *
## factor(cohort)Q    5.459      6.680    0.82   0.4138
```

```
## factor(cohort)R   15.431      8.300    1.86   0.0631 .
## factor(cohort)S   25.034     10.325    2.42   0.0154 *
## factor(cohort)T       NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 58.3 on 4037 degrees of freedom
## Multiple R-squared:  0.397,  Adjusted R-squared:  0.394
## F-statistic:  127 on 21 and 4037 DF,  p-value: <2e-16
```

When adding dummy variables for each cohort (without the cohort specific variable), we can see that almost all dummy variables are statistically significant. However, when the cohort specific variable is added to the model, we can see that a lot of dummy variables are not statistically significant anymore. This is because the cohort specific variable already explains some of the variability between the cohort studies. Moreover, for one of the dummy variables the estimate is non-available (NA). This is due because the variable is linearly related to another one.

When using the model structure where the cohort studies are represented by dummy variables, one assumes that the observations are still independent of each other. However, children from one cohort study might be more similar than to a randomly choosing other child from one of the other cohort studies. Or to put it more simple, a child from two parents is probably more similar to another child from the same parents (brother or sister) then a randomly chosen other child. Thus, adding dummy variables for each cohort study does not take this correlation into account. This could potentially lead to wrongly calculated standard errors (too low) leading to overstatement of the statistical significance.

One way to take this correlation structure into account is by means of a mixed effects model. We will in this workshop only look at the random intercepts model, since the workshop tries to explain why some methods might be needed for RECAP and not go into full detail of these methods. With the random intercepts model each cohort study has its own intercept consisting of a fixed part (which is similar for each cohort study) and a random part. This random part is different for each cohort study, however will on average be equal to zero. Running the following chunk of code will estimate a random intercepts model:

```
randint.model <- lmer(birthweight ~ gestational.age + gender + homebirth +
(1|cohort), data=child.data)

summary(randint.model)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: birthweight ~ gestational.age + gender + homebirth + (1 | cohor
t)
##    Data: child.data
##
## REML criterion at convergence: 44566
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -3.851 -0.651  0.034  0.688  3.335
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  cohort   (Intercept)  210     14.5
##  Residual             3403     58.3
## Number of obs: 4059, groups:  cohort, 20
##
## Fixed effects:
##                 Estimate Std. Error t value
## (Intercept)     -660.056     44.685   -14.8
## gestational.age   65.472      1.461    44.8
## genderM          -14.081      1.938    -7.3
## homebirth          0.692      0.277     2.5
##
## Correlation of Fixed Effects:
##             (Intr) gsttn. gendrM
## gestatinl.g -0.965
## genderM     -0.066  0.046
## homebirth   -0.184 -0.070  0.011
```

Note, that all the coefficient estimates are significant (|t-value| > 2). Moreover, the coefficient estimate for the cohort specific variable is now much smaller than in the model with dummy variables for each cohort study (140.5824). Thus, in the model with a random intercept the variation between the cohort studies is explained by the random intercepts instead of this variable.

To get the random parts of the intercept for each of the cohort studies we can run the following chunk of code:

```
ranef(randint.model)
```

```
## $cohort
##   (Intercept)
## A      18.438
## B      11.298
## C      -2.289
## D     -17.353
## E     -19.372
## F     -24.114
## G      -9.648
## H      -7.456
## I      16.265
## J       4.135
## K       3.331
## L      -1.847
## M      -9.361
## N      -2.059
## O      -7.588
## P      27.190
## Q      20.036
## R       0.651
## S       0.238
## T      -0.495
```

We can also calculate the ICC. Remember the rule of thumb: if the ICC > 5% it is advised to use a mixed effects model.

```
varcor <- as.data.frame(VarCorr(randint.model))

ICC <- varcor[1,4]/(varcor[1,4] + varcor[2,4])

ICC
```

```
## [1] 0.0581
```

Hence, for this study it was a good choice to use a mixed effects model.