S23 Summer School

Advanced Course on using Mplus

Caspar J. van Lissa[1] (only this GitBook; rest of the course by colleagues)

[1]Utrecht University, Methodology & Statistics

Contents

Course

This course material is part of the Advanced Course on using Mplus, a five-day summer school course hosted by Utrecht University's department of Methodology and Statistics. If you already know how to analyse your data in Mplus but want to learn more about what you are actually doing, and especially if you want to know more about advanced longitudinal analyses, this course is for you. The course consists of in-depth lectures on the fundamentals of Mplus and advanced longitudinal models.

Chapter 1

Preparing for the course

This Chapter helps you prepare for the course. It shows how to install R and RStudio on your computer. We'll also provide some general information on R, and how you can get help if you get error messages.

If you're already using R, all of this might be nothing new for you. You may **skip** this chapter then.

If you have **never used R before, this Chapter is essential**, as it gives you some input on how R works, and how we can use it for our data analyses.

1.1 Installing software

If you use R on your own computer, you will need to install it yourself. You should first:

1. Install R from https://CRAN.R-project.org
2. Install 'RStudio' Desktop (Free) from https://rstudio.com

1.1.1 Installing packages

As a prerequisite for this guide, you need to have a few essential **R packages** installed.

1. Open RStudio
2. Inside RStudio, find the window named **Console** on the bottom left corner of your screen (it might fill the entire left side of the screen).

3. We will now install a few packages using R Code. Here's an overview of the packages, and why we need them:

Package	Description
MplusAutomation	Control Mplus from R and parse model output
ggplot2	A flexible and user-friendly plotting package
tidySEM	Plotting and tabulating the output of SEM-models
semTools	Comparing models, establishing measurement invariance across groups

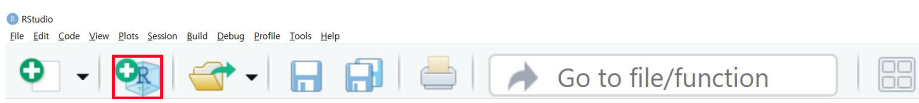
To install these packages, we use the `install.packages()` function in R. One package after another, our code should look like this:

```
install.packages("MplusAutomation")
install.packages("ggplot2")
install.packages("tidySEM")
install.packages("semTools")
```

1.1.2 Get started

1.1.3 Starting a new project in Rstudio

To keep all your work organized, you should use a **project**. In Rstudio, click on the *New project* button:



In the pop-up dialog, click *New directory*, and again *New project*.

type the desired directory name in the dialog (give it a meaningful name, e.g. "TCSM_course"), and use 'Browse' if you need to change the directory where you store your projects. Now, in your project, click *File > New file > R script*. This script file works just like notepad, or the syntax editor in SPSS: You type plain text, but you can run it any time you want. Conduct all of the exercises in this script file.

1.1.4 Code conventions

Throughout the guide, a consistent set of conventions is used to refer to code:

- Functions are in a code font and followed by parentheses, like `sum()` or `mean()`.
- Other R objects (like data or function arguments) are in a code font, without parentheses, like `seTE` or `method.tau`.

- Sometimes, we'll use the package name followed by two colons, like `lavaan::sem()`. This is valid R code and will run. The `lavaan::` part indicates that the function `sem()` comes from the package `lavaan`.

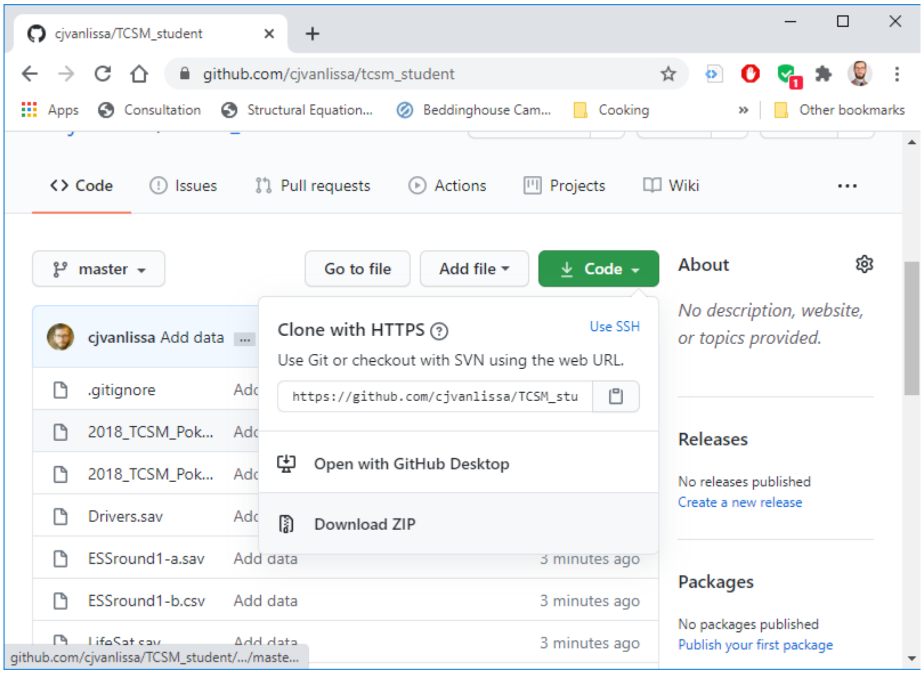
1.1.5 Getting Help

As you start to apply the techniques described in this guide to your data you will soon find questions that the guide does not answer. This section describes a few tips on how to get help.

1. Every function in R has documentation (a help file). To see it, select the name of the function and press F1, or run the command `?` followed by the name of the function, e.g.: `?aov`. I have been using R for 10 years, and I still press F1 all the time to see how a function works.
2. If you get stuck, start with **Google**. Typically, adding "R" to a search is enough to restrict it to relevant results, e.g.: "exploratory factor analysis R." Google is particularly useful for error messages. If you get an error message and you have no idea what it means, try googling it. Chances are that someone else has been confused by it in the past, and there will be help somewhere on the web. (If the error message isn't in English, run `Sys.setenv(LANGUAGE = "en")` and re-run the code; you're more likely to find help for English error messages.)
3. If Google doesn't help, try stackoverflow. Start by spending a little time searching for an existing answer; including [R] restricts your search to questions and answers that use R.
4. Lastly, if you stumble upon an error (or typos!) in this guide's text or R syntax, feel free to contact **Caspar van Lissa** at **c.j.vanlissa@uu.nl**.

1.2 Getting the course data

All of the course data files are available on a GitHub repository. You can download them all at once by going to https://github.com/cjvanlissa/S23_student, clicking the green button labeled 'Code,' and downloading a ZIP archive of the repository.



After unzipping the archive, you can open the RStudio project 'S23_student.Rproj,' and the script 'run_me.R.' This script contains a few lines of code to help you install the required R-packages for the course.

1.3 R tutorial for beginners (optional)

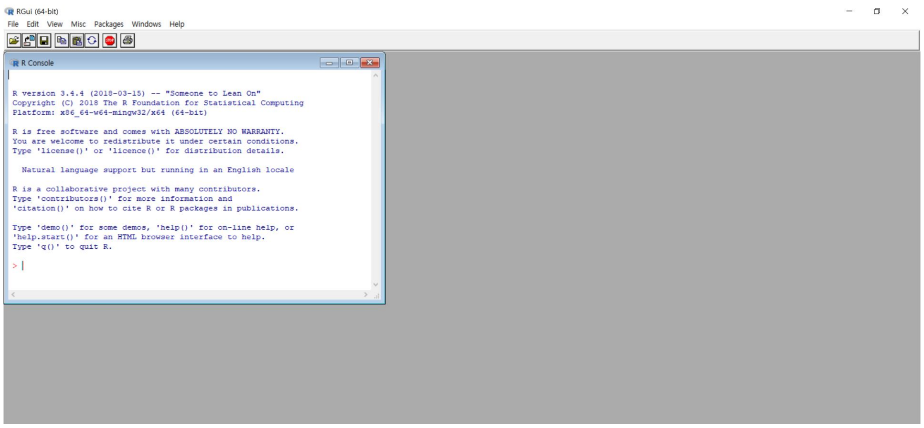
Welcome to the world of R! This tutorial is based on the tutorial "R: How to get started" by Ihnwhi Heo, Duco Veen, and Rens van de Schoot, and adapted for TCSM.

1.3.1 Who R you?

R is...

- Free programming software for statistical computation and graphics
- Open source: everyone (even you!) can improve, develop, and contribute to R
- The official manual by the R Core Team: An introduction to R

R itself looks a bit old-fashioned and tedious:



1.3.2 RStudio

Thankfully, we have a great user interface for R, called RStudio!

- RStudio helps users to use and learn R easier
- If you are using RStudio, this means you are using R.
- From now on, all tutorials will go with RStudio.

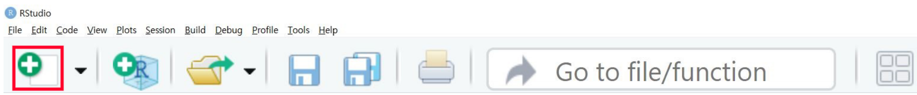
1.3.2.1 No 'pane,' no gain!

When you open RStudio, the screen may look like this. You may notice that the screen is divided into A 'panes' (a pane is a division of a window):

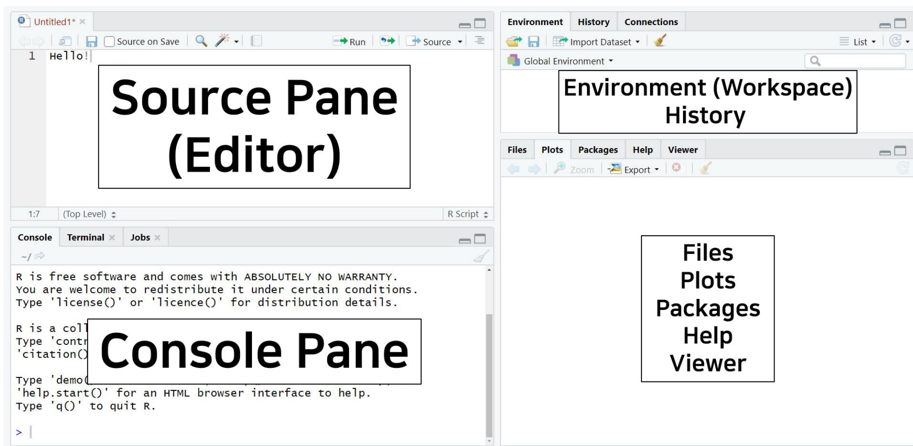
Before we explain these three panes - I want you to add the fourth one, which you will see if you open an R script. An R script is like a "new document" in Microsoft Word. When you open an R script, the fourth pane appears.

1.3.2.2 Create a new R script

Click the icon with a plus sign on the paper. Click the icon highlighted by the red square:



When you click the icon, a new script appears in a fourth pane on the upper left side of the screen



The four panes really help become organized. In RStudio, you can do everything all together on one screen. Thus, four panes make the work efficient (indeed, no ‘pain!’).

1.3.2.3 What do the four panes do?

- Out of four panes, the two on the left side are the panes you will use a lot.
  - Source pane: located at the top left side of the screen. It is also called the “editor,” because this is where we edit scripts. We will usually type our code in the source pane.
  - Console pane: located at the bottom left side of the screen. This panel is for direct communication with R. We can type commands here that are *immediately* evaluated (whereas a script is only evaluated when we run it). Furthermore, all output of our commands is printed in this console pane.

- The panels on the right side of the screen contain various tabs. Among those tabs, it is worth looking at the Environment tab at the upper pane and the Plots tab at the lower pane.

- The Environment tab contains all the 'objects' currently loaded in your R session. In SPSS, you can have only one data file open. In R, you can have as many data 'objects' as you like. They will be listed here. You can always check what objects are loaded under the environment tab. The environment is also called the 'workspace.'
- The Plots tab shows various graphs and figures we draw. If you click Zoom with the magnifying glass, you can see plots in a bigger size.

Chapter 2

Day 2: Latent Growth Models

This computer lab session demonstrates to run latent growth models in batch, using the R-package `MplusAutomation`. Note that, if you do not want to automate part of your workflow (like making plots and tables), you can also use Mplus exclusively. All of the input files for the exercises described in this GitBook are provided with the course materials.

To get started with today's computer lab, first open the project file called `"S23_student.Rproj"`. It should load in the program "RStudio." The bottom right panel has several tabs, including one called "Files." Click on "Files" in this bottom right tab, and click the file "growth_exercises.R."

2.1 Exercise 1: burn survivors

The file `"PTSD.dat"` contains data on burn survivors. An incomplete, basic Mplus syntax can be found in the file `"PTSD - M0.inp"`.

Specifying that same syntax could be done in R, using `MplusAutomation`, as well. First, load the file `PTSD.dat` into the R environment. For convenience's sake, rename the columns of the data object to something a human would understand:

```
data <- read.table("PTSD.dat", na.strings = -999)
names(data) <- c("gender", "tvlo",
                 "W1", "W2", "W3", "W4", "W5", "W6", "W7", "W8",
                 "pain")
```

To specify a basic, *incomplete* syntax, use:

```
basic <- mplusObject(
  TITLE = "exercise 1",
  MODEL = "",
  OUTPUT = "standardized;",
  PLOT = "SERIES = w1-w8 (s);
          TYPE = PLOT3;",
  rdata = data,
  usevariables =
    c("W1", "W2", "W3", "W4", "W5", "W6", "W7", "W8")
)
```

To evaluate this model, you can use:

```
result <- mplusModeler(basic, modelout = "basic.inp", run = 1L)
```

The argument `run = 1L` creates the Mplus input file and data, evaluates the input file, thus creating an output file, and reads the results into R.

To summarize the results, you can use:

```
# For one model
SummaryTable(result)
# For more models
SummaryTable(list(result1, result2))
```

2.1.1 Exercise 1a

Specify a latent growth model. Consider different specifications discussed in the lecture, and try to find the best specification.

Use only the time measurements, not including additional predictor variables. Think about:

- which metric of time to use (see the SPSS file for more information about the variables);
- the shape of the function (linear or quadratic); and base your decision of the best model on:

- model fit indices;
- model comparison tools;
- plots;
- interpretation of the model parameters.

Click to show answers

Deciding on the metric of time

From the SPSS file variable descriptions:

- SVL wave 1 (2 weeks after burn injury)
- SVL wave 2 (4 weeks)
- SVL wave 3 (2 months)
- SVL wave 4 (4 months)
- SVL wave 5 (6 months)
- SVL wave 6 (9 months)
- SVL wave 7 (12 months)
- SVL wave 8 (18 months)

Based on these descriptions, I've chosen for the following specification of time in the LGM:

`"i s | W1@0.5 W2@1 W3@2 W4@4 W5@6 W6@9 W7@12 W8@18;"`

In this specification I set the first time point to 0.5 months after burn injury (approximation of 2 weeks after burn injury), the second time point to 1 month after burn injury (approximation of 4 weeks after burn injury), etc.

Deciding on Linear v.s. Linear + Quadratic slope

Some example syntaxes for running models with different trajectory shapes are shown below, along with a table of the resulting fit statistics.

For the model with a quadratic slope, it was necessary to fix the variance of Q at 0 to ensure convergence (you can do this by adding q@0; to the syntax). For both models, only CFI and TLI indicate adequate fit.

```
# First, create a linear model from the basic one
linear <- basic
linear$MODEL <-
  "i s | W1@0.5 W2@1 W3@2 W4@4 W5@6 W6@9 W7@12 W8@18;"
result_linear <-
  mplusModeler(linear, modelout = "linear.inp", run = 1L)
# Then, a quadratic one
quad <- basic
quad$MODEL <-
  "i s q | W1@0.5 W2@1 W3@2 W4@4 W5@6 W6@9 W7@12 W8@18;"
result_quad <-
  mplusModeler(quad, modelout = "quad.inp", run = 1L)
# Then, a quadratic one with fixed variance
quad0 <- basic
quad0$MODEL <-
  "i s q | W1@0.5 W2@1 W3@2 W4@4 W5@6 W6@9 W7@12 W8@18;
   q@0;"
```

```
result_quad0 <-
  mplusModeler(quad0, modelout = "quad0.inp", run = 1L)
# Combine them both in a list:
results <- list(result_linear, result_quad, result_quad0)
# Compare the fit:
SummaryTable(results,
             keepCols = c("Filename", "Parameters",
                          "AIC", "BIC", "RMSEA_Estimate",
                          "CFI", "TLI", "SRMR"))
```

Filename	Parameters	AIC	BIC	RMSEA_Estimate	CFI	TLI	SRMR
linear.out	13	14051	14096	0.14	0.93	0.94	0.08
quad.out	NA	NA	NA	NA	NA	NA	NA
quad0.out	14	14037	14085	0.13	0.94	0.94	0.08

Model comparison tools: To see which model fit the data better, we can do a Chi-square difference test, using the function `chisq_sb()` from the `tidySEM` package. It works directly on MplusAutomation's `SummaryTables`:

```
library(tidySEM)
chisq_sb(SummaryTable(list(result_linear, result_quad0), keepCols = c("Filename", "Chi
```

Filename	ChiSqM_Value	ChiSqM_DF	Dchisq	Dchisq_df	Dchisq_p
linear.out	173	31	NA	NA	NA
quad0.out	157	30	16	1	0

Note that the Furthermore, both AIC and BIC are lower in the model with a quadratic slope. Thus, model misfit is significantly lower when the quadratic slope is added.

Plots

To examine the plots, you must open Mplus and examine its native plots.

Model parameters

The added quadratic slope is significant ($Q = 0.02$, $p < .001$), indicating that, on average, the growth curve does follow a quadratic curve.

```
tab <- table_results(result_quad0, columns = NULL)
tab[tab$paramheader == "Means", c("param", "est_sig", "pval")]
```

param	est_sig	pval
I	18.73***	0.00
Q	0.02***	0.00
S	-0.65***	0.00

2.1.2 Exercise 1b: Add covariates

Using the best fitting LGM model found above, regress the growth parameters on TVLO and regress Pain on the growth parameters (see examples from slides 195 and 199). Are there gender differences in the regression of the growth parameters on TVLO and in the regression of Pain on the growth parameters?

Click to show answers

To answer this question, we use multi-group analysis. To do this with MplusAutomation, we add the following syntax to the VARIABLE argument:

`VARIABLE = "GROUPING IS gender (1 = male 2 = female);"`

Since we needed to fix the quadratic slope variance to 0, we cannot estimate any regressions on the quadratic slope or use the quadratic slope as a predictor of some outcome. We therefore focus on the intercept and linear slope.

Here is how we can specify the full syntax (starting with the quad0 model as base):

```
# Assuming quad0 was the best, start with that
covs <- quad0
# Add to the existing model
covs$MODEL <- c(covs$MODEL,
                "
                i s on TVLO;
                pain on i s;
                MODEL male:
                i on TVLO (m1);
                s on TVLO (m2);
                pain on i (m3);
                pain on s (m4);
                MODEL female:
                i on TVLO (f1);
                s on TVLO (f2);
                pain on i (f3);
                pain on s (f4);")
# Add new usevariables:
covs$usevariables <- c(covs$usevariables,
                       "tvlo", "gender", "pain")
covs$rdata
# Add grouping variable:
covs$VARIABLE <- "GROUPING IS gender (1 = male 2 = female);"
# Add parameter tests
covs$MODELTEST <- "m3 = f3; m4 = f4;"

result_covs <- mplusModeler(covs,
```

```
                            modelout = "covs.inp",
                            run = 1L)
# Obtain the model summaries; you need the Wald test for
# the parameter difference tests.
get_summaries(result_covs)
```

```
##   Mplus.version       Title AnalysisType   DataType Estimator Observations
## 1           8.6  exercise 1      GENERAL INDIVIDUAL        ML          240
##   NGroups NDependentVars NIndependentVars NContinuousLatentVars Parameters
## 1       2              9                1                     3         38
##   ChiSqM_Value ChiSqM_DF ChiSqM_PValue ChiSqBaseline_Value ChiSqBaseline_DF
## 1          285        88             0                2316               90
##   ChiSqBaseline_PValue    LL UnrestrictedLL  CFI  TLI   AIC   BIC  aBIC
## 1                    0 -7959          -7817 0.91 0.91 15994 16127 16006
##   RMSEA_Estimate RMSEA_90CI_LB RMSEA_90CI_UB RMSEA_pLT05 WaldChiSq_Value
## 1           0.14          0.12          0.15           0             0.5
##   WaldChiSq_DF WaldChiSq_PValue  SRMR  AICC Filename
## 1            2             0.78 0.089 16009 covs.out
```

This is only an illustrative example for how to approach this analysis; your specific execution may differ.

Note that in this example, the Wald χ^2 p-value is not significant. That means that there are no significant sex differences in the effect of the growth trajectory on pain.

Note that the Wald test is an overall test of **all** comparisons that we specify in `MODELTEST`.

Thus, if you want a separate test for the regression of TVLO on the growth parameters, you need to re-run the analysis but with a different `MODELTEST` argument.

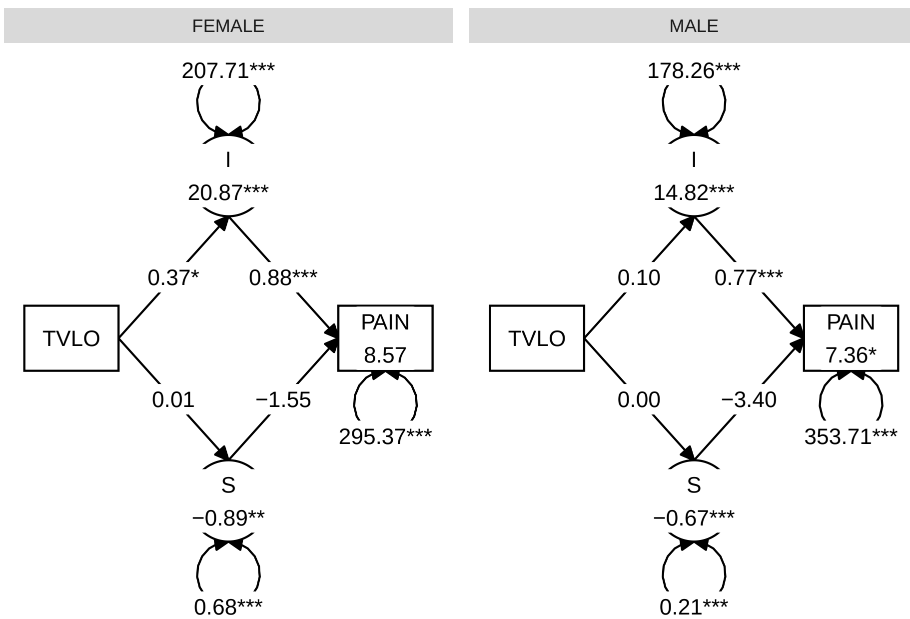
What is your conclusion about the other research question (regarding TVLO)?

Note that there are also other ways to test these hypotheses, aside from a Wald test. Model comparisons would also be a feasible way; either using AIC/BIC, or using a Chi square / Likelihood ratio test to compare models with parameters free vs constrained. For this, you could use the function `chisq_sb()` as before.

2.1.2.1 Visualization

If you want to plot the model for these two groups, you can use the SEM graphing package `tidySEM`. This flexible package produces fully customizable plots based on the R graphing package `ggplot2` for Mplus (and `lavaan`) models. If you want to make publication quality graphs, here is an online tutorial for graph customization. The script below demonstrates how to plot a model using `tidySEM`. Assuming that we only want to visualize the regression part of the model, you could specify:

```
library(tidySEM)
library(dplyr)
lo <- get_layout("",     "I",   "",
                 "TVLO", "",    "PAIN",
                 "",     "S",   "", rows = 3)
graph_sem(result_covs, layout = lo)
```



2.1.2.2 Tabulating results

In addition to graphing, it is also possible to tabulate the results using `tidySEM`. Here is a brief example:

```
# Get all columns of results
tab <- table_results(result_covs, columns = NULL)
# Retain regression parameters
tab <- tab[tab$op == "~", ]
# Remove group name from the label
tab$label <- gsub("\\.(FE)?MALE", "", tab$label)
# Make a wide table with both groups next to each other
tab <- reshape(tab,
```

```
                timevar = "group",
                idvar = "label",
                direction = "wide")
# Retain only the standardized estimate, pvalue, and 95% CI
tab[, c(1, grep("(est_sig|pval|confint)_std", names(tab)))]
```

label	pval_std.FEMALE	est_sig_std.FEMALE	confint_std.FEMALE	pval_std.MAL
I.ON.TVLO	0.01	0.31*	[0.06, 0.56]	0.20
PAIN.ON.I	0.00	0.61***	[0.43, 0.79]	0.00
PAIN.ON.S	0.63	-0.06	[-0.31, 0.19]	0.35
S.ON.TVLO	0.14	0.22	[-0.07, 0.50]	0.47

2.2 Exercise 2: Alcohol use

The figure below depicts the basic Latent Growth model for the alcohol use data from Duncan, Duncan & Strycker example 8_1.

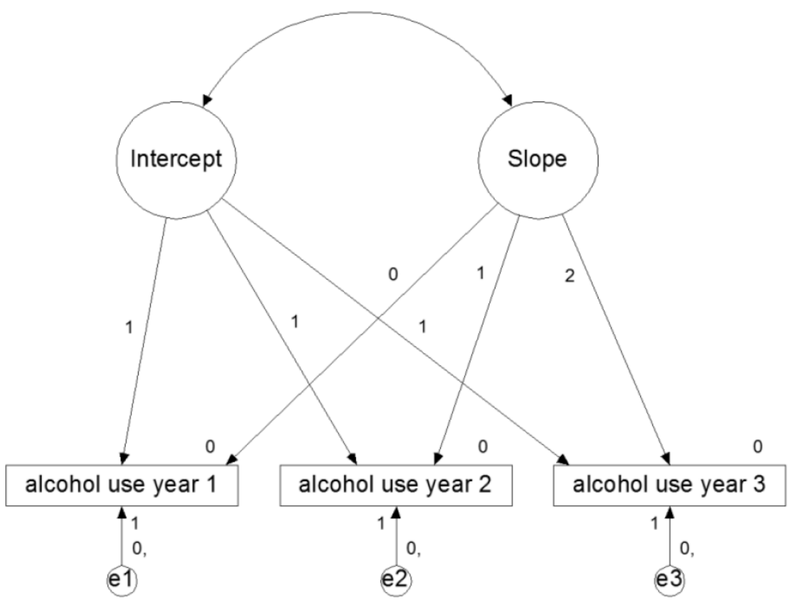


Figure 2.1: Latent Growth model for alcohol

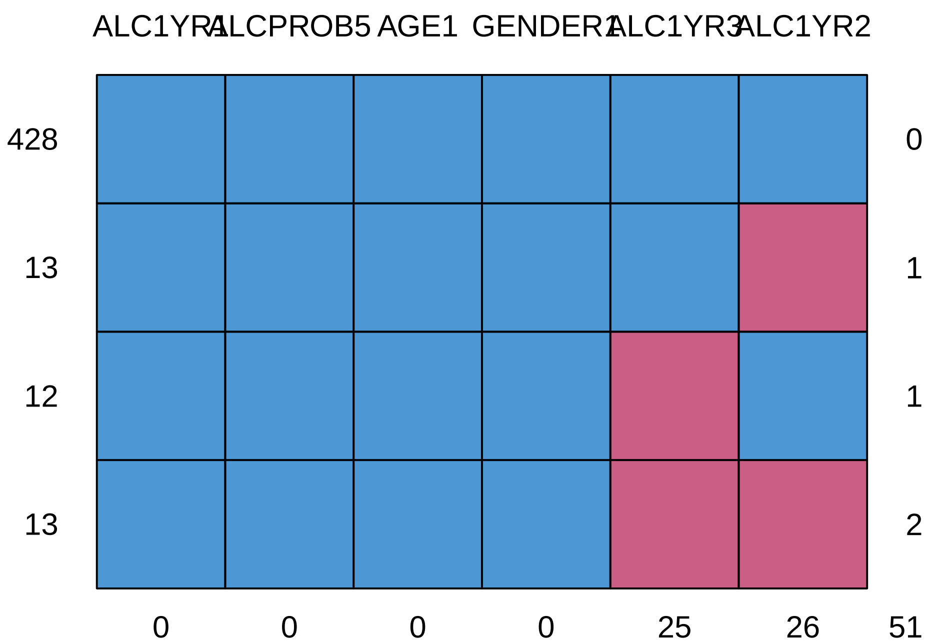
The data are in the file `DDS8_1.dat`, with variables ALC1YR1 ALC1YR2 ALC1YR3 ALCPROB5 AGE1 and GENDER1. Missing values are coded as -99. The variable ALCPROB5 is categorical, it indicates alcohol problems in year 5 of the study (0=no, 1=yes).

First, load the file `DDS8_1.dat` into the R environment. For convenience's sake, rename the columns of the data object to something a human would understand:

```
data <- read.table("DDS8_1.dat", na.strings = -99)
names(data) <- c("ALC1YR1", "ALC1YR2", "ALC1YR3",
                 "ALCPROB5", "AGE1", "GENDER1")
```

Now, examine the patterns of missing data. For this, you could use Mplus, or in R, you can use the `mice` package:

```
install.packages("mice")
library(mice)
md.pattern(data)
```



ALC1YR1 ALCPROB5 AGE1 GENDER1 ALC1YR3 ALC1YR2
428 1 1 1 1 1 1 0 13 1 1 1 1 1 0 1 12 1 1 1 1 0 1 1 13 1 1 1 1 0 0 2 0 0 0 0 25 26 51

The missing data pattern shows that the majority of the cases is complete, there is a small amount of attrition over time (panel dropout).

2.2.1 Exercise 2a

Set up the growth curve model as depicted in the Figure in Mplus. As a starting point, use the `MplusAutomation` code below.

- Add the necessary syntax statements to finalize the syntax.
- Request sample statistics and standardized (STDYX) output.
- Inspect the output carefully with special attention for

    1. how well the model fits
    2. interpretation of the output; how well does the model predict alcohol use over the years?

```
m0 <- mplusObject(
  TITLE = "LGA MODEL",
  MODEL = "",
  OUTPUT = "",
  rdata = data,
  usevariables = c("ALC1YR1", "ALC1YR2", "ALC1YR3"))
```

Click to show answers

Here is an example of how to approach the problem:

```
m0 <- mplusObject(
  TITLE = "LGA MODEL",
  MODEL = "i s | ALC1YR1@0 ALC1YR2@1 ALC1YR3@2;",
  OUTPUT = "SAMPSTAT standardized;",
  PLOT = "SERIES = ALC1YR1 ALC1YR2 ALC1YR3 (s);
          TYPE = PLOT3;",
  rdata = data,
  usevariables = c("ALC1YR1", "ALC1YR2", "ALC1YR3"),
  modelout = "m0.inp",
  run = 1L)

# Missing data patterns:
get_data_summary(m0)
# Fit
SummaryTable(m0, keepCols = c("Filename", "Parameters", "ChiSqM_Value", "ChiSqM_DF",
"ChiSqM_PValue", "LL", "CFI", "TLI", "RMSEA_Estimate", "SRMR"))
# Parameters
tab <- tidySEM::table_results(m0, columns = NULL)
tab[tab$paramheader == "Means", c("label", "est_sig", "pval")]
```

The model fit should be very good, with non-significant chi-square, and good fit according to CFL/TLI. The intercept and slope means indicate a relatively high starting point (3.68) and a growth of 0.92 per year. The Intercept and Slope show considerable variance, indicating that the starting points and rates of growth differ considerably across individuals.

2.2.2 Exercise 2b

We will now *explore* how different predictor variables affect the model fit. Include gender and age in the model as predictors of the intercept and slope. Interpret the fit of the model and the output. Feel free to estimate several models, including or excluding certain covariates. Either make a model fit table by hand in a spreadsheet, or use `SummaryTable()` to request the fit indices you deem to be appropriate. Which model do you consider to be best?

2.2.2.1 Exploratory vs confirmatory research

Note that when you conduct *confirmatory* research, and are testing theoretical hypotheses, you should not add and omit paths based on exploratory analyses and model fit.

It is fine to add and remove paths in *exploratory* research. Model fit indices, like AIC and BIC, are suitable for selecting well-fitting models in exploratory research. P-values are not designed for variable selection, and using them for that purpose may lead to suboptimal models.

It is good scientific practice to clearly separate confirmatory and exploratory research. When you conduct exploratory research, you should not perform inference on the resulting parameters based on p-values (because inference generalizes your findings to the population, and exploratory findings tend to be tailored toward this specific sample). You should also not present exploratory results as if they were testing a post-hoc theory ("Hypothesizing After the Results are Known," or HARKing, is a questionable research practice and can lead to false-positive (spurious) findings.

Click to show answers

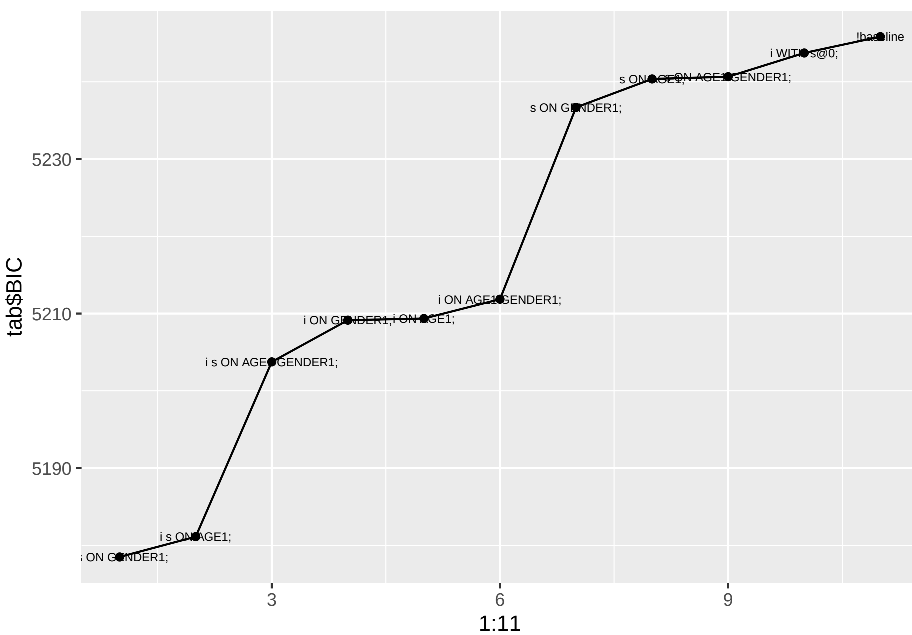
The answers to Exercise 1a demonstrate how to approach this. Estimate multiple slightly different models, put them in a list, and run `SummaryTable()`. Then use the AIC and BIC to identify the best-fitting model, and assess how different the model fits are. Also consider using RMSEA, CFI, TLI, and SRMR to make sure that your best-fitting model has acceptable objective fit.

```
# Create a vector with three "additional syntaxes"
# for my three different models
mod = c("i s ON GENDER1;",
        "i s ON AGE1;",
        "i s ON AGE1 GENDER1;",
        "i ON GENDER1;",
        "i ON AGE1;",
        "i ON AGE1 GENDER1;",
        "s ON GENDER1;",
        "s ON AGE1;",
```

```
        "s ON AGE1 GENDER1;",
        "i WITH s@0;",
        "!baseline")
# Create a list with the additional usevariables
# used in the three models above
vars = list("GENDER1",
            "AGE1",
            c("GENDER1", "AGE1"),
            "GENDER1",
            "AGE1",
            c("GENDER1", "AGE1"),
            "GENDER1",
            "AGE1",
            c("GENDER1", "AGE1"),
            NULL,
            NULL)
# Make a list of exploratory models by modifying m0
models <- lapply(1:length(mod), function(i){
  # append element i of mod
  m0$MODEL <- paste0(m0$MODEL, mod[i])
  # append element i of vars
  m0$usevariables <- c(m0$usevariables, vars[[i]])
  # Add unique filename
  m0$modelout = paste0("Model", i, ".inp")
  # return the modified model
  m0
})
# Run all models and store results in a list called results
results <- lapply(models, mplusModeler, run = 1L)
# Get summary table and store it in 'tab'
tab <- SummaryTable(results,
                    keepCols = c("Parameters", "AIC", "BIC",
                                 "RMSEA_Estimate", "CFI",
                                 "TLI", "SRMR"),
                    sortBy = NULL)
# Order by BIC
tab <- tab[order(tab$BIC), ]
# Add model syntax to the table
tab <- cbind(Model = mod, tab)
tab
```

Model	Parameters	AIC	BIC	RMSEA_Estimate	CFI	TLI	SRMR
i s ON GENDER1;	12	5129	5178	0.03	0.99	0.98	0.02
i s ON AGE1;	10	5140	5181	0.03	1.00	0.98	0.02
i s ON AGE1 GENDER1;	9	5166	5204	0.12	0.81	0.71	0.07
i ON GENDER1;	8	5176	5209	0.06	0.99	0.97	0.02
i ON AGE1;	10	5168	5209	0.05	0.99	0.96	0.02
i ON AGE1 GENDER1;	8	5179	5212	0.15	0.80	0.69	0.08
s ON GENDER1;	9	5199	5237	0.16	0.66	0.48	0.10
s ON AGE1;	8	5207	5240	0.15	0.76	0.63	0.08
s ON AGE1 GENDER1;	8	5208	5241	0.20	0.66	0.48	0.11
i WITH s@0;	8	5211	5244	0.16	0.74	0.60	0.08
!baseline	7	5217	5246	0.22	0.74	0.61	0.09

```r
# Plot the BICs and annotate with the syntax to see which is best
library(ggplot2)
qplot(x = 1:11, y = tab$BIC) +
  geom_line() +
  geom_text(label = tab$Model, size = 2)
```



It looks like, paradoxically, predicting I and S from either age or gender has the best fit. However, note that there is only a small difference between the smallest and the largest BIC: 67.36. I would either go for the simplest model (i WITH s@0), or go for the best-fitting model (i s ON GENDER1).

Note that the covariance between intercept and slope disappears from the model when you add predictors, as this turns it into a covariance between a latent variable and a residual. Mplus automatically constrains these to zero. If we add the statement I WITH S to the model, we obtain a good fit with significant effects of both gender and age on the intercept. NOTE: This illustrates the importance of checking the output carefully to find out if Mplus is actually doing what you think it does!

2.2.3 Exercise 2c

Include alcohol problems in year 5 in the model: let the intercept and slope factors predict alcohol problems year 5. Declare the variable as categorical in the variable section (CATEGORICAL = ALCPROB5). Inspect if the effect of age and gender on alcohol problems year 5 is completely mediated by the growth factors, or if there are additional direct paths from age and gender on the alcohol problems.

Click to show answers

The model fit is still good. Note that after adding a categorical dependent variable to the model, Mplus switches to a robust estimator (MLR). Both intercept and slope predict alcohol problems. Age also predicts alcohol problems directly. Since age predicts alcohol problems both directly and via the intercept, a mediation analysis is in order. This shows that the indirect effect of age via the intercept on alcohol problems is still significant when the direct effect is added to the model.

To test whether there is full mediation or not, we may want to test whether the direct effects are equal to zero or not.

If the analysis had not included a categorical dependent variable, then we would have been able to compute the difference test using the `tidySEM` function `chisq_sb(SummaryTable(keepCols = NULL))`.

However, in the presence of a categorical dependent variable, we must use Mplus' option `difftest`.

Again, we can start from m0:

```
# Specify model with only indirect effects
m2c_indirect <- m0
m2c_indirect$usevariables <- c(
  m2c_indirect$usevariables,
  "AGE1", "GENDER1", "ALCPROB5")
m2c_indirect$VARIABLE <- "CATEGORICAL = ALCPROB5;"
m2c_indirect$MODEL <- paste0(
  m0$MODEL,
  "i on AGE1 GENDER1;
  i WITH s;
```

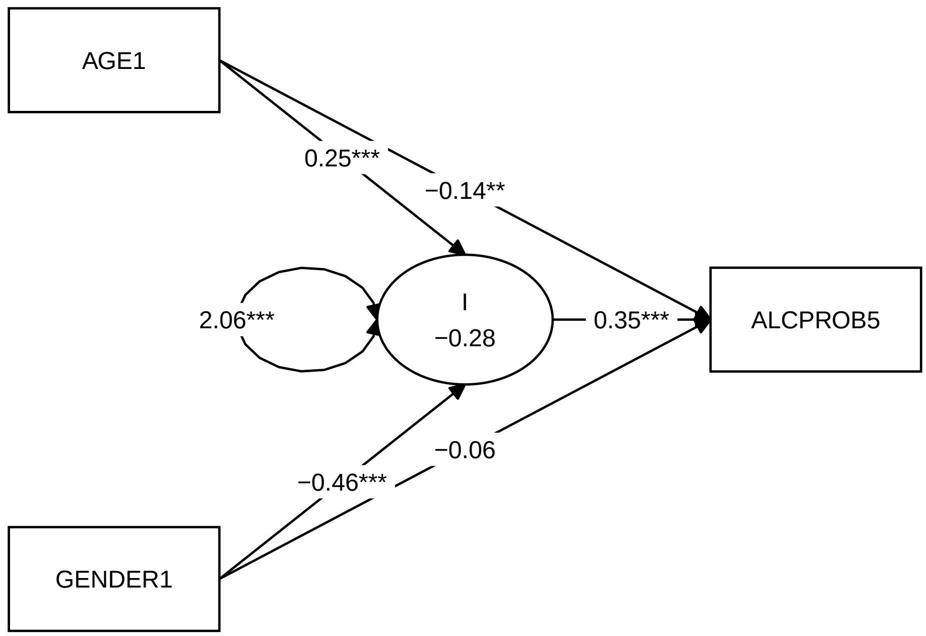
```
  ALCPROB5 on i s;
  ALCPROB5 on AGE1 GENDER1;")
m2c_indirect$SAVEDATA <- "difftest is mediation.dat;"
m2c_indirect$modelout <- "m2c_indirect.inp"
# Run all models and store results in a list called results
result_ind <- mplusModeler(m2c_indirect, run = 1L)

# Specify model with direct effects too
m2c_direct <- m2c_indirect
# Constrain direct effects to 0 using gsub (replace)
m2c_direct$MODEL <- gsub("ALCPROB5 on AGE1 GENDER1;",
                         "ALCPROB5 on AGE1 GENDER1@0;",
                         m2c_direct$MODEL,
                         fixed = TRUE)
m2c_direct$ANALYSIS <- "difftest = mediation.dat;"
m2c_direct$modelout <- "m2c_direct.inp"
# Run the direct model, which includes the difference test
result_dir <- mplusModeler(m2c_direct, run = 1L)
# Look at the model summaries
get_summaries(result_dir)
```

```
##   Mplus.version      Title AnalysisType   DataType Estimator Observations
## 1           8.6  LGA MODEL      GENERAL INDIVIDUAL     WLSMV          466
##   NGroups NDependentVars NIndependentVars NContinuousLatentVars Parameters
## 1       1              4                2                     2         14
##   ChiSqM_Value ChiSqM_DF ChiSqM_PValue ChiSqBaseline_Value ChiSqBaseline_DF
## 1          5.1         7          0.64                 215               14
##   ChiSqBaseline_PValue ChiSqDiffTest_Value ChiSqDiffTest_DF
## 1                    0                0.21                1
##   ChiSqDiffTest_PValue CFI TLI RMSEA_Estimate RMSEA_90CI_LB RMSEA_90CI_UB
## 1                 0.64   1   1              0             0         0.047
##   RMSEA_pLT05  SRMR       Filename
## 1        0.96 0.057 m2c_direct.out
```

Note that the ChiSqDiffTest_PValue is non-significant, which means we can prefer the simpler (no direct effects) model. There is full mediation.

Let's use a graph to examine the unconstrained model too:

```
lo <- get_layout("AGE1",    "", "",
                 "",        "I",  "ALCPROB5",
                 "GENDER1", "", "", rows = 3)
graph_sem(result_ind, layout = lo)
```



2.3 Exercise 3: Level and Shape Parameterization

The file GPA.dat holds the GPA data (GPA = grade point average) with GPA scores of 200 students in 6 consecutive semesters. There are also time-invariant covariates: high school GPA and gender; and the outcome variable: admitted to university of choice (missing if not applied for university). Use the GPA data to set up a level and shape model.

First, load the data and name the variables:

```
data <- read.table("GPA.dat")
names(data) <- c("STUDENT", "SEX", "HIGHGPA",
                 "GPA1", "GPA2", "GPA3", "GPA4", "GPA5", "GPA6")
```

2.3.1 Exercise 3a

Use a parameterization with GPA1@0 and GPA6@1. The loadings for the other timepoints should be freely estimated. This can be done with, for example, the syntax GPA2* as shown in the handout.
Interpret the factor loadings and estimate for S.

Click to show answers

This can be done as follows:

```
m3 <- mplusObject(
  MODEL = "i s | gpa1@0 gpa2* gpa3* gpa4* gpa5* gpa6@1;",
  rdata = data,
  usevariables = c("GPA1", "GPA2", "GPA3", "GPA4", "GPA5", "GPA6"),
  modelout = "m3.inp"
)

res <- mplusModeler(m3, run = 1L)

# Get all columns of results
tab <- table_results(res, columns = NULL)
# Retain factor loadings and intercepts using op %in% c("=~", "~1")
# for all parameters involving S (lhs == "S")
tab <- tab[tab$lhs == "S" & tab$op %in% c("=~", "~1"), ]
tab[, c("label", "est_sig", "pval", "confint")]
```

The factor loading indicates the proportion of change from the starting time point to the current one. Thus, 24% of the total change occurs between GPA1 and GPA2. Mean S = 0.55. This indicates the total change between GPA1 and GPA6. The intercept at GPA1 = 2.575. So the estimated score at GPA2 = 2.575 + 0.239*0.549 = 2.706, the estimated score at GPA3 = 2.575 + 0.450*0.549 = 2.822, etcetera.

label	est_sig	pval	confint
S.|.GPA1	0.00	NA	NA
S.|.GPA2	0.24***	0.00	[0.16, 0.32]
S.|.GPA3	0.45***	0.00	[0.38, 0.52]
S.|.GPA4	0.65***	0.00	[0.60, 0.70]
S.|.GPA5	0.81***	0.00	[0.77, 0.86]
S.|.GPA6	1.00	NA	NA
Means.S	0.55***	0.00	[0.49, 0.61]

2.3.2 Exercise 3b

Now use a parameterization with GPA1@0 and GPA2@1. The other GPA's should be freely estimated. Interpret the factor loadings and estimate for S.

Click to show answers

This can be done as follows:

```
m3b <- m3
m3b$MODEL <- "i s | gpa1@0 gpa2@1 gpa3* gpa4* gpa5* gpa6*;"

res_3b <- mplusModeler(m3b, run = 1L)

# Get all columns of results
tab <- table_results(res_3b, columns = NULL)
# Retain factor loadings and intercepts using op %in% c("=~", "~1")
# for all parameters involving S (lhs == "S")
tab <- tab[tab$lhs == "S" & tab$op %in% c("=~", "~1"), ]
tab[, c("label", "est_sig", "pval", "confint")]
```

Mean S now indicates the difference between GPA1 and GPA2. The estimated factor loadings indicate the distance in units from the starting point, where 1 unit is S. You could also say that every distance compares to the increase between GPA1 and GPA2.

label	est_sig	pval	confint
S.|.GPA1	0.00	NA	NA
S.|.GPA2	1.00	NA	NA
S.|.GPA3	1.88***	0.00	[1.31, 2.46]
S.|.GPA4	2.73***	0.00	[1.89, 3.58]
S.|.GPA5	3.41***	0.00	[2.33, 4.49]
S.|.GPA6	4.18***	0.00	[2.82, 5.55]
Means.S	0.13***	0.00	[0.08, 0.18]

Which parameterization do you like best?

2.3.3 Exercise 3c

Draw the development of GPA over time based on your own calculations (by hand). Compare this to the estimated means plot that you can get with the plot command: PLOT: SERIES = GPA1-GPA6 (s); TYPE = PLOT3;
However, don't forget that you need to 'rescale' that plot, since S is linear while the location of the estimated points is based on the factor loadings.

Click to show answers

This can be done as follows. Note that you will have to load the plot in Mplus:

```
m3b$PLOT <- "SERIES = GPA1-GPA6 (s);
             TYPE = PLOT3;"

res_3b <- mplusModeler(m3b, run = 1L)
```

2.3.4 Exercise 3d

Use sex as a predictor of the intercept and slope and interpret the result (with 0 = boys, 1 = girls).

Click to show answers

This can be done as follows:

```
m3_sex <- m3
# Make 0 = boys and 1 = girls
m3_sex$rdata$SEX <- m3_sex$rdata$SEX - 1
m3_sex$usevariables <- c(m3_sex$usevariables, "SEX")
m3_sex$MODEL <- paste0(m3_sex$MODEL, "i s on sex;")
res_3sex <- mplusModeler(m3_sex, run = 1L)
# Get all columns of results
tab <- table_results(res_3sex, columns = NULL)
# Retain only regressions on I and S
tab <- tab[tab$lhs %in% c("I", "S") & tab$op == "~", ]
tab[, c("label", "est_sig", "pval", "confint")]
```

label	est_sig	pval	confint
I.ON.SEX	0.08*	0.03	[0.01, 0.15]
S.ON.SEX	0.14*	0.01	[0.03, 0.24]

Sex is a significant predictor of the intercept and a significant predictor of development, with girls having a higher initial level, and a greater development over time.

2.4 Exercise 4: latent growth model on GPA data

2.4.1 Exercise 4a

Continuing with the data used for the previous exercise, set up a latent growth model for GPA for the 6 consecutive occasions and run this model. Obtain the following parameters:

- AIC/BIC, Chi Square, RMSEA, CFI/TLI
- Mean Intercept and Slope
- Variance of the Intercept and Slope

Click to show answers

This can be done as follows, using m3 as starting point:

```
m4 <- m3
m4$MODEL <- "i s | gpa1@0 gpa2@1 gpa3@2 gpa4@3 gpa5@4 gpa6@5;"
m4$modelout <- "m4.inp"
res_4 <- mplusModeler(m4, run = 1L)
# Get the requested fit indices:
get_summaries(res_4)[c("AIC", "BIC", "ChiSqM_Value",
                       "RMSEA_Estimate", "CFI", "TLI")]
# Get all columns of results
tab <- table_results(res_4, columns = NULL)
# Retain only regressions on I and S
tab <- tab[tab$lhs %in% c("I", "S") & tab$op %in% c("~~", "~1"), ]
tab[, c("label", "est_sig", "pval", "confint")]
```

AIC BIC ChiSqM_Value RMSEA_Estimate CFI TLI 1 121 157 44 0.093 0.96 0.97

label	est_sig	pval	confint
S.WITH.I	0.00	0.12	[-0.00, 0.01]
Means.I	2.60***	0.00	[2.56, 2.63]
Means.S	0.11***	0.00	[0.10, 0.12]
Variances.I	0.04***	0.00	[0.02, 0.05]
Variances.S	0.00***	0.00	[0.00, 0.00]

2.4.2 Exercise 4b

Then, set up a latent growth model for 3 years where each year is a latent variable measured by the GPA of two consecutive semesters.

The factor loadings for GPA2, GPA4 and GPA6 ought to be constrained to be equal with a label (a) behind the loading in the syntax. As such, the scores relate in the same way to the year score over time. The GPA intercepts are constrained at 0.

If you get the error message below, can you find out what the problem is?

```
WARNING:  THE LATENT VARIABLE COVARIANCE MATRIX (PSI) IS NOT POSITIVE     DEFINITE. THI
COULD INDICATE A NEGATIVE VARIANCE/RESIDUAL VARIANCE FOR A LATENT VARIABLE, A CORRELATI
GREATER OR EQUAL TO ONE BETWEEN TWO LATENT VARIABLES, OR A LINEAR DEPENDENCY AMONG MORE
THAN TWO LATENT VARIABLES. CHECK THE TECH4 OUTPUT FOR MORE INFORMATION.
```

A rough way to deal with this problem may be to fix the problematic parameter to a particular value (ie. .001), try this and re-run the model.

Now examine the same parameters as for exercise 4a, and compare the two. Are there major differences?

- AIC/BIC, Chi Square, RMSEA, CFI/TLI
- Mean Intercept and Slope
- Variance of the Intercept and Slope

Click to show answers

This can be done as follows, using m4 as starting point:

```
m4b <- m4
m4b$MODEL <- "year1 by gpa1@1 gpa2 (a);
             year2 by gpa3@1 gpa4 (a);
             year3 by gpa5@1 gpa6 (a);
             [gpa1@0 gpa2@0 gpa3@0 gpa4@0 gpa5@0 gpa6@0];
             i s | year1@0 year2@1 year3@2;
             [i s];
             year3@.001;"
res_4b <- mplusModeler(m4b, run = 1L)
# Get the requested fit indices:
get_summaries(res_4b)[c("AIC", "BIC", "ChiSqM_Value",
                        "RMSEA_Estimate", "CFI", "TLI")]
# Get all columns of results
tab <- table_results(res_4b, columns = NULL)
# Retain only regressions on I and S
tab <- tab[tab$lhs %in% c("I", "S") & tab$op %in% c("~~", "~1"), ]
tab[, c("label", "est_sig", "pval", "confint")]
```

Note that, without `year3@.001`, this code gives an error message: The variance of the latent variable year3 is estimated negatively which is problematic since variances should always be positive. A simple way to deal with the problem of the latent variance of year3 is to fix it to a very small value (.001) for instance, as it would also be illogical to fix a variance to 0. To do this, simply add this to your input file under model: year3@.001;

AIC BIC ChiSqM_Value RMSEA_Estimate CFI TLI 1 130 177 48 0.12 0.96 0.95

label	est_sig	pval	confint
S.WITH.I	0.01***	0.00	[0.01, 0.02]
Means.I	2.60***	0.00	[2.56, 2.64]
Means.S	0.21***	0.00	[0.19, 0.23]
Variances.I	0.03***	0.00	[0.02, 0.04]
Variances.S	0.01**	0.00	[0.00, 0.01]

Compare the estimates from the two models. Are there major differences? If you inspect the output carefully (and provided you have requested standardized estimates) you will notice that the latent variables year2 and year3 have a correlation of 1. So the negative variance is the result of a multicollinearity problem. It is apparently better to analyze these data using only the observed variables gpa1-gpa6. Creating latent variables per year does not work well. In line with this interpretation, the fit and results of the simple latent growth model look better than the 2nd order latent growth curve model.

Chapter 3

Day 3: Latent Growth (Mixture) Modeling

This computer lab session demonstrates to run latent growth mixture models in batch, using the R-package `MplusAutomation`. Note that, if you do not want to automate part of your workflow (like making plots and tables), you can also use Mplus exclusively. All of the input files for the exercises described in this GitBook are provided with the course materials.

To get started with today's computer lab, first open the project file called `"S23_student.Rproj"`. It should load in the program "RStudio." The bottom right panel has several tabs, including one called "Files." Click on "Files" in this bottom right tab, and click the file "mixture_exercises.R."

3.1 Exercise 1: Latent Growth (Mixture) Modeling

3.1.1 1a. Latent Class Growth Models.

Set up the growth curve model as depicted in Figure 1 in Mplus, using the | notation. Use the file DDS8_1.dat for this.

First, load the file `DDS8_1.dat` into the R environment. For convenience's sake, rename the columns of the data object to something a human would understand:

```
data <- read.table("DDS8_1.dat", na.strings = -99)
names(data) <- c("ALC1YR1", "ALC1YR2", "ALC1YR3",
                 "ALCPROB5", "AGE1", "GENDER1")
```

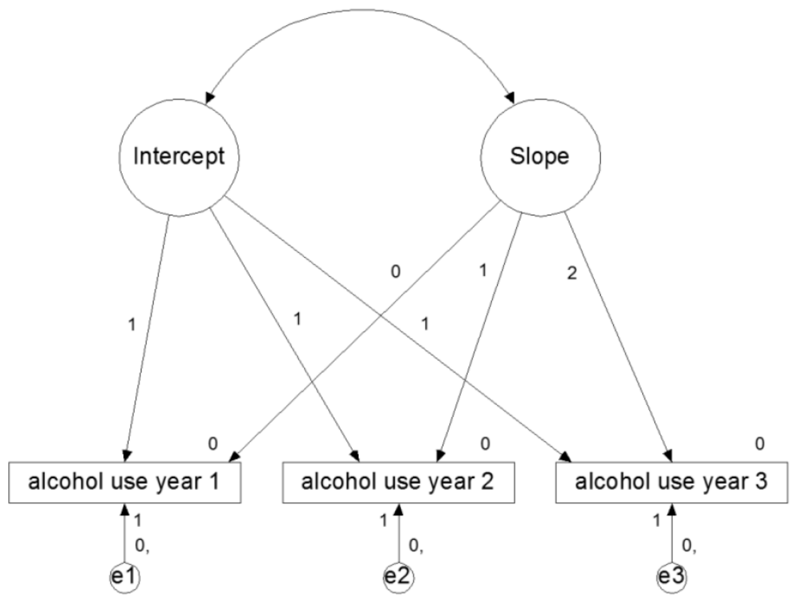


Figure 3.1: Latent Growth model for alcohol

Use the function `createMixtures` to define the latent class growth model as displayed in Figure 1. To see how the function `createMixtures()` works, type its name, select it, and press F1.

You can specify the model using the "|" notation. Constrain the variance of the intercept and slope factors to be equal to zero, using the Mplus syntax `i@0; s@0;`.

In createMixtures, you can specify the overall model using the argument `model_overall = "i s | ALC1YR1@0 ALC1YR2@1 ALC1YR3@2;  i@0; s@0;"`. The double spaces are converted to newline characters, which results in a nicely formatted Mplus file.

Request 1 to 6 classes, by specifying the argument `classes = 1:6` for createMixtures. Request tech8, tech11 and tech14 output by specifying the argument: `OUTPUT = "tech8 tech11 tech14;"`.

To run the analysis, add the argument `run = 1L` to the `createMixtures()` call. Make sure to store the resulting output in an object, the same way you stored the data in an object called `data` when using the function `read.table()`.

Click to show answers

The resulting syntax should look like this:

```
results_1a <- createMixtures(
  classes = 1:6,
  filename_stem = "1a",
  model_overall = "i s | ALC1YR1@0 ALC1YR2@1 ALC1YR3@2;
                   i@0;  s@0;",
  rdata = data,
  usevariables = c("ALC1YR1", "ALC1YR2", "ALC1YR3"),
  OUTPUT = "tech8 tech11 tech14;",
  run = 1L)
```

The function `createModels()` can run all of the analyses in batch, thus taking a lot of work out of your hands. The function essentially performs three steps:

1. Create Mplus syntax for each of the latent class models based on `model_overall` and `model_class_specific`
2. Use the function `mplusObject()` to turn this syntax into a model that can be evaluated
3. Use the function `mplusModeler()` to create the Mplus `.inp` files, run them, and return the results

If you use MplusAutomation for other types of models, you won't need `createMixtures()`. Instead, you can use `mplusObject()` and `mplusModeler()`.

To see what the function `createMixtures()` is doing, you should inspect each of the automatically generated input files (`.inp`). You can even run one by hand.

Finally, print a summary table to the R console by calling `mixtureSummaryTable()` on the results object.

Click to show answers

The resulting syntax should look like this:

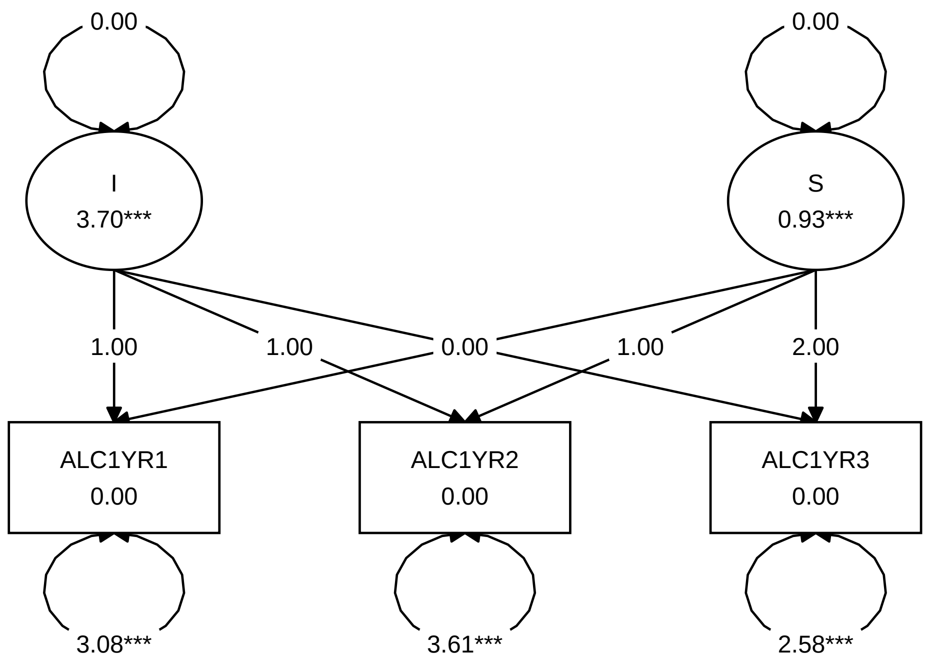
```
mixtureSummaryTable(results_1a)
```

Title	Classes	AIC	BIC	aBIC	Entropy	T11_VLMR_PValue	T11_LMR_PValue	BLRT_PVal
1 classes	1	5339	5360	5344	NA	NA	NA	N
2 classes	2	5191	5224	5199	0.73	0.00	0.00	
3 classes	3	4995	5040	5005	0.82	0.00	0.00	
4 classes	4	4971	5029	4984	0.80	0.01	0.01	
5 classes	5	4962	5032	4978	0.75	0.11	0.12	
6 classes	6	4938	5020	4957	0.76	0.00	0.00	

3.1.1.1 Visualization

To verify that the estimated model corresponds to the Figure above, you can use the SEM graphing package `tidySEM`. This flexible package produces fully customizable plots based on the R graphing package `ggplot2` for Mplus (and `lavaan`) models. If you want to make publication quality graphs, here is an online tutorial for graph customization. The script below demonstrates how to plot a model using `tidySEM`.

```
install.packages("tidySEM")
library(tidySEM)
lo <- get_layout("I",       "",        "S",
                 "ALC1YR1", "ALC1YR2", "ALC1YR3", rows = 2)
graph_sem(results_1a[[1]], layout = lo, angle = 179)
```



3.1.2 1b. Increasing random starts

These models use random starting values. Several independent random starts are made, to ensure that the model converges on the proper solution. The default is 20 random sets of starting values, of which 4 are run to completion. Inspect the output, and look carefully if the model estimation has converged, especially for the larger number of classes. Look for warning and error messages, make sure you understand what they are telling you.

The STARTS option is used to specify the number of initial random starting values and final stage optimizations. Now, increase the number of starts to ensure proper convergence. For createMixtures, the argument is `ANALYSIS = "STARTS = .. ..;"`.

Click to show answers

The resulting syntax should look like this:

```
results_1b <-
  createMixtures(classes = 1:6,
                 filename_stem = "1b",
                 model_overall = "i s | ALC1YR1@0 ALC1YR2@1 ALC1YR3@2;
                                  i@0;  s@0;",
                 ANALYSIS = "STARTS = 50 10;",
                 rdata = data,
                 usevariables = c("ALC1YR1", "ALC1YR2", "ALC1YR3"),
                 OUTPUT = "tech8 tech11 tech14;",
                 run = 1L)
```

Make a mixture summary table to compare the fit information of the models with 1-6 classes, with the increased number of starts, using `mixtureSummaryTable()`. Also open the output files, and inspect the estimates in each class. Which model do you prefer, and why?

Click to show answers

The resulting syntax should look like this:

```
mixtureSummaryTable(results_1b)
```

Title	Classes	AIC	BIC	aBIC	Entropy	T11_VLMR_PValue	T11_LMR_PValue	BLRT_PVal
1 classes	1	5339	5360	5344	NA	NA	NA	N
2 classes	2	5191	5224	5199	0.73	0.00	0.00	
3 classes	3	4995	5040	5005	0.82	0.00	0.00	
4 classes	4	4971	5029	4984	0.80	0.01	0.01	
5 classes	5	4962	5032	4978	0.75	0.11	0.12	
6 classes	6	4938	5020	4957	0.76	0.00	0.00	

Based on the table, I would select a 3-class model. The fit indices and (V)LMR tests essentially indicate that you can keep adding classes and improve the model, which makes it difficult to decide. However, if we look at min_N, we see that from 4 classes onward, the smallest class has less than 10% of cases assigned to it. The minimum posterior classification probability and entropy are best for the 3-class model, which means that this model can reasonably accurately assign individuals to classes.

3.1.3 1c. Latent Growth Mixture Models.

Set up the same models as analyzed in the previous exercise, but now allow the means and variances of the intercept and slope factors to be freely estimated in each class. You do this by mentioning the intercept and slope explicitly in the class-specific part of the syntax. This is a more complex model, and we might therefore expect that we will need fewer classes for a good description of the data. This analysis will also take more computing time, so add `PROCESSORS=4` to the analysis section. Make a table of the fit indices, look at AIC, BIC, and the Bootstrapped LRT value (Tech 14).

Click to show answers

The resulting syntax should look like this:

```
results_1c <-
  createMixtures(classes = 1:4,
                 filename_stem = "1c",
                 model_overall = "i s | ALC1YR1@0 ALC1YR2@1 ALC1YR3@2;
                                  i@0;  s@0;",
                 model_class_specific = "i;  s;",
                 rdata = data,
                 OUTPUT = "tech8 tech11 tech14;",
                 usevariables = c("ALC1YR1", "ALC1YR2", "ALC1YR3"),
                 ANALYSIS = "PROCESSORS = 4;",
                 run = 1L)
```

3.1.4 1d. Visualizing growth models.

Plotting the model-predicted trajectories makes it easier to interpret the model. Moreover, visualizing the raw data provides yet another way to evaluate the fit of your mixture model to the data. With this in mind, plot the four models you created in exercise *1c*, and interpret what you see. First, plot only the predicted trajectories. Then, plot raw data as well. Explain the benefit of plotting the raw data in your own words.

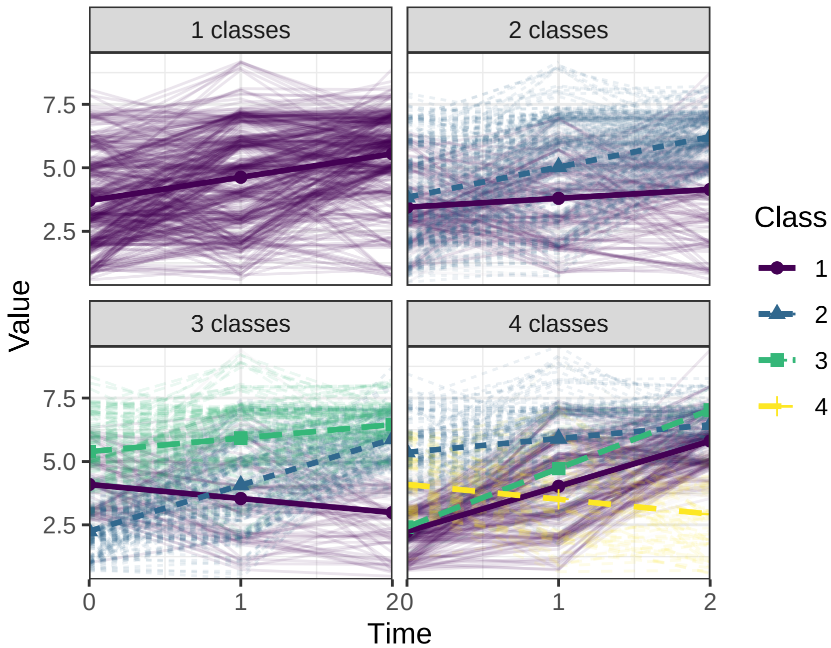
Tip: Because the scales in the alcohol data are ordinal, many of the observed trajectories overlap. To prevent 'overplotting,' you can jitter the observed trajectories by a fraction of their standard deviation. Even as little as jitter_lines = .1, jittering the positions on the y-axis by 1/10th of a standard deviation, can make a difference.

Click to show answers

Plotting the raw data helps us understand how representative the average trajectory for each class captures the individual trajectories of individuals in that class. It helps us see how separable the classes are visually, instead of just relying on statistics like entropy.

The resulting plot should look like this:

```
plotGrowthMixtures(results_1c, rawdata = TRUE, jitter_lines = .1)
```



3.1.5 1e. The 3-step model.

Covariates are often added to mixture models, to predict 1) class membership 2) to explain variation in the growth parameters within the classes or 3) as a distal outcome.

Whenever covariates are however added to the model, they change the Latent Class solution. Sometimes, this is fine, as the covariates can help to improve the classification. In other cases, you would use a 3-step approach, which Mplus has automated:

1. Fit an unconditional Latent Class Model (without covariates)
2. A "most likely class variable" is created using the posterior distribution of step 1.
3. This most likely class variable is then regressed on (a) covariate.

There are a few options for how to do 3-step analysis. They all rely on adding to the `Variable:` command. For more info, see https://www.statmodel.com/download/webnotes/webnote15.pdf.

3.1.5.1 Commands for conducting a 3-step model

You can add the following options to the `Variable:` command:

1. `Auxiliary = x(R);`
   This is actually a 1-step method for predicting latent class memberships using Pseudo-Class draws.
2. `Auxiliary = x(R3step);`
   A 3 step procedure, where covariates predict the latent class
3. `Auxiliary = y(e)`
   A 1-step method, where the latent class predicts a continuous distal outcome.
4. `Auxiliary = y(de3step);`
   A 3 step procedure, where latent class predicts continuous covariates (distal outcome) with unequal means and equal variances
5. `Auxiliary = y(du3step);`
   A 3 step procedure, where latent class predicts continuous covariates (distal outcome) with unequal means and variances
6. `Auxiliary = Y(dcon);`
   Procedure for continuous distal outcomes as suggested by Lanza et al (2013)
7. `Auxiliary = Y(dcon);`
   Procedure for categorical distal outcomes as suggested by Lanza et al (2013)
8. `Auxiliary = y(BCH);`
   Improved and currently best 3-step procedure with continuous covariates as distal outcomes

Pick your final model from 1c, and add both age and gender as auxiliary variables in the model. Try to think what 3-step model you want, and if you are not sure, run different models, so you can evaluate how the different procedures make a difference. You can do this by editing the Mplus file, or by adding the `VARIABLE = "Auxiliary = ...;",` to your call to `createMixtures()`. What is the effect of both age and gender?

Note: The results can be extracted using the `get_lcCondMeans()` *function.*

Click to show answers

I'm providing an example using the BCH 3-step procedure below. It can be seen, from the overall test and the pairwise comparisons, that the third group is significantly older, and has a significantly lower proportion of girls than the other two classes.

The resulting syntax and output should look like this:

```
results_1e <-
  createMixtures(classes = 3,
                 filename_stem = "1e",
                 model_overall = "i s | ALC1YR1@0 ALC1YR2@1 ALC1YR3@2;
                                  i@0;  s@0;",
                 model_class_specific = "i;  s;",
                 rdata = data,
                 OUTPUT = "tech8 tech11 tech14;",
                 usevariables = c("ALC1YR1", "ALC1YR2", "ALC1YR3",
                                  "AGE1", "GENDER1"),
                 VARIABLE = "Auxiliary = AGE1(BCH)  GENDER1(BCH);",
                 ANALYSIS = "PROCESSORS = 4;",
                 run = 1L)

# The results of the conditional means test are inside the output object
# But you have to dig a little bit. The code to get them is:
get_lcCondMeans(results_1e)
```

```
## $overall
##       var   m.1  se.1   m.2  se.2  m.3  se.3 chisq df     p
## 1    AGE1 14.81 0.227 15.31 0.114 15.9 0.107  25.7  2     0
## 2 GENDER1  0.48 0.063  0.46 0.037  0.3 0.042 9.421  2 0.009
##
## $pairwise
##       var classA classB chisq df     p
## 1    AGE1      1      2  3.56 NA 0.059
## 2    AGE1      1      3 18.34 NA 0.000
## 3    AGE1      2      3 15.46 NA 0.000
## 4 GENDER1      1      2  0.11 NA 0.741
## 5 GENDER1      1      3  5.56 NA 0.018
## 6 GENDER1      2      3  7.79 NA 0.005
##
## attr(,"class")
## [1] "mplus.auxE" "list"
```

3.2 Exercise 2: Latent Transition Analysis (LTA)

3.2.1 2a. Latent transition analysis with probability parameterization

The file DatingSex.dat holds data on five indicators measured at two occasions (see MPlus example on LTA in user guide), as well as the variable gender. Read the file into memory, and name the variables:

```
data <- read.table("DatingSex.dat", na.strings = -99)
names(data) <- c("u11", "u12", "u13", "u14", "u15",
                 "u21", "u22", "u23", "u24", "u25",
                 "gender")
```

The u-variables represent five yes/no items (second digit represents the item) measured at two time points (first digit represents time point). Set up a model with two latent class variables for the two time points. Exclude the variable gender for now. Assume there are 2 latent classes.

Set up and run a model that restricts the thresholds (and hence response probabilities) across the two time points by first repeating the thresholds for each Latent Class (2), in both Model C1: and Model C2. To be sure Mplus does what you want, include equality constraints on the five thresholds of c1#1 and c2#1, and similarly for c1#2 and c2#2. Use the lecture slides for help in specifying the model. Check the Mplus input files manually.

Using MplusAutomation, the model can be specified as follows:

```
results_2a <-
createMixtures(
# Create only one model, with two classes
classes = 2,
# Name the generated files "2a"
filename_stem = "2a",
# Specify the autoregressive effect
model_overall = "c2 ON c1;",
# Specify two class-specific models;
# one for each categorical latent variable
model_class_specific = c(
"[u11$1] (a{C});  [u12$1] (b{C});  [u13$1] (c{C});
 [u14$1] (d{C});  [u15$1] (e{C});",
"[u21$1] (a{C});  [u22$1] (b{C});  [u23$1] (c{C});
 [u24$1] (d{C});  [u25$1] (e{C});"
),
rdata = data,
usevariables = names(data)[-11],
# Speed up analysis by using 2 processors; increase random starts
# Use probability parametrerization
ANALYSIS = "PROCESSORS IS 2;  LRTSTARTS (0 0 40 20);
            PARAMETERIZATION = PROBABILITY;",
# Specify that the items are categorical (binary)
VARIABLE = "CATEGORICAL = u11-u15 u21-u25;",
run = 1L
)
```

After running the analysis, inspect the proportions of yes/no answers for each of the indicators in the latent classes (look at probability scale in Mplus output).

Click to show answers

You can easily extract the parameters in probability scale from the output, using the $ sign. They will be printed to the console as a table.

Here, we examine just the first few rows of the table:

```
results_2a$results$parameters$probability.scale
```

param	category	est	se	est_se	pval	LatentClass
U11	1	0.43	0.02	18	0	C1#1
U11	2	0.57	0.02	23	0	C1#1
U12	1	0.43	0.03	16	0	C1#1
U12	2	0.57	0.03	22	0	C1#1
U13	1	0.25	0.02	12	0	C1#1
U13	2	0.75	0.02	35	0	C1#1

3.2.2 2b. Class proportions and transition probabilities

Examine the proportions of participants in class, based on the estimated model. Note that for each latent variable, the total proportions add up to 1. Next, examine the latent transition probabilities based on the estimated model. What do these probabilities signify?

Click to show answers

Again, you can extract this information from the output, using the $ sign. They will be printed to the console as a table.

Your results should look like this:

```
results_2a$results$class_counts$modelEstimated
```

variable	class	count	proportion
C1	1	554	0.55
C1	2	446	0.45
C2	1	667	0.67
C2	2	333	0.33

These probabilites represent the proportion of the total sample that is assigned to each class. Note that an individual can have a non-zero probability of being assigned to both classes. E.g., there might be a 70% probability that the person belongs to class 1, and a 30% probability that the person belongs to class 2.

The proportions here are a sum across those probabilities for all participants. Thus, this person would contribute for 30% to the proportion of the sample in class 2.

```
results_2a$results$class_counts$transitionProbs
```

from	to	probability
C1.1	C2.1	0.76
C1.2	C2.1	0.56
C1.1	C2.2	0.24
C1.2	C2.2	0.44

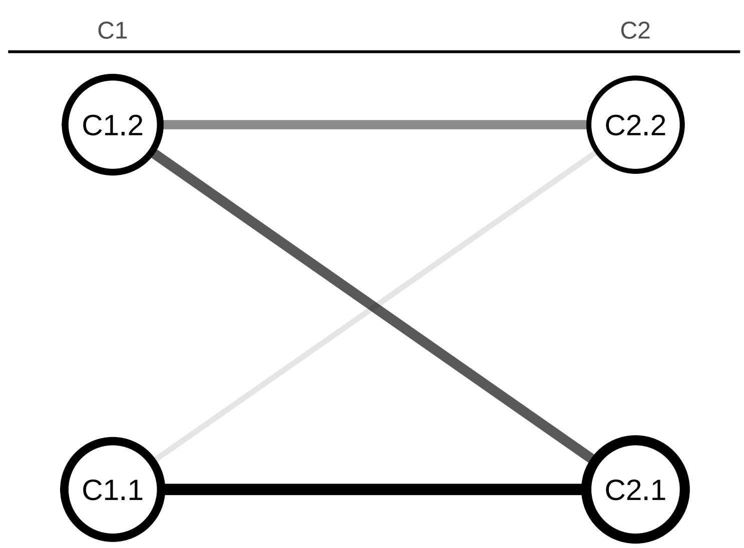
These probabilites represent the probability that an individual assigned to one class at time one, will be assigned to another class at time 2. So for example, we see that people in class 1 at time 1 also tend to be in class 1 at time 2 (.76 probability).

These probabilities can be visualized as a nodes-and-edges plot, using `plotLTA()`. How are the numeric results from the output reflected in the graph generated by this function?

Click to show answers

The probabilities are mapped to the line width of the circles and lines in the plot. We see that the nodes with biggest proportion of the sample (classes C1.1 and C2.1) have thicker circles, and the edges connecting C1.1 and C2.1, and C1.2 to C2.1, are thickest, because they are the most common transitions.

```
plotLTA(results_2a)
```



3.2.3 2c. Adding control variables (optional)

If there is time, you can conduct additional analyses, including gender as a control variable.

Click to show answers

You could include gender as a control variable on the observed variables, by adding it to the `usevariables`, and including the following lines:

`u11-u15 ON gender; u21-u25 ON gender;`

Alternatively, you could regress class membership on gender, to see whether men are more likely to be in a particular class than women, or vice versa. This is only allowed when you're NOT using probability parametrization. So, you would have to remove this line:

`PARAMETERIZATION = PROBABILITY;`

And add this line:

`c1 c2 ON gender;`

3.2.4 2d. Extend the model to a mover-stayer model (optional)

This Mplus FAQ explains how to extend the model to a mover-stayer model. At this point, we have to forget about the function `createMixtures`: It is not suited to estimate the mover-stayer model; it can only estimate simple latent class (growth) models for a varying number of classes. However, we can use this as an opportunity to look at another function in the `MplusAutomation` package: `mplusModeler`. This function creates and runs the Mplus syntax in one step:

First, specify the Mplus model. Every argument of the function `mplusObject` corresponds to a section of an Mplus input file:

```
mover_stayer_model <- mplusObject(
  VARIABLE = "CATEGORICAL = u11-u15 u21-u25;
              CLASSES = move(2) c1(2) c2(2);",
  ANALYSIS = "TYPE = mixture;
              PROCESSORS IS 2;
              LRTSTARTS (0 0 40 20);
              PARAMETERIZATION = PROBABILITY;",
  MODEL = "%OVERALL%
           c1 ON move;
           MODEL move:
           %move#1% ! Movers
           c2 on c1;
```

```
            %move#2% ! Stayers
            c2#1 ON c1#1@1;
            c2#1 ON c1#2@0;
            MODEL c1:
            %c1#1%
            [u11$1] (a1);
            [u12$1] (b1);
            [u13$1] (c1);
            [u14$1] (d1);
            [u15$1] (e1);
            %c1#2%
            [u11$1] (a2);
            [u12$1] (b2);
            [u13$1] (c2);
            [u14$1] (d2);
            [u15$1] (e2);
            MODEL c2:
            %c2#1%
            [u21$1] (a1);
            [u22$1] (b1);
            [u23$1] (c1);
            [u24$1] (d1);
            [u25$1] (e1);
            %c2#2%
            [u21$1] (a2);
            [u22$1] (b2);
            [u23$1] (c2);
            [u24$1] (d2);
            [u25$1] (e2);",
  OUTPUT = "tech15;",
  rdata = data,
  usevariables = names(data)[-11])
```

Next, run the model we created above. You specify an input file, which will be created, and the argument `run = 1L` means you want to run the model. Set it to `run = 0L` (the default) to create syntax without running it.

```
result <- mplusModeler(object = mover_stayer_model,
                       modelout = "mover_stayer.inp",
                       run = 1L)
```

Look in the output for the information about the response probabilities for the various latent classes. Also, look for the transition table. Which classes are the movers and which are the stayers?

```
get_class_counts(result)
```

variable	class	count	proportion
MOVE	1	588	0.59
MOVE	2	412	0.41
C1	1	554	0.55
C1	2	446	0.45
C2	1	667	0.67
C2	2	333	0.33

3.2.5 2e. Selecting the number of classes (optional)

We have not investigated whether 2 classes is the right number for this dataset. Investigate how many classes at each timepoint you would choose.

Think about:

- Whether you think there should be an equal number of classes at both timepoints (this is mostly a theoretical decision).
- How to build the model. Should you start by comparing unconstrained or constrained models?
- How to decide what solution you prefer.

Chapter 4

Day 4: Cross-lagged relations

These exercises belong to day 4 of S23, “Cross-lagged relations". The exercises consist roughly of 2 parts: first we will focus on the quasi-simplex model, and second we will focus on the random intercept cross-lagged panel model.

For the quasi-simplex model we are going to study the concept of life satisfaction. The covariance matrix is in the file `Coenders.dat` and comes from 1724 children and adolescents that participated in the *National Survey of Child and Adolescent Well-Being* (NSCAW) in Russia. They indicated how satisfied they were with their lives as a whole on a 10-point scale (1 = not at all satisfied, 10 = very satisfied). There were three waves (1993, 1994 and 1995). At the third wave, the question was asked twice (with 40 minutes in between). Hence, in total there are four measurements obtained at three waves. The data and variables commands for these data should read:

```
DATA:
  TYPE = COVARIANCE;
  FILE = Coenders.dat;
  NOBSERVATIONS = 1724;

VARIABLE:
  NAMES = Y1 Y2 Y3 Y4;
```

The researchers are interested in fitting a quasi-simplex model to these data, that is, a simplex model at the latent level, thus accounting for measurement error in the observations. This model is graphically represented in slide 19. Use the menu on the left to navigate to the exercises. The Mplus input files can found in the Summer School Dropbox-folder (`./Dropbox/Lab material - Day 4/S23 - Day 4 - 4MplusInputFiles`).

4.1 Quasi-simplex model

4.1.1 Exercise A

Provide the names of the variances (i.e., indicate in which model matrix, and which position in this matrix they have) in the quasi-simplex graph on slide 19. What is the difference between the e's and the ζ's?

Click to show answers

e's are residuals at the measurement level and can be found in the θ-matrix. They only influence the observation at a single occasion in time.

ζ's are residuals of the simplex process and can be found in the ψ-matrix. Their effect is carried forward to future observation through the autoregressive relationships.

4.1.2 Exercise B

How would you specify the model in Mplus?

Click to show answers

See the Mplus input file `Exercise B.inp`.

4.1.3 Exercise C

Determine the number of degrees of freedom for this model (indicate how you obtained this number). Is it possible to estimate this model?

Click to show answers

There are $\frac{4\times5}{2} = 10$ unique elements in S.

Free parameters:

- 4 residual variances at measurement level
- 1 factor variance
- 3 residual factor variances
- 3 regression parameters
- 11 parameters in total.

Therefore, we have $10 - 11 = -1$ *df*. It is not possible to estimate this model because we are trying to estimate more parameters than we have information in the data.

4.1.4 Exercise D

To make sure a quasi-simplex model is identified, often the variances of the measurement errors are constrained to be equal over time. How can you do this in Mplus? How many *df* does this model have?

Click to show answers

See the Mplus input file `Exercise E.inp` for how to constrain the measurement error variances over time. With constrained measurment error variances, we estimate 3 parameters less. So, we estimate $11 - 3 = 8$ and therefore have $10 - 8 = 2$ *df*.

4.1.5 Exercise E

Run the model and report on the model fit.

Click to show answers

We find the below fit indices:

- $\chi^2 = 13.29$ with $df = 2$, $p = .0013$,
- RMSEA = .057,
- CFI = .994, and
- TLI = .981.

Except for χ^2 test of model fit, the model seems to fit the data well. Note, however, that the sample size is very large and therefore the χ^2 is likely to be significant, even for minor problems with model fit.

4.1.6 Exercise F

The quasi-simplex model you just ran, led to the following warning:

```
WARNING: THE LATENT VARIABLE COVARIANCE MATRIX (PSI) IS NOT POSITIVE
DEFINITE. THIS COULD INDICATE A NEGATIVE VARIANCE/RESIDUAL VARIANCE FOR A
LATENT VARIABLE, A CORRELATION GREATER OR EQUAL TO ONE BETWEEN TWO LATENT
VARIABLES, OR A LINEAR DEPENDENCY AMONG MORE THAN TWO LATENT VARIABLES.
CHECK THE TECH4 OUTPUT FOR MORE INFORMATION.
PROBLEM INVOLVING VARIABLE ETA4.
```

What is the problem?

Click to show answers

In your output, look at the reported (estimated) residual variances. We find that the residual variance of `ETA4` is estimated to be negative. This is a Heywood case and it is causing the warning to appear. Note however, that it is significant, so “just" fixing it to 0 as a solution is probably not warrented here.

4.1.7 Exercise G

As indicated in the description of the data, the third and fourth measurement were obtained at the same measurement wave (with only 40 minutes in between). Hence, the researchers proposed the following model instead of the regular quasi-simplex model. Explain why this model makes more sense for these data than the regular quasi-simplex model. Tip: check the description of the study at the beginning of this exercise.

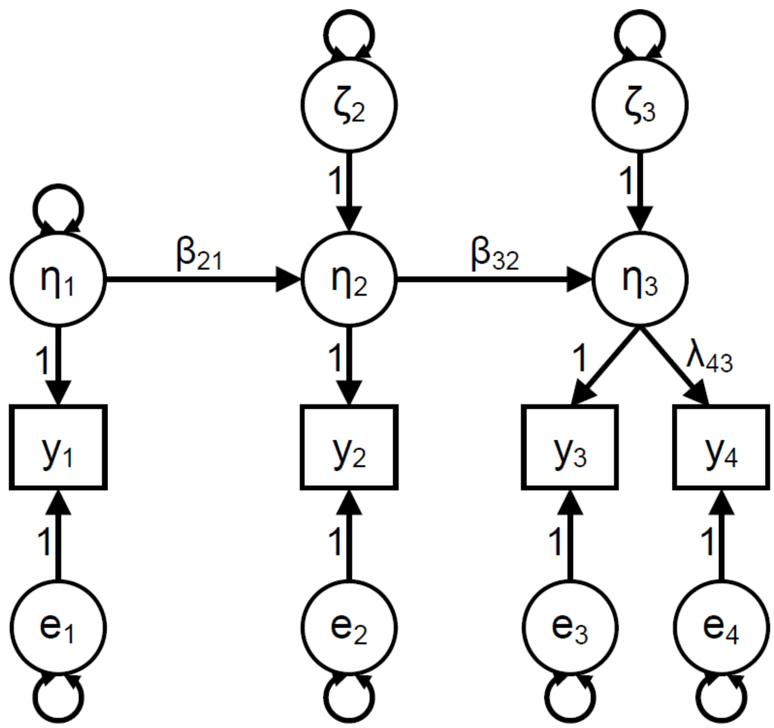


Figure 4.1: Adjusted quasi-simplex model.

Click to show answers

At each occasion there is a latent variable which represents Life Satisfaction. At the first two occasions there was only a single indicator of this latent variable, but at the third occasion there were two indicators.

4.1.8 Exercise H

How many *df* does this model have? Note that we keep the constraint on the variances of the measurement errors.

Click to show answers

There are $\frac{4*5}{2} = 10$ unique elements in S. We freely estimate:

- 1 constrained residual variances at measurement level
- 1 factor variance
- 2 residual factor variances
- 2 regression parameters
- 1 factor loading
- 7 parameters in total.

Therefore, we have $10 - 7 = 3$ *df*.

4.1.9 Exercise I

Are these two models nested? If so, how? If not, why not, and how could we compare them?

Click to show answers

Yes, they are nested: this model is a special case of the previous model, as it is based on having `ETA3` and `ETA4` from the previous model now being a single latent variable. That is, we can constrain the residual variance of `ETA4` to zero to get the alternative model. This gives us 1 *df* for the difference.

4.1.10 Exercise J

Specify this model in Mplus and run it. Report on the model fit.

Click to show answers

See the Mplus input file `Excercise J.inp` for the model specification in Mplus. Apart from the χ^2-test of model fit, the model fits well:

- $\chi^2(3) = 27.37$, with $p < .001$,
- RMSEA = 0.069,
- CFI = 0.987,
- TLI = 0.973, and
- SRMR = 0.039.

4.1.11 Exercise K

Compare the two models to each other. What can you conclude?

Click to show answers

Comparing both models using the $\Delta\chi^2$-test gives us 27.37˘13.29 = 14.0 with 1 *df* such that $p < .001$. This implies that imposing the restriction is not tenable. You can calculate the p-value of the $\Delta\chi^2$ using the `pchisq()`-function in R (with the `lower.tail` argument set to `FALSE`), or an online tool

Comparing the models using information criteria gives us AIC = 29593 and BIC = 29637 for the first model, and AIC = 29605 and BIC = 29643 for the second. In conclusion, all measures indicate the first model is better. However, the current model makes more theoretical sense, and the negative variance estimate in the first model is a problem. For these 2 reasons, we should prefer the current model.

4.1.12 Exercise L

Can you improve the second model in any way? Indicate which parameter you would add to your model, and what this parameter represents in substantive terms.

Click to show answers

You can get the modification indices by adding `MOD` to the `OUTPUT` command. Here, the suggested BY statements make no sense (later life satisfaction as an indicator of previous life satisfaction). With regards to the ON statement, only the suggested effect of `ETA3 ON ETA1` makes sense as we then predict forwards in time. The WITH statement suggests adding a covariance between the residuals of `y3` and `y4`. If we add this covariance and look at the standardized results, we get a correlation. This correlation actually quite high: .522 (SE = .044), $p < .001$.

4.1.13 Exercise M

Run a model in which you include the `Y3 WITH Y4` parameter. Where will this relationship end up in the model? Does it lead to a significant improvement? How would you interpret this additional parameter?

Click to show answers

See `Exercise M.inp` for the Mplus specification of this model. The `Y3 WITH Y4` parameter is an additional covariance between the residuals of `y3` and `y4` (so not between `y3` and `y4` themselves). Model fit is quite good (except for the χ^2-test of model fit):

- $\chi^2(2) = 7.077$, $p = .0291$,
- RMSEA = 0.038,
- CFI = 0.997,
- TLI = 0.992, and
- SRMR = 0.011.

To compare this model to the previous model, we can do a the $\Delta\chi^2$-test: 27.37˘7.08 = 20.29, with 1 *df* such that $p < .001$, which implies that adding the covariance between the residuals leads to a significant improvement in model fit. This additional parameter implies that `y3` and `y4` have more in common with each other than what would be expected based on their common dependence on `ETA3`. Note that in the standardized results, the WITH statement can be interpreted as a correlation, and it is quite high: .522 (SE = .044), $p < .001$.

4.2 CLPM & RI-CLPM

For the cross-lagged panel model (CLPM) and the random intercept cross-lagged panel model (RI-CLPM) we are going to analyze data that were reported in Davies, Martin, Coe and Cummings (2016). The summary data (means, standard deviations and correlation matrix) are included in Davies.dat, and contains the means, standard deviations, and the correlation matrix. The number of observations is 232. There are 5 waves of data, taken when the child was 7, 8, 13, 14, and 15 years old. The order of the variables is:

- Child gender
- Parental education
- Interparental hostility (waves 1-5): composite score based on observational data and questionnaires, reflecting the degree of hostility between the parents
- Interparental dysphoria (waves 1-5): based on composite score based on observational data and questionnaires, reflecting the degree of dysphoria
- Child/adolescent insecurity in the relationship with the parents (waves 1-5)
- Psychological problems (waves 1-5): based on the subscales anxious/depressed, withdrawal, aggressive behaviors, and delinquency scales of the Child Behavior Checklist (CBCL), filled out by both parents.

Here we will focus on *Interparental dysphoria* and *Psychological problems* of the child. The DATA and VARIABLE commands should be:

```
DATA:
  TYPE = MEANS STDEVIATIONS CORRELATION;
  FILE = Davies.dat;
```

```
  NOBSERVATIONS = 232;

VARIABLE:
  NAMES = ChildGen ParentEd
  Hos1 Hos2 Hos3 Hos4 Hos5 Dys1 Dys2 Dys3 Dys4 Dys5
  Ins1 Ins2 Ins3 Ins4 Ins5 PsPr1 PsPr2 PsPr3 PsPr4 PsPr5;
  USEVARIABLES = Dys1-Dys5 PsPr1-PsPr5;
```

4.2.1 Exercise A

How many sample statistics are there for this data set (focusing on the 5 measures of dysphoria and the 5 measures of psychological problems?

Click to show answers

There are 10 observed variables such that there are $\frac{10*11}{2} = 55$ unique elements in the observed covariance matrix S, and 10 observed means in M. Therefore, there are 55 sample statistics in total.

4.2.2 Exercise B

We begin with an RI-CLPM (see slide 53). For now, do not impose any constraints on the parameters across time. Draw the model, and indicate which parameters will be estimated freely. How many parameters will be estimated in total? So how many *df* are there?

Click to show answers

In the RI-CLPM we estimate:

- 2 variances for the random intercepts,
- 1 covariance between the random intercepts,
- 2 variances for the within-person centered variables at wave 1,
- 1 covariance between the within-person centered variables at wave 1,
- 8 residual variances (for the dynamic errors of both variables at wave 2-5),
- 4 covariances between the residuals (for the dynamics errors at waves 2-5),
- 16 lagged parameters (4 for each interval), and
- 10 means.

In total, we estimate 44 parameters such that we have $65 - 44 = 21$ *df*.

4.2.3 Exercise C

Run the model. Check whether the number of *df* is correct. Also look at the TECH1 output, to see if you understand where the free parameters are. What is the model fit?

Click to show answers

The input for this model is in `RICLPM.inp`. The model means are estimated in the ν-matrix, no parameters are estimated in the θ-matrix (measurement error variances), λ-matrix (factor loadings), or α-matrix (means/intercepts of the latent variables). The variances and covariance of the random intercepts, the within-person centered variables at wave 1, and the dynamic errors at subsequent waves are all estimated in the ψ-matrix. The lagged regression coefficients are estimated in β.

Apart from the χ^2-test of model, all fit indices indicate at least acceptable fit.

- $\chi^2(21) = 41.451$, $p = .005$,
- RMSEA = 0.065,
- CFI = 0.979,
- TLI = 0.956, and
- SRMR = 0.029.

4.2.4 Exercise D

Include the significant standardized parameter estimates for the covariances (i.e., the WITH statements) and the lagged regression parameters (i.e., the ON statements) in the figure below. Indicate which part of the model is considered the between-person part, and which part is the within-person part.

4.2.5 Exercise E

Omit the random intercepts. How many parameters and *df* does this model have? What is the model fit?

Click to show answers

The input for this model is `CLPMasRICLPM.inp`. The model has three parameters less (and thus 3 *df* more) than the previous model: 2 variances and the covariance for the random intercepts.

The model fit indices show that this model does not fit well:

- $\chi^2(24) = 73.374$, $p < .001$,
- RMSEA = 0.094,
- CFI = 0.950,
- TLI = 0.907, and
- SRMR = 0.061.

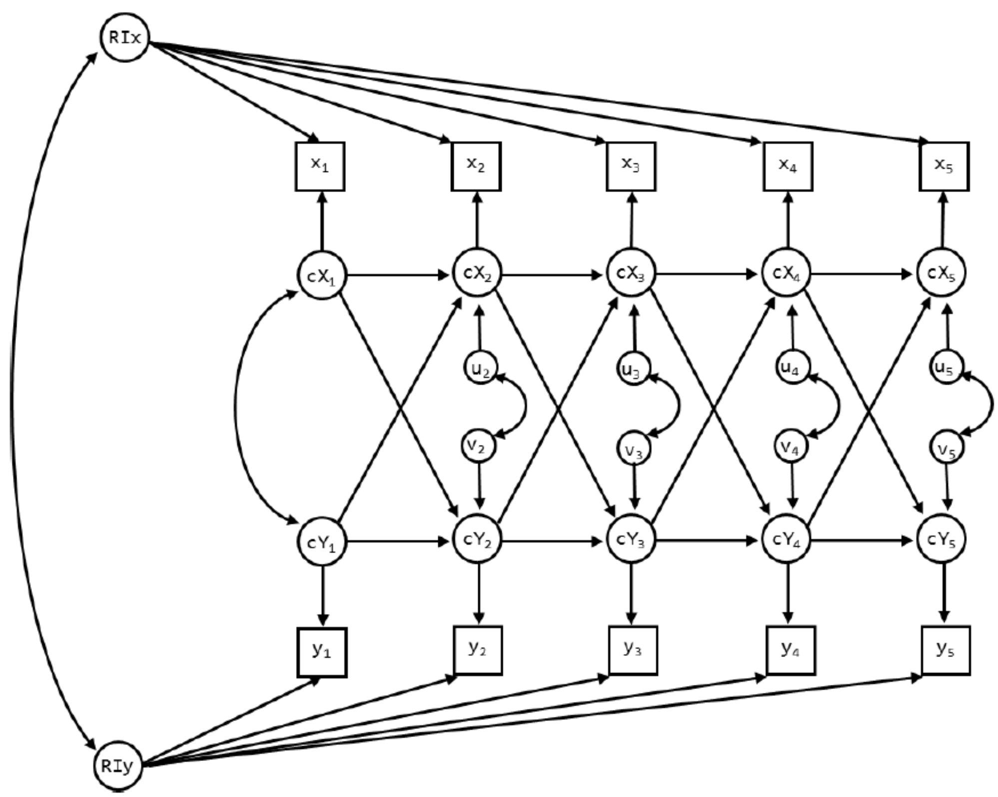


Figure 4.2: The bivariate random-intercept cross-lagged panel model with 5 repeated measures (waves).

4.2.6 Exercise F

Specify the CLPM and run this model. Compare it to the previous two models. How are these models related?

Click to show answers

The input for this model is in `CLPM.inp`. This model is statistically identical to the previous model; these are different parameterizations of the same model. The model fit is therefore also exactly the same. Hence, this model is a special case of the RI-CLPM.

Comparing the two models using a chi-square difference test gives: $\Delta\chi^2 = 73.37 - 41.45 = 31.92$ with $24\check{\ }21 = 3$ *df*, $p < .001$. Hence, the random intercepts should not be omitted; put differently, there are stable, trait-like difference between families in the two variables (parental dysphoria and psychological problems).

However, when constraints are placed on the bound of the parameter space (which is the case here, fixing a variance to 0 is its absolute minimum value), we should actually use the chi-bar-square test ($\bar{\chi}^2$-test; Stoel et al. 2006). The traditional $\Delta\chi^2$-test does not take into account that variances can only be positive and is therefore conservative. This means that if it is significant, we are certain that the correct test (i.e., the $\bar{\chi}^2$ test) would also be significant. On the other hand, when the usual chi-square test is not significant, we do not know anything about the result of the correct test (it can be significant or not significant).

If you are working in R with the lavaan-package, you can find more information about the $\bar{\chi}^2$-test at jeroendmulder.github.io/RI-CLPM/lavaan.html#(bar{chi}^{2})-test. For Mplus users, there is a Shiny app by Rebecca Kuiper available as well.

4.2.7 Exercise G

Include the significant standardized parameter estimates for the covariances and the lagged regression parameters in the figure below.

4.2.8 Exercise H

Discuss how the model results differ.

Click to show answers

Cross-lagged relationships In the RI-CLPM none of the cross-lagged parameters are significant. In contrast, in the CLPM there is a positive relationship from PsPr1 to Dys2. This implies that higher levels of children's psychological problems at age 7 are followed by higher levels of interparental dysphoria at age 8. Moreover, from age 14 to 15 both cross-lagged parameters are significant

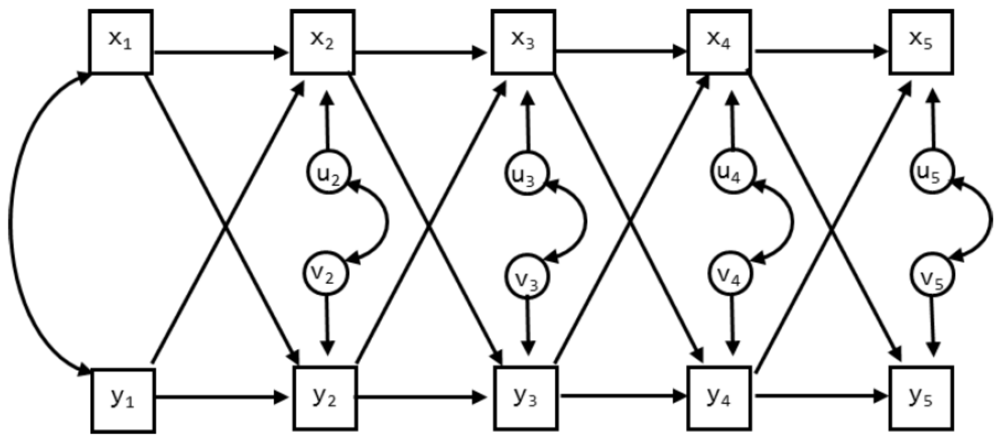


Figure 4.3: The bivariate cross-lagged panel model with 5 repeated measures (waves).

and positive, indicating that psychological problems are followed by increases in interparental dysphoria, but also that increased interparental dysphoria is followed by an increase in psychological problems for the adolescent.

Autoregressive parameters The autoregressive parameters in the RI-CLPM are lower, and have larger SE's, such that fewer reach significance. This is expected as within-person stability is now captures in the random intercepts, rather than in the autoregressive effects in the CLPM.

Correlations In the CLPM only the residual correlation at wave 2 is significant; it is negative, indicating that external effects tend to have an opposite effect on these two processes; increases in Dysphoria are accompanied by decreases in psychological problems and vice versa. In the RI-CLPM, the within-person correlation at wave 1 is not significantly different from zero; however, at waves 2, 3 and 4 the correlations between the residuals is significant and negative. At wave 5 the residual variance is not significant.

In the RI-CLPM there is also the correlation between the random intercepts (i.e., the trait-like difference between families). This turns out to be a very substantial correlation of .63: Hence, in contrast to the results from the CLPM and the within-level results from the RI-CLPM, there is a strong positive relationship between trait-like levels of interparental dysphoria and trait-like levels of psychological problems.