`semnova`: An R package for latent repeated measures ANOVA

Benedikt Langenberg and Axel Mayer

2021-01-04

Contents

Preface

About `semnova`

`semnova` is an `R` package for analyzing data from repeated measures experimental designs. `semnova` implements latent repeated measures analysis of variance (RM-ANOVA) which is a structural equation modeling based alternative to tradtional RM-ANOVA. *Traditional* RM-ANOVA is a widely used statistical tool in the field of psychology and social sciences. RM-ANOVA can analyze data such as test scores, questionnaire items, reaction times, attitudes, characteristics, or motives. Oftentimes, the construct of interest cannot be observed directly and the aforementioned measures contain measurement errors. *Latent* RM-ANOVA can include multiple indicators that measure the same latent construct of and prune of measurement error. *Latent* RM-ANOVA can furthermore be used to examine interindividual differences in main and interaction effects of experimental factors and introduces a whole lot of other advantages from the structural equation modeling framework.

This very short tutorial is mainly based on the article by Langenberg et al. (2020). This tutorial, however, only covers how to use `semnova` for latent RM-ANOVA. If you would like to know more about the methods behind the software, check out the article.

Citing `semnova`

Please use this reference to cite the software package:

Langenberg, B., Helm, J. L., & Mayer, A. (2020). Repeated measures ANOVA with latent variables to analyze interindividual differences in contrasts. *Multivariate Behavioral Research*, 1–19. https://doi.org/10.1080/00273171.2020.1803038

Changelog

- 2019-06-13: First Commit

Chapter 1

Installation

`semnova` is available on `cran`. It is, however, recommended to install the latest version from github via the `devtools` package. In the installation process, a couple of other packages will be installed. In particular, `semnova` strongly depends on the structural equation modeling software `lavaan` (Rosseel, 2012).

1.1 Installation via `devtools` (recommended)

Make sure to install `devtools` first. Just copy and run the following lines within `R`.

```
if (!require("devtools", character.only = T)) {
  install.packages("devtools")
}

devtools::install_github("langenberg/semnova")
```

1.2 Installation via `cran`

Simply run this code to install `semnova` via cran.

```
install.packages("semnova")
```

Chapter 2

Example

semnova is a package intended for latent RM-ANOVA. In the following, we will download a publicly available dataset by Qu et al. (2015) and perform latent RM-ANOVA on a subset of the data.

2.1 Getting the Data

The data set can be downloaded from the *PLoS ONE* website. We use the openxlsx package for importing the data and the tidyverse collection (Wickham et al., 2019) for further data manipulations.

```
if (!require("openxlsx", character.only = T)) {
  install.packages("openxlsx")
  library(openxlsx)
}

if (!require("tidyverse", character.only = T)) {
  install.packages("tidyverse")
  library(tidyverse)
}

d_raw <- read.xlsx("https://doi.org/10.1371/journal.pone.0125279.s001") %>%
  as_tibble()
```

2.2 Introducing the Data

Let's briefly introduce the study conducted by Qu et al. (2015). The authors investigated a bystander effect in a virtual classroom setting. In particular, participants first witnessed virtual students answering questions from a teacher in a training for spoken English. The participants afterwards answered questions themselves. During both parts of the experiment, virtual students (bystanders) whispered and commented on the virtual speaker and the human speaker, respectively. The whispering was either positive or negative. This resulted in a two by two within subject repeated measures design. Every participant received every condition. The relevant sub scale used in our analysis is the *virtual students' performance*. We only use two items from this sub scale measuring attitudes towards the virtual bystanders and the performance of the virtual bystanders on an 11-point scale reaching from 0 (very bad) to 11 (very good).

A graphical illustration of these items can be found in Figure 2.1

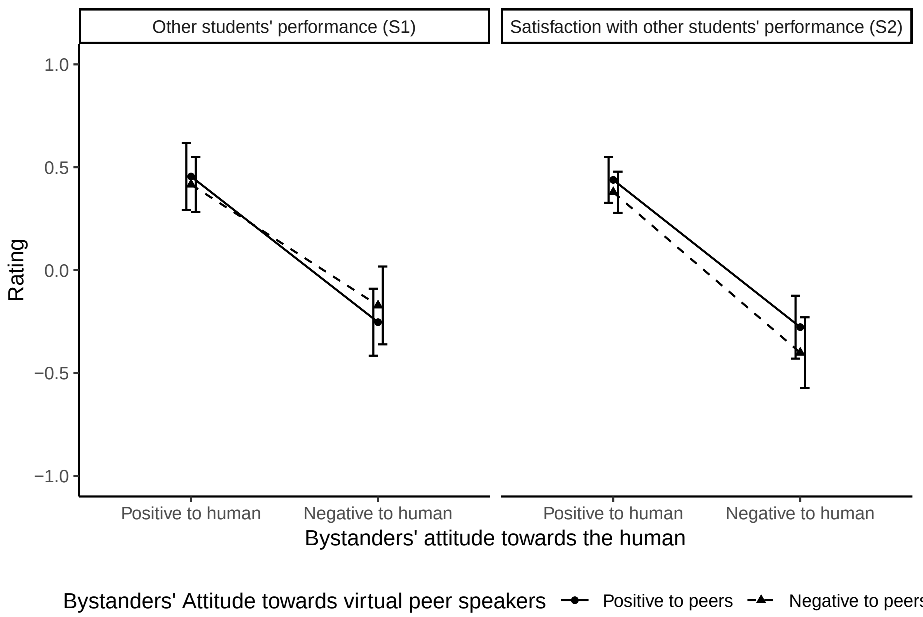

Figure 2.1: Means and standard errors for the the investigated items across the four experimental conditions.

2.3 Preparing the Data

The dataset contains a lot of variables. In the next step, we select only the relevant variables.

```
d_wide <- d_raw %>%
  select(ID, S1PP:S2NN)

d_wide
```

```
## # A tibble: 26 x 9
##       ID   S1PP   S1NP   S1PN    S1NN   S2PP  S2NP   S2PN    S2NN
##    <dbl>  <dbl>  <dbl>  <dbl>   <dbl>  <dbl> <dbl>  <dbl>   <dbl>
##  1     1  0.913 0.177  -0.927 -0.927   0.913 0.177 -0.927 -0.927
##  2     2  0.752 1.56    0.128 -1.81    0.565 0.939 -0.246 -2.12
##  3     3  0.238 0.238  -1.32   0.0275  0.660 0.238 -1.49  -0.689
##  4     4 -1.54  1.47    1.09   1.60   -0.552 1.09   0.655  1.52
##  5     5 -0.269 0.120  -0.269  0.120  -0.269 0.120 -0.269  0.509
##  6     6  0.463 0.0686  0.108  0.858   0.463 0.858  0.463  0.0686
##  7     7  1.10  0.707  -0.701 -1.49    0.663 0.179 -0.657 -0.657
##  8     8  0.200 0.777   0.392  0.585  -0.665 0.873 -0.665  1.07
##  9     9  1.88  0.356   0.297  0.297   1.47  1.17  -0.287 -0.521
## 10    10  0.600 1.44    0.676  0.216   0.753 0.753  0.830 -0.168
## # ... with 16 more rows
```

Chapter 3

Latent Repeated Measures ANOVA

3.1 Arguments

The input for `semnova` is very similar to the `car` package for analysis of variance (Fox and Weisberg, 2019). It takes the following arguments:

- `formula`: A formula object.
- `idesign`: A design object describing the factorial design.
- `idata`: A table mapping the dependent variables to the experimental conditions.
- `data`: A dataset.
- `mmodel` (optional): A measurement model mapping the variables from the dataset to the variables in the formula object. If not given, the variables from the formula object will be used.

`formula`

The formula object contains the names of the dependent variables (i.e., the combinations of the levels of the factors).

```
formula <- cbind(NN, PN, NP, PP) ~ 1
```

NN, for instance, represents the condition Peer = negative and Human = negative.

idata

The idata object is a dataframe that has as many rows as variables in the formula object. The columns represent the experimental factors. The rows represent the possible combinations of the factor levels.

```
idata <- expand.grid(Peer = c("negative", "positive"),
                     Human = c("negative", "positive"))

idata
```

```
##       Peer    Human
## 1 negative negative
## 2 positive negative
## 3 negative positive
## 4 positive positive
```

The idata object maps the combinations of the factorial levels to the variables in the formula object.

idesign

The design object is a formula object that represents the effects of interest. That is, when expanded, the formula contains the terms (i.e., the main and interaction effects) that the user is interested in. Currently, `semnova` can only model saturated models, that is, all possible terms are inlcuded. Subsets are currently not possible, but will this feature will be added in the future.

```
idesign <- ~Peer*Human
```

mmodel

The measurement model is optional. If the dependent variables are measure via multiple indicators, the `create_mmodel()` can be used to map variables from the dataset to the variables from the formula object. By default, the effects-coding indentification method introduced by Little et al. (2006) is used. If you prefer the referent-indicator method, add the argument `lv_scaling = "referent`.

```
library(semnova)

mmodel <- create_mmodel(
  NN = c("S1NN", "S2NN"),
  PN = c("S1PN", "S2PN"),
  NP = c("S1NP", "S2NP"),
  PP = c("S1PP", "S2PP")
)
```

3.2 The First Analysis

The function `semnova()` performs the analysis and takes the aforementioned arguments:

```
library(semnova)

fit <- semnova(
  formula = cbind(NN, PN, NP, PP) ~ 1,
  data = d_wide,
  idata = idata,
  idesign = idesign,
  mmodel = mmodel
)
```

Hypothesis tests of default main and interaction effects can be printed using the `summary()` function:

```
summary(fit)
```

```
## ------------------------------------------
##
## term: (Intercept)
##
## Response transformation matrix:
##                   (Intercept)
## negative.negative         0.5
## positive.negative         0.5
## negative.positive         0.5
## positive.positive         0.5
##
## multiv. tests:
##       Df test stat approx F num Df den Df Pr(>F)
## Wald   1   0.85307  0.85307      1     25 0.3645
## Wilks  1   0.96595  0.88125      1     25 0.3568
##
## univ. test:
##         Sum Sq num Df Error SS den Df F value Pr(>F)
## F-test 0.7295      1   20.695     25  0.8812 0.3568
##
## ------------------------------------------
##
## term: Peer
##
## Response transformation matrix:
##                   Peer1
## negative.negative  -0.5
```

```
## positive.negative   0.5
## negative.positive  -0.5
## positive.positive   0.5
##
## multiv. tests:
##       Df test stat approx F num Df den Df Pr(>F)
## Wald   1   0.18200  0.18200      1     25 0.6733
## Wilks  1   0.99124  0.22083      1     25 0.6425
##
## univ. test:
##          Sum Sq num Df Error SS den Df F value Pr(>F)
## F-test 0.045471      1   5.1477     25  0.2208 0.6425
##
## ------------------------------------------
##
## term: Human
##
## Response transformation matrix:
##                   Human1
## negative.negative   -0.5
## positive.negative   -0.5
## negative.positive    0.5
## positive.positive    0.5
##
## multiv. tests:
##       Df test stat approx F num Df den Df    Pr(>F)
## Wald   1   22.6149   22.615      1     25 7.021e-05 ***
## Wilks  1    0.5058   24.431      1     25 4.322e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## univ. test:
##        Sum Sq num Df Error SS den Df F value    Pr(>F)
## F-test 13.782      1   14.102     25  24.431 4.322e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ------------------------------------------
##
## term: Peer:Human
##
## Response transformation matrix:
##                   Peer1:Human1
## negative.negative          0.5
## positive.negative         -0.5
## negative.positive         -0.5
```

```
## positive.positive          0.5
##
## multiv. tests:
##       Df test stat approx F num Df den Df Pr(>F)
## Wald   1   0.06643 0.066430      1     25 0.7987
## Wilks  1   0.99670 0.082832      1     25 0.7759
##
## univ. test:
##          Sum Sq num Df Error SS den Df F value Pr(>F)
## F-test 0.015346      1   4.6316     25  0.0828 0.7759
##
## ------------------------------------------
```

The output is very similar to the output produced by the `car` package. The contrast matrix (or response transformation matrix) as well as test statistics for each of the effects are printed. Univariate and multivariate statistics are given. As latent RM-ANOVA relys on structural equation modeling, maximum likelihood is used for estimating the model parameters. The Wald test is a common test for testing parameters in the structural equation modeling framework. Wilk's lambda is based on the means and variances of the effect variables (i.e., contrasts). That is, Wilk's lambda is not based on an OLS estimator, but on the maximum likelihood estimates. This is also valid for the univariate test statistics.

Chapter 4

Traditional ANOVA

In this section, traditional RM-ANOVA is performed for a brief comparison with latent RM-ANOVA. For this analysis, the two indicators are averaged for each subject within each of the experimental conditions as RM-ANOVA can only deal with manifest variables. The `car` package is used for this analysis.

First of all, a new dataset is needed containing the averaged indicators. This is being done using the `tidyverse` collection. In the next chunk, the long format dataset is only an auxiliary dataset as averaging more convenient with long data.

```
if (!require("car", character.only = T)) {
  install.packages("car")
}

library(car)
library(tidyverse)

d_long <- d_wide %>%
  gather(condition, value, -ID) %>%
  mutate(item = substr(condition, 1, 2),
         condition = substr(condition, 3, 4))

d_mean_wide <- d_long %>%
  group_by(ID, condition) %>%
  summarise(value = mean(value)) %>%
  spread(condition, value)

d_mean_wide
```

```
## # A tibble: 26 x 5
## # Groups:   ID [26]
```

```
##       ID      NN    NP       PN     PP
##    <dbl>   <dbl> <dbl>    <dbl>  <dbl>
##  1      1 -0.927  0.177 -0.927    0.913
##  2      2 -1.96   1.25  -0.0588   0.659
##  3      3 -0.331  0.238 -1.40     0.449
##  4      4  1.56   1.28   0.870   -1.05
##  5      5  0.314  0.120 -0.269   -0.269
##  6      6  0.463  0.463  0.286    0.463
##  7      7 -1.07   0.443 -0.679    0.883
##  8      8  0.825  0.825 -0.136   -0.232
##  9      9 -0.112  0.765  0.00506  1.67
## 10     10  0.0236 1.10   0.753    0.676
## # ... with 16 more rows
```

The idata and the idesign objects are just the same as for latent RM-ANOVA.

```
idata
```

```
##       Peer    Human
## 1 negative negative
## 2 positive negative
## 3 negative positive
## 4 positive positive
```

```
idesign
```

```
## ~Peer * Human
```

Now, let's run the analysis using the `Anova` function. Note that the variables in the formula now are actual variables from the dataset (average scores).

```
anova_model <- lm(cbind(NN, PN, NP, PP) ~ 1, d_mean_wide)
anova_fit <- Anova(anova_model,
                   idata = idata,
                   idesign = idesign,
                   type = "III")
summary(anova_fit)
```

```
##
## Type III Repeated Measures MANOVA Tests:
##
## ------------------------------------------
##
## Term: (Intercept)
##
##  Response transformation matrix:
##    (Intercept)
## NN           1
## PN           1
```

```
## NP            1
## PP            1
##
## Sum of squares and products for the hypothesis:
##             (Intercept)
## (Intercept)    2.234811
##
## Multivariate Tests: (Intercept)
##                  Df test stat  approx F num Df den Df  Pr(>F)
## Pillai            1 0.0233974 0.5989488      1     25 0.44624
## Wilks             1 0.9766026 0.5989488      1     25 0.44624
## Hotelling-Lawley  1 0.0239580 0.5989488      1     25 0.44624
## Roy               1 0.0239580 0.5989488      1     25 0.44624
##
## ------------------------------------------
##
## Term: Peer
##
##  Response transformation matrix:
##    Peer1
## NN     1
## PN    -1
## NP     1
## PP    -1
##
## Sum of squares and products for the hypothesis:
##           Peer1
## Peer1 0.1304386
##
## Multivariate Tests: Peer
##                  Df test stat  approx F num Df den Df Pr(>F)
## Pillai            1 0.0047895 0.1203125      1     25 0.7316
## Wilks             1 0.9952105 0.1203125      1     25 0.7316
## Hotelling-Lawley  1 0.0048125 0.1203125      1     25 0.7316
## Roy               1 0.0048125 0.1203125      1     25 0.7316
##
## ------------------------------------------
##
## Term: Human
##
##  Response transformation matrix:
##    Human1
## NN      1
## PN      1
## NP     -1
## PP     -1
```

```
##
## Sum of squares and products for the hypothesis:
##          Human1
## Human1 50.63608
##
## Multivariate Tests: Human
##                  Df test stat approx F num Df den Df     Pr(>F)
## Pillai            1 0.4442502 19.98427      1     25 0.00014706 ***
## Wilks             1 0.5557498 19.98427      1     25 0.00014706 ***
## Hotelling-Lawley  1 0.7993709 19.98427      1     25 0.00014706 ***
## Roy               1 0.7993709 19.98427      1     25 0.00014706 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ------------------------------------------
##
## Term: Peer:Human
##
##  Response transformation matrix:
##    Peer1:Human1
## NN            1
## PN           -1
## NP           -1
## PP            1
##
## Sum of squares and products for the hypothesis:
##              Peer1:Human1
## Peer1:Human1   0.02058027
##
## Multivariate Tests: Peer:Human
##                  Df test stat   approx F num Df den Df  Pr(>F)
## Pillai            1 0.0009096 0.02276124      1     25 0.88129
## Wilks             1 0.9990904 0.02276124      1     25 0.88129
## Hotelling-Lawley  1 0.0009104 0.02276124      1     25 0.88129
## Roy               1 0.0009104 0.02276124      1     25 0.88129
##
## Univariate Type III Repeated-Measures ANOVA Assuming Sphericity
##
##              Sum Sq num Df Error SS den Df F value    Pr(>F)
## (Intercept)  0.5587      1  23.3201     25  0.5989 0.4462367
## Peer         0.0326      1   6.7760     25  0.1203 0.7315986
## Human       12.6590      1  15.8362     25 19.9843 0.0001471 ***
## Peer:Human   0.0051      1   5.6511     25  0.0228 0.8812904
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Chapter 5

Model Inspection

5.1 lavaan Object

Currently, semnova relies on the lavaan package which uses maximum likelihood for estimation. In later releases more estimators will be available. A Bayesian estimation method is on top of the to-do list.

lavaan provides a lot of functions and methods to investigate the estimated model. These functions can also be used with semnova. The underlying lavaan object can be accessed via the @sem_obj slot of the semnova object.

```
library(lavaan)

fit@sem_obj
```

```
## lavaan 0.6-6 ended normally after 50 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         38
##   Number of equality constraints                    14
##
##   Number of observations                            26
##
## Model Test User Model:
##
##   Test statistic                                29.975
##   Degrees of freedom                                20
##   P-value (Chi-square)                           0.070
```

For a comprehensive summary of the model, use the summary function on the lavaan object.

```r
summary(fit@sem_obj)
```

```
## lavaan 0.6-6 ended normally after 50 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         38
##   Number of equality constraints                    14
##
##   Number of observations                            26
##
## Model Test User Model:
##
##   Test statistic                                29.975
##   Degrees of freedom                                20
##   P-value (Chi-square)                           0.070
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                      Estimate  Std.Err  z-value  P(>|z|)
##   NN =~
##     S1NN     (.l1)    1.038    0.044   23.684    0.000
##     S2NN     (.l2)    0.962    0.044   21.936    0.000
##   PN =~
##     S1PN     (.l1)    1.038    0.044   23.684    0.000
##     S2PN     (.l2)    0.962    0.044   21.936    0.000
##   NP =~
##     S1NP     (.l1)    1.038    0.044   23.684    0.000
##     S2NP     (.l2)    0.962    0.044   21.936    0.000
##   PP =~
##     S1PP     (.l1)    1.038    0.044   23.684    0.000
##     S2PP     (.l2)    0.962    0.044   21.936    0.000
##   .pi1 =~
##     NN                0.500
##     PN                0.500
##     NP                0.500
##     PP                0.500
##   .pi2 =~
##     NN               -0.500
```

```
##     PN                0.500
##     NP               -0.500
##     PP                0.500
##   .pi3 =~
##     NN               -0.500
##     PN               -0.500
##     NP                0.500
##     PP                0.500
##   .pi4 =~
##     NN                0.500
##     PN               -0.500
##     NP               -0.500
##     PP                0.500
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##  .NN ~~
##    .PN                0.000
##    .NP                0.000
##  .PN ~~
##    .NP                0.000
##  .NN ~~
##    .PP                0.000
##  .PN ~~
##    .PP                0.000
##  .NP ~~
##    .PP                0.000
##   .pi1 ~~
##     .pi2    (.__2)   -0.109    0.095   -1.149    0.251
##     .pi3  (.__3_1)   -0.390    0.160   -2.429    0.015
##   .pi2 ~~
##     .pi3  (.__3_2)    0.212    0.089    2.394    0.017
##   .pi1 ~~
##     .pi4  (.__4_1)   -0.113    0.090   -1.253    0.210
##   .pi2 ~~
##     .pi4  (.__4_2)    0.036    0.050    0.728    0.466
##   .pi3 ~~
##     .pi4  (.__4_3)    0.088    0.076    1.166    0.244
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .S1NN     (.i1)    0.034    0.026    1.308    0.191
##    .S2NN     (.i2)   -0.034    0.026   -1.308    0.191
##    .S1PN     (.i1)    0.034    0.026    1.308    0.191
##    .S2PN     (.i2)   -0.034    0.026   -1.308    0.191
##    .S1NP     (.i1)    0.034    0.026    1.308    0.191
```

```
##    .S2NP     (.i2)   -0.034    0.026   -1.308    0.191
##    .S1PP     (.i1)    0.034    0.026    1.308    0.191
##    .S2PP     (.i2)   -0.034    0.026   -1.308    0.191
##    .NN                0.000
##    .PN                0.000
##    .NP                0.000
##    .PP                0.000
##     .pi1     (.m1)    0.168    0.181    0.924    0.356
##     .pi2     (.m2)    0.042    0.098    0.427    0.670
##     .pi3     (.m3)    0.728    0.153    4.756    0.000
##     .pi4     (.m4)    0.024    0.094    0.258    0.797
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .NN                0.000
##    .PN                0.000
##    .NP                0.000
##    .PP                0.000
##     .pi1    (.__1)    0.796    0.235    3.383    0.001
##     .pi2    (.__2)    0.198    0.071    2.786    0.005
##     .pi3    (.__3)    0.542    0.165    3.282    0.001
##     .pi4    (.__4)    0.178    0.065    2.723    0.006
##    .S1NN              0.168    0.090    1.863    0.062
##    .S2NN              0.147    0.078    1.885    0.059
##    .S1PN              0.049    0.055    0.895    0.371
##    .S2PN              0.167    0.065    2.571    0.010
##    .S1NP              0.208    0.073    2.833    0.005
##    .S2NP              0.055    0.042    1.324    0.186
##    .S1PP              0.342    0.120    2.835    0.005
##    .S2PP              0.041    0.065    0.638    0.523
##
## Constraints:
##                                                |Slack|
##     .l1+.l2 - (2)                                0.000
##     .i1+.i2 - 0                                  0.000
```

5.2 Model String

In some cases, the model specified by **semnova** does not completely fit the needs of the user. It is possible to plot the model string that is passed to lavaan and change it according to one's needs.

```
model_string <- fit@model_string
```

```r
cat(model_string)
```

```
##
##   # loadings
##     NN =~ .l1*S1NN + NA*S1NN + .l2*S2NN
##     PN =~ .l1*S1PN + NA*S1PN + .l2*S2PN
##     NP =~ .l1*S1NP + NA*S1NP + .l2*S2NP
##     PP =~ .l1*S1PP + NA*S1PP + .l2*S2PP
##   # intercepts
##     S1NN ~ .i1*1
##     S2NN ~ .i2*1
##     S1PN ~ .i1*1
##     S2PN ~ .i2*1
##     S1NP ~ .i1*1
##     S2NP ~ .i2*1
##     S1PP ~ .i1*1
##     S2PP ~ .i2*1
##     NN ~ 0*1
##     PN ~ 0*1
##     NP ~ 0*1
##     PP ~ 0*1
##   # variances
##     NN ~~ 0*NN
##     PN ~~ 0*NN + 0*PN
##     NP ~~ 0*NN + 0*PN + 0*NP
##     PP ~~ 0*NN + 0*PN + 0*NP + 0*PP
##     .pi1 ~~ .vcov_pi_1_1*.pi1
##     .pi2 ~~ .vcov_pi_2_1*.pi1 + .vcov_pi_2_2*.pi2
##     .pi3 ~~ .vcov_pi_3_1*.pi1 + .vcov_pi_3_2*.pi2 + .vcov_pi_3_3*.pi3
##     .pi4 ~~ .vcov_pi_4_1*.pi1 + .vcov_pi_4_2*.pi2 + .vcov_pi_4_3*.pi3 + .vcov_pi_4_4*.pi4
##   # struc_coeff
##     .pi1 =~ 0.5*NN + 0.5*PN + 0.5*NP + 0.5*PP
##     .pi2 =~ -0.5*NN + 0.5*PN + -0.5*NP + 0.5*PP
##     .pi3 =~ -0.5*NN + -0.5*PN + 0.5*NP + 0.5*PP
##     .pi4 =~ 0.5*NN + -0.5*PN + -0.5*NP + 0.5*PP
##   # regressions
##     .pi1 ~ .m1*1
##     .pi2 ~ .m2*1
##     .pi3 ~ .m3*1
##     .pi4 ~ .m4*1
##   # constraints
##     .l1 + .l2 == 2
##     .i1 + .i2 == 0
```

Using the `sem()` function from the `lavaan` package, we see that the summary gives just the same output. Feel free to modify the model.

```r
fit_lavaan <- sem(model_string, data = d_wide)

summary(fit_lavaan)
```

```
## lavaan 0.6-6 ended normally after 50 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         38
##   Number of equality constraints                    14
##
##   Number of observations                            26
##
## Model Test User Model:
##
##   Test statistic                                29.975
##   Degrees of freedom                                20
##   P-value (Chi-square)                           0.070
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   NN =~
##     S1NN     (.l1)    1.038    0.044   23.684    0.000
##     S2NN     (.l2)    0.962    0.044   21.936    0.000
##   PN =~
##     S1PN     (.l1)    1.038    0.044   23.684    0.000
##     S2PN     (.l2)    0.962    0.044   21.936    0.000
##   NP =~
##     S1NP     (.l1)    1.038    0.044   23.684    0.000
##     S2NP     (.l2)    0.962    0.044   21.936    0.000
##   PP =~
##     S1PP     (.l1)    1.038    0.044   23.684    0.000
##     S2PP     (.l2)    0.962    0.044   21.936    0.000
##   .pi1 =~
##     NN                0.500
##     PN                0.500
##     NP                0.500
##     PP                0.500
##   .pi2 =~
```

```
##     NN               -0.500
##     PN                0.500
##     NP               -0.500
##     PP                0.500
##   .pi3 =~
##     NN               -0.500
##     PN               -0.500
##     NP                0.500
##     PP                0.500
##   .pi4 =~
##     NN                0.500
##     PN               -0.500
##     NP               -0.500
##     PP                0.500
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##  .NN ~~
##    .PN                0.000
##    .NP                0.000
##  .PN ~~
##    .NP                0.000
##  .NN ~~
##    .PP                0.000
##  .PN ~~
##    .PP                0.000
##  .NP ~~
##    .PP                0.000
##   .pi1 ~~
##     .pi2    (.__2)   -0.109    0.095   -1.149    0.251
##     .pi3  (.__3_1)   -0.390    0.160   -2.429    0.015
##   .pi2 ~~
##     .pi3  (.__3_2)    0.212    0.089    2.394    0.017
##   .pi1 ~~
##     .pi4  (.__4_1)   -0.113    0.090   -1.253    0.210
##   .pi2 ~~
##     .pi4  (.__4_2)    0.036    0.050    0.728    0.466
##   .pi3 ~~
##     .pi4  (.__4_3)    0.088    0.076    1.166    0.244
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .S1NN     (.i1)    0.034    0.026    1.308    0.191
##    .S2NN     (.i2)   -0.034    0.026   -1.308    0.191
##    .S1PN     (.i1)    0.034    0.026    1.308    0.191
##    .S2PN     (.i2)   -0.034    0.026   -1.308    0.191
```

```
##    .S1NP     (.i1)    0.034    0.026    1.308    0.191
##    .S2NP     (.i2)   -0.034    0.026   -1.308    0.191
##    .S1PP     (.i1)    0.034    0.026    1.308    0.191
##    .S2PP     (.i2)   -0.034    0.026   -1.308    0.191
##    .NN                0.000
##    .PN                0.000
##    .NP                0.000
##    .PP                0.000
##     .pi1     (.m1)    0.168    0.181    0.924    0.356
##     .pi2     (.m2)    0.042    0.098    0.427    0.670
##     .pi3     (.m3)    0.728    0.153    4.756    0.000
##     .pi4     (.m4)    0.024    0.094    0.258    0.797
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .NN                0.000
##    .PN                0.000
##    .NP                0.000
##    .PP                0.000
##     .pi1    (.__1)    0.796    0.235    3.383    0.001
##     .pi2    (.__2)    0.198    0.071    2.786    0.005
##     .pi3    (.__3)    0.542    0.165    3.282    0.001
##     .pi4    (.__4)    0.178    0.065    2.723    0.006
##    .S1NN              0.168    0.090    1.863    0.062
##    .S2NN              0.147    0.078    1.885    0.059
##    .S1PN              0.049    0.055    0.895    0.371
##    .S2PN              0.167    0.065    2.571    0.010
##    .S1NP              0.208    0.073    2.833    0.005
##    .S2NP              0.055    0.042    1.324    0.186
##    .S1PP              0.342    0.120    2.835    0.005
##    .S2PP              0.041    0.065    0.638    0.523
##
## Constraints:
##                                                |Slack|
##     .l1+.l2 - (2)                                0.000
##     .i1+.i2 - 0                                  0.000
```

It is also possible to append lavaan syntax to the model string using the `semnova()` function. This can be achieved via the `append` argument. The following chunk of code estimates the same model as in the previous section by constraints the residual variances of the indicators to be equal across the experimental conditions. This is just an example how this works, this constraint is not necessarily meaningful.

```
model_constraints <- "
  # constraints
```

```
    S1NN ~~ var1*S1NN
    S2NN ~~ var2*S2NN
    S1PN ~~ var1*S1PN
    S2PN ~~ var2*S2PN
    S1NP ~~ var1*S1NP
    S2NP ~~ var2*S2NP
    S1PP ~~ var1*S1PP
    S2PP ~~ var2*S2PP
"

fit_constrained <- semnova(
  formula = cbind(NN, PN, NP, PP) ~ 1,
  data = d_wide,
  idata = idata,
  idesign = idesign,
  mmodel = mmodel,
  append = model_constraints
)

summary(fit_constrained@sem_obj)
```

```
## lavaan 0.6-6 ended normally after 35 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         38
##   Number of equality constraints                    20
##
##   Number of observations                            26
##
## Model Test User Model:
##
##   Test statistic                                37.349
##   Degrees of freedom                                26
##   P-value (Chi-square)                           0.070
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   NN =~
```

```
##     S1NN     (.l1)    1.012    0.046   21.988    0.000
##     S2NN     (.l2)    0.988    0.046   21.457    0.000
##   PN =~
##     S1PN     (.l1)    1.012    0.046   21.988    0.000
##     S2PN     (.l2)    0.988    0.046   21.457    0.000
##   NP =~
##     S1NP     (.l1)    1.012    0.046   21.988    0.000
##     S2NP     (.l2)    0.988    0.046   21.457    0.000
##   PP =~
##     S1PP     (.l1)    1.012    0.046   21.988    0.000
##     S2PP     (.l2)    0.988    0.046   21.457    0.000
##   .pi1 =~
##     NN                0.500
##     PN                0.500
##     NP                0.500
##     PP                0.500
##   .pi2 =~
##     NN               -0.500
##     PN                0.500
##     NP               -0.500
##     PP                0.500
##   .pi3 =~
##     NN               -0.500
##     PN               -0.500
##     NP                0.500
##     PP                0.500
##   .pi4 =~
##     NN                0.500
##     PN               -0.500
##     NP               -0.500
##     PP                0.500
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##  .NN ~~
##    .PN                0.000
##    .NP                0.000
##  .PN ~~
##    .NP                0.000
##  .NN ~~
##    .PP                0.000
##  .PN ~~
##    .PP                0.000
##  .NP ~~
##    .PP                0.000
##   .pi1 ~~
```

```
##      .pi2    (.__2)   -0.101    0.095   -1.067    0.286
##      .pi3  (.__3_1)   -0.372    0.156   -2.385    0.017
##    .pi2 ~~
##      .pi3  (.__3_2)    0.196    0.089    2.212    0.027
##    .pi1 ~~
##      .pi4  (.__4_1)   -0.073    0.080   -0.904    0.366
##    .pi2 ~~
##      .pi4  (.__4_2)    0.018    0.046    0.390    0.696
##    .pi3 ~~
##      .pi4  (.__4_3)    0.060    0.069    0.871    0.384
##
## Intercepts:
##                   Estimate  Std.Err  z-value  P(>|z|)
##     .S1NN     (.i1)    0.038    0.027    1.404    0.160
##     .S2NN     (.i2)   -0.038    0.027   -1.404    0.160
##     .S1PN     (.i1)    0.038    0.027    1.404    0.160
##     .S2PN     (.i2)   -0.038    0.027   -1.404    0.160
##     .S1NP     (.i1)    0.038    0.027    1.404    0.160
##     .S2NP     (.i2)   -0.038    0.027   -1.404    0.160
##     .S1PP     (.i1)    0.038    0.027    1.404    0.160
##     .S2PP     (.i2)   -0.038    0.027   -1.404    0.160
##     .NN                0.000
##     .PN                0.000
##     .NP                0.000
##     .PP                0.000
##      .pi1     (.m1)    0.147    0.179    0.820    0.412
##      .pi2     (.m2)    0.061    0.103    0.590    0.555
##      .pi3     (.m3)    0.724    0.152    4.747    0.000
##      .pi4     (.m4)   -0.006    0.087   -0.072    0.943
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|)
##     .NN                0.000
##     .PN                0.000
##     .NP                0.000
##     .PP                0.000
##      .pi1    (.__1)    0.757    0.228    3.319    0.001
##      .pi2    (.__2)    0.214    0.077    2.771    0.006
##      .pi3    (.__3)    0.539    0.167    3.223    0.001
##      .pi4    (.__4)    0.138    0.057    2.433    0.015
##     .S1NN    (var1)    0.214    0.046    4.693    0.000
##     .S2NN    (var2)    0.081    0.035    2.331    0.020
##     .S1PN    (var1)    0.214    0.046    4.693    0.000
##     .S2PN    (var2)    0.081    0.035    2.331    0.020
##     .S1NP    (var1)    0.214    0.046    4.693    0.000
##     .S2NP    (var2)    0.081    0.035    2.331    0.020
```

```
##    .S1PP    (var1)    0.214    0.046    4.693    0.000
##    .S2PP    (var2)    0.081    0.035    2.331    0.020
##
## Constraints:
##                                                |Slack|
##     .l1+.l2 - (2)                                0.000
##     .i1+.i2 - 0                                  0.000
```

5.3 Fit Measures

Fit measures can be inspected via the `fitmeasures()` function. Let's have a look at the fit measures for the original unconstrained model:

```
fitmeasures(fit@sem_obj)
```

```
##               npar                fmin               chisq                  df
##             24.000               0.576              29.975              20.000
##             pvalue      baseline.chisq         baseline.df     baseline.pvalue
##              0.070             142.345              28.000               0.000
##                cfi                 tli                nnfi                 rfi
##              0.913               0.878               0.878               0.705
##                nfi                pnfi                 ifi                 rni
##              0.789               0.564               0.918               0.913
##               logl   unrestricted.logl                 aic                 bic
##           -171.986            -156.998             391.972             422.166
##             ntotal                bic2               rmsea      rmsea.ci.lower
##             26.000             347.672               0.139               0.000
##     rmsea.ci.upper        rmsea.pvalue                 rmr          rmr_nomean
##              0.235               0.106               0.052               0.050
##               srmr        srmr_bentler srmr_bentler_nomean                crmr
##              0.086               0.086               0.090               0.089
##        crmr_nomean          srmr_mplus   srmr_mplus_nomean               cn_05
##              0.093               0.084               0.086              28.245
##              cn_01                 gfi                agfi                pgfi
##             33.584               0.867               0.707               0.394
##                mfi                ecvi
##              0.825               2.999
```

The p-value for the χ^2-statistic is almost below the common 5% level. The RMSEA is also quite high. However, the 5% confidence interval of the RMSEA includes the common threshold of 0.06. The fit can therefore be considered satisfactory. The small sample size might also be a reason why the fit is not "perfect".

Chapter 6

[under construction] Interindividual Differences

Chapter 7

[under construction] Custom Contrasts

Chapter 8

[under construction] Between-Subject Factors

Chapter 9

[under construction] Covariance Structures

Chapter 10

[under construction] Visualization

Chapter 11

Outlook

11.1 Roadmap

- enhanced output
  - effect sizes
  - variance estimates of the effect variables
- between subject designs
- stochastic group sizes
- plots
- covariates
- Bayesian estimation

Chapter 12

Bibliography

Fox, J. and Weisberg, S. (2019). *An R companion to applied regression.* Sage, Thousand Oaks, CA, 3 edition.

Langenberg, B., Helm, J. L., and Mayer, A. (2020). Repeated measures ANOVA with latent variables to analyze interindividual differences in contrasts. *Multivariate Behavioral Research*, pages 1–19.

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.

Qu, C., Ling, Y., Heynderickx, I., and Brinkman, W. P. (2015). Virtual bystanders in a language lesson: Examining the effect of social evaluation, vicarious experience, cognitive consistency and praising on students' beliefs, self-efficacy and anxiety in a virtual reality environment. *PLoS ONE*, 10(4):1–26.

Rosseel, Y. (2012). Lavaan: an r package for structural equation modeling. *Journal of Statistical Software*, 48(2):1–20.

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., Takahashi, K., Vaughan, D., Wilke, C., Woo, K., and Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686.