# How to evaluate theory-based hypotheses in a (RI-)CLPM using the GORICA

Rebecca M. Kuiper

11 February 2025

## Contents

This is a tutorial for using GORICA for (Random Intercept) Cross-lagged Panel Models ((RI-)CLPMs). The GORICA is an information criterion that can be used to evaluate theory-driven hypotheses.

(RI-)CLPMs are a type of statistical models used in longitudinal data research to analyze the relations between variables measured at multiple time points. Panel data can be analyzed at the construct level and the dimension level. In the construct level model, the focus is on the latent constructs that the observed variables represent. In the dimension model, the focus is on the observed variables themselves, rather than the latent constructs.

Here, two examples are presented for the use of the `goric` function in the restriktor package to evaluate hypotheses about a CLPM. These are based on the analysis in:
Snijders, I., Wijnia, L., Kuiper, R. M., Rikers, R. M. J. P., & Loyens, S. M. M. (2021). Relationship quality in higher education and the interplay with student engagement and loyalty. *British Journal of Educational Psychology.* https://doi.org/10.1111/bjep.12455
The first example covers analysis at the construct level, while the second example covers analysis at the dimension level.
**Other example files** for evaluating (causal dominance) hypotheses in RI-CLPMs can be found on 'https://github.com/rebeccakuiper/Tutorials/tree/main/GORICA%20in%20RI-CLPM'.

Note: For (more) information regarding interpreting the GORIC(A) output, see 'Guidelines_output_GORIC' (https://github.com/rebeccakuiper/Tutorials).

# Example 1: Construct Level Analysis

## R packages

First, install and call the `lavaan` library and the `restriktor` library (to load the `goric` function). If needed, it is possible to view the description of the function with the `?` operator or the `help` command.

The code presented here also requires the `tidyverse` package for data manipulation.

```r
# To install restriktor in R:
#if (!require("restriktor")) install.packages("restriktor")

# To install restriktor from github:
# if (!require("devtools")) install.packages("devtools")
# library(devtools)
# install_github("LeonardV/restriktor")
library(restriktor)

# print docs in the help-tab to view arguments and explanations for the function
#?goric

# To install lavaan in R:
# if (!require("lavaan")) install.packages("lavaan")
library(lavaan)

# To install tidyverse in R:
# if (!require("tidyverse")) install.packages("tidyverse")
library(tidyverse)
```

## Data

Upload the data set to the R environment and select the columns used for analysis. The id column is renamed to *ID* and the code in the data set for missing numbers -999.00 is replaced with `NA`s.

```r
data <- read.table("data/CLPM.dat", header = T)
colnames(data)[1] <- "ID"
data <- replace(data , data == -999.00, NA)

data_subset <- select(data,
                  THT1_SS,
                  TBT1_SS,
                  ACOMT1_SS,
                  SATT1_SS,
                  AB_T1_SS,
                  DE_T1_SS,
                  VI_T1_SS,
                  SLT1_SS,
                  TH_T2_SS,
                  TB_T2_SS,
                  ACOMT2SS,
                  SAT_T2SS,
                  ABT2_SS,
                  DET2_SS,
                  VIT2_SS,
                  SLT2SS)
```

## Measurement invariance

Next, we fit the CLPM using `lavaan`. The ‘RQ’ dimension is split into two sub-dimensions (cf. Snijders et al., 2021). Model 1 is fit to investigate configural invariance. The model is specified and fit in the following two steps.

```
CLPM_M1 <- '

  #####################
  # MEASUREMENT MODEL #
  #####################

  # Factor models for RQ1 at 2 waves.
  RQ11 =~ THT1_SS  + TBT1_SS
  RQ12 =~ TH_T2_SS + TB_T2_SS
  #RQ1  =~ 1 * RQ11 + 1 * RQ12

  # Factor models for RQ2 at 2 waves.
  RQ21 =~ ACOMT1_SS + SATT1_SS
  RQ22 =~ ACOMT2SS  + SAT_T2SS
  #RQ2  =~ 1 * RQ21 + 1 * RQ22

  # Factor models for SE at 2 waves.
  SE1 =~ AB_T1_SS + DE_T1_SS + VI_T1_SS
  SE2 =~ ABT2_SS  + DET2_SS  + VIT2_SS


  ############
  # DYNAMICS #
  ############

  # Specify the lagged effects between the latent variables.
  RQ12 + RQ22 + SE2 + SLT2SS ~ RQ11 + RQ21 + SE1 + SLT1_SS

  # Estimate the correlations within the same wave.
  # T1
  RQ11 ~~ RQ21 + SE1 + SLT1_SS
  RQ21 ~~ SE1 + SLT1_SS
  SE1 ~~ SLT1_SS
  # T2
  RQ12 ~~ RQ22 + SE2 + SLT2SS
  RQ22 ~~ SE2 + SLT2SS
  SE2 ~~ SLT2SS

'
```

```
CLPM_M1.fit <- sem(CLPM_M1, data = data_subset, missing = 'ML')
```

When fitting the model R returns the following warning message:

```
Warning message:  In lav_object_post_check(object) :  lavaan WARNING: covariance matrix of
latent variables is not positive definite;
```

So, we use `lavInspect(fit, "cov.lv")` to investigate further.

```
lavInspect(CLPM_M1.fit, "cov.lv")
```

```
       RQ11   RQ12   RQ21   RQ22    SE1    SE2
RQ11  7.757
RQ12  4.985  8.573
RQ21  8.547  5.629  9.481
RQ22  5.317  9.563  7.673 11.391
SE1   4.433  3.928  6.694  6.074  9.840
SE2   3.768  5.091  5.077  7.265  9.183 11.039
```

```
lavInspect(CLPM_M1.fit, "cor.lv")
```

```
      RQ11  RQ12  RQ21  RQ22   SE1   SE2
RQ11 1.000
RQ12 0.611 1.000
RQ21 0.997 0.624 1.000
RQ22 0.566 0.968 0.738 1.000
SE1  0.507 0.428 0.693 0.574 1.000
SE2  0.407 0.523 0.496 0.648 0.881 1.000
```

The correlations between RQ11 & RQ21 and between RQ22 & RQ12 are very high, which is to be expected considered that these two sub-dimensions belong to one dimension. Given that the warning does not point to a model misspecification, we continue the analysis.

```
fitMeasures(CLPM_M1.fit)[c("chisq","df")]
```

```
  chisq      df
715.867  78.000
```

The output reports the following: `chisq       df 715.867    78.000`

Based on these results we continue to Model 2, which investigates weak factorial invariance.

```
CLPM_M2 <- '

  #####################
  # MEASUREMENT MODEL #
  #####################

  # Factor models for RQ1 at 2 waves.
  RQ11 =~ L1 * THT1_SS  + L2 * TBT1_SS
  RQ12 =~ L1 * TH_T2_SS + L2 * TB_T2_SS

  # Factor models for RQ2 at 2 waves.
  RQ21 =~ L3 * ACOMT1_SS + L4 * SATT1_SS
  RQ22 =~ L3 * ACOMT2SS  + L4 * SAT_T2SS

  # Factor models for SE at 2 waves.
  SE1 =~ L5 * AB_T1_SS + L6 * DE_T1_SS + L7 * VI_T1_SS
  SE2 =~ L5 * ABT2_SS  + L6 * DET2_SS  + L7 * VIT2_SS


  ############
  # DYNAMICS #
  ############

  # Specify the lagged effects between the latent variables.
  RQ12 + RQ22 + SE2 + SLT2SS ~ RQ11 + RQ21 + SE1 + SLT1_SS
  # Estimate the correlations within the same wave.
  # T1
  RQ11 ~~ RQ21 + SE1 + SLT1_SS
  RQ21 ~~ SE1 + SLT1_SS
  SE1 ~~ SLT1_SS
  # T2
  RQ12 ~~ RQ22 + SE2 + SLT2SS
  RQ22 ~~ SE2 + SLT2SS
  SE2 ~~ SLT2SS

'
```

```
CLPM_M2.fit <- sem(CLPM_M2, data = data_subset, missing = 'ML')
```

R returns the same warning as before; so, we check the correlations again.

```
lavInspect(CLPM_M2.fit, "cov.lv")
```

```
       RQ11   RQ12   RQ21   RQ22    SE1    SE2
RQ11  7.734
RQ12  4.986  8.595
RQ21  8.460  5.589  9.308
RQ22  5.342  9.632  7.651 11.528
SE1   4.481  4.000  6.718  6.212 10.282
SE2   3.729  5.031  4.984  7.210  9.183 10.540
```

```
lavInspect(CLPM_M2.fit, "cor.lv")
```

```
      RQ11  RQ12  RQ21  RQ22   SE1   SE2
RQ11 1.000
RQ12 0.611 1.000
RQ21 0.997 0.625 1.000
RQ22 0.566 0.968 0.739 1.000
SE1  0.502 0.425 0.687 0.571 1.000
SE2  0.413 0.529 0.503 0.654 0.882 1.000
```

Again, there is no sign the model needs revision, so we continue.

```
fitMeasures(CLPM_M2.fit)[c("chisq","df")]
```

```
  chisq      df
721.021  82.000
```

We obtain these results: `chisq      df 721.021    82.000`

We perform a Chi-square difference test to check whether Models 1 and 2 differ significantly.

Df = 82 - 78 = 4
Check the constrained factor loadings = 1 + 1 + 2 = 4
Chi-square difference = 721.021 - 715.867 = 5.154
https://www.socscistatistics.com/pvalues/chidistribution.aspx

The *p* value is .271858. Hence, the result is not significant at $\alpha = .05$.

When the chi-square test is non-significant, this implies the factor loadings are not significantly different from each other over time. In other words, we can assume weak factorial invariance holds.

So, we move on to strong factorial invariance using Model 3:

```
CLPM_M3 <- '

  #####################
  # MEASUREMENT MODEL #
  #####################

  # Factor models for RQ1 at 2 waves.
  RQ11 =~ L1 * THT1_SS  + L2 * TBT1_SS
  RQ12 =~ L1 * TH_T2_SS + L2 * TB_T2_SS

  # Factor models for RQ2 at 2 waves.
  RQ21 =~ L3 * ACOMT1_SS + L4 * SATT1_SS
  RQ22 =~ L3 * ACOMT2SS  + L4 * SAT_T2SS

  # Factor models for SE at 2 waves.
  SE1 =~ L5 * AB_T1_SS + L6 * DE_T1_SS + L7 * VI_T1_SS
  SE2 =~ L5 * ABT2_SS  + L6 * DET2_SS  + L7 * VIT2_SS


  # Constrained intercepts over time
  THT1_SS ~ int_th*1
  TH_T2_SS ~ int_th*1
  TBT1_SS ~ int_tb*1
  TB_T2_SS ~ int_tb*1
  ACOMT1_SS ~ int_acom*1
  ACOMT2SS ~ int_acom*1
  SATT1_SS ~ int_sat*1
  SAT_T2SS ~ int_sat*1
  #
  AB_T1_SS ~ int_ab*1
  ABT2_SS  ~ int_ab*1
  DE_T1_SS ~ int_de*1
  DET2_SS ~ int_de*1
  VI_T1_SS ~ int_vi*1
  VIT2_SS ~ int_vi*1
  #
  SLT1_SS ~ int_sl*1
  SLT2SS ~ int_sl*1


  # Free latent means on t=2
  RQ12 + RQ22 + SE2 + RQ11 + RQ21 + SE1 ~ 1


  ############
  # DYNAMICS #
  ############

  # Specify the lagged effects between the latent variables.
  RQ12 + RQ22 + SE2 + SLT2SS ~ RQ11 + RQ21 + SE1 + SLT1_SS

  # Estimate the correlations within the same wave.
  # T1
  RQ11 ~~ RQ21 + SE1 + SLT1_SS
  RQ21 ~~ SE1 + SLT1_SS
  SE1 ~~ SLT1_SS
  # T2
  RQ12 ~~ RQ22 + SE2 + SLT2SS
  RQ22 ~~ SE2 + SLT2SS
  SE2 ~~ SLT2SS

'
```

```
CLPM_M3.fit <- sem(CLPM_M3, data = data_subset, missing = 'ML')
```

Given the warning, we investigate correlations again.

```
lavInspect(CLPM_M3.fit, "cov.lv")
```

```
       RQ11   RQ12   RQ21   RQ22    SE1    SE2
RQ11  7.764
RQ12  5.006  8.632
RQ21  8.487  5.608  9.328
RQ22  5.358  9.665  7.670 11.553
SE1   4.489  4.009  6.727  6.219 10.280
SE2   3.735  5.042  4.991  7.218  9.181 10.537
```

```
lavInspect(CLPM_M3.fit, "cor.lv")
```

```
      RQ11  RQ12  RQ21  RQ22   SE1   SE2
RQ11 1.000
RQ12 0.611 1.000
RQ21 0.997 0.625 1.000
RQ22 0.566 0.968 0.739 1.000
SE1  0.502 0.426 0.687 0.571 1.000
SE2  0.413 0.529 0.503 0.654 0.882 1.000
```

Then, move on to the results of the model:

```
fitMeasures(CLPM_M3.fit)[c("chisq","df")]
```

```
   chisq       df
725.4913  84.0000
```

Because Models 2 and 3 are also nested, we perform another Chi-square difference test:

Df = 84 - 82 = 2
Check the constrained parameters = 8 - 6 = 2
Chi-square difference = 725.4913 - 721.021 = 4.4703
https://www.socscistatistics.com/pvalues/chidistribution.aspx

The $p$ value is .106976. Hence, the result is not significant: $p > .05$.

If this chi-square difference test is non-significant, this means we can assume that strong factorial invariance holds over time. In that case we could consider investigating whether the means change over time. This is just optional.

Model 4 investigates strong factorial invariance without free latent means, meaning they are constrained over time). We repeat similar steps as above:

```
CLPM_M4 <- '

  #####################
```

```
# MEASUREMENT MODEL #
#####################

# Factor models for RQ1 at 2 waves.
RQ11 =~ L1 * THT1_SS  + L2 * TBT1_SS
RQ12 =~ L1 * TH_T2_SS + L2 * TB_T2_SS

# Factor models for RQ2 at 2 waves.
RQ21 =~ L3 * ACOMT1_SS + L4 * SATT1_SS
RQ22 =~ L3 * ACOMT2SS  + L4 * SAT_T2SS

# Factor models for SE at 2 waves.
SE1 =~ L5 * AB_T1_SS + L6 * DE_T1_SS + L7 * VI_T1_SS
SE2 =~ L5 * ABT2_SS  + L6 * DET2_SS  + L7 * VIT2_SS


# Constrained intercepts over time
THT1_SS ~ int_th*1
TH_T2_SS ~ int_th*1
TBT1_SS ~ int_tb*1
TB_T2_SS ~ int_tb*1
ACOMT1_SS ~ int_acom*1
ACOMT2SS ~ int_acom*1
SATT1_SS ~ int_sat*1
SAT_T2SS ~ int_sat*1
#
AB_T1_SS ~ int_ab*1
ABT2_SS  ~ int_ab*1
DE_T1_SS ~ int_de*1
DET2_SS ~ int_de*1
VI_T1_SS ~ int_vi*1
VIT2_SS ~ int_vi*1
#
SLT1_SS ~ int_sl*1
SLT2SS ~ int_sl*1


############
# DYNAMICS #
############

# Specify the lagged effects between the latent variables.
RQ12 + RQ22 + SE2 + SLT2SS ~ RQ11 + RQ21 + SE1 + SLT1_SS

# Estimate the correlations within the same wave.
# T1
RQ11 ~~ RQ21 + SE1 + SLT1_SS
RQ21 ~~ SE1 + SLT1_SS
SE1 ~~ SLT1_SS
# T2
RQ12 ~~ RQ22 + SE2 + SLT2SS
RQ22 ~~ SE2 + SLT2SS
SE2 ~~ SLT2SS
```

```
'
```

Fit the model:

```
CLPM_M4.fit <- sem(CLPM_M4, data = data_subset, missing = 'ML')
```

Inspect the correlations:

```
lavInspect(CLPM_M4.fit, "cov.lv")
```

```
       RQ11   RQ12   RQ21   RQ22    SE1    SE2
RQ11  7.813
RQ12  4.945  8.645
RQ21  8.391  5.831  9.761
RQ22  5.396  9.651  7.340 11.492
SE1   4.494  4.012  6.627  6.171 10.258
SE2   3.694  5.065  5.132  7.205  9.182 10.552
```

```
lavInspect(CLPM_M4.fit, "cor.lv")
```

```
      RQ11  RQ12  RQ21  RQ22   SE1   SE2
RQ11 1.000
RQ12 0.602 1.000
RQ21 0.961 0.635 1.000
RQ22 0.570 0.968 0.693 1.000
SE1  0.502 0.426 0.662 0.568 1.000
SE2  0.407 0.530 0.506 0.654 0.883 1.000
```

Obtain the results:

```
fitMeasures(CLPM_M4.fit)[c("chisq","df")]
```

```
   chisq       df
757.1568  90.0000
```

We proceed with the Chi-squared difference test with the previous model:
Df = 90 - 84 = 6
Check the constrained / freed means = 6
Chi-square difference = 757.1568 - 725.4913 = 31.6655
https://www.socscistatistics.com/pvalues/chidistribution.aspx

The $p$ value is .000019. Hence, the result is significant: $p < .05$.

Thus, we reject Model 4 and proceed with Model 3 (i.e., strong factorial invariance - with freed means = CLPM_M3.fit).

We can now move further by specifying the lagged effects between the latent variables.

## CLPM

```
clpmModel <- '

  #####################
  # MEASUREMENT MODEL #
  #####################

  # Factor models for RQ1 at 2 waves.
  RQ11 =~ L1 * THT1_SS  + L2 * TBT1_SS
```

```
RQ12 =~ L1 * TH_T2_SS + L2 * TB_T2_SS

# Factor models for RQ2 at 2 waves.
RQ21 =~ L3 * ACOMT1_SS + L4 * SATT1_SS
RQ22 =~ L3 * ACOMT2SS  + L4 * SAT_T2SS

# Factor models for SE at 2 waves.
SE1 =~ L5 * AB_T1_SS + L6 * DE_T1_SS + L7 * VI_T1_SS
SE2 =~ L5 * ABT2_SS  + L6 * DET2_SS  + L7 * VIT2_SS


# Constrained intercepts over time
THT1_SS ~ int_th*1
TH_T2_SS ~ int_th*1
TBT1_SS ~ int_tb*1
TB_T2_SS ~ int_tb*1
ACOMT1_SS ~ int_acom*1
ACOMT2SS ~ int_acom*1
SATT1_SS ~ int_sat*1
SAT_T2SS ~ int_sat*1
#
AB_T1_SS ~ int_ab*1
ABT2_SS  ~ int_ab*1
DE_T1_SS ~ int_de*1
DET2_SS ~ int_de*1
VI_T1_SS ~ int_vi*1
VIT2_SS ~ int_vi*1
#
SLT1_SS ~ int_sl*1
SLT2SS ~ int_sl*1


# Free latent means on t=2
RQ12 + RQ22 + SE2 + RQ11 + RQ21 + SE1 ~ 1


############
# DYNAMICS #
############

# Specify the lagged effects between the latent variables.
RQ12 ~ Phi11 * RQ11 + Phi12 * RQ21 + Phi13 * SE1 + Phi14 * SLT1_SS
RQ22 ~ Phi21 * RQ11 + Phi22 * RQ21 + Phi23 * SE1 + Phi24 * SLT1_SS
#
SE2 ~ Phi31 * RQ11 + Phi32 * RQ21 + Phi33 * SE1 + Phi34 * SLT1_SS
SLT2SS ~ Phi41 * RQ11 + Phi42 * RQ21 + Phi43 * SE1 + Phi44 * SLT1_SS


# Estimate the correlations within the same wave.
# T1
RQ11 ~~ RQ21 + SE1 + SLT1_SS
RQ21 ~~ SE1 + SLT1_SS
SE1 ~~ SLT1_SS
```

```
  # T2
  RQ12 ~~ RQ22 + SE2 + SLT2SS
  RQ22 ~~ SE2 + SLT2SS
  SE2 ~~ SLT2SS

'
```

Next, we fit the model with the lagged relations:

```
clpmUnc <- sem(clpmModel, data = data_subset, missing = 'ML')
```

Using the `summary` function we obtain the results of the model fit and estimates. The standardized solution contains the p-values of standardized effects.

```
fitMeasures(clpmUnc)
```

```
                     npar                      fmin                     chisq 
                   68.000                     0.220                   725.491 
                     srmr              srmr_bentler       srmr_bentler_nomean 
                    0.053                     0.053                     0.056 
```

```
stdClpmUnc <- standardizedsolution(clpmUnc, type = "std.all", se = TRUE, zstat = TRUE,
                                   pvalue = TRUE, ci = TRUE, level = 0.95, cov.std = TRUE,
                                   remove.eq = TRUE, remove.ineq = TRUE, remove.def = FALSE,
                                   partable = NULL, GLIST = NULL, est = NULL)
stdClpmUnc
```

```
         lhs op        rhs    label est.std    se       z pvalue ci.lower ci.upper
1       RQ11 =~    THT1_SS       L1   0.860 0.010  87.184  0.000    0.841    0.880
2       RQ11 =~    TBT1_SS       L2   0.892 0.009  98.450  0.000    0.874    0.909
3       RQ12 =~   TH_T2_SS       L1   0.891 0.009 104.577  0.000    0.874    0.907
4       RQ12 =~   TB_T2_SS       L2   0.933 0.007 132.024  0.000    0.920    0.947
5       RQ21 =~ ACOMT1_SS       L3   0.755 0.015  50.807  0.000    0.726    0.784
6       RQ21 =~   SATT1_SS       L4   0.797 0.014  57.832  0.000    0.770    0.824
7       RQ22 =~   ACOMT2SS       L3   0.850 0.011  77.025  0.000    0.829    0.872
8       RQ22 =~   SAT_T2SS       L4   0.863 0.011  81.286  0.000    0.843    0.884
9        SE1 =~    AB_T1_SS       L5   0.843 0.011  74.549  0.000    0.821    0.866
10       SE1 =~    DE_T1_SS       L6   0.840 0.012  68.560  0.000    0.816    0.864
11       SE1 =~    VI_T1_SS       L7   0.847 0.011  74.234  0.000    0.825    0.869
12       SE2 =~    ABT2_SS       L5   0.856 0.012  71.260  0.000    0.833    0.880
13       SE2 =~    DET2_SS       L6   0.836 0.013  66.520  0.000    0.811    0.860
14       SE2 =~    VIT2_SS       L7   0.858 0.012  72.268  0.000    0.835    0.881
15   THT1_SS ~1             int_th   9.880 0.732  13.505  0.000    8.446   11.314
16  TH_T2_SS ~1             int_th   9.700 0.740  13.105  0.000    8.249   11.150
17   TBT1_SS ~1             int_tb   9.375 0.758  12.376  0.000    7.891   10.860
18  TB_T2_SS ~1             int_tb   9.309 0.774  12.034  0.000    7.793   10.825
19 ACOMT1_SS ~1           int_acom   7.678 0.307  24.994  0.000    7.076    8.281
20  ACOMT2SS ~1           int_acom   7.773 0.317  24.505  0.000    7.151    8.395
21  SATT1_SS ~1            int_sat   8.035 0.325  24.716  0.000    7.398    8.672
22  SAT_T2SS ~1            int_sat   7.824 0.322  24.279  0.000    7.192    8.456
23  AB_T1_SS ~1             int_ab   9.533 0.345  27.664  0.000    8.857   10.208
24   ABT2_SS ~1             int_ab   9.558 0.386  24.730  0.000    8.801   10.316
25  DE_T1_SS ~1             int_de  10.664 0.351  30.340  0.000    9.975   11.353
26   DET2_SS ~1             int_de  10.484 0.396  26.507  0.000    9.709   11.259
27  VI_T1_SS ~1             int_vi   9.709 0.348  27.927  0.000    9.028   10.391
28   VIT2_SS ~1             int_vi   9.714 0.391  24.834  0.000    8.947   10.480
```

```
29   SLT1_SS ~1           int_sl   4.075 0.094  43.518  0.000    3.891    4.258
30    SLT2SS ~1           int_sl   3.864 0.098  39.350  0.000    3.672    4.057
31      RQ12 ~1                   -2.610 1.541  -1.694  0.090   -5.629    0.410
32      RQ22 ~1                   -2.196 1.921  -1.143  0.253   -5.962    1.569
33       SE2 ~1                    0.759 1.806   0.420  0.674   -2.782    4.300
34      RQ11 ~1                   -5.844 0.833  -7.013  0.000   -7.477   -4.210
35      RQ21 ~1                   -5.302 0.390 -13.588  0.000   -6.067   -4.538
36       SE1 ~1                   -7.281 0.387 -18.821  0.000   -8.039   -6.523
37      RQ12  ~       RQ11  Phi11  0.149 0.542   0.275  0.783   -0.913    1.211
38      RQ12  ~       RQ21  Phi12  0.415 0.685   0.606  0.544   -0.927    1.758
39      RQ12  ~        SE1  Phi13  0.034 0.147   0.232  0.817   -0.254    0.322
40      RQ12  ~    SLT1_SS  Phi14  0.046 0.142   0.324  0.746   -0.233    0.325
41      RQ22  ~       RQ11  Phi21  1.721 0.889   1.936  0.053   -0.021    3.462
42      RQ22  ~       RQ21  Phi22 -1.808 1.137  -1.591  0.112   -4.036    0.420
43      RQ22  ~        SE1  Phi23  0.597 0.223   2.671  0.008    0.159    1.034
44      RQ22  ~    SLT1_SS  Phi24  0.517 0.215   2.408  0.016    0.096    0.937
45       SE2  ~       RQ11  Phi31 -1.018 0.663  -1.534  0.125   -2.318    0.282
46       SE2  ~       RQ21  Phi32  1.319 0.848   1.556  0.120   -0.342    2.981
47       SE2  ~        SE1  Phi33  0.752 0.168   4.467  0.000    0.422    1.082
48       SE2  ~    SLT1_SS  Phi34 -0.389 0.163  -2.394  0.017   -0.708   -0.071
49    SLT2SS  ~       RQ11  Phi41  0.716 0.544   1.316  0.188   -0.351    1.783
50    SLT2SS  ~       RQ21  Phi42 -0.708 0.693  -1.023  0.306   -2.066    0.649
51    SLT2SS  ~        SE1  Phi43  0.289 0.142   2.027  0.043    0.009    0.568
52    SLT2SS  ~    SLT1_SS  Phi44  0.596 0.136   4.388  0.000    0.330    0.862
53      RQ11 ~~       RQ21         0.997 0.013  75.177  0.000    0.971    1.023
54      RQ11 ~~        SE1         0.502 0.028  17.884  0.000    0.447    0.558
55      RQ11 ~~    SLT1_SS         0.675 0.019  34.754  0.000    0.637    0.713
56      RQ21 ~~        SE1         0.687 0.026  26.784  0.000    0.637    0.737
57      RQ21 ~~    SLT1_SS         0.815 0.017  47.545  0.000    0.782    0.849
58       SE1 ~~    SLT1_SS         0.681 0.020  34.112  0.000    0.642    0.720
59      RQ12 ~~       RQ22         0.805 0.102   7.909  0.000    0.605    1.004
60      RQ12 ~~        SE2         0.484 0.121   3.987  0.000    0.246    0.722
61      RQ12 ~~     SLT2SS         0.509 0.068   7.511  0.000    0.376    0.642
62      RQ22 ~~        SE2         0.140 0.223   0.628  0.530   -0.297    0.577
63      RQ22 ~~     SLT2SS         0.661 0.068   9.773  0.000    0.528    0.794
64       SE2 ~~     SLT2SS         0.382 0.173   2.208  0.027    0.043    0.721
65   THT1_SS ~~    THT1_SS         0.260 0.017  15.290  0.000    0.226    0.293
66   TBT1_SS ~~    TBT1_SS         0.205 0.016  12.694  0.000    0.173    0.237
67  TH_T2_SS ~~   TH_T2_SS         0.207 0.015  13.624  0.000    0.177    0.236
68  TB_T2_SS ~~   TB_T2_SS         0.129 0.013   9.748  0.000    0.103    0.155
69 ACOMT1_SS ~~  ACOMT1_SS         0.430 0.022  19.201  0.000    0.387    0.474
70  SATT1_SS ~~   SATT1_SS         0.365 0.022  16.631  0.000    0.322    0.408
71  ACOMT2SS ~~   ACOMT2SS         0.277 0.019  14.767  0.000    0.240    0.314
72  SAT_T2SS ~~   SAT_T2SS         0.254 0.018  13.873  0.000    0.219    0.290
73  AB_T1_SS ~~   AB_T1_SS         0.289 0.019  15.128  0.000    0.251    0.326
74  DE_T1_SS ~~   DE_T1_SS         0.295 0.021  14.329  0.000    0.254    0.335
75  VI_T1_SS ~~   VI_T1_SS         0.283 0.019  14.635  0.000    0.245    0.321
76   ABT2_SS ~~    ABT2_SS         0.267 0.021  12.976  0.000    0.227    0.307
77   DET2_SS ~~    DET2_SS         0.301 0.021  14.345  0.000    0.260    0.343
78   VIT2_SS ~~    VIT2_SS         0.264 0.020  12.971  0.000    0.224    0.304
79    SLT2SS ~~     SLT2SS         0.510 0.064   7.992  0.000    0.385    0.635
80   SLT1_SS ~~    SLT1_SS         1.000 0.000      NA     NA    1.000    1.000
81      RQ11 ~~       RQ11         1.000 0.000      NA     NA    1.000    1.000
82      RQ12 ~~       RQ12         0.611 0.055  11.157  0.000    0.504    0.719
```

```
83       RQ21 ~~       RQ21             1.000 0.000       NA      NA     1.000     1.000
84       RQ22 ~~       RQ22             0.707 0.164    4.308  0.000     0.385     1.029
85        SE1 ~~        SE1             1.000 0.000       NA      NA     1.000     1.000
86        SE2 ~~        SE2             0.292 0.099    2.946  0.003     0.098     0.486
```

## GORICA

Next, you will find two ways of applying the GORICA to a lavaan model.

### Input option 1

**Extract estimates of interest and their covariance matrix** Next, we extract the standardized estimates of interest (thus, based on the parameters mentioned in our set of hypotheses) and their covariance matrix, which can be used as input for the `goric` function.

```r
# indices of estimates of interest
indices <- 37:52

# select estimates from the column 'Std.all' in the results summary above
est <- stdClpmUnc[indices, 'est.std']

names(est) <- c("RQ12_RQ11", "RQ12_RQ21", "RQ12_SE1", "RQ12_SL1",
                "RQ22_RQ11", "RQ22_RQ21", "RQ22_SE1", "RQ22_SL1",
                "SE2_RQ11", "SE2_RQ21", "SE2_SE1", "SE2_SL1",
                "SL2_RQ11", "SL2_RQ21", "SL2_SE1", "SL2_SL1"
)

# the covariance matrix for these estimates
vcov <- lavInspect(clpmUnc, "vcov.std.all")[indices-6, indices-6]
```

**Hypotheses** Next, we specify the hypotheses to be evaluated (which are known before seeing the data). Note the use of the `abs` function: This is because we are interested in the size of the relations and we want to compare absolute effects. In cases where the sign of the values is of interest, one should look at the regular/raw/non-absolute effect (e.g., estimate_x > .3 or estimate_y < 0).

Here, there are two sets of hypotheses, $H1\_Q1$ and $H1\_Q2$, which focus on different relations in the model. The decisions of whether multiple hypotheses should be split in different sets and how to divide them are driven by theory, and depend on what the researchers intend to evaluate. When multiple hypotheses are included in one set, as in $H1\_Q2$, they are handled by the `goric` function as a whole, not individually.

```r
# Q1: Phi_21 > Phi_12
H1_Q1 <- "
abs(RQ22_RQ11) > abs(RQ12_RQ21)
"
#
# Q2
H1_Q2 <- "
abs(SE2_RQ11) > abs(RQ12_SE1);
abs(SL2_RQ11) > abs(RQ12_SL1);
abs(SE2_RQ21) > abs(RQ22_SE1);
abs(SL2_RQ21) > abs(RQ22_SL1)
"
```

**GORICA** We obtain the GORICA results for $H1\_Q1$ and $H1\_Q2$ in two steps (where each is evaluated against their own complement). Note the use of `set.seed` to ensure that the results are reproducible.

```r
set.seed(123)

#H1_Q1 vs its complement, using GORICA
goricaResults_Q1 <- goric(est, VCOV = vcov, # then, default: type = "gorica"
                          hypotheses = list(H1_Q1=H1_Q1))

goricaResults_Q1
```

```
restriktor (0.6-10): generalized order-restricted information criterion approximation:

Results:
       model  loglik  penalty   gorica  loglik.weights  penalty.weights  gorica.weights
1      H1_Q1  22.439   15.500  -13.878           0.632            0.500           0.632
2 complement  21.898   15.500  -12.796           0.368            0.500           0.368

Conclusion:
The order-restricted hypothesis 'H1_Q1' has 1.72 times more support than its complement.
```

```r
#summary(goricaResults_Q1)
```

The output shows that the order-restricted hypothesis $H1\_Q1$ has 1.7 times more support than its complement.

```r
set.seed(123)

#H1_Q2 vs its complement, using GORICA
goricaResults_Q2 <- goric(est, VCOV = vcov,
                          hypotheses = list(H1_Q2=H1_Q2))

goricaResults_Q2
```

```
restriktor (0.6-10): generalized order-restricted information criterion approximation:

Results:
       model  loglik  penalty   gorica  loglik.weights  penalty.weights  gorica.weights
1      H1_Q2  22.439   14.856  -15.167           0.513            0.568           0.581
2 complement  22.387   15.131  -14.512           0.487            0.432           0.419

Conclusion:
The order-restricted hypothesis 'H1_Q2' has 1.39 times more support than its complement.
```

```r
#summary(goricaResults_Q2)
```

Furthermore, the order-restricted hypothesis $H1\_Q2$ has 1.4 times more support than its complement.

Note 1: The results hold for the chosen time interval. That is, the results are time-interval dependent. At the end, more information is given.

Note 2: The log-likelihood (loglik) weights seem to be quite close. This could indicate that one or more of the inequality constraints can be replaced by (about-)equality constraints. One could investigate with the benchmarks function, using 'output_type = "rlw" ', whether the loglik weights indeed are close.
For more information, see the guidelines ('Guidelines_output_GORIC.html') and/or the benchmark tutorial on https://github.com/rebeccakuiper/Tutorials.

### Input option 2

Instead of extracting the (standardized) estimates and their covariance matrix, you can use the lavaan object. The easiest is to label the estimates (of interest); like done earlier in this example. You can then use these labels in specifying your hypotheses. To make a fair comparison, we need to look at the standardized estimates, so we should include 'standardized = T'.

The R code to do it this ways follows next.
Note the use of `set.seed` to ensure that the results are reproducible.

```
# Q1
H1_Q1_lav <- "
abs(Phi21) > abs(Phi12)
"
#
# Q2
H1_Q2_lav <- "
abs(Phi31) > abs(Phi13);
abs(Phi41) > abs(Phi14);
abs(Phi32) > abs(Phi23);
abs(Phi42) > abs(Phi24)
"

set.seed(123)
#H1_Q2_lav vs its complement, using GORICA
goricaResults_Q2_lav <- goric(clpmUnc, # then, default: type = "gorica"
                              standardized = T,
                              hypotheses = list(H1_Q2_lav=H1_Q2_lav))

goricaResults_Q2_lav
```

```
restriktor (0.6-10): generalized order-restricted information criterion approximation:

Results:
        model  loglik  penalty   gorica  loglik.weights  penalty.weights  gorica.weights
1   H1_Q2_lav  48.468   26.856  -43.224           0.513            0.568           0.581
2  complement  48.415   27.131  -42.568           0.487            0.432           0.419

Conclusion:
The order-restricted hypothesis 'H1_Q2_lav' has 1.39 times more support than its complement.
```

```
#summary(goricaResults_Q2_lav)
```

This of course renders the same results as above (and thus also the same notes hold true).

# Example 2: Measurement Level Analysis

## R packages

First, install and call the `lavaan` library to create a CLPM and the `restriktor` library to load the `goric` function. If needed, it is possible to view the description of the function with the `?` operator or the `help` command.

The code presented here also requires the `tidyverse` package for data manipulation.

```
# To install restriktor in R:
#if (!require("restriktor")) install.packages("restriktor")

# To install restriktor from github:
# if (!require("devtools")) install.packages("devtools")
# library(devtools)
# install_github("LeonardV/restriktor")
library(restriktor)

# print docs in the help-tab to view arguments and explanations for the function
#?goric

# To install lavaan in R:
# if (!require("lavaan")) install.packages("lavaan")
library(lavaan)

# To install tidyverse in R:
# if (!require("tidyverse")) install.packages("tidyverse")
library(tidyverse)
```

## Data

Upload the data set to the R environment and select the columns used for analysis. The id column is renamed to $ID$ and the code in the data set for missing numbers -999.00 is replaced with `NAs`.

```
data <- read.table("data/CLPM.dat", header = T)
colnames(data)[1] <- "ID"
data <- replace(data , data == -999.00, NA)

data_subset <- select(data,
                  THT1_SS,
                  TBT1_SS,
                  ACOMT1_SS,
                  SATT1_SS,
                  AB_T1_SS,
                  DE_T1_SS,
                  VI_T1_SS,
                  SLT1_SS,
                  TH_T2_SS,
                  TB_T2_SS,
                  ACOMT2SS,
                  SAT_T2SS,
                  ABT2_SS,
                  DET2_SS,
                  VIT2_SS,
                  SLT2SS)
```

## CLPM

Next, we fit the CLPM on sum scores using the `lavaan` package. Here we specify all the relations of the model.

```
clpmModel_2 <- '

  ############
  # DYNAMICS #
  ############

  # Specify the lagged effects between the latent variables.
  TH_T2_SS ~ THT1_SS + TBT1_SS + ACOMT1_SS + SATT1_SS + AB_T1_SS + DE_T1_SS + VI_T1_SS + SLT1_SS
  TB_T2_SS ~ THT1_SS + TBT1_SS + ACOMT1_SS + SATT1_SS + AB_T1_SS + DE_T1_SS + VI_T1_SS + SLT1_SS
  ACOMT2SS ~ THT1_SS + TBT1_SS + ACOMT1_SS + SATT1_SS + AB_T1_SS + DE_T1_SS + VI_T1_SS + SLT1_SS
  SAT_T2SS ~ THT1_SS + TBT1_SS + ACOMT1_SS + SATT1_SS + AB_T1_SS + DE_T1_SS + VI_T1_SS + SLT1_SS
  #
  ABT2_SS ~ THT1_SS + TBT1_SS + ACOMT1_SS + SATT1_SS + AB_T1_SS + DE_T1_SS + VI_T1_SS + SLT1_SS
  DET2_SS ~ THT1_SS + TBT1_SS + ACOMT1_SS + SATT1_SS + AB_T1_SS + DE_T1_SS + VI_T1_SS + SLT1_SS
  VIT2_SS ~ THT1_SS + TBT1_SS + ACOMT1_SS + SATT1_SS + AB_T1_SS + DE_T1_SS + VI_T1_SS + SLT1_SS
  #
  SLT2SS ~ THT1_SS + TBT1_SS + ACOMT1_SS + SATT1_SS + AB_T1_SS + DE_T1_SS + VI_T1_SS + SLT1_SS


  # Estimate the correlations within the same wave.
  # T1
   ACOMT1_SS ~~ THT1_SS + TBT1_SS + SATT1_SS + AB_T1_SS + DE_T1_SS + VI_T1_SS + SLT1_SS
   THT1_SS ~~ TBT1_SS + SATT1_SS + AB_T1_SS + DE_T1_SS + VI_T1_SS + SLT1_SS
   TBT1_SS ~~ SATT1_SS + AB_T1_SS + DE_T1_SS + VI_T1_SS + SLT1_SS
   SATT1_SS ~~ AB_T1_SS + DE_T1_SS + VI_T1_SS + SLT1_SS
   AB_T1_SS ~~ DE_T1_SS + VI_T1_SS + SLT1_SS
   DE_T1_SS ~~ VI_T1_SS + SLT1_SS
   VI_T1_SS ~~ SLT1_SS
   # T2
   TH_T2_SS ~~ TB_T2_SS + SAT_T2SS + ACOMT2SS + ABT2_SS + DET2_SS + VIT2_SS + SLT2SS
   TB_T2_SS ~~ SAT_T2SS + ACOMT2SS + ABT2_SS + DET2_SS + VIT2_SS + SLT2SS
   SAT_T2SS ~~ ACOMT2SS + ABT2_SS + DET2_SS + VIT2_SS + SLT2SS
   ACOMT2SS ~~ ABT2_SS + DET2_SS + VIT2_SS + SLT2SS
   ABT2_SS ~~ DET2_SS + VIT2_SS + SLT2SS
   DET2_SS ~~ VIT2_SS + SLT2SS
   VIT2_SS ~~ SLT2SS

'
```

We fit the model using the sem function:

```
clpmUnc_2 <- sem(clpmModel_2, data = data_subset, missing = 'ML')
```

Using the summary function we obtain the results of the model fit and estimates. The standardized solution contains the p-values of standardized effects.

```
fitMeasures(clpmUnc_2)
```

```
                        npar                      fmin                    chisq
                     152.000                     0.000                    0.000
                        srmr              srmr_bentler      srmr_bentler_nomean
                       0.000                     0.000                    0.000
```

```
stdClpmUnc_2 <- standardizedsolution(clpmUnc_2, type = "std.all", se = TRUE, zstat = TRUE,
                                   pvalue = TRUE, ci = TRUE, level = 0.95, cov.std = TRUE,
                                   remove.eq = TRUE, remove.ineq = TRUE, remove.def = FALSE,
                                   partable = NULL, GLIST = NULL, est = NULL)
stdClpmUnc_2
```

```
          lhs op       rhs est.std    se      z pvalue ci.lower ci.upper
1    TH_T2_SS  ~   THT1_SS   0.191 0.094  2.033  0.042    0.007    0.374
2    TH_T2_SS  ~   TBT1_SS   0.072 0.098  0.739  0.460   -0.120    0.265
3    TH_T2_SS  ~ ACOMT1_SS  -0.017 0.076 -0.225  0.822   -0.167    0.132
4    TH_T2_SS  ~  SATT1_SS   0.122 0.093  1.319  0.187   -0.059    0.304
```

```
5    TH_T2_SS  ~  AB_T1_SS  -0.030 0.083 -0.362  0.718   -0.192    0.132
6    TH_T2_SS  ~  DE_T1_SS   0.075 0.101  0.738  0.460   -0.124    0.273
7    TH_T2_SS  ~  VI_T1_SS   0.086 0.088  0.981  0.327   -0.086    0.259
8    TH_T2_SS  ~   SLT1_SS   0.140 0.086  1.617  0.106   -0.030    0.309
9    TB_T2_SS  ~   THT1_SS   0.019 0.092  0.203  0.839   -0.162    0.200
10   TB_T2_SS  ~   TBT1_SS   0.384 0.094  4.084  0.000    0.200    0.568
11   TB_T2_SS  ~ ACOMT1_SS   0.019 0.074  0.257  0.797   -0.126    0.164
12   TB_T2_SS  ~  SATT1_SS   0.051 0.091  0.562  0.574   -0.127    0.230
13   TB_T2_SS  ~  AB_T1_SS  -0.106 0.081 -1.299  0.194   -0.265    0.054
14   TB_T2_SS  ~  DE_T1_SS   0.037 0.101  0.361  0.718   -0.162    0.235
15   TB_T2_SS  ~  VI_T1_SS   0.127 0.087  1.456  0.145   -0.044    0.297
16   TB_T2_SS  ~   SLT1_SS   0.142 0.085  1.679  0.093   -0.024    0.308
17   ACOMT2SS  ~   THT1_SS  -0.027 0.087 -0.315  0.753   -0.198    0.143
18   ACOMT2SS  ~   TBT1_SS   0.002 0.091  0.023  0.981   -0.176    0.180
19   ACOMT2SS  ~ ACOMT1_SS   0.329 0.068  4.838  0.000    0.195    0.462
20   ACOMT2SS  ~  SATT1_SS   0.018 0.086  0.204  0.838   -0.151    0.186
21   ACOMT2SS  ~  AB_T1_SS   0.022 0.076  0.288  0.773   -0.128    0.171
22   ACOMT2SS  ~  DE_T1_SS   0.115 0.094  1.230  0.219   -0.068    0.299
23   ACOMT2SS  ~  VI_T1_SS   0.078 0.082  0.959  0.337   -0.082    0.238
24   ACOMT2SS  ~   SLT1_SS   0.202 0.079  2.547  0.011    0.047    0.357
25   SAT_T2SS  ~   THT1_SS  -0.081 0.090 -0.899  0.368   -0.257    0.095
26   SAT_T2SS  ~   TBT1_SS   0.126 0.094  1.343  0.179   -0.058    0.310
27   SAT_T2SS  ~ ACOMT1_SS   0.033 0.073  0.456  0.648   -0.109    0.176
28   SAT_T2SS  ~  SATT1_SS   0.325 0.087  3.751  0.000    0.155    0.494
29   SAT_T2SS  ~  AB_T1_SS  -0.088 0.079 -1.111  0.266   -0.242    0.067
30   SAT_T2SS  ~  DE_T1_SS   0.098 0.097  1.013  0.311   -0.092    0.288
31   SAT_T2SS  ~  VI_T1_SS   0.125 0.084  1.487  0.137   -0.040    0.289
32   SAT_T2SS  ~   SLT1_SS   0.147 0.082  1.796  0.073   -0.013    0.308
33    ABT2_SS  ~   THT1_SS   0.001 0.079  0.014  0.989   -0.154    0.156
34    ABT2_SS  ~   TBT1_SS   0.072 0.085  0.845  0.398   -0.094    0.238
35    ABT2_SS  ~ ACOMT1_SS  -0.063 0.064 -0.974  0.330   -0.189    0.064
36    ABT2_SS  ~  SATT1_SS  -0.019 0.082 -0.237  0.813   -0.180    0.141
37    ABT2_SS  ~  AB_T1_SS   0.645 0.063 10.167  0.000    0.521    0.770
38    ABT2_SS  ~  DE_T1_SS   0.118 0.086  1.368  0.171   -0.051    0.286
39    ABT2_SS  ~  VI_T1_SS   0.028 0.074  0.378  0.706   -0.116    0.172
40    ABT2_SS  ~   SLT1_SS   0.018 0.070  0.259  0.795   -0.120    0.156
41    DET2_SS  ~   THT1_SS  -0.124 0.080 -1.550  0.121   -0.280    0.033
42    DET2_SS  ~   TBT1_SS   0.214 0.085  2.500  0.012    0.046    0.381
43    DET2_SS  ~ ACOMT1_SS  -0.105 0.065 -1.625  0.104   -0.232    0.022
44    DET2_SS  ~  SATT1_SS   0.062 0.082  0.751  0.453   -0.100    0.223
45    DET2_SS  ~  AB_T1_SS   0.054 0.070  0.771  0.441   -0.083    0.192
46    DET2_SS  ~  DE_T1_SS   0.685 0.078  8.737  0.000    0.531    0.838
47    DET2_SS  ~  VI_T1_SS   0.037 0.075  0.498  0.619   -0.109    0.184
48    DET2_SS  ~   SLT1_SS  -0.051 0.071 -0.708  0.479   -0.190    0.089
49    VIT2_SS  ~   THT1_SS   0.025 0.079  0.311  0.756   -0.131    0.180
50    VIT2_SS  ~   TBT1_SS   0.151 0.085  1.770  0.077   -0.016    0.318
51    VIT2_SS  ~ ACOMT1_SS  -0.155 0.064 -2.417  0.016   -0.281   -0.029
52    VIT2_SS  ~  SATT1_SS  -0.021 0.082 -0.255  0.798   -0.183    0.141
53    VIT2_SS  ~  AB_T1_SS   0.230 0.069  3.338  0.001    0.095    0.365
54    VIT2_SS  ~  DE_T1_SS   0.072 0.088  0.823  0.411   -0.100    0.244
55    VIT2_SS  ~  VI_T1_SS   0.574 0.069  8.372  0.000    0.440    0.708
56    VIT2_SS  ~   SLT1_SS  -0.074 0.070 -1.057  0.291   -0.212    0.064
57     SLT2SS  ~   THT1_SS  -0.026 0.081 -0.318  0.751   -0.184    0.133
58     SLT2SS  ~   TBT1_SS   0.021 0.086  0.244  0.807   -0.148    0.190
```

```
59     SLT2SS  ~ ACOMT1_SS  -0.013 0.066 -0.203  0.839    -0.142    0.116
60     SLT2SS  ~  SATT1_SS   0.215 0.082  2.631  0.009     0.055    0.375
61     SLT2SS  ~  AB_T1_SS   0.083 0.070  1.186  0.236    -0.054    0.221
62     SLT2SS  ~  DE_T1_SS   0.134 0.087  1.537  0.124    -0.037    0.306
63     SLT2SS  ~  VI_T1_SS  -0.043 0.075 -0.568  0.570    -0.190    0.104
64     SLT2SS  ~   SLT1_SS   0.468 0.068  6.840  0.000     0.334    0.602
65    THT1_SS ~~ ACOMT1_SS   0.611 0.019 31.414  0.000     0.573    0.649
66    TBT1_SS ~~ ACOMT1_SS   0.637 0.018 34.602  0.000     0.601    0.673
67  ACOMT1_SS ~~  SATT1_SS   0.603 0.020 30.608  0.000     0.564    0.641
68  ACOMT1_SS ~~  AB_T1_SS   0.467 0.024 19.248  0.000     0.419    0.515
69  ACOMT1_SS ~~  DE_T1_SS   0.586 0.020 28.750  0.000     0.546    0.626
70  ACOMT1_SS ~~  VI_T1_SS   0.528 0.022 23.450  0.000     0.484    0.572
71  ACOMT1_SS ~~   SLT1_SS   0.651 0.018 36.260  0.000     0.616    0.686
72    THT1_SS ~~   TBT1_SS   0.767 0.013 59.792  0.000     0.742    0.792
73    THT1_SS ~~  SATT1_SS   0.710 0.015 46.179  0.000     0.680    0.741
74    THT1_SS ~~  AB_T1_SS   0.281 0.029  9.855  0.000     0.225    0.337
75    THT1_SS ~~  DE_T1_SS   0.449 0.025 18.208  0.000     0.401    0.498
76    THT1_SS ~~  VI_T1_SS   0.343 0.027 12.482  0.000     0.289    0.397
77    THT1_SS ~~   SLT1_SS   0.592 0.020 29.322  0.000     0.552    0.631
78    TBT1_SS ~~  SATT1_SS   0.740 0.014 52.720  0.000     0.713    0.768
79    TBT1_SS ~~  AB_T1_SS   0.323 0.028 11.659  0.000     0.269    0.377
80    TBT1_SS ~~  DE_T1_SS   0.454 0.024 18.541  0.000     0.406    0.502
81    TBT1_SS ~~  VI_T1_SS   0.377 0.027 14.158  0.000     0.325    0.429
82    TBT1_SS ~~   SLT1_SS   0.594 0.020 29.545  0.000     0.555    0.633
83   SATT1_SS ~~  AB_T1_SS   0.311 0.028 11.178  0.000     0.257    0.366
84   SATT1_SS ~~  DE_T1_SS   0.467 0.024 19.374  0.000     0.420    0.514
85   SATT1_SS ~~  VI_T1_SS   0.390 0.026 14.824  0.000     0.338    0.441
86   SATT1_SS ~~   SLT1_SS   0.621 0.019 32.617  0.000     0.584    0.659
87   AB_T1_SS ~~  DE_T1_SS   0.688 0.016 42.342  0.000     0.656    0.720
88   AB_T1_SS ~~  VI_T1_SS   0.760 0.013 58.074  0.000     0.735    0.786
89   AB_T1_SS ~~   SLT1_SS   0.482 0.024 20.323  0.000     0.435    0.528
90   DE_T1_SS ~~  VI_T1_SS   0.680 0.017 40.772  0.000     0.647    0.712
91   DE_T1_SS ~~   SLT1_SS   0.703 0.016 44.880  0.000     0.672    0.734
92   VI_T1_SS ~~   SLT1_SS   0.535 0.022 24.146  0.000     0.491    0.578
93   TH_T2_SS ~~  TB_T2_SS   0.781 0.019 42.220  0.000     0.745    0.818
94   TH_T2_SS ~~  SAT_T2SS   0.685 0.024 28.242  0.000     0.637    0.732
95   TH_T2_SS ~~  ACOMT2SS   0.643 0.029 22.348  0.000     0.586    0.699
96   TH_T2_SS ~~   ABT2_SS   0.231 0.054  4.268  0.000     0.125    0.338
97   TH_T2_SS ~~   DET2_SS   0.353 0.050  7.097  0.000     0.256    0.451
98   TH_T2_SS ~~   VIT2_SS   0.229 0.055  4.180  0.000     0.122    0.337
99   TH_T2_SS ~~    SLT2SS   0.494 0.041 12.049  0.000     0.414    0.575
100  TB_T2_SS ~~  SAT_T2SS   0.727 0.023 32.276  0.000     0.683    0.772
101  TB_T2_SS ~~  ACOMT2SS   0.698 0.027 26.170  0.000     0.646    0.750
102  TB_T2_SS ~~   ABT2_SS   0.303 0.055  5.496  0.000     0.195    0.412
103  TB_T2_SS ~~   DET2_SS   0.368 0.052  7.086  0.000     0.266    0.470
104  TB_T2_SS ~~   VIT2_SS   0.259 0.057  4.519  0.000     0.147    0.371
105  TB_T2_SS ~~    SLT2SS   0.502 0.043 11.688  0.000     0.418    0.586
106  ACOMT2SS ~~  SAT_T2SS   0.652 0.028 23.001  0.000     0.596    0.708
107  SAT_T2SS ~~   ABT2_SS   0.254 0.054  4.706  0.000     0.148    0.360
108  SAT_T2SS ~~   DET2_SS   0.328 0.051  6.469  0.000     0.229    0.427
109  SAT_T2SS ~~   VIT2_SS   0.249 0.055  4.533  0.000     0.141    0.356
110  SAT_T2SS ~~    SLT2SS   0.494 0.041 12.185  0.000     0.415    0.574
111  ACOMT2SS ~~   ABT2_SS   0.346 0.052  6.706  0.000     0.245    0.447
112  ACOMT2SS ~~   DET2_SS   0.453 0.046  9.769  0.000     0.362    0.544
```

```
113  ACOMT2SS ~~   VIT2_SS   0.358 0.052  6.850  0.000    0.256    0.461
114  ACOMT2SS ~~    SLT2SS   0.549 0.039 14.179  0.000    0.474    0.625
115   ABT2_SS ~~   DET2_SS   0.568 0.041 13.762  0.000    0.487    0.648
116   ABT2_SS ~~   VIT2_SS   0.621 0.037 16.797  0.000    0.549    0.693
117   ABT2_SS ~~    SLT2SS   0.314 0.055  5.762  0.000    0.207    0.421
118   DET2_SS ~~   VIT2_SS   0.547 0.044 12.402  0.000    0.461    0.633
119   DET2_SS ~~    SLT2SS   0.508 0.045 11.328  0.000    0.420    0.596
120   VIT2_SS ~~    SLT2SS   0.377 0.054  7.009  0.000    0.272    0.483
121  TH_T2_SS ~~  TH_T2_SS   0.722 0.043 16.911  0.000    0.639    0.806
122  TB_T2_SS ~~  TB_T2_SS   0.646 0.044 14.620  0.000    0.559    0.733
123  ACOMT2SS ~~  ACOMT2SS   0.600 0.044 13.611  0.000    0.513    0.686
124  SAT_T2SS ~~  SAT_T2SS   0.649 0.043 15.050  0.000    0.565    0.734
125   ABT2_SS ~~   ABT2_SS   0.434 0.040 10.971  0.000    0.356    0.511
126   DET2_SS ~~   DET2_SS   0.455 0.041 11.069  0.000    0.374    0.535
127   VIT2_SS ~~   VIT2_SS   0.420 0.038 11.004  0.000    0.345    0.495
128    SLT2SS ~~    SLT2SS   0.464 0.042 11.136  0.000    0.382    0.545
129   THT1_SS ~~   THT1_SS   1.000 0.000     NA     NA    1.000    1.000
130   TBT1_SS ~~   TBT1_SS   1.000 0.000     NA     NA    1.000    1.000
131 ACOMT1_SS ~~ ACOMT1_SS   1.000 0.000     NA     NA    1.000    1.000
132  SATT1_SS ~~  SATT1_SS   1.000 0.000     NA     NA    1.000    1.000
133  AB_T1_SS ~~  AB_T1_SS   1.000 0.000     NA     NA    1.000    1.000
134  DE_T1_SS ~~  DE_T1_SS   1.000 0.000     NA     NA    1.000    1.000
135  VI_T1_SS ~~  VI_T1_SS   1.000 0.000     NA     NA    1.000    1.000
136   SLT1_SS ~~   SLT1_SS   1.000 0.000     NA     NA    1.000    1.000
137  TH_T2_SS ~1             1.839 0.286  6.420  0.000    1.277    2.400
138  TB_T2_SS ~1             1.186 0.275  4.308  0.000    0.646    1.726
139  ACOMT2SS ~1             0.759 0.258  2.941  0.003    0.253    1.265
140  SAT_T2SS ~1             0.925 0.264  3.499  0.000    0.407    1.443
141   ABT2_SS ~1             0.305 0.237  1.284  0.199   -0.160    0.770
142   DET2_SS ~1             1.058 0.244  4.340  0.000    0.580    1.536
143   VIT2_SS ~1             0.466 0.240  1.943  0.052   -0.004    0.937
144    SLT2SS ~1             0.365 0.237  1.540  0.124   -0.100    0.830
145   THT1_SS ~1             4.869 0.111 43.722  0.000    4.651    5.088
146   TBT1_SS ~1             4.140 0.096 43.136  0.000    3.952    4.329
147 ACOMT1_SS ~1             3.686 0.086 42.940  0.000    3.518    3.854
148  SATT1_SS ~1             3.805 0.089 42.896  0.000    3.631    3.979
149  AB_T1_SS ~1             3.429 0.080 42.754  0.000    3.272    3.586
150  DE_T1_SS ~1             4.508 0.102 44.130  0.000    4.308    4.708
151  VI_T1_SS ~1             3.542 0.083 42.581  0.000    3.379    3.705
152   SLT1_SS ~1             4.082 0.094 43.588  0.000    3.899    4.266
```

In this case, the results shows we obtain a ‘perfect’ model fit, that is because the degrees of freedom are 0, meaning the model is saturated (i.e., you have as much parameters as you have data points).

## GORICA

We select the estimates relevant to our hypotheses in order to use the goric function.

```r
# indices of estimates of interest
indices_2 <- 1:64

# select estimates from the column 'Std.all' in the results summary above
est_2 <- stdClpmUnc_2[indices_2, 'est.std']

names(est_2) <- c("TH2_TH1", "TH2_TB1", "TH2_ACOM1", "TH2_SAT1", "TH2_AB1", "TH2_DE1", "TH2_VI1", "TH2_
                "TB2_TH1", "TB2_TB1", "TB2_ACOM1", "TB2_SAT1", "TB2_AB1", "TB2_DE1", "TB2_VI1", "TB2_SL
                "ACOM2_TH1", "ACOM2_TB1", "ACOM2_ACOM1", "ACOM2_SAT1", "ACOM2_AB1", "ACOM2_DE1", "ACOM2_
                "SAT2_TH1", "SAT2_TB1", "SAT2_ACOM1", "SATM2_SAT1", "SAT2_AB1", "SAT2_DE1", "SAT2_VI1",
                #
                "AB2_TH1", "AB2_TB1", "AB2_ACOM1", "AB2_SAT1", "AB2_AB1", "AB2_DE1", "AB2_VI1", "AB2_SL
                "DE2_TH1", "DE2_TB1", "DE2_ACOM1", "DE2_SAT1", "DE2_AB1", "DE2_DE1", "DE2_VI1", "DE2_SL
                "VI2_TH1", "VI2_TB1", "VI2_ACOM1", "VI2_SAT1", "VI2_AB1", "VI2_DE1", "VI2_VI1", "VI2_SL
                #
                "SL2_TH1", "SL2_TB1", "SL2_ACOM1", "SL2_SAT1", "SL2_AB1", "SL2_DE1", "SL2_VI1", "SL2_SL
)

# the covariance matrix for these estimates
vcov_2 <- lavInspect(clpmUnc_2, "vcov.std.all")[indices_2, indices_2]
```

Next, we specify the hypotheses to be evaluated. Note the use of the use of the `abs` function; that is because we are interested in the size of the relations and we want to compare absolute effects. In cases where the sign of the values is of interest, one should not use absolute values (e.g., estimate_x > .3 or estimate_y < 0).

Here, there are two sets of hypotheses, $H1\_Q1$ and $H1\_Q2$, which focus on different relations in the model. The decisions of whether multiple hypotheses should be split in different sets and how to divide them are driven by theory, and depend on what the researchers intend to evaluate. When multiple hypotheses are included in one set they are handled by the `goric` function as a whole, not individually.

```
# Q1
H2_Q1 <- "
abs(ACOM2_TH1) > abs(TH2_ACOM1); abs(SAT2_TH1) > abs(TH2_SAT1);
abs(ACOM2_TB1) > abs(TB2_ACOM1); abs(SAT2_TB1) > abs(TB2_SAT1)
"

# Q2
H2_Q2 <- "
abs(AB2_TH1) > abs(TH2_AB1); abs(DE2_TH1) > abs(TH2_DE1); abs(VI2_TH1) > abs(TH2_VI1); abs(SL2_TH1) > a
abs(AB2_TB1) > abs(TB2_AB1); abs(DE2_TB1) > abs(TB2_DE1); abs(VI2_TB1) > abs(TB2_VI1); abs(SL2_TB1) > a
abs(AB2_ACOM1) > abs(ACOM2_AB1); abs(DE2_ACOM1) > abs(ACOM2_DE1); abs(VI2_ACOM1) > abs(ACOM2_VI1); abs(
abs(AB2_SAT1) > abs(SAT2_AB1); abs(DE2_SAT1) > abs(SAT2_DE1); abs(VI2_SAT1) > abs(SAT2_VI1); abs(SL2_SA
"
```

We obtain the GORICA results for $H2\_Q1$ and $H2\_Q2$ in two steps. Note the use of `set.seed` to ensure that the results are reproducible.

```
set.seed(123)

#H2_Q1 vs its complement, using GORICA
goricaResults_H2_Q1 <- goric(est_2, VCOV = vcov_2,
                             hypotheses = list(H2_Q1=H2_Q1))

goricaResults_H2_Q1
```

```
restriktor (0.6-10): generalized order-restricted information criterion approximation:

Results:
        model   loglik  penalty    gorica  loglik.weights  penalty.weights  gorica.weights
1       H2_Q1  134.232   62.451  -143.562           0.483            0.735           0.722
2  complement  134.301   63.473  -141.656           0.517            0.265           0.278

Conclusion:
```

```
The order-restricted hypothesis 'H2_Q1' has 2.59 times more support than its complement.
```

```
#summary(goricaResults_H2_Q1)
```

The output shows that the order-restricted hypothesis $H2\_Q1$ has 2.6 times more support than its complement. However, the log-likelihood (loglik) weights seem to be quite close. This could indicate that one or more of the inequality constraints can be replaced by (about-)equality constraints. One could investigate with the benchmarks function, using 'output_type = "rlw" ', whether the loglik weights indeed are close.
For more information, see the guidelines ('Guidelines_output_GORIC.html') and/or the benchmark tutorial on https://github.com/rebeccakuiper/Tutorials.

We can proceed in the same manner for $H2\_Q2$; however, because the default method takes too long to calculate the penalty of the GORICA, we use the bootstrap method. When using the bootstrapping the results do not change, but the computation time may decrease.

```
set.seed(123)

##H2_Q2 vs its complement, using GORICA
#goricaResults_H2_Q2 <- goric(est_2, VCOV = vcov_2,
#                             hypotheses = list(H2_Q2=H2_Q2))
#goricaResults_H2_Q2
##summary(goricaResults_H2_Q2)

# if (!require("parallel")) install.packages("parallel")
#library(parallel)
#
#nrCPUcores <- detectCores(all.tests = FALSE, logical = TRUE)
#
goricaResults_H2_Q2_b <- goric(est_2, VCOV = vcov_2,
                               hypotheses = list(H2_Q2=H2_Q2),
                               mix_weights = "boot")

goricaResults_H2_Q2_b
```

```
restriktor (0.6-10): generalized order-restricted information criterion approximation:

Results:
       model   loglik  penalty    gorica  loglik.weights  penalty.weights  gorica.weights
1      H2_Q2  130.969   58.028  -145.881           0.034            0.997           0.933
2 complement  134.301   63.995  -140.612           0.966            0.003           0.067

Conclusion:
The order-restricted hypothesis 'H2_Q2' has 13.94 times more support than its complement.
```

```
#summary(goricaResults_H2_Q2_b)
```

The order-restricted hypothesis $H2\_Q2$ has 14 times more support than its complement.

Note that the results hold for the chosen time interval. That is, the results are time-interval dependent. Next, more information is given.

## Note on time-interval dependency

The parameter estimates in a (RI-)CLPM are time-interval dependent, and thus the GORICA results as well. By using the CTmeta package:

```
# Install and load packages
#
#library(devtools)
#if (!require("CTmeta")) install_github("rebeccakuiper/CTmeta") ##install_github("rebeccakuiper/CTmeta"
library(CTmeta)
#?PhiPlot
```

one can plot the lagged-effects parameter estimates for different choices of time intervals. Based on this plot (and/or on other information), one can evaluate the hypotheses using the GORICA for different choices of time intervals.