

# **semnova**: An R package for latent repeated measures ANOVA

Benedikt Langenberg, Jonathan L. Helm, and Axel Mayer

2023-01-31



# Contents

<b>Preface</b>	<b>5</b>
About this Tutorial . . . . .	6
<b>1 Motivating Example</b>	<b>7</b>
<b>2 Installation</b>	<b>11</b>
<b>3 Within-Subjects Designs</b>	<b>13</b>
3.1 Sphericity . . . . .	15
3.2 Latent Variables . . . . .	18
<b>4 Between-Subjects Designs</b>	<b>25</b>
4.1 Latent Variables . . . . .	26
<b>5 Covariates</b>	<b>31</b>
<b>6 Power Analysis</b>	<b>35</b>
<b>7 Further Arguments</b>	<b>41</b>
<b>8 Further Developments</b>	<b>43</b>
<b>A Power Script Example</b>	<b>45</b>
<b>B Filled Power Script Example</b>	<b>53</b>
<b>C Bibliography</b>	<b>61</b>



# Preface

*semnova* is an R package for analyzing data from repeated measures experimental designs. *semnova* implements latent repeated measures analysis of variance (L-RM-ANOVA) an alternative approach to traditional repeated measures ANOVA (RM-ANOVA). Traditional RM-ANOVA are commonplace for investigating phenomena in the social and behavioral sciences, and researchers often use RM-ANOVA as a statistical method for analyzing data from repeated measures designs. The popularity of RM-ANOVA is, to some extent, due to its generality (i.e., many hypotheses may be tested via main and/or interaction effects), ease of application (i.e., virtually all statistical packages conduct RM-ANOVA), and desirable statistical properties (since these are optimal when all assumptions are fulfilled). RM-ANOVA, however, has several shortcomings including that it (1) focuses on drawing conclusions about average (causal) effects and does not consider interindividual differences, (2) assumes both perfectly reliable measures and measurement invariance for latent constructs, (3) assumes homogeneity of variance and fixed group sizes when including between-subject factors, (4) cannot handle missing data, and (5) relies on the normality assumption.

L-RM-ANOVA addresses the aforementioned issues. It is a structural equation modeling (SEM) based approach – as opposed to RM-ANOVA which relies on the general linear model – and extends the latent growth components model. (1) L-RM-ANOVA enables the researcher to not only consider mean differences between experimental conditions (i.e., main and interaction effects) but also to investigate variances of effect variables that represent interindividual differences. L-RM-ANOVA estimates variances of effect variables across individuals and also allows to predict effect variables by covariates which explain a share of the variances of the effect variables. (2) As L-RM-ANOVA is an SEM based approach, it can incorporate a measurement model for both the dependent variables and the predictors to explicitly model measurement error. It is also possible to test the oftentimes implicit assumption of measurement invariance across experimental conditions. (3) The approach can be further extended to include between-subject factors (i.e., groups) through dummy-coded variables or by using a multi-group SEM approach. The multi-group approach allows for the relaxation of the homogeneity of variance assumption across groups and

to estimate the group sizes as a model parameter, thus accounting for uncertainty in the group sizes. (4+5) Lastly, L-RM-ANOVA profits from a variety of SEM benefits, namely fit indices, handling missing data through full information maximum likelihood, robust estimators, and robustness against violations of normality.

## **About this Tutorial**

This very short tutorial is mainly based on a chapter of dissertation (see Langenberg, 2022, Chapter 7 “Software”). If you would like to know more about L-RM-ANOVA, check out my dissertation which can be found by following the DOI in the references.

# Chapter 1

## Motivating Example

The open-source R (R Core Team, 2021) software package *semnova* implements many of the methods demonstrated in this dissertation. The *semnova* package is based on the open-source R software package *lavaan* (Rosseel, 2012) which implements a wide range of methods for estimating structural equation models. This chapter gives an overview of the available features and explains the output by means of code examples using data from the motivating example about reading development in children from the introduction (see Langenberg, 2022, Section 1.1.1 “Motivating Example”). Recall, in the study,  $N = 267$  children had to read two types of sentences: regular sentences or Landolt sentences in which characters are replaced by circles. The study included three measurements in the first, second, and fourth grade of school. Female and male children participated in the study. The dependent variables were re-fixation time<sup>1</sup>, initial landing position<sup>2</sup>, and fixation count<sup>3</sup>. The experimental design consists of a  $3 \times 2 \times 2$  mixed design, with the within-subjects factors grade (Factor A: A1 = grade one, A2 = grade two, A3 = grade four) and sentence type (Factor B: B1 = regular sentences, B2 = Landolt sentences). In this tutorial we additionally introduce the between-subjects factor sex (Factor C: C1 = female, C2 = male). Figure 1.1 shows the means and standard errors of re-fixation time, initial landing position, and fixation count across all six conditions for both genders. The data are included in the *semnova* package as a manifest and a latent version (`reading_manifest` and `reading_latent`). The data sets do not need to be explicitly loaded and can immediately be accessed after loading the package.

We will use the same contrast matrix for the within-subjects design as in the

---

<sup>1</sup>Re-fixation time includes the total duration of fixations during the first gaze at a word excluding saccades in-between fixations. The variable is measured in milliseconds.

<sup>2</sup>Initial landing position is the position of the letter of a word that was fixated first. The variable ranges from 0 to 6.

<sup>3</sup>Fixation count is the number of fixations on a word during all gazes. The variable is a positive integer.

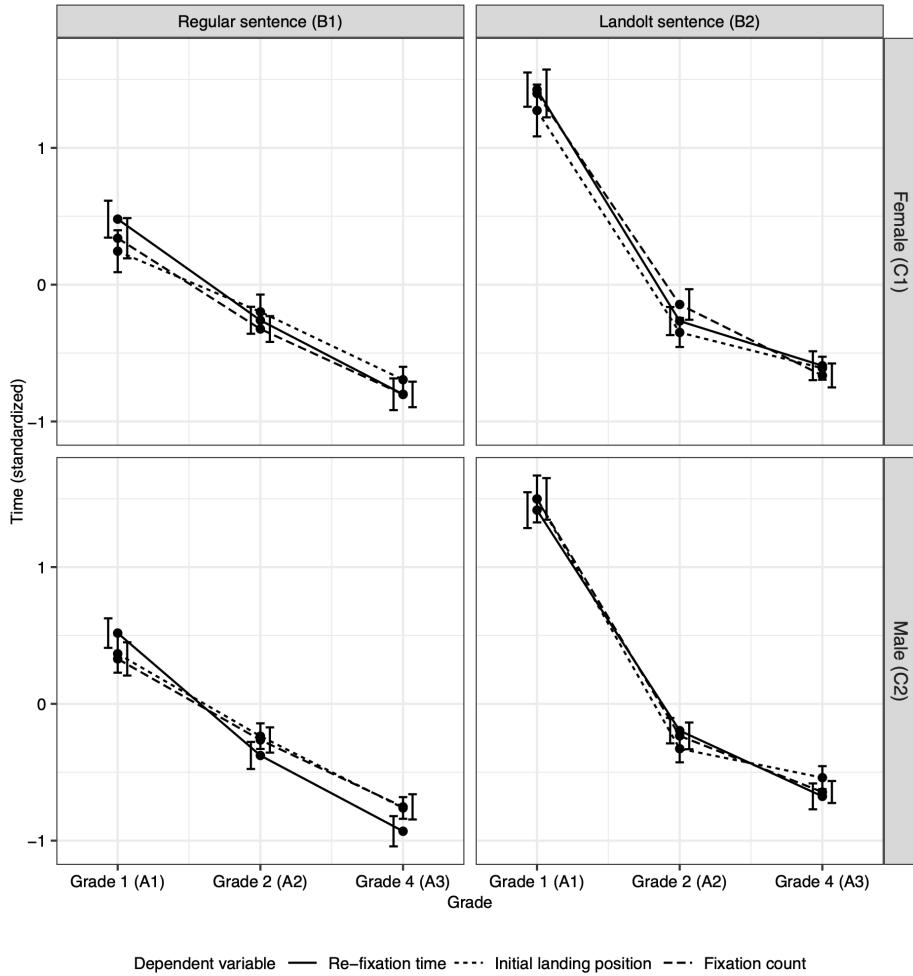


Figure 1.1: Re-fixation time (solid line), initial landing position (dotted line), and fixation count (dashed line) depending on sentence type (left panel: regular sentences; right panel: Landolt sentences), and sex (top panel: female; bottom panel: male) for the three measurement occasions (grade 1, grade 2, grade 4). The three variables were first averaged within children, then log-transformed and standardized. Shown are means across children. Error bars indicate standard errors.

introduction (see Langenberg, 2022, Section 1.2 “Latent Repeated Measures ANOVA”). It is an orthogonal (i.e., linearly independent rows with length equal to one) polynomial contrast matrix and is given by:

$$\mathbf{C} = \begin{pmatrix} 0.408 & 0.408 & 0.408 & 0.408 & 0.408 & 0.408 \\ -0.5 & 0 & 0.5 & -0.5 & 0 & 0.5 \\ 0.289 & -0.577 & 0.289 & 0.289 & -0.577 & 0.289 \\ -0.408 & -0.408 & -0.408 & 0.408 & 0.408 & 0.408 \\ 0.5 & 0 & -0.5 & -0.5 & 0 & 0.5 \\ -0.289 & 0.577 & -0.289 & 0.289 & -0.577 & 0.289 \end{pmatrix}$$

In the following subsections, we will show numerous structural equation model (SEM) path diagrams that represent the estimated models. The entries from the inverse of the  $\mathbf{C}$  matrix (i.e. the  $\mathbf{B}^*$  matrix) can be found on the regression arrows from the  $\pi$  variables to the  $\eta$  variables. It is noteworthy that the inverse of the  $\mathbf{C}$  matrix equals its transpose because it is an orthonormal matrix. This is a convenient property because the regression coefficients from the path diagrams can be found in both the  $\mathbf{B}^*$  and the (“rotated”)  $\mathbf{C}$  matrix. Thus, no inversion is needed to compare the path diagrams with the  $\mathbf{C}$  matrix.



# Chapter 2

## Installation

This chapter will use Version 0.2.0 of the *semnova* package. *semnova* can be installed from *CRAN*<sup>1</sup>. However, we recommend to download the latest version from *GitHub*<sup>2</sup>. The following lines of code can be used to install the package.

```
install.packages("devtools")  
  
devtools::install_github("langenberg/semnova@v0.2.0")  
  
library(semnova)
```

Note that the *devtools* package must be installed in order to be able to install software from *GitHub*.

---

<sup>1</sup><https://cran.r-project.org>

<sup>2</sup><https://github.com>



# Chapter 3

## Within-Subjects Designs

For the first code example, we use the two within-subjects factors sentence type and grade from the motivating example with the manifest variable re-fixation time neglecting the between-subjects factor sex. The function `semnova()` offers a range of features and is used to estimate the first model:

```
fit <- semnova(  
  data = reading_manifest,  
  id = "id",  
  dv = "re_fix_time",  
  within = c("grade", "sentence")  
)
```

The function takes at least three arguments: `id` is a character vector of variables that uniquely identify the subjects, `dv` takes the name of the dependent variable, and `data` takes a data set in the long format (also called narrow format). Furthermore, `within` takes a vector of variables containing experimental within-subjects factors.

The next command is the `summary()` function which we use to extract the results from the model:

```
summary(fit)  
  
## Main and Interaction Effects:  
##                                Chisq df Pr(>Chisq)  
## Intercept          4.711  1     0.03 *  
## grade            832.066  2    <2e-16 ***  
## sentence         515.317  1    <2e-16 ***  
## grade:sentence  234.919  2    <2e-16 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The function prints hypothesis tests for each of the main and interaction effects. By default, the output contains a Wald test for the effects, future releases will include more statistics such as the likelihood ratio test and an approximate  $F$ -test. The factor grade has three levels and the test consequently has two degrees of freedom. All tests are highly significant, leading to the rejection of the null hypothesis and suggesting that the re-fixation time is neither constant across grade nor across the sentence types.

We can further use the `print()` function to print parameter estimates for each of the freely estimated model parameters. The following output is truncated after seven rows due to the large number of parameters.

```
print(fit)
```

##	##	## parameter	est	se	pvalue
##	-----	## alpha_{pi,1,1}	-0.0677313	0.0312067	0.0299760
##	## sigma_{pi,1,1,1}^2	0.1636084	0.0178511	0.0000000	
##	## sigma_{pi,1,2,1}^2	-0.0354949	0.0116314	0.0022760	
##	## sigma_{pi,1,3,1}^2	-0.0165968	0.0082027	0.0430381	
##	## sigma_{pi,1,4,1}^2	-0.0275617	0.0138830	0.0471122	
##	## sigma_{pi,1,5,1}^2	0.0400701	0.0143246	0.0051533	
##	## sigma_{pi,1,6,1}^2	0.0063447	0.0101050	0.5300847	
##	## alpha_{pi,2,1}	-0.7971504	0.0279478	0.0000000	
##	## sigma_{pi,2,2,1}^2	0.1312214	0.0143174	0.0000000	
##	## sigma_{pi,2,3,1}^2	-0.0249054	0.0075061	0.0009066	
##	## sigma_{pi,2,4,1}^2	0.0258745	0.0124476	0.0376466	
##	## sigma_{pi,2,5,1}^2	-0.0147721	0.0125781	0.2402218	
##	## sigma_{pi,2,6,1}^2	0.0255615	0.0092517	0.0057293	
##	## alpha_{pi,3,1}	0.0681849	0.0200307	0.0006640	
##	## sigma_{pi,3,3,1}^2	0.0674062	0.0073546	0.0000000	
##	## sigma_{pi,3,4,1}^2	0.0096321	0.0088372	0.2757335	
##	## sigma_{pi,3,5,1}^2	0.0341557	0.0093566	0.0002618	
##	## sigma_{pi,3,6,1}^2	0.0085963	0.0065123	0.1868360	
##	## alpha_{pi,4,1}	-0.7699475	0.0339175	0.0000000	
##	## sigma_{pi,4,4,1}^2	0.1932668	0.0210871	0.0000000	
##	## sigma_{pi,4,5,1}^2	-0.0326321	0.0154091	0.0341988	
##	## sigma_{pi,4,6,1}^2	-0.0105359	0.0110000	0.3381557	
##	## alpha_{pi,5,1}	0.5290506	0.0345798	0.0000000	
##	## sigma_{pi,5,5,1}^2	0.2008881	0.0219187	0.0000000	
##	## sigma_{pi,5,6,1}^2	-0.0264462	0.0113687	0.0200060	
##	## alpha_{pi,6,1}	-0.0470649	0.0249530	0.0592762	
##	## sigma_{pi,6,6,1}^2	0.1046057	0.0114134	0.0000000	

The `plot()` function further plots a path diagram for the model. shows two

path diagrams. Per default, the diagram will only show regression arrows and measurement model arrows (see Figure 3.1). The `what` argument can be used to also draw intercepts, variances, covariances, residual variances and residual covariances (see Figure 3.2).

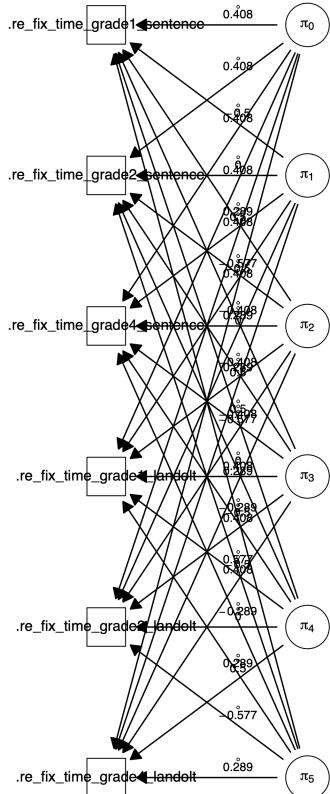


Figure 3.1: Path diagram for a  $2 \times 3$  (sentence  $\times$  grade) within-subjects design with the manifest dependent variables re-fixation time. Created by `plot(fit)`.

### 3.1 Sphericity

The next example shows how `semnova` can be used to test for sphericity as described in Chapter 4 “Understanding, Testing, and Relaxing Sphericity of Repeated Measures ANOVA with Manifest and Latent Variables Using SEM” (Langenberg, 2022). The `semanova()` function offers the `sphericity` argument to impose sphericity for a set of experimental factors. By default, the argument is `FALSE`. The argument can also take a list of vectors of experimental factors. Each vector describes an effect consisting of factor names (i.e., variable names) for which sphericity is to be assumed. For the factor `grade`, the code looks as follows:

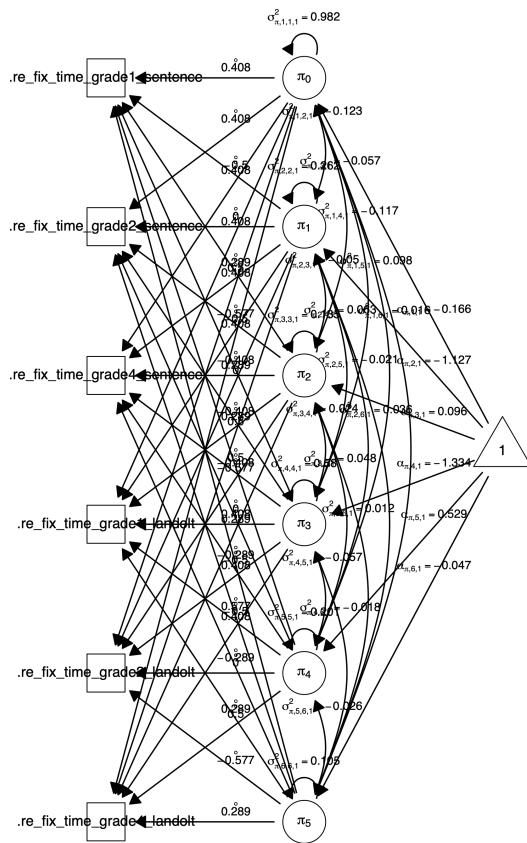


Figure 3.2: Path diagram for a  $2 \times 3$  (sentence  $\times$  grade) within-subjects design with the manifest dependent variables re-fixation time. Created by `plot(fit, what = "all")`.

```
## Chi-Squared Difference Test
##
##          Df      AIC      BIC   Chisq   Chisq diff Df diff Pr(>Chisq)
## fit           0 1494.6 1579.0  0.000
## fit_sph_grade 2 1521.1 1599.2 30.518     30.518      2 2.361e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All arguments are the same as for the previous example except for the `sphericity` argument. The factor grade has three levels and thus two effect variables. This command will force the variances of the effect variables  $\pi_1$  and  $\pi_2$  to be equal and their covariance to be zero. The `anova()` function can compare multiple models. Comparing a model with unconstrained covariance matrix to a model that imposes sphericity yields a formal test for sphericity.

The test is significant, suggesting that sphericity cannot be assumed. That is, either the variances of the effect variables are unequal, or the covariance is not zero. Thus, univariate test statistics cannot be used and multivariate tests should be preferred (e.g., Lane, 2016). The test presents an alternative to the often-used Mauchly's sphericity test (Mauchly, 1940).

The same test can be performed for the interaction of grade and sentence type because the effect consists of the two contrast variables  $\pi_4$  and  $\pi_5$  (i.e., the effect has two degrees of freedom):

```
fit_sph_grade_sentence <- semnova(
  data = reading_manifest,
  id = "id",
  dv = "re_fix_time",
  within = c("grade", "sentence"),
  sphericity = list(c("grade", "sentence"))
)

anova(fit, fit_sph_grade_sentence)

## Chi-Squared Difference Test
##
##          Df      AIC      BIC   Chisq   Chisq diff Df diff Pr(>Chisq)
## fit           0 1494.6 1579.0  0.000
## fit_sph_grade_sentence 2 1513.9 1592 23.262     23.262      2 8.884e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The `sphericity` argument is now a list consisting of a character vector with two elements indicating that sphericity is to be assumed for the interaction effect of grade and sentence type. The test is again significant and sphericity does not hold for the interaction effect.

Lastly, it is possible to perform an omnibus test for sphericity. This test assumes

sphericity for all effects that have two or more degrees of freedom. Note that the variances and covariances of, and between, effect variables belonging to different effects (e.g., the contrast variable  $\pi_1$  from the main effect of grade and the contrast variable  $\pi_4$  from the interaction effect) are not constrained and are freely estimated. The following test will have four degrees of freedom (i.e., the sum of the degrees of freedom of the separate tests).

```
fit_sph_all <- semnova(
  data = reading_manifest,
  id = "id",
  dv = "re_fix_time",
  within = c("grade", "sentence"),
  sphericity = list("grade", c("grade", "sentence"))
)

anova(fit, fit_sph_all)

## Chi-Squared Difference Test
##
##          Df      AIC    BIC   Chisq   Chisq diff Df diff Pr(>Chisq)
## fit          0 1494.6 1579  0.000
## fit_sph_all  4 1536.1 1608 49.511     49.511      4  4.567e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The test is also significant. We can use the test as an omnibus test to avoid multiple testing issues such as cumulative Type 1 errors. If the test is non-significant, we can assume sphericity for all effects and proceed with the more strict model which may give us a gain in statistical power (e.g., Lane, 2016; Muller and Barton, 1989). However, if the test is significant, we cannot say which of the effects violates the assumption and we have to perform both sphericity tests separately.

Note that, instead of `list(c("grade", "sentence"), "grade")`, the `sphericity` argument can also be set to `TRUE` imposing sphericity for all of the effects. The argument can also take a formula such as `FALSE ~ grade:sentence` which is equivalent to using `list(c("grade", "sentence"))`. The formula reads as “do not assume sphericity (`FALSE`) except (~) for the interaction effect of sentence and grade (`sentence:grade`)”.

## 3.2 Latent Variables

### 3.2.1 Measurement Model

The `semnova()` function can also incorporate latent variables. The input looks the same as for the very first example. Only this time, the `indicator` argument is provided, which takes the variable name of the data set that contains the names of the indicators. The variable named “`indicator`” from the example

data set contains the values `re_fix_dur`, `"ini_land_pos"` and `"fix_count"`. Furthermore, the argument `equal_resid_cov` can be used to allow for residual covariances between the indicators. Residual covariances between indicators are forced to be equal across experimental conditions. For instance, the covariances between the manifest variables of re-fixation time will be the same across all of the six experimental conditions. In this example, we use the variables re-fixation time, initial landing position and fixation count to form a latent variable that measures reading ability. By default, `semnova` uses a congeneric measurement model with strong measurement invariance and the effect-code identification scheme (Little et al., 2006) to set the scale for the latent variables (see also Langenberg, 2022, Section 2.4.1 “Defining the Model”). We will further use full information maximum likelihood for this example because of several missing values by using `missing = "fiml"`. Full information maximum likelihood is implemented in the `lavaan` package and the argument is passed on to `lavaan`.

```
fit_latent <- semnova(
  data = reading_latent,
  id = "id",
  dv = "dv",
  indicator = "indicator",
  equal_resid_cov = list("ini_land_pos", "fix_count"),
  within = c("grade", "sentence"),
  missing = "fiml"
)
summary(fit_latent)

## Fit measures:
##
## Chisq(138) = 461.551, p = 0
## CFI = 0.861
## TLI = 0.845
## RMSEA = 0.094 [0.084, 0.103]
##
## #####
##
## Main and Interaction Effects:
##          Chisq df Pr(>Chisq)
## Intercept      0.416  1     0.519
## grade         1432.334  2    <2e-16 ***
## sentence      1037.801  1    <2e-16 ***
## grade:sentence 607.323  2    <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The output now contains a section for fit indices in addition to hypothesis tests. The hypothesis tests are again significant. The fit is not very good

but the example is intended to demonstrate the basic usage of the software package. The fit can be poor for several reasons. For instance, measurement invariance across the experimental conditions may not hold. This assumption will be examined in the next section. As a consequence, we should be very cautious when interpreting parameter estimates and hypothesis tests.

In complex latent variables models, negative (residual) variances are also an issue. This can happen if the variances of the latent variables are very small as compared to residual variances (i.e., very low reliability of manifest variables) of the manifest variables or vice versa (i.e., manifest variables are highly correlated). The `reliabilities()` function serves as a diagnostic tool and prints the reliabilities and variances of both the manifest and latent variables:

```
reliabilities(fit_latent)
```

## indicator	sentence	grade	reliability	sigma_eta	lambda	eps_y
## -----	-----	-----	-----	-----	-----	-----
## fix_count	grade1	sentence	0.6363140	0.4724375	0.9959292	0.26782
## ini_land_pos	grade1	sentence	0.4610548	0.4724375	0.9304984	0.47815
## re_fix_time	grade1	sentence	0.6672992	0.4724375	1.0735725	0.27148
## fix_count	grade2	sentence	0.7140289	0.3885014	0.9959292	0.15433
## ini_land_pos	grade2	sentence	0.5883054	0.3885014	0.9304984	0.23539
## re_fix_time	grade2	sentence	0.9671821	0.3885014	1.0735725	0.01519
## fix_count	grade4	sentence	0.5436482	0.1850009	0.9959292	0.15403
## ini_land_pos	grade4	sentence	0.4849815	0.1850009	0.9304984	0.17009
## re_fix_time	grade4	sentence	0.8347808	0.1850009	1.0735725	0.04220
## fix_count	grade1	landolt	0.5371139	0.1755438	0.9959292	0.15005
## ini_land_pos	grade1	landolt	0.4112846	0.1755438	0.9304984	0.21756
## re_fix_time	grade1	landolt	0.6084315	0.1755438	1.0735725	0.13021
## fix_count	grade2	landolt	0.4555433	0.1123143	0.9959292	0.13314
## ini_land_pos	grade2	landolt	0.3650680	0.1123143	0.9304984	0.16912
## re_fix_time	grade2	landolt	0.4975977	0.1123143	1.0735725	0.13069
## fix_count	grade4	landolt	0.3834018	0.0987020	0.9959292	0.15744
## ini_land_pos	grade4	landolt	0.3831102	0.0987020	0.9304984	0.13760
## re_fix_time	grade4	landolt	0.3894837	0.0987020	1.0735725	0.17831

“sigma\_eta” contains the model implied variances of the latent dependent variables  $\eta$  (i.e., the variances are fixed to zero in latent repeated measures ANOVA [L-RM-ANOVA] and can be calculated from the variances of the  $\pi$  variables). “lambda” contains the factor loadings of the manifest variables on the latent variables. “eps\_y” contains the residual variances of the manifest variables. We see that none of the residual variances is negative. The smallest reliability is .37 which is rather low. In contrast, the highest reliability is .97 which is very high. The analysis serves as an example and we will not further examine the reliabilities and residual variances. In practice, however, we should further investigate possible reasons for the poor fit.

Figure 3.3 shows a path diagram of the model including all of the parameters.

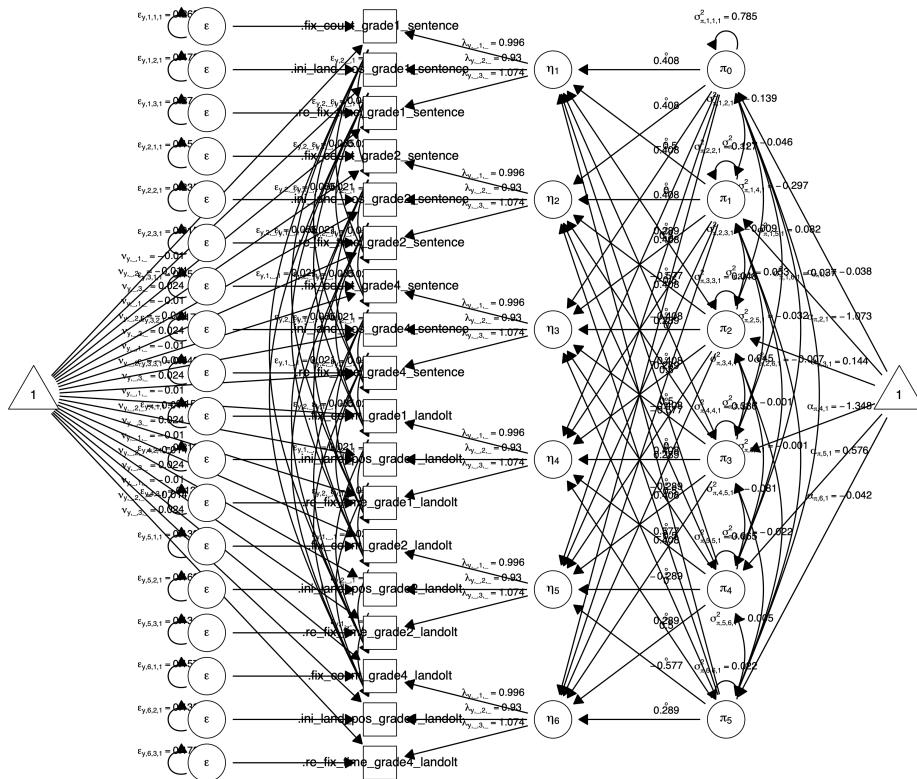


Figure 3.3: Path diagram created by `plot(fit_latent, what = "all")` for a  $2 \times 3$  (sentence  $\times$  grade) within-subjects design with the latent dependent variable reading ability measured by the three manifest variables re-fixation time, initial landing position and fixation count.

### 3.2.2 Measurement Invariance

Measurement invariance is an important assumption when using latent variables. This assumption is oftentimes implicitly assumed when using traditional repeated measures (RM-ANOVA). RM-ANOVA cannot handle latent variables and indicators are averaged to manifest composite or sum scores in order to be able to use RM-ANOVA. This approach implicitly assumes a parallel measurement model across the six experimental conditions but the assumption is not explicitly tested. In order to be able to compare the means of latent variables, we need strong measurement invariance (see Langenberg, 2022, Section 2.4 “Latent Repeated Measures ANOVA (L-RM-ANOVA)”), that is, equal loadings and equal intercepts across the experimental conditions. With *semnova*, we can actually test for measurement invariance. As described in the previous example, *semnova* uses a congeneric measurement model. The *invariance\_within* argument can be set to either of “*configural*” (equal model structure), “*weak*” (equal loadings), “*strong*” (equal loadings and intercepts), or “*strict*” (equal loadings, intercepts and residual variances):

```
fit_latent_configMI <- semnova(
  data = reading_latent,
  id = "id",
  dv = "dv",
  indicator = "indicator",
  equal_resid_cov = list("ini_land_pos", "fix_count"),
  within = c("grade", "sentence"),
  missing = "fiml",
  invariance_within = "configural"
)

fit_latent_weakMI <- semnova(
  data = reading_latent,
  id = "id",
  dv = "dv",
  indicator = "indicator",
  equal_resid_cov = list("ini_land_pos", "fix_count"),
  within = c("grade", "sentence"),
  missing = "fiml",
  invariance_within = "weak"
)

fit_latent_strongMI <- semnova(
  data = reading_latent,
  id = "id",
  dv = "dv",
  indicator = "indicator",
  equal_resid_cov = list("ini_land_pos", "fix_count"),
```

```

within = c("grade", "sentence"),
missing = "fiml",
invariance_within = "strong"
)

fit_latent_strictMI <- semnova(
  data = reading_latent,
  id = "id",
  dv = "dv",
  indicator = "indicator",
  equal_resid_cov = list("ini_land_pos", "fix_count"),
  within = c("grade", "sentence"),
  missing = "fiml",
  invariance_within = "strict"
)

anova(
  fit_latent_configMI,
  fit_latent_weakMI,
  fit_latent_strongMI,
  fit_latent_strictMI
)

## Chi-Squared Difference Test
##
##          Df      AIC      BIC   Chisq Chisq diff Df diff Pr(>Chisq)
## fit_latent_configMI 118 5467.5 5722.2 248.11
## fit_latent_weakMI   128 5579.8 5798.6 380.44    132.332     10 < 2.2e-16 ***
## fit_latent_strongMI 138 5640.9 5823.9 461.55    81.114     10 3.036e-13 ***
## fit_latent_strictMI 153 5875.1 6004.2 725.68   264.131     15 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Using the `anova()` function, we can compare the four models. All tests are highly significant indicating that they do not fit equally well. Thus, we cannot assume any degree of measurement invariance. As a consequence, we should be very cautious when interpreting parameter estimates and hypothesis tests.



# Chapter 4

## Between-Subjects Designs

The `semnova()` function can also be used with between-subjects factors and covariates. The `between` argument takes a vector of between-subjects factors (i.e., grouping variables). We estimate the same manifest model as in the very first example with the additional between-subjects factor `sex`. Using the `between` argument will cause `semnova` to estimate a multi-group model:

```
fit_btwn <- semnova(
  data = reading_manifest,
  id = "id",
  dv = "re_fix_time",
  within = c("grade", "sentence"),
  between = "sex"
)

summary(fit_btwn)

## Main and Interaction Effects:
##                               Chisq df Pr(>Chisq)
## Intercept                  1.505  1    0.220
## grade                      354.755 2    <2e-16 ***
## sentence                   381.882 1    <2e-16 ***
## grade:sentence              84.370  2    <2e-16 ***
## sex                         0.192  1    0.661
## grade:sex                   1.969  2    0.374
## sentence:sex                1.800  1    0.180
## grade:sentence:sex          1.959  2    0.376
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

When including between-subjects effects, hypothesis tests correspond to Type

III sums of squares in ANOVA. Type III sums of squares do not take unbalanced group sizes into account as apposed to average effects (e.g., Gräfe et al., 2022). Tests for average effects remain work in progress and may be included in future releases.

Main and interaction effects for the within-subjects factors are again significant. Main and interaction effects involving the between-subjects factor sex are non-significant indicating that no differences exist across genders.

## 4.1 Latent Variables

### 4.1.1 Measurement Model

The multi-group approach can also be used with latent variables. The command is identical to the previous purely within-subjects example that includes latent variables, except for the additional `between` argument.

```
fit_btwn_latent <- semnova(
  data = reading_latent,
  id = "id",
  dv = "dv",
  indicator = "indicator",
  equal_resid_cov = list("ini_land_pos", "fix_count"),
  between = "sex",
  within = c("grade", "sentence"),
  missing = "fiml"
)

summary(fit_btwn_latent)

## Fit measures:
##
## Chisq(280) = 693.254, p = 0
## CFI = 0.833
## TLI = 0.817
## RMSEA = 0.105 [0.095, 0.115]
##
## #####
## 
## Main and Interaction Effects:
##          Chisq df Pr(>Chisq)
## Intercept      0.035  1     0.852
## grade         765.984  2    <2e-16 ***
## sentence       651.753  1    <2e-16 ***
## grade:sentence 310.771  2    <2e-16 ***
## sex            0.253  1     0.615
## grade:sex      1.446  2     0.485
```

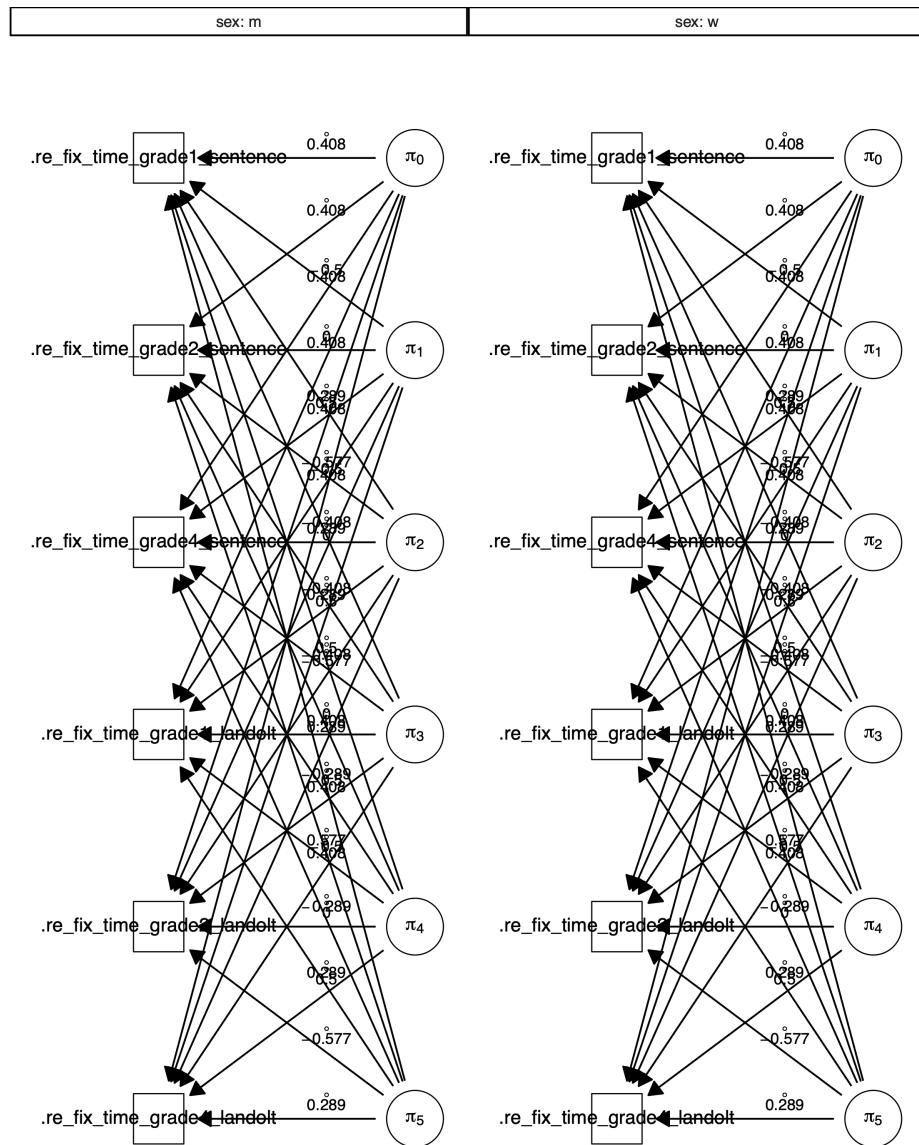


Figure 4.1: Path diagram created by `plot(fit_btw)` for a  $2 \times 3 \times 2$  (sentence  $\times$  grade  $\times$  sex) mixed design design with the manifest dependent variable re-fixation time.

```

## sentence:sex      1.469 1     0.226
## grade:sentence:sex 1.993 2     0.369
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

The model fit is again not very good. Hypothesis tests show the same pattern as for the model using only the manifest variable re-fixation time. A path diagram of the model is shown in Figure 4.2.

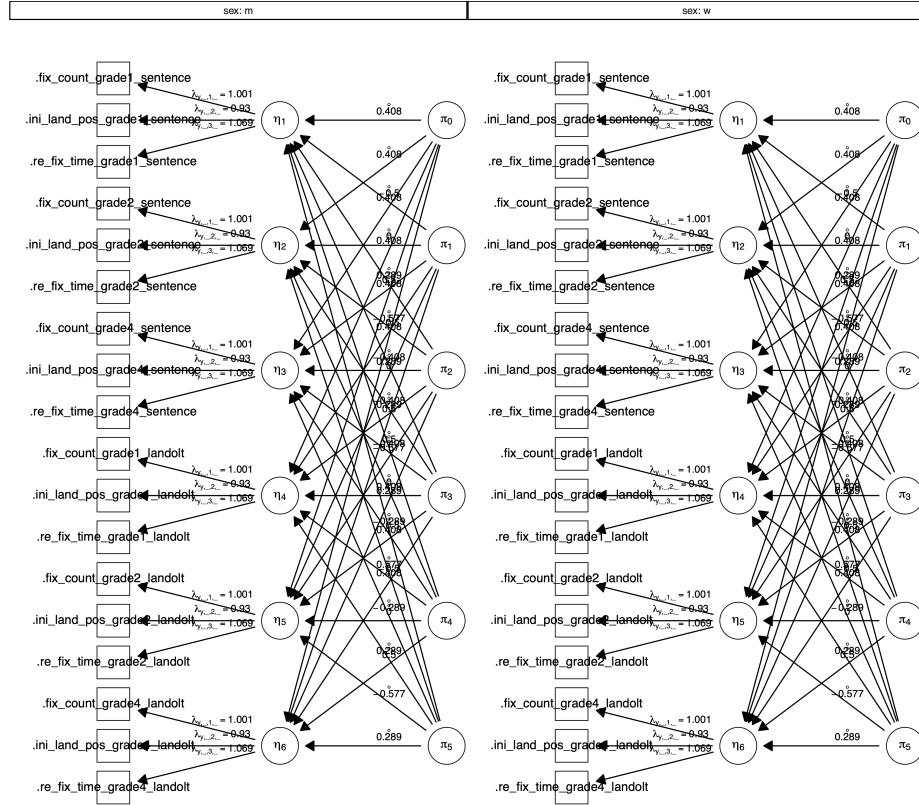


Figure 4.2: Path diagram created by `plot(fit_btwn_latent)` for a  $2 \times 3 \times 2$  (sentence  $\times$  grade  $\times$  sex) mixed design with the latent dependent variable reading ability measured by the three manifest variables re-fixation time, initial landing position and fixation count.

#### 4.1.2 Measurement Invariance

Measurement invariance must also be tested across groups. In order to be able to compare means between groups, strong measurement invariance is needed (i.e., equal loadings and intercepts between groups). This can be done by sequentially comparing models that assume different degrees of measurement invariance (i.e.,

configural, weak, strong, or strict). The `invariance_between` argument can be set to "configural", "weak", "strong", or "strict".

```
fit_btwn_latent_configMI <- semnova(
  data = reading_latent,
  id = "id",
  dv = "dv",
  indicator = "indicator",
  equal_resid_cov = list("ini_land_pos", "fix_count"),
  between = "sex",
  within = c("grade", "sentence"),
  missing = "fiml",
  invariance_between = "configural"
)

fit_btwn_latent_weakMI <- semnova(
  data = reading_latent,
  id = "id",
  dv = "dv",
  indicator = "indicator",
  equal_resid_cov = list("ini_land_pos", "fix_count"),
  between = "sex",
  within = c("grade", "sentence"),
  missing = "fiml",
  invariance_between = "weak"
)

fit_btwn_latent_strongMI <- semnova(
  data = reading_latent,
  id = "id",
  dv = "dv",
  indicator = "indicator",
  equal_resid_cov = list("ini_land_pos", "fix_count"),
  between = "sex",
  within = c("grade", "sentence"),
  missing = "fiml",
  invariance_between = "strong"
)

fit_btwn_latent_strictMI <- semnova(
  data = reading_latent,
  id = "id",
  dv = "dv",
  indicator = "indicator",
  equal_resid_cov = list("ini_land_pos", "fix_count"),
  between = "sex",
```

```

within = c("grade", "sentence"),
missing = "fiml",
invariance_between = "strict"
)

anova(
  fit_bt_w_latent_configMI,
  fit_bt_w_latent_weakMI,
  fit_bt_w_latent_strongMI,
  fit_bt_w_latent_strictMI
)

## Chi-Squared Difference Test
##
##          Df      AIC      BIC   Chisq Chisq diff Df diff Pr(>Chisq)
## fit_bt_w_latent_configMI 272 5636.4 6016.6 686.14
## fit_bt_w_latent_weakMI    276 5633.9 5999.8 691.62      5.487     4  0.2408
## fit_bt_w_latent_strongMI 280 5627.5 5979.1 693.25      1.630     4  0.8034
## fit_bt_w_latent_strictMI 298 5677.5 5964.5 779.30     86.043    18 7.356e-11
##
## fit_bt_w_latent_configMI
## fit_bt_w_latent_weakMI
## fit_bt_w_latent_strongMI
## fit_bt_w_latent_strictMI ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

The fit of the first model is not very good. These results are in line with the results for the latent model from the previous examples. However, we can see that weak and strong measurement invariance explain the data equally well as compared to configural measurement invariance. The strict measurement invariance model, however, differs in terms of fit as compared to the other three models.

# Chapter 5

## Covariates

The `semnova()` function can further incorporate manifest and latent covariates. The `covariates` argument takes a list of character vectors representing the variable names of covariates. If a character vector from the list consists of only one element, the covariate is treated as manifest, and as latent otherwise. If a manifest covariate points to a variable that contains a factor or characters, the variable will be dummy coded and included in the model. Interactions between manifest covariates will also be included. Interactions between latent covariates are not possible at this moment. Manifest variables measuring latent covariates must further be numerical.

In the next example, we estimate a  $2 \times 3$  within-subjects design for re-fixation time and include sex as a dummy-coded covariate and the latent covariate child behavior measured at the end of the study by the two scales *internalizing* and *externalizing* behavior. The scales are taken from the child behavior checklist (CBLB, Achenbach, 1991).

```
fit_covariates <- semnova(
  data = reading_manifest,
  id = "id",
  dv = "re_fix_time",
  within = c("grade", "sentence"),
  covariates = list("sex", CBCL = c("cbcl_int_verh_rw_t5", "cbcl_ext_verh_rw_t5"))
)

summary(fit_covariates)

## Fit measures:
##
## Chisq(9) = 25.066, p = 0.003
## CFI = 0.962
```

```

## TLI = 0.847
## RMSEA = 0.107 [0.059, 0.158]
##
## #####
## Main and Interaction Effects:
##          Chisq df Pr(>Chisq)
## Intercept      1.105  1    0.293
## grade         213.637  2   <2e-16 ***
## sentence      129.699  1   <2e-16 ***
## grade:sentence  57.760  2   <2e-16 ***
## sex            0.002  1    0.964
## grade:sex      2.263  2    0.323
## sentence:sex    2.620  1    0.106
## grade:sentence:sex 2.058  2    0.357
## CBCL           0.042  1    0.837
## grade:CBCL     3.674  2    0.159
## sentence:CBCL   0.257  1    0.612
## grade:sentence:CBCL 2.648  2    0.266
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Residual covariances between re-fixation time and the internalizing and externalizing child behavior are fixed to zero which explains the nine degrees of freedom. The fit is not perfect but can be considered satisfactory because the confidence interval of the root mean squared error of approximation (RMSEA) includes the established threshold of 0.06 (e.g., tze Hu and Bentler, 1999). Hypothesis tests for within-subjects factors are again significant. Effects involving the covariate sex are non-significant which is in line with the multi-group model from the previous sections. Effects involving child behavior are also non-significant. That is, child behavior does neither predict differences between genders nor does it predict differences in re-fixation time between grades or sentence types. Figure 5.1 shows the path diagram of the model.

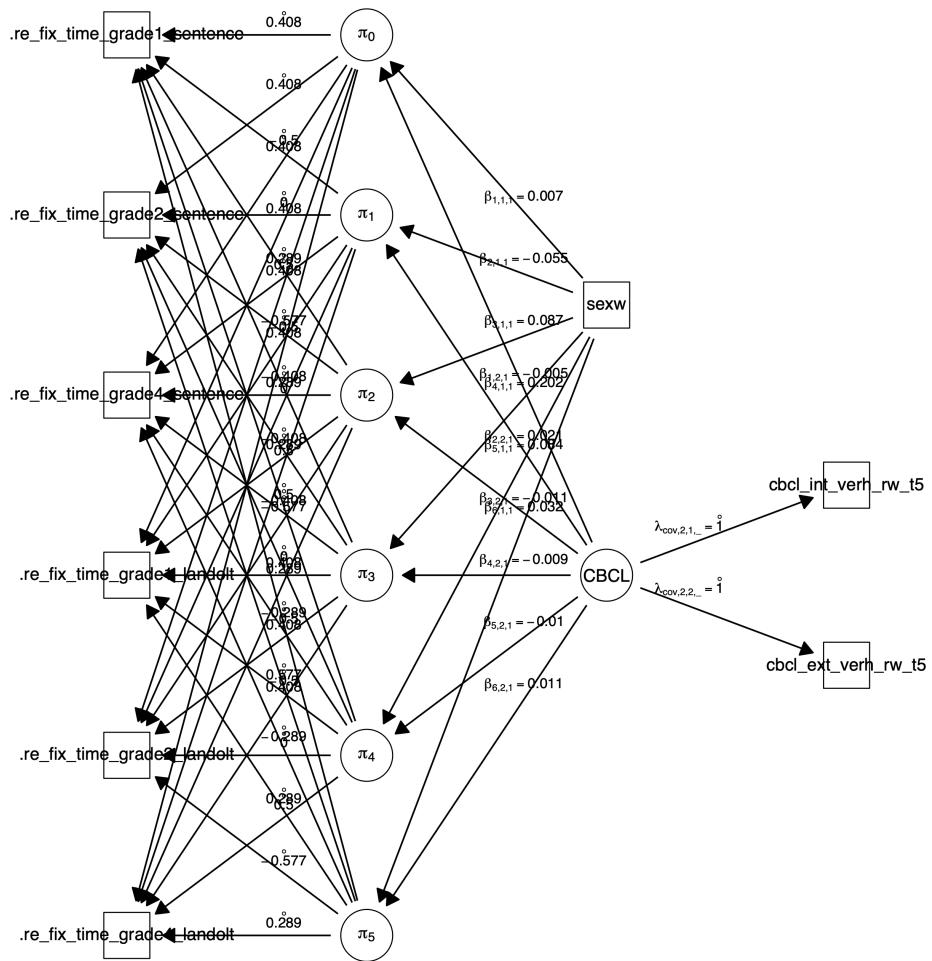


Figure 5.1: Path diagram created by `plot(fit_covariates)` for a *2imes3* (sentence *imes* grade) within-subjects design with the manifest dependent variable re-fixation time, the manifest categorical covariate sex, and the latent covariate CBCL measured by the two manifest variables internalizing and externalizing child behavior.



# Chapter 6

## Power Analysis

The *semnova* package enables the researcher to perform power calculations for within- and between-subjects designs of any complexity. The `power_script_semnova()` function creates a script template making it very easy to specify population parameters in form of *lavaan* model syntax. The `n_within` argument takes an integer vector whose entries each represent a within-subjects factor with the number indicating the number of levels. `n_between` takes the same input for the between-subjects design. `n_indicator` takes a single integer representing the number of indicators. `n_manifest_covariates` takes the same input as `n_within` and `n_between`. However, if an entry equals one, the covariate is assumed to be continuous; if the entry is greater than or equal to two the covariate is assumed to be categorical with the number indicating the number of levels. `n_latent_covariates` take a vector of integers each representing a latent variable with the number indicating the number of manifest variables that measure the latent variable. Latent covariates are always assumed to be continuous. The `sphericity` argument takes a list of integer vectors and can be used to impose sphericity on effect variables of main or interaction effects. Each vector contains indices that refer to the within-subjects factors specified by `n_within`. `out_file` takes a path that the script file will be saved to.

What follows is an example for a  $2 \times 3 \times 2 \times 2$  (within  $\times$  within  $\times$  between  $\times$  between) mixed design with a latent dependent variable measured by three manifest variables, one continuous manifest covariate and one latent covariate measured by two manifest variables.

```
power_script_semnova(  
  n_within = c(2,3),  
  n_between = c(2,2),  
  n_indicator = 3,  
  n_manifest_covariate = c(1),
```

```

n_latent_covariate = c(2),
out_file = "power_script.R",
sphericity = list(c(1,2))
)

```

The function will automatically name the factors, factor levels and covariates. Alternatively, the names can be given manually. The following code will produce exactly the same result as the previous example.

```

power_script_semnova(
  within = list(c(W = c("W1_1", "W1_2"), W2 = c("W2_1", "W2_2", "W2_3"))),
  between = list(c(B = c("B1_1", "B1_2"), B2 = c("B2_1", "B2_2"))),
  indicator = c("Y1", "Y2", "Y3"),
  manifest_covariate = c("covariate1"),
  latent_covariate = list(covariate2 = c("covariate2_1", "covariate2_1")),
  out_file = "power_script.R",
  sphericity = list(c("W1", "W2"))
)

```

Within-subjects factors are automatically given the prefix "W", between subjects factors have the prefix "B", covariates begin with the word "covariate", indicators of the dependent variables have the prefix "Y", indicators of the covariates start with the covariate name.

The function `power_script_semnova()` will create a script file that consists of three main parts. First, the script contains a *lavaan* model syntax that represents the specified model. The user has to replace parameter labels in the model syntax by population parameters. Second, the script contains a call to the `power_analysis_semnova()` function which will perform the power analysis and save the results to an object called `power`. The function takes the same input that was used to create the script file plus three additional parameters: `data_syntax` takes the lavaan model syntax containing the population parameters; `sample_size` takes a vector of sample sizes for the groups specified by the `n_between` or `between` (the length of the vector must be one or equal to the number of groups); `replications` takes the number of replications for the power analysis. Third, the three functions `summary()`, `print()`, and `plot()` are suggested to extract information from the resulting object. The following code is the result from the previous call to `power_script_semnova` saved to the specified output file.

```

data_syntax <- "
[...]
"

power <- power_analysis_semnova(
  n_within = c(2, 3),
  n_between = c(2, 2),

```

```

n_indicator = 3,
n_manifest_covariates = 1,
n_latent_covariates = c(2),
sphericity = list(2),
data_syntax = data_syntax,
sample_size = c(100, 100, 100, 100),
replications = 500
)

# hypothesis tests
summary(power)

# parameters
print(power)

# path diagram
plot(power)

```

The contents of the `data_syntax` variable are omitted due to the large number of lines. The complete syntax can be found in Appendix A.

Appendix B further contains a copy of the previously created power script filled with example population parameters. Means of latent variables and regression coefficients are set to 0.1. Variances of latent variables are set to 1 and covariances to 0.5. Loadings of manifest variables are set to 1 and intercepts to 0. Residual variances are 0.4 and residual covariances are set to 0.

Depending on the number of replications and the complexity of the model, the power analysis may need some time to finish. After the analysis, the `summary(power)` function can be used to print a table that contains the power for the main and interaction effects of the model. In the following output, the results are truncated after the sixth hypothesis.

```

## Table: Hypotheses
##
##   hypothesis      power
##   ----- -----
##   B1              0.0456349
##   B1:B2           0.0456349
##   B1:B2:covariate1 0.0357143
##   B1:B2:covariate2 0.0634921
##   B1:covariate1    0.0515873
##   B1:covariate2    0.0595238
##   [...]

```

We can further use the `print(power)` function to print the power and descriptive statistics for each of the model parameters separated by between-subjects factors

(i.e., groups).

## parameter	mean	sd	pctNA	power	pct2.5	
## alpha_{pi,1,1}	0.0958489	0.1104171	0	0.1686508	-0.1392029	0
## alpha_{pi,2,1}	0.0977477	0.1118119	0	0.1666667	-0.1278233	0
## alpha_{pi,3,1}	0.0982346	0.1079209	0	0.1488095	-0.1170303	0
## alpha_{pi,4,1}	0.0994142	0.1099875	0	0.1428571	-0.1081087	0
## alpha_{pi,5,1}	0.1043942	0.1102061	0	0.1865079	-0.1095376	0
## alpha_{pi,6,1}	0.0997237	0.1035536	0	0.1329365	-0.0902241	0
## [...]						

Lastly, the `plot(power)` function plots a path diagram for the model. The diagram will include parameter labels but no estimates.

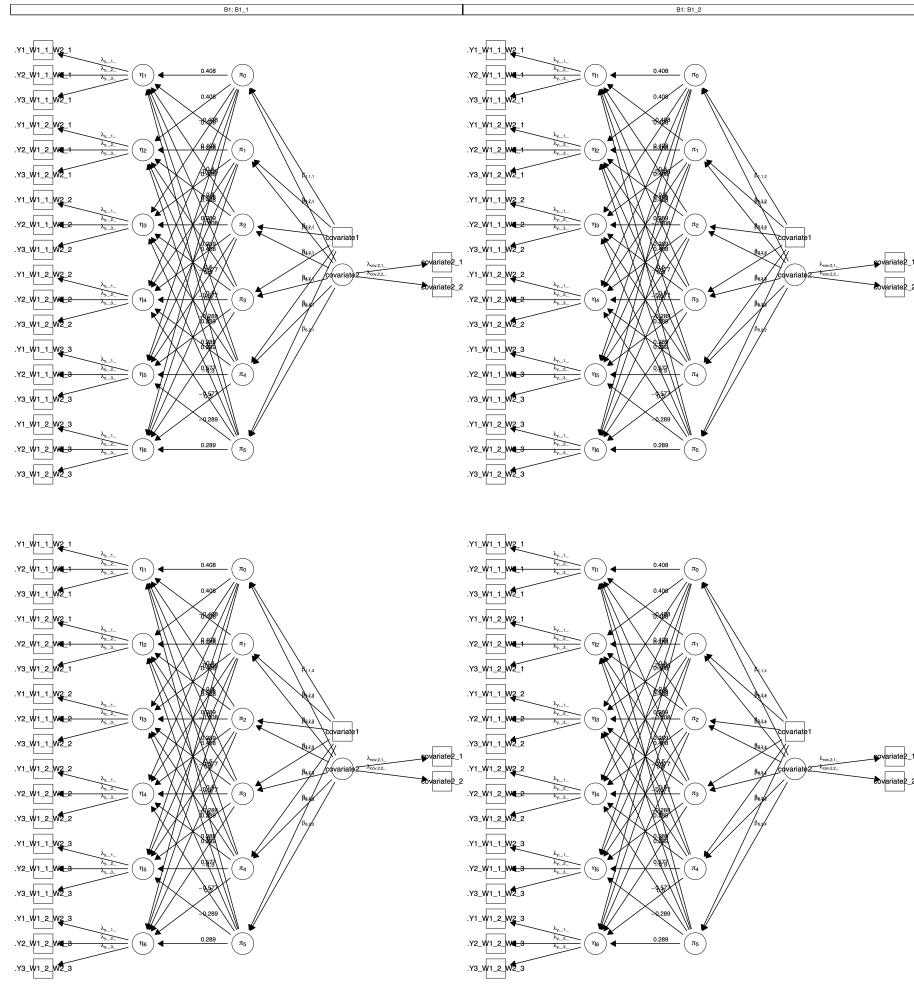


Figure 6.1: Path diagram created by `plot(power)` for a  $2 \times 3 \times 2 \times 2$  (within  $i_{mes}$  within  $i_{mes}$  between  $i_{mes}$  between  $i_{mes}$ ) mixed design with a latent dependent variable, one manifest covariates, and one latent covariate measured by two manifest variables.



## Chapter 7

# Further Arguments

The `semnova()` function accepts the following additional arguments which have not been described in this tutorial.

---

<code>contrasts_arg</code>	Named list that specifies the contrast coding scheme. E.g.: <code>contrasts_arg = list(grade = "contr.poly")</code>
<code>variance_homogeneity</code>	Forces equal variances across groups. Can be used to test the assumption of variance homogeneity. Default is FALSE. E.g.: <code>variance_homogeneity = FALSE</code>
<code>lv_scaling</code>	Specifies the latent variable identification scheme. Possible schemes are effect-code identification ("effect", default) and referent-indicator identification ("referent"). E.g.: <code>lv_scaling = "effect"</code>
<code>compound_symmetry</code>	Forces the variances and covariances to be equal across the dependent variables. Can be used to test the assumption of compound symmetry. Default is FALSE. E.g.: <code>compound_symmetry = FALSE</code>
<code>normalize_contrasts</code>	Scales the length of the contrasts to one. Default is FALSE. E.g.: <code>normalize_contrasts = FALSE</code>
<code>append</code>	Appends the given character to the *lavaan* syntax. E.g.: <code>append = "any lavaan syntax"</code>

---

## Chapter 8

# Further Developments

The *semnova* package offers a wide range of features but remains work in progress. The road map for future developments is long and includes:

1. Graphical user interface
2. Plotting features
3. Output for means and variances estimates
4. Output for effect size estimates
5. Average and conditional effects
6. Ordinal outcomes
7. Multilevel extension
8. Bayesian estimation
9. Different test statistics (e.g., likelihood ratio test, approximate  $F$ -test)



## Appendix A

# Power Script Example

```
data_syntax <- "
.eta1 =~
  c(.lambda_y_1_, .lambda_y_1_, .lambda_y_1_, .lambda_y_1_)*.Y1_W1_1_W2_1 +
  c(.lambda_y_2_, .lambda_y_2_, .lambda_y_2_, .lambda_y_2_)*.Y2_W1_1_W2_1 +
  c(.lambda_y_3_, .lambda_y_3_, .lambda_y_3_, .lambda_y_3_)*.Y3_W1_1_W2_1
.eta2 =~
  c(.lambda_y_1_, .lambda_y_1_, .lambda_y_1_, .lambda_y_1_)*.Y1_W1_2_W2_1 +
  c(.lambda_y_2_, .lambda_y_2_, .lambda_y_2_, .lambda_y_2_)*.Y2_W1_2_W2_1 +
  c(.lambda_y_3_, .lambda_y_3_, .lambda_y_3_, .lambda_y_3_)*.Y3_W1_2_W2_1
.eta3 =~
  c(.lambda_y_1_, .lambda_y_1_, .lambda_y_1_, .lambda_y_1_)*.Y1_W1_1_W2_2 +
  c(.lambda_y_2_, .lambda_y_2_, .lambda_y_2_, .lambda_y_2_)*.Y2_W1_1_W2_2 +
  c(.lambda_y_3_, .lambda_y_3_, .lambda_y_3_, .lambda_y_3_)*.Y3_W1_1_W2_2
.eta4 =~
  c(.lambda_y_1_, .lambda_y_1_, .lambda_y_1_, .lambda_y_1_)*.Y1_W1_2_W2_2 +
  c(.lambda_y_2_, .lambda_y_2_, .lambda_y_2_, .lambda_y_2_)*.Y2_W1_2_W2_2 +
  c(.lambda_y_3_, .lambda_y_3_, .lambda_y_3_, .lambda_y_3_)*.Y3_W1_2_W2_2
.eta5 =~
  c(.lambda_y_1_, .lambda_y_1_, .lambda_y_1_, .lambda_y_1_)*.Y1_W1_1_W2_3 +
  c(.lambda_y_2_, .lambda_y_2_, .lambda_y_2_, .lambda_y_2_)*.Y2_W1_1_W2_3 +
  c(.lambda_y_3_, .lambda_y_3_, .lambda_y_3_, .lambda_y_3_)*.Y3_W1_1_W2_3
.eta6 =~
  c(.lambda_y_1_, .lambda_y_1_, .lambda_y_1_, .lambda_y_1_)*.Y1_W1_2_W2_3 +
  c(.lambda_y_2_, .lambda_y_2_, .lambda_y_2_, .lambda_y_2_)*.Y2_W1_2_W2_3 +
  c(.lambda_y_3_, .lambda_y_3_, .lambda_y_3_, .lambda_y_3_)*.Y3_W1_2_W2_3
.pi1 =~
  c(1, 1, 1, 1)*.eta1 +
  c(1, 1, 1, 1)*.eta2 +
  c(1, 1, 1, 1)*.eta3 +
```

```

c(1, 1, 1, 1)*.eta4 +
c(1, 1, 1, 1)*.eta5 +
c(1, 1, 1, 1)*.eta6
.pi2 =~
c(-0.707106781186547, -0.707106781186547, -0.707106781186547, -0.707106781186547)*
c(0.707106781186547, 0.707106781186547, 0.707106781186547, 0.707106781186547)*.eta1
c(-0.707106781186547, -0.707106781186547, -0.707106781186547, -0.707106781186547)*
c(0.707106781186547, 0.707106781186547, 0.707106781186547, 0.707106781186547)*.eta2
c(-0.707106781186547, -0.707106781186547, -0.707106781186547, -0.707106781186547)*
c(0.707106781186547, 0.707106781186547, 0.707106781186547, 0.707106781186547)*.eta3
.pi3 =~
c(-0.707106781186548, -0.707106781186548, -0.707106781186548, -0.707106781186548)*
c(-0.707106781186548, -0.707106781186548, -0.707106781186548, -0.707106781186548)*
c(0, 0, 0, 0)*.eta3 +
c(0, 0, 0, 0)*.eta4 +
c(0.707106781186547, 0.707106781186547, 0.707106781186547, 0.707106781186547)*.eta1
c(0.707106781186547, 0.707106781186547, 0.707106781186547, 0.707106781186547)*.eta2
.pi4 =~
c(0.408248290463863, 0.408248290463863, 0.408248290463863, 0.408248290463863)*.eta1
c(0.408248290463863, 0.408248290463863, 0.408248290463863, 0.408248290463863)*.eta2
c(-0.816496580927726, -0.816496580927726, -0.816496580927726, -0.816496580927726)*
c(-0.816496580927726, -0.816496580927726, -0.816496580927726, -0.816496580927726)*
c(0.408248290463863, 0.408248290463863, 0.408248290463863, 0.408248290463863)*.eta1
c(0.408248290463863, 0.408248290463863, 0.408248290463863, 0.408248290463863)*.eta2
.pi5 =~
c(0.5, 0.5, 0.5, 0.5)*.eta1 +
c(-0.5, -0.5, -0.5, -0.5)*.eta2 +
c(0, 0, 0, 0)*.eta3 +
c(0, 0, 0, 0)*.eta4 +
c(-0.5, -0.5, -0.5, -0.5)*.eta5 +
c(0.5, 0.5, 0.5, 0.5)*.eta6
.pi6 =~
c(-0.288675134594813, -0.288675134594813, -0.288675134594813, -0.288675134594813)*
c(0.288675134594813, 0.288675134594813, 0.288675134594813, 0.288675134594813)*.eta1
c(0.577350269189626, 0.577350269189626, 0.577350269189626, 0.577350269189626)*.eta2
c(-0.577350269189626, -0.577350269189626, -0.577350269189626, -0.577350269189626)*
c(-0.288675134594813, -0.288675134594813, -0.288675134594813, -0.288675134594813)*
c(0.288675134594813, 0.288675134594813, 0.288675134594813, 0.288675134594813)*.eta1
covariate2 =~
c(.lambda_cov_2_1_, .lambda_cov_2_1_, .lambda_cov_2_1_, .lambda_cov_2_1_)*covariate1
c(.lambda_cov_2_2_, .lambda_cov_2_2_, .lambda_cov_2_2_, .lambda_cov_2_2_)*covariate2
.pi1 ~
c(.beta_1_1_1, .beta_1_1_2, .beta_1_1_3, .beta_1_1_4)*covariate1 +
c(.beta_1_2_1, .beta_1_2_2, .beta_1_2_3, .beta_1_2_4)*covariate2
.pi2 ~

```

```

    c(.beta_2_1_1, .beta_2_1_2, .beta_2_1_3, .beta_2_1_4)*covariate1 +
    c(.beta_2_2_1, .beta_2_2_2, .beta_2_2_3, .beta_2_2_4)*covariate2
.pi3 ~
    c(.beta_3_1_1, .beta_3_1_2, .beta_3_1_3, .beta_3_1_4)*covariate1 +
    c(.beta_3_2_1, .beta_3_2_2, .beta_3_2_3, .beta_3_2_4)*covariate2
.pi4 ~
    c(.beta_4_1_1, .beta_4_1_2, .beta_4_1_3, .beta_4_1_4)*covariate1 +
    c(.beta_4_2_1, .beta_4_2_2, .beta_4_2_3, .beta_4_2_4)*covariate2
.pi5 ~
    c(.beta_5_1_1, .beta_5_1_2, .beta_5_1_3, .beta_5_1_4)*covariate1 +
    c(.beta_5_2_1, .beta_5_2_2, .beta_5_2_3, .beta_5_2_4)*covariate2
.pi6 ~
    c(.beta_6_1_1, .beta_6_1_2, .beta_6_1_3, .beta_6_1_4)*covariate1 +
    c(.beta_6_2_1, .beta_6_2_2, .beta_6_2_3, .beta_6_2_4)*covariate2
.eta1 ~~
    c(0, 0, 0, 0)*.eta1 +
    c(0, 0, 0, 0)*.eta2 +
    c(0, 0, 0, 0)*.eta3 +
    c(0, 0, 0, 0)*.eta4 +
    c(0, 0, 0, 0)*.eta5 +
    c(0, 0, 0, 0)*.eta6
.eta2 ~~
    c(0, 0, 0, 0)*.eta2 +
    c(0, 0, 0, 0)*.eta3 +
    c(0, 0, 0, 0)*.eta4 +
    c(0, 0, 0, 0)*.eta5 +
    c(0, 0, 0, 0)*.eta6
.eta3 ~~
    c(0, 0, 0, 0)*.eta3 +
    c(0, 0, 0, 0)*.eta4 +
    c(0, 0, 0, 0)*.eta5 +
    c(0, 0, 0, 0)*.eta6
.eta4 ~~
    c(0, 0, 0, 0)*.eta4 +
    c(0, 0, 0, 0)*.eta5 +
    c(0, 0, 0, 0)*.eta6
.eta5 ~~
    c(0, 0, 0, 0)*.eta5 +
    c(0, 0, 0, 0)*.eta6
.eta6 ~~
    c(0, 0, 0, 0)*.eta6
.pi1 ~~
    c(.sigma_pi_1_1_1, .sigma_pi_1_1_2, .sigma_pi_1_1_3, .sigma_pi_1_1_4)*.pi1 +
    c(.sigma_pi_1_2_1, .sigma_pi_1_2_2, .sigma_pi_1_2_3, .sigma_pi_1_2_4)*.pi2 +
    c(.sigma_pi_1_3_1, .sigma_pi_1_3_2, .sigma_pi_1_3_3, .sigma_pi_1_3_4)*.pi3 +

```

```

c(.sigma_pi_1_4_1, .sigma_pi_1_4_2, .sigma_pi_1_4_3, .sigma_pi_1_4_4)*.pi4 +
c(.sigma_pi_1_5_1, .sigma_pi_1_5_2, .sigma_pi_1_5_3, .sigma_pi_1_5_4)*.pi5 +
c(.sigma_pi_1_6_1, .sigma_pi_1_6_2, .sigma_pi_1_6_3, .sigma_pi_1_6_4)*.pi6
.pi2 ~~
  c(.sigma_pi_2_2_1, .sigma_pi_2_2_2, .sigma_pi_2_2_3, .sigma_pi_2_2_4)*.pi2 +
  c(.sigma_pi_2_3_1, .sigma_pi_2_3_2, .sigma_pi_2_3_3, .sigma_pi_2_3_4)*.pi3 +
  c(.sigma_pi_2_4_1, .sigma_pi_2_4_2, .sigma_pi_2_4_3, .sigma_pi_2_4_4)*.pi4 +
  c(.sigma_pi_2_5_1, .sigma_pi_2_5_2, .sigma_pi_2_5_3, .sigma_pi_2_5_4)*.pi5 +
  c(.sigma_pi_2_6_1, .sigma_pi_2_6_2, .sigma_pi_2_6_3, .sigma_pi_2_6_4)*.pi6
.pi3 ~~
  c(.sigma_pi_1_1_1, .sigma_pi_1_1_2, .sigma_pi_1_1_3, .sigma_pi_1_1_4)*.pi3 +
  c(0, 0, 0, 0)*.pi4 +
  c(.sigma_pi_3_5_1, .sigma_pi_3_5_2, .sigma_pi_3_5_3, .sigma_pi_3_5_4)*.pi5 +
  c(.sigma_pi_3_6_1, .sigma_pi_3_6_2, .sigma_pi_3_6_3, .sigma_pi_3_6_4)*.pi6
.pi4 ~~
  c(.sigma_pi_1_1_1, .sigma_pi_1_1_2, .sigma_pi_1_1_3, .sigma_pi_1_1_4)*.pi4 +
  c(.sigma_pi_4_5_1, .sigma_pi_4_5_2, .sigma_pi_4_5_3, .sigma_pi_4_5_4)*.pi5 +
  c(.sigma_pi_4_6_1, .sigma_pi_4_6_2, .sigma_pi_4_6_3, .sigma_pi_4_6_4)*.pi6
.pi5 ~~
  c(.sigma_pi_5_5_1, .sigma_pi_5_5_2, .sigma_pi_5_5_3, .sigma_pi_5_5_4)*.pi5 +
  c(.sigma_pi_5_6_1, .sigma_pi_5_6_2, .sigma_pi_5_6_3, .sigma_pi_5_6_4)*.pi6
.pi6 ~~
  c(.sigma_pi_6_6_1, .sigma_pi_6_6_2, .sigma_pi_6_6_3, .sigma_pi_6_6_4)*.pi6
.Y1_W1_1_W2_1 ~~
  c(.epsilon_y_1_1_1, .epsilon_y_1_1_2, .epsilon_y_1_1_3, .epsilon_y_1_1_4)*.Y1_W1_1
.Y1_W1_1_W2_2 ~~
  c(.epsilon_y_3_1_1, .epsilon_y_3_1_2, .epsilon_y_3_1_3, .epsilon_y_3_1_4)*.Y1_W1_1
.Y1_W1_1_W2_3 ~~
  c(.epsilon_y_5_1_1, .epsilon_y_5_1_2, .epsilon_y_5_1_3, .epsilon_y_5_1_4)*.Y1_W1_1
.Y1_W1_2_W2_1 ~~
  c(.epsilon_y_2_1_1, .epsilon_y_2_1_2, .epsilon_y_2_1_3, .epsilon_y_2_1_4)*.Y1_W1_2
.Y1_W1_2_W2_2 ~~
  c(.epsilon_y_4_1_1, .epsilon_y_4_1_2, .epsilon_y_4_1_3, .epsilon_y_4_1_4)*.Y1_W1_2
.Y1_W1_2_W2_3 ~~
  c(.epsilon_y_6_1_1, .epsilon_y_6_1_2, .epsilon_y_6_1_3, .epsilon_y_6_1_4)*.Y1_W1_2
.Y2_W1_1_W2_1 ~~
  c(.epsilon_y_1_2_1, .epsilon_y_1_2_2, .epsilon_y_1_2_3, .epsilon_y_1_2_4)*.Y2_W1_1
.Y2_W1_1_W2_2 ~~
  c(.epsilon_y_3_2_1, .epsilon_y_3_2_2, .epsilon_y_3_2_3, .epsilon_y_3_2_4)*.Y2_W1_1
.Y2_W1_1_W2_3 ~~
  c(.epsilon_y_5_2_1, .epsilon_y_5_2_2, .epsilon_y_5_2_3, .epsilon_y_5_2_4)*.Y2_W1_1
.Y2_W1_2_W2_1 ~~
  c(.epsilon_y_2_2_1, .epsilon_y_2_2_2, .epsilon_y_2_2_3, .epsilon_y_2_2_4)*.Y2_W1_2
.Y2_W1_2_W2_2 ~~
  c(.epsilon_y_4_2_1, .epsilon_y_4_2_2, .epsilon_y_4_2_3, .epsilon_y_4_2_4)*.Y2_W1_2

```

```

.Y2_W1_2_W2_3 ~
  c(.epsilon_y_6_2_1, .epsilon_y_6_2_2, .epsilon_y_6_2_3, .epsilon_y_6_2_4)*.Y2_W1_2_W2_3
.Y3_W1_1_W2_1 ~
  c(.epsilon_y_1_3_1, .epsilon_y_1_3_2, .epsilon_y_1_3_3, .epsilon_y_1_3_4)*.Y3_W1_1_W2_1
.Y3_W1_1_W2_2 ~
  c(.epsilon_y_3_3_1, .epsilon_y_3_3_2, .epsilon_y_3_3_3, .epsilon_y_3_3_4)*.Y3_W1_1_W2_2
.Y3_W1_1_W2_3 ~
  c(.epsilon_y_5_3_1, .epsilon_y_5_3_2, .epsilon_y_5_3_3, .epsilon_y_5_3_4)*.Y3_W1_1_W2_3
.Y3_W1_2_W2_1 ~
  c(.epsilon_y_2_3_1, .epsilon_y_2_3_2, .epsilon_y_2_3_3, .epsilon_y_2_3_4)*.Y3_W1_2_W2_1
.Y3_W1_2_W2_2 ~
  c(.epsilon_y_4_3_1, .epsilon_y_4_3_2, .epsilon_y_4_3_3, .epsilon_y_4_3_4)*.Y3_W1_2_W2_2
.Y3_W1_2_W2_3 ~
  c(.epsilon_y_6_3_1, .epsilon_y_6_3_2, .epsilon_y_6_3_3, .epsilon_y_6_3_4)*.Y3_W1_2_W2_3
covariate1 ~
  c(.epsilon_cov_1__1, .epsilon_cov_1__2, .epsilon_cov_1__3, .epsilon_cov_1__4)*covariate1
covariate2 ~
  c(.sigma_cov_2__1, .sigma_cov_2__2, .sigma_cov_2__3, .sigma_cov_2__4)*covariate2
covariate2_1 ~
  c(.epsilon_cov_2_1_1, .epsilon_cov_2_1_2, .epsilon_cov_2_1_3, .epsilon_cov_2_1_4)*covariate2_1
covariate2_2 ~
  c(.epsilon_cov_2_2_1, .epsilon_cov_2_2_2, .epsilon_cov_2_2_3, .epsilon_cov_2_2_4)*covariate2_2
.eta1 ~
  c(0, 0, 0, 0)*1
.eta2 ~
  c(0, 0, 0, 0)*1
.eta3 ~
  c(0, 0, 0, 0)*1
.eta4 ~
  c(0, 0, 0, 0)*1
.eta5 ~
  c(0, 0, 0, 0)*1
.eta6 ~
  c(0, 0, 0, 0)*1
.pi1 ~
  c(.alpha_pi_1_1, .alpha_pi_1_2, .alpha_pi_1_3, .alpha_pi_1_4)*1
.pi2 ~
  c(.alpha_pi_2_1, .alpha_pi_2_2, .alpha_pi_2_3, .alpha_pi_2_4)*1
.pi3 ~
  c(.alpha_pi_3_1, .alpha_pi_3_2, .alpha_pi_3_3, .alpha_pi_3_4)*1
.pi4 ~
  c(.alpha_pi_4_1, .alpha_pi_4_2, .alpha_pi_4_3, .alpha_pi_4_4)*1
.pi5 ~
  c(.alpha_pi_5_1, .alpha_pi_5_2, .alpha_pi_5_3, .alpha_pi_5_4)*1
.pi6 ~

```

```

c(.alpha_pi_6_1, .alpha_pi_6_2, .alpha_pi_6_3, .alpha_pi_6_4)*1
.Y1_W1_1_W2_1 ~
  c(.nu_y__1_, .nu_y__1_, .nu_y__1_, .nu_y__1_)*1
.Y1_W1_1_W2_2 ~
  c(.nu_y__1_, .nu_y__1_, .nu_y__1_, .nu_y__1_)*1
.Y1_W1_1_W2_3 ~
  c(.nu_y__1_, .nu_y__1_, .nu_y__1_, .nu_y__1_)*1
.Y1_W1_2_W2_1 ~
  c(.nu_y__1_, .nu_y__1_, .nu_y__1_, .nu_y__1_)*1
.Y1_W1_2_W2_2 ~
  c(.nu_y__1_, .nu_y__1_, .nu_y__1_, .nu_y__1_)*1
.Y1_W1_2_W2_3 ~
  c(.nu_y__1_, .nu_y__1_, .nu_y__1_, .nu_y__1_)*1
.Y2_W1_1_W2_1 ~
  c(.nu_y__2_, .nu_y__2_, .nu_y__2_, .nu_y__2_)*1
.Y2_W1_1_W2_2 ~
  c(.nu_y__2_, .nu_y__2_, .nu_y__2_, .nu_y__2_)*1
.Y2_W1_1_W2_3 ~
  c(.nu_y__2_, .nu_y__2_, .nu_y__2_, .nu_y__2_)*1
.Y2_W1_2_W2_1 ~
  c(.nu_y__2_, .nu_y__2_, .nu_y__2_, .nu_y__2_)*1
.Y2_W1_2_W2_2 ~
  c(.nu_y__2_, .nu_y__2_, .nu_y__2_, .nu_y__2_)*1
.Y2_W1_2_W2_3 ~
  c(.nu_y__2_, .nu_y__2_, .nu_y__2_, .nu_y__2_)*1
.Y3_W1_1_W2_1 ~
  c(.nu_y__3_, .nu_y__3_, .nu_y__3_, .nu_y__3_)*1
.Y3_W1_1_W2_2 ~
  c(.nu_y__3_, .nu_y__3_, .nu_y__3_, .nu_y__3_)*1
.Y3_W1_1_W2_3 ~
  c(.nu_y__3_, .nu_y__3_, .nu_y__3_, .nu_y__3_)*1
.Y3_W1_2_W2_1 ~
  c(.nu_y__3_, .nu_y__3_, .nu_y__3_, .nu_y__3_)*1
.Y3_W1_2_W2_2 ~
  c(.nu_y__3_, .nu_y__3_, .nu_y__3_, .nu_y__3_)*1
.Y3_W1_2_W2_3 ~
  c(.nu_y__3_, .nu_y__3_, .nu_y__3_, .nu_y__3_)*1
covariate1 ~
  c(.nu_cov_1__1, .nu_cov_1__2, .nu_cov_1__3, .nu_cov_1__4)*1
covariate2 ~
  c(.alpha_cov_2__1, .alpha_cov_2__2, .alpha_cov_2__3, .alpha_cov_2__4)*1
covariate2_1 ~
  c(.nu_cov_2_1_, .nu_cov_2_1_, .nu_cov_2_1_, .nu_cov_2_1_)*1
covariate2_2 ~
  c(.nu_cov_2_2_, .nu_cov_2_2_, .nu_cov_2_2_, .nu_cov_2_2_)*1

```

```
""

power <- power_analysis_semnova(
  n_within = c(2, 3),
  n_between = c(2, 2),
  n_indicator = 3,
  n_manifest_covariate = 1,
  n_latent_covariate = 2,
  sphericity = list(2),
  data_syntax = data_syntax,
  sample_size = c(100, 100, 100, 100),
  replications = 500
)

# hypothesis tests
summary(power)

# parameters
print(power)

# path diagram
plot(power)
```



## Appendix B

# Filled Power Script Example

```
data_syntax <- "
.eta1 =~
  c(1, 1, 1, 1)*.Y1_W1_1_W2_1 +
  c(1, 1, 1, 1)*.Y2_W1_1_W2_1 +
  c(1, 1, 1, 1)*.Y3_W1_1_W2_1
.eta2 =~
  c(1, 1, 1, 1)*.Y1_W1_2_W2_1 +
  c(1, 1, 1, 1)*.Y2_W1_2_W2_1 +
  c(1, 1, 1, 1)*.Y3_W1_2_W2_1
.eta3 =~
  c(1, 1, 1, 1)*.Y1_W1_1_W2_2 +
  c(1, 1, 1, 1)*.Y2_W1_1_W2_2 +
  c(1, 1, 1, 1)*.Y3_W1_1_W2_2
.eta4 =~
  c(1, 1, 1, 1)*.Y1_W1_2_W2_2 +
  c(1, 1, 1, 1)*.Y2_W1_2_W2_2 +
  c(1, 1, 1, 1)*.Y3_W1_2_W2_2
.eta5 =~
  c(1, 1, 1, 1)*.Y1_W1_1_W2_3 +
  c(1, 1, 1, 1)*.Y2_W1_1_W2_3 +
  c(1, 1, 1, 1)*.Y3_W1_1_W2_3
.eta6 =~
  c(1, 1, 1, 1)*.Y1_W1_2_W2_3 +
  c(1, 1, 1, 1)*.Y2_W1_2_W2_3 +
  c(1, 1, 1, 1)*.Y3_W1_2_W2_3
.pi1 =~
```

```

c(0.408248290463863, 0.408248290463863, 0.408248290463863, 0.408248290463863)*.eta1
.pi2 =~
c(-0.408248290463863, -0.408248290463863, -0.408248290463863, -0.408248290463863)*.eta1
c(0.408248290463863, 0.408248290463863, 0.408248290463863, 0.408248290463863)*.eta1
c(-0.408248290463863, -0.408248290463863, -0.408248290463863, -0.408248290463863)*.eta1
c(0.408248290463863, 0.408248290463863, 0.408248290463863, 0.408248290463863)*.eta1
c(-0.408248290463863, -0.408248290463863, -0.408248290463863, -0.408248290463863)*.eta1
c(0.408248290463863, 0.408248290463863, 0.408248290463863, 0.408248290463863)*.eta1
.pi3 =~
c(-0.5, -0.5, -0.5, -0.5)*.eta1 +
c(-0.5, -0.5, -0.5, -0.5)*.eta2 +
c(0, 0, 0, 0)*.eta3 +
c(0, 0, 0, 0)*.eta4 +
c(0.5, 0.5, 0.5, 0.5)*.eta5 +
c(0.5, 0.5, 0.5, 0.5)*.eta6
.pi4 =~
c(0.288675134594813, 0.288675134594813, 0.288675134594813, 0.288675134594813)*.eta1
c(0.288675134594813, 0.288675134594813, 0.288675134594813, 0.288675134594813)*.eta1
c(-0.577350269189626, -0.577350269189626, -0.577350269189626, -0.577350269189626)*.eta1
c(-0.577350269189626, -0.577350269189626, -0.577350269189626, -0.577350269189626)*.eta1
c(0.288675134594813, 0.288675134594813, 0.288675134594813, 0.288675134594813)*.eta1
c(0.288675134594813, 0.288675134594813, 0.288675134594813, 0.288675134594813)*.eta1
.pi5 =~
c(0.5, 0.5, 0.5, 0.5)*.eta1 +
c(-0.5, -0.5, -0.5, -0.5)*.eta2 +
c(0, 0, 0, 0)*.eta3 +
c(0, 0, 0, 0)*.eta4 +
c(-0.5, -0.5, -0.5, -0.5)*.eta5 +
c(0.5, 0.5, 0.5, 0.5)*.eta6
.pi6 =~
c(-0.288675134594813, -0.288675134594813, -0.288675134594813, -0.288675134594813)*.eta1
c(0.288675134594813, 0.288675134594813, 0.288675134594813, 0.288675134594813)*.eta1
c(0.577350269189626, 0.577350269189626, 0.577350269189626, 0.577350269189626)*.eta1
c(-0.577350269189626, -0.577350269189626, -0.577350269189626, -0.577350269189626)*.eta1
c(-0.288675134594813, -0.288675134594813, -0.288675134594813, -0.288675134594813)*.eta1
c(0.288675134594813, 0.288675134594813, 0.288675134594813, 0.288675134594813)*.eta1
covariate2 =~
c(1, 1, 1, 1)*covariate2_1 +
c(1, 1, 1, 1)*covariate2_2
.pi1 ~

```

```

    c(0.1, 0.1, 0.1, 0.1)*covariate1 +
    c(0.1, 0.1, 0.1, 0.1)*covariate2
.pi2 ~
    c(0.1, 0.1, 0.1, 0.1)*covariate1 +
    c(0.1, 0.1, 0.1, 0.1)*covariate2
.pi3 ~
    c(0.1, 0.1, 0.1, 0.1)*covariate1 +
    c(0.1, 0.1, 0.1, 0.1)*covariate2
.pi4 ~
    c(0.1, 0.1, 0.1, 0.1)*covariate1 +
    c(0.1, 0.1, 0.1, 0.1)*covariate2
.pi5 ~
    c(0.1, 0.1, 0.1, 0.1)*covariate1 +
    c(0.1, 0.1, 0.1, 0.1)*covariate2
.pi6 ~
    c(0.1, 0.1, 0.1, 0.1)*covariate1 +
    c(0.1, 0.1, 0.1, 0.1)*covariate2
.eta1 ~-
    c(0, 0, 0, 0)*.eta1 +
    c(0, 0, 0, 0)*.eta2 +
    c(0, 0, 0, 0)*.eta3 +
    c(0, 0, 0, 0)*.eta4 +
    c(0, 0, 0, 0)*.eta5 +
    c(0, 0, 0, 0)*.eta6
.eta2 ~-
    c(0, 0, 0, 0)*.eta2 +
    c(0, 0, 0, 0)*.eta3 +
    c(0, 0, 0, 0)*.eta4 +
    c(0, 0, 0, 0)*.eta5 +
    c(0, 0, 0, 0)*.eta6
.eta3 ~-
    c(0, 0, 0, 0)*.eta3 +
    c(0, 0, 0, 0)*.eta4 +
    c(0, 0, 0, 0)*.eta5 +
    c(0, 0, 0, 0)*.eta6
.eta4 ~-
    c(0, 0, 0, 0)*.eta4 +
    c(0, 0, 0, 0)*.eta5 +
    c(0, 0, 0, 0)*.eta6
.eta5 ~-
    c(0, 0, 0, 0)*.eta5 +
    c(0, 0, 0, 0)*.eta6
.eta6 ~-
    c(0, 0, 0, 0)*.eta6
.pi1 ~-

```

```

c(1, 1, 1, 1)*.pi1 +
c(0.5, 0.5, 0.5, 0.5)*.pi2 +
c(0.5, 0.5, 0.5, 0.5)*.pi3 +
c(0.5, 0.5, 0.5, 0.5)*.pi4 +
c(0.5, 0.5, 0.5, 0.5)*.pi5 +
c(0.5, 0.5, 0.5, 0.5)*.pi6
.pi2 ~~
  c(1, 1, 1, 1)*.pi2 +
  c(0.5, 0.5, 0.5, 0.5)*.pi3 +
  c(0.5, 0.5, 0.5, 0.5)*.pi4 +
  c(0.5, 0.5, 0.5, 0.5)*.pi5 +
  c(0.5, 0.5, 0.5, 0.5)*.pi6
.pi3 ~~
  c(1, 1, 1, 1)*.pi3 +
  c(0, 0, 0, 0)*.pi4 +
  c(0.5, 0.5, 0.5, 0.5)*.pi5 +
  c(0.5, 0.5, 0.5, 0.5)*.pi6
.pi4 ~~
  c(1, 1, 1, 1)*.pi4 +
  c(0.5, 0.5, 0.5, 0.5)*.pi5 +
  c(0.5, 0.5, 0.5, 0.5)*.pi6
.pi5 ~~
  c(1, 1, 1, 1)*.pi5 +
  c(0.5, 0.5, 0.5, 0.5)*.pi6
.pi6 ~~
  c(1, 1, 1, 1)*.pi6
.Y1_W1_1_W2_1 ~~
  c(0.4, 0.4, 0.4, 0.4)*.Y1_W1_1_W2_1
.Y1_W1_1_W2_2 ~~
  c(0.4, 0.4, 0.4, 0.4)*.Y1_W1_1_W2_2
.Y1_W1_1_W2_3 ~~
  c(0.4, 0.4, 0.4, 0.4)*.Y1_W1_1_W2_3
.Y1_W1_2_W2_1 ~~
  c(0.4, 0.4, 0.4, 0.4)*.Y1_W1_2_W2_1
.Y1_W1_2_W2_2 ~~
  c(0.4, 0.4, 0.4, 0.4)*.Y1_W1_2_W2_2
.Y1_W1_2_W2_3 ~~
  c(0.4, 0.4, 0.4, 0.4)*.Y1_W1_2_W2_3
.Y2_W1_1_W2_1 ~~
  c(0.4, 0.4, 0.4, 0.4)*.Y2_W1_1_W2_1
.Y2_W1_1_W2_2 ~~
  c(0.4, 0.4, 0.4, 0.4)*.Y2_W1_1_W2_2
.Y2_W1_1_W2_3 ~~
  c(0.4, 0.4, 0.4, 0.4)*.Y2_W1_1_W2_3
.Y2_W1_2_W2_1 ~~

```

```

    c(0.4, 0.4, 0.4, 0.4)*.Y2_W1_2_W2_1
.Y2_W1_2_W2_2 ~
    c(0.4, 0.4, 0.4, 0.4)*.Y2_W1_2_W2_2
.Y2_W1_2_W2_3 ~
    c(0.4, 0.4, 0.4, 0.4)*.Y2_W1_2_W2_3
.Y3_W1_1_W2_1 ~
    c(0.4, 0.4, 0.4, 0.4)*.Y3_W1_1_W2_1
.Y3_W1_1_W2_2 ~
    c(0.4, 0.4, 0.4, 0.4)*.Y3_W1_1_W2_2
.Y3_W1_1_W2_3 ~
    c(0.4, 0.4, 0.4, 0.4)*.Y3_W1_1_W2_3
.Y3_W1_2_W2_1 ~
    c(0.4, 0.4, 0.4, 0.4)*.Y3_W1_2_W2_1
.Y3_W1_2_W2_2 ~
    c(0.4, 0.4, 0.4, 0.4)*.Y3_W1_2_W2_2
.Y3_W1_2_W2_3 ~
    c(0.4, 0.4, 0.4, 0.4)*.Y3_W1_2_W2_3
covariate1 ~
    c(0.4, 0.4, 0.4, 0.4)*covariate1
covariate2 ~
    c(1, 1, 1, 1)*covariate2
covariate2_1 ~
    c(0.4, 0.4, 0.4, 0.4)*covariate2_1
covariate2_2 ~
    c(0.4, 0.4, 0.4, 0.4)*covariate2_2
.eta1 ~
    c(0, 0, 0, 0)*1
.eta2 ~
    c(0, 0, 0, 0)*1
.eta3 ~
    c(0, 0, 0, 0)*1
.eta4 ~
    c(0, 0, 0, 0)*1
.eta5 ~
    c(0, 0, 0, 0)*1
.eta6 ~
    c(0, 0, 0, 0)*1
.pi1 ~
    c(0.1, 0.1, 0.1, 0.1)*1
.pi2 ~
    c(0.1, 0.1, 0.1, 0.1)*1
.pi3 ~
    c(0.1, 0.1, 0.1, 0.1)*1
.pi4 ~
    c(0.1, 0.1, 0.1, 0.1)*1

```

```

.pi5 ~
  c(0.1, 0.1, 0.1, 0.1)*1
.pi6 ~
  c(0.1, 0.1, 0.1, 0.1)*1
.Y1_W1_1_W2_1 ~
  c(0, 0, 0, 0)*1
.Y1_W1_1_W2_2 ~
  c(0, 0, 0, 0)*1
.Y1_W1_1_W2_3 ~
  c(0, 0, 0, 0)*1
.Y1_W1_2_W2_1 ~
  c(0, 0, 0, 0)*1
.Y1_W1_2_W2_2 ~
  c(0, 0, 0, 0)*1
.Y1_W1_2_W2_3 ~
  c(0, 0, 0, 0)*1
.Y2_W1_1_W2_1 ~
  c(0, 0, 0, 0)*1
.Y2_W1_1_W2_2 ~
  c(0, 0, 0, 0)*1
.Y2_W1_1_W2_3 ~
  c(0, 0, 0, 0)*1
.Y2_W1_2_W2_1 ~
  c(0, 0, 0, 0)*1
.Y2_W1_2_W2_2 ~
  c(0, 0, 0, 0)*1
.Y2_W1_2_W2_3 ~
  c(0, 0, 0, 0)*1
.Y3_W1_1_W2_1 ~
  c(0, 0, 0, 0)*1
.Y3_W1_1_W2_2 ~
  c(0, 0, 0, 0)*1
.Y3_W1_1_W2_3 ~
  c(0, 0, 0, 0)*1
.Y3_W1_2_W2_1 ~
  c(0, 0, 0, 0)*1
.Y3_W1_2_W2_2 ~
  c(0, 0, 0, 0)*1
.Y3_W1_2_W2_3 ~
  c(0, 0, 0, 0)*1
covariate1 ~
  c(0, 0, 0, 0)*1
covariate2 ~
  c(0.1, 0.1, 0.1, 0.1)*1
covariate2_1 ~

```

```
    c(0, 0, 0, 0)*1
covariate2_2 ~
    c(0, 0, 0, 0)*1
""

power <- power_analysis_semnova(
  n_within = c(2, 3),
  n_between = c(2, 2),
  n_indicator = 3,
  n_manifest_covariate = 1,
  n_latent_covariate = 2,
  sphericity = list(2),
  normalize_contrasts = TRUE,
  data_syntax = data_syntax,
  sample_size = c(100, 100, 100, 100),
  replications = 500
)

# hypothesis tests
summary(power)

# parameters
print(power)

# path diagram
plot(power)
```



## Appendix C

# Bibliography

- Achenbach, T. M. (1991). *Manual for the child behavior checklist/4-18 and 1991 profile*. University of Vermont, Department of Psychiatry, Burlington, VT.
- Gräfe, L., Hahn, S., and Mayer, A. (2022). On the relationship between ANOVA main effects and average treatment effects. *Multivariate Behavioral Research*.
- Lane, D. M. (2016). The assumption of sphericity in repeated-measures designs: what it means and what to do when it is violated. *The Quantitative Methods for Psychology*, 12(2):114–122.
- Langenberg, B. (2022). *Repeated measures ANOVA with latent variables: A new approach based on structural equation modeling*. Dissertation, Bielefeld University.
- Little, T. D., Siegers, D. W., and Card, N. A. (2006). A non-arbitrary method of identifying and scaling latent variables in SEM and MACS models. *Structural Equation Modeling*, 13(1):59–72.
- Mauchly, J. W. (1940). Significance test for sphericity of a normal  $n$ -variate distribution. 11(2):204–209.
- Muller, K. E. and Barton, C. N. (1989). Approximate power for repeated-measures anova lacking sphericity. *Journal of the American Statistical Association*, 84(406):549–555.
- R Core Team (2021). R: a language and environment for statistical computing. Computer software.
- Rosseel, Y. (2012). lavaan: an R package for structural equation modeling. *Journal of Statistical Software*, 48(2):1–20.
- tze Hu, L. and Bentler, P. M. (1999). Cutoff criteria for fit indexes in covariance structure analysis: conventional criteria versus new alternatives. *Structural Equation Modeling: A Multidisciplinary Journal*, 6(1):1–55.