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.

Here, two examples are presented for the use of the `goric` function in the restriktor package to evaluate hypotheses about a RI-CLPM. These are based on the analysis in:
Sukpan, C., & Kuiper, R. M. (2024). How to Evaluate Causal Dominance Hypotheses in Lagged Effects Models. Structural Equation Modeling, 31(3), 404-419. https://doi.org/10.1080/10705511.2023.2265065
The corresponding R files 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).

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.

```
# 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)
```

Example 1: 'wave-independent' parameters model

Next, you find the R code to evaluate causal dominance in lagged-effects 'wave-independent' parameters model using the GORICA (using the goric function). This is an example using a bivariate RI-CLPM with 2 variables and 5 time points.

In this example, I will use the lavaan object with user-specified parameter labels.

```
# Load the data set into R: Traditional RI-CLPM
dat <- read.table("data/RICLPM.dat",
                  col.names = c(
                    "x1", "x2", "x3", "x4", "x5",
                    "y1", "y2", "y3", "y4", "y5")
)

# Standardize the data
dat <- scale(dat)

# Hypothesis w.r.t. cross-lagged effects (as specified in the model)
H1 <- "abs(b) < abs(c)"
# versus it complement, that is, versus all other possibilities
# (here: versus abs(b) > abs(c))
# default in case of one hypothesis

# Fitting a RI-CLPM; here, a bivariate RI-CLPM with wave-independent parameters:
RICLPM5 <- '
  # Create between components (random intercepts)
  RIx =~ 1*x1 + 1*x2 + 1*x3 + 1*x4 + 1*x5
  RIy =~ 1*y1 + 1*y2 + 1*y3 + 1*y4 + 1*y5

  # Create within-person centered variables
  wx1 =~ 1*x1
  wx2 =~ 1*x2
  wx3 =~ 1*x3
  wx4 =~ 1*x4
  wx5 =~ 1*x5
  wy1 =~ 1*y1
  wy2 =~ 1*y2
  wy3 =~ 1*y3
  wy4 =~ 1*y4
  wy5 =~ 1*y5

  # Estimate lagged effects between within-person centered variables
  # (constrained)
  wx2 ~ a*wx1 + b*wy1
  wy2 ~ c*wx1 + d*wy1
  wx3 ~ a*wx2 + b*wy2
  wy3 ~ c*wx2 + d*wy2
```

```r
  wx4 ~ a*wx3 + b*wy3
  wy4 ~ c*wx3 + d*wy3
  wx5 ~ a*wx4 + b*wy4
  wy5 ~ c*wx4 + d*wy4

  # Estimate covariances between residuals of within-person centered variables
  # (i.e., innovations, constrained)
  wx2 ~~ cov*wy2
  wx3 ~~ cov*wy3
  wx4 ~~ cov*wy4
  wx5 ~~ cov*wy5

  # Estimate covariance between within-person centered variables at first wave
  wx1 ~~ wy1 # Covariance

  # Estimate variance and covariance of random intercepts
  RIx ~~ RIx
  RIy ~~ RIy
  RIx ~~ RIy

  # Estimate (residual) variance of within-person centered variables
  # (constrained)
  wx1 ~~ wx1 # Variance
  wy1 ~~ wy1
  wx2 ~~ vx*wx2 # Residual variance
  wy2 ~~ vy*wy2
  wx3 ~~ vx*wx3
  wy3 ~~ vy*wy3
  wx4 ~~ vx*wx4
  wy4 ~~ vy*wy4
  wx5 ~~ vx*wx5
  wy5 ~~ vy*wy5

  # Constrain grand means over time
  x1 + x2 + x3 + x4 + x5 ~ mx*1
  y1 + y2 + y3 + y4 + y5 ~ my*1
'
RICLPM5.fit <- lavaan(RICLPM5,
                      data = dat,
                      missing = 'ML',
                      meanstructure = T,
                      int.ov.free = T
)
summary(RICLPM5.fit, standardized = T)
```

```
lavaan 0.6-19 ended normally after 30 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        44
  Number of equality constraints                    29

  Number of observations                          1189
  Number of missing patterns                         1
```

```
Model Test User Model:

  Test statistic                               111.561
  Degrees of freedom                                50
  P-value (Chi-square)                           0.000

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Observed
  Observed information based on                Hessian

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  RIx =~
    x1                1.000                               0.449    0.447
    x2                1.000                               0.449    0.450
    x3                1.000                               0.449    0.450
    x4                1.000                               0.449    0.450
    x5                1.000                               0.449    0.450
  RIy =~
    y1                1.000                               0.557    0.539
    y2                1.000                               0.557    0.560
    y3                1.000                               0.557    0.562
    y4                1.000                               0.557    0.563
    y5                1.000                               0.557    0.563
  wx1 =~
    x1                1.000                               0.899    0.894
  wx2 =~
    x2                1.000                               0.892    0.893
  wx3 =~
    x3                1.000                               0.891    0.893
  wx4 =~
    x4                1.000                               0.891    0.893
  wx5 =~
    x5                1.000                               0.891    0.893
  wy1 =~
    y1                1.000                               0.869    0.842
  wy2 =~
    y2                1.000                               0.823    0.828
  wy3 =~
    y3                1.000                               0.819    0.827
  wy4 =~
    y4                1.000                               0.818    0.827
  wy5 =~
    y5                1.000                               0.818    0.827

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  wx2 ~
    wx1        (a)    0.282    0.022   13.049    0.000    0.284    0.284
    wy1        (b)    0.007    0.020    0.367    0.713    0.007    0.007
  wy2 ~
```

```
    wx1         (c)    0.090    0.018    4.991    0.000    0.098    0.098
    wy1         (d)    0.220    0.023    9.542    0.000    0.232    0.232
  wx3 ~
    wx2         (a)    0.282    0.022   13.049    0.000    0.282    0.282
    wy2         (b)    0.007    0.020    0.367    0.713    0.007    0.007
  wy3 ~
    wx2         (c)    0.090    0.018    4.991    0.000    0.098    0.098
    wy2         (d)    0.220    0.023    9.542    0.000    0.221    0.221
  wx4 ~
    wx3         (a)    0.282    0.022   13.049    0.000    0.282    0.282
    wy3         (b)    0.007    0.020    0.367    0.713    0.007    0.007
  wy4 ~
    wx3         (c)    0.090    0.018    4.991    0.000    0.098    0.098
    wy3         (d)    0.220    0.023    9.542    0.000    0.220    0.220
  wx5 ~
    wx4         (a)    0.282    0.022   13.049    0.000    0.282    0.282
    wy4         (b)    0.007    0.020    0.367    0.713    0.007    0.007
  wy5 ~
    wx4         (c)    0.090    0.018    4.991    0.000    0.098    0.098
    wy4         (d)    0.220    0.023    9.542    0.000    0.220    0.220

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
 .wx2 ~~
   .wy2       (cov)    0.146    0.013   11.594    0.000    0.217    0.217
 .wx3 ~~
   .wy3       (cov)    0.146    0.013   11.594    0.000    0.217    0.217
 .wx4 ~~
   .wy4       (cov)    0.146    0.013   11.594    0.000    0.217    0.217
 .wx5 ~~
   .wy5       (cov)    0.146    0.013   11.594    0.000    0.217    0.217
  wx1 ~~
    wy1                0.279    0.029    9.617    0.000    0.357    0.357
  RIx ~~
    RIy                0.153    0.019    8.235    0.000    0.611    0.611

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .x1         (mx)   -0.000    0.019   -0.000    1.000   -0.000   -0.000
   .x2         (mx)    0.000    0.019    0.000    1.000    0.000    0.000
   .x3         (mx)   -0.000    0.019   -0.000    1.000   -0.000   -0.000
   .x4         (mx)    0.000    0.019    0.000    1.000    0.000    0.000
   .x5         (mx)    0.000    0.019    0.000    1.000    0.000    0.000
   .y1         (my)    0.000    0.021    0.000    1.000    0.000    0.000
   .y2         (my)    0.000    0.021    0.000    1.000    0.000    0.000
   .y3         (my)   -0.000    0.021   -0.000    1.000   -0.000   -0.000
   .y4         (my)    0.000    0.021    0.000    1.000    0.000    0.000
   .y5         (my)   -0.000    0.021   -0.000    1.000   -0.000   -0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
    RIx                0.202    0.023    8.738    0.000    1.000    1.000
    RIy                0.310    0.025   12.530    0.000    1.000    1.000
    wx1                0.807    0.040   20.170    0.000    1.000    1.000
```

```
    wy1               0.755    0.037   20.302    0.000    1.000    1.000
   .wx2       (vx)    0.730    0.018   40.589    0.000    0.918    0.918
   .wy2       (vy)    0.624    0.017   37.190    0.000    0.920    0.920
   .wx3       (vx)    0.730    0.018   40.589    0.000    0.919    0.919
   .wy3       (vy)    0.624    0.017   37.190    0.000    0.931    0.931
   .wx4       (vx)    0.730    0.018   40.589    0.000    0.920    0.920
   .wy4       (vy)    0.624    0.017   37.190    0.000    0.931    0.931
   .wx5       (vx)    0.730    0.018   40.589    0.000    0.920    0.920
   .wy5       (vy)    0.624    0.017   37.190    0.000    0.931    0.931
   .x1                0.000                               0.000    0.000
   .x2                0.000                               0.000    0.000
   .x3                0.000                               0.000    0.000
   .x4                0.000                               0.000    0.000
   .x5                0.000                               0.000    0.000
   .y1                0.000                               0.000    0.000
   .y2                0.000                               0.000    0.000
   .y3                0.000                               0.000    0.000
   .y4                0.000                               0.000    0.000
   .y5                0.000                               0.000    0.000
```

```
# Compute GORICA values and weights
set.seed(123)
GORICA.Result <- goric(RICLPM5.fit,
                       hypotheses = list(H1))
# Defaults: comparison = "complement"
#           type = "gorica"
#
GORICA.Result
```

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

Results:
        model  loglik  penalty   gorica  loglik.weights  penalty.weights  gorica.weights
1          H1  28.317    8.500  -39.634           0.999            0.500           0.999
2  complement  21.651    8.500  -26.301           0.001            0.500           0.001

Conclusion:
The order-restricted hypothesis 'H1' has 785.56 times more support than its complement.
```

```
#summary(GORICA.Result)
```

The order-restricted hypothesis H_1 has 786 times more support than its complement.

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

Example 2: 'wave-specific' parameters model

Next, you find the R code to evaluate causal dominance in lagged-effects 'wave-independent' parameters model using the GORICA (using the goric function). This is an example using a bivariate RI-CLPM with 2 variables and 5 time points.

Two types of input will be shown:
- the lavaan object with user-specified parameter labels;
- the extracted standardized estimates and their covariance matrix.

Input option 1: lavaan object

In this example, I will use the lavaan object with user-specified parameter labels.

```r
# Load the data set into R: Traditional RI-CLPM
dat <- read.table("data/RICLPM.dat",
                  col.names = c(
                    "x1", "x2", "x3", "x4", "x5",
                    "y1", "y2", "y3", "y4", "y5")
)

# Hypothesis w.r.t. wave-specific cross-lagged effects (as specified in the model)
H1ws.l <- "abs(b2) < abs(c2); abs(b3) < abs(c3);
         abs(b4) < abs(c4); abs(b5) < abs(c5)"
# versus it complement, that is, versus all other possibilities
# default in case of one hypothesis

# Fitting a RI-CLPM; here, a bivariate RI-CLPM with wave-specific parameters:
RICLPM.l <- '
  # Create between components (random intercepts)
  RIx =~ 1*x1 + 1*x2 + 1*x3 + 1*x4 + 1*x5
  RIy =~ 1*y1 + 1*y2 + 1*y3 + 1*y4 + 1*y5

  # Create within-person centered variables
  wx1 =~ 1*x1
  wx2 =~ 1*x2
  wx3 =~ 1*x3
  wx4 =~ 1*x4
  wx5 =~ 1*x5
  wy1 =~ 1*y1
  wy2 =~ 1*y2
  wy3 =~ 1*y3
  wy4 =~ 1*y4
  wy5 =~ 1*y5

  # Estimate lagged effects between within-person centered variables
  wx2 ~ a2*wx1 + b2*wy1
  wy2 ~ c2*wx1 + d2*wy1
  wx3 ~ a3*wx2 + b3*wy2
  wy3 ~ c3*wx2 + d3*wy2
  wx4 ~ a4*wx3 + b4*wy3
  wy4 ~ c4*wx3 + d4*wy3
  wx5 ~ a5*wx4 + b5*wy4
  wy5 ~ c5*wx4 + d5*wy4


  # Estimate covariance between within-person centered variables at first wave
  wx1 ~~ wy1 # Covariance

  # Estimate covariances between residuals of within-person centered variables
  # (i.e., innovations)
  wx2 ~~ wy2
  wx3 ~~ wy3
  wx4 ~~ wy4
  wx5 ~~ wy5
```

```
  # Estimate variance and covariance of random intercepts
  RIx ~~ RIx
  RIy ~~ RIy
  RIx ~~ RIy

  # Estimate (residual) variance of within-person centered variables
  wx1 ~~ wx1 # Variances
  wy1 ~~ wy1
  wx2 ~~ wx2 # Residual variances
  wy2 ~~ wy2
  wx3 ~~ wx3
  wy3 ~~ wy3
  wx4 ~~ wx4
  wy4 ~~ wy4
  wx5 ~~ wx5
  wy5 ~~ wy5
'
RICLPM.fit.l <- lavaan(RICLPM.l,
                     data = dat,
                     missing = "ML",
                     meanstructure = T,
                     int.ov.free = T
)
summary(RICLPM.fit.l, standardized = T)
```

```
lavaan 0.6-19 ended normally after 116 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        44

  Number of observations                          1189
  Number of missing patterns                         1

Model Test User Model:

  Test statistic                                25.806
  Degrees of freedom                                21
  P-value (Chi-square)                           0.214

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Observed
  Observed information based on                Hessian

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  RIx =~
    x1                1.000                               0.096    0.390
    x2                1.000                               0.096    0.473
    x3                1.000                               0.096    0.475
    x4                1.000                               0.096    0.461
```

```
    x5                1.000                               0.096    0.465
  RIy =~
    y1                1.000                               0.178    0.569
    y2                1.000                               0.178    0.558
    y3                1.000                               0.178    0.535
    y4                1.000                               0.178    0.525
    y5                1.000                               0.178    0.533
  wx1 =~
    x1                1.000                               0.227    0.921
  wx2 =~
    x2                1.000                               0.179    0.881
  wx3 =~
    x3                1.000                               0.178    0.880
  wx4 =~
    x4                1.000                               0.185    0.887
  wx5 =~
    x5                1.000                               0.183    0.885
  wy1 =~
    y1                1.000                               0.257    0.822
  wy2 =~
    y2                1.000                               0.265    0.830
  wy3 =~
    y3                1.000                               0.281    0.845
  wy4 =~
    y4                1.000                               0.288    0.851
  wy5 =~
    y5                1.000                               0.282    0.846

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  wx2 ~
    wx1       (a2)    0.232    0.028    8.314    0.000    0.294    0.294
    wy1       (b2)    0.009    0.026    0.329    0.742    0.012    0.012
  wy2 ~
    wx1       (c2)    0.174    0.045    3.888    0.000    0.149    0.149
    wy1       (d2)    0.004    0.046    0.092    0.927    0.004    0.004
  wx3 ~
    wx2       (a3)    0.241    0.037    6.509    0.000    0.242    0.242
    wy2       (b3)    0.026    0.024    1.082    0.279    0.039    0.039
  wy3 ~
    wx2       (c3)    0.156    0.054    2.871    0.004    0.099    0.099
    wy2       (d3)    0.262    0.039    6.747    0.000    0.247    0.247
  wx4 ~
    wx3       (a4)    0.279    0.038    7.267    0.000    0.269    0.269
    wy3       (b4)    0.010    0.023    0.431    0.666    0.015    0.015
  wy4 ~
    wx3       (c4)    0.185    0.055    3.367    0.001    0.114    0.114
    wy3       (d4)    0.296    0.035    8.362    0.000    0.288    0.288
  wx5 ~
    wx4       (a5)    0.290    0.035    8.244    0.000    0.293    0.293
    wy4       (b5)   -0.004    0.022   -0.186    0.852   -0.006   -0.006
  wy5 ~
    wx4       (c5)    0.124    0.048    2.612    0.009    0.082    0.082
    wy4       (d5)    0.392    0.031   12.644    0.000    0.400    0.400
```

```
Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  wx1 ~~
    wy1               0.021    0.002    9.372    0.000    0.364    0.364
 .wx2 ~~
   .wy2               0.009    0.002    5.168    0.000    0.196    0.196
 .wx3 ~~
   .wy3               0.013    0.002    7.837    0.000    0.274    0.274
 .wx4 ~~
   .wy4               0.013    0.002    8.177    0.000    0.277    0.277
 .wx5 ~~
   .wy5               0.007    0.001    4.916    0.000    0.160    0.160
  RIx ~~
    RIy               0.010    0.001    7.992    0.000    0.587    0.587

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .x1                0.241    0.007   33.687    0.000    0.241    0.977
   .x2                0.173    0.006   29.331    0.000    0.173    0.851
   .x3                0.186    0.006   31.646    0.000    0.186    0.918
   .x4                0.117    0.006   19.288    0.000    0.117    0.559
   .x5                0.111    0.006   18.427    0.000    0.111    0.534
   .y1                0.336    0.009   37.099    0.000    0.336    1.076
   .y2                0.348    0.009   37.686    0.000    0.348    1.093
   .y3                0.319    0.010   33.098    0.000    0.319    0.960
   .y4                0.384    0.010   39.097    0.000    0.384    1.134
   .y5                0.388    0.010   40.056    0.000    0.388    1.162

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
    RIx               0.009    0.001    8.722    0.000    1.000    1.000
    RIy               0.032    0.003   12.351    0.000    1.000    1.000
    wx1               0.052    0.002   21.067    0.000    1.000    1.000
    wy1               0.066    0.004   17.985    0.000    1.000    1.000
   .wx2               0.029    0.001   20.793    0.000    0.911    0.911
   .wy2               0.068    0.004   17.503    0.000    0.977    0.977
   .wx3               0.030    0.001   20.467    0.000    0.935    0.935
   .wy3               0.072    0.003   21.324    0.000    0.918    0.918
   .wx4               0.032    0.001   21.445    0.000    0.925    0.925
   .wy4               0.074    0.003   21.876    0.000    0.884    0.884
   .wx5               0.031    0.001   21.680    0.000    0.915    0.915
   .wy5               0.065    0.003   22.446    0.000    0.813    0.813
   .x1                0.000                               0.000    0.000
   .x2                0.000                               0.000    0.000
   .x3                0.000                               0.000    0.000
   .x4                0.000                               0.000    0.000
   .x5                0.000                               0.000    0.000
   .y1                0.000                               0.000    0.000
   .y2                0.000                               0.000    0.000
   .y3                0.000                               0.000    0.000
   .y4                0.000                               0.000    0.000
   .y5                0.000                               0.000    0.000
```

```r
# Compute GORICA values and weights
# Make sure to use: standardized = T
set.seed(123)
GORICA.Result.ws.l <- goric(RICLPM.fit.l,
                            standardized = T,
                            hypotheses = list(H1ws.l = H1ws.l))
# Defaults: comparison = "complement"
#           type = "gorica"
#
GORICA.Result.ws.l
```

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

Results:
        model  loglik  penalty   gorica  loglik.weights  penalty.weights  gorica.weights
1      H1ws.l  39.996   13.966  -52.061           0.684            0.858           0.929
2  complement  39.222   15.767  -46.910           0.316            0.142           0.071

Conclusion:
The order-restricted hypothesis 'H1ws.l' has 13.14 times more support than its complement.
```

```r
#summary(GORICA.Result.ws.l)
```

The order-restricted hypothesis $H1ws.l$ has 13 times more support than its complement.

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

Input option 2: extracted standardized estimates and their covariance matrix

In this example, I will use the extracted standardized estimates and their covariance matrix as input.

```r
# Hypothesis w.r.t. wave-specific cross-lagged effects
H1ws <- "abs(beta2) < abs(gamma2); abs(beta3) < abs(gamma3);
         abs(beta4) < abs(gamma4); abs(beta5) < abs(gamma5)"
# versus it complement, that is, versus all other possibilities
# default in case of one hypothesis

# Fitting a RI-CLPM; here, a bivariate RI-CLPM with wave-specific parameters:
RICLPM <- '
  # Create between components (random intercepts)
  RIx =~ 1*x1 + 1*x2 + 1*x3 + 1*x4 + 1*x5
  RIy =~ 1*y1 + 1*y2 + 1*y3 + 1*y4 + 1*y5

  # Create within-person centered variables
  wx1 =~ 1*x1
  wx2 =~ 1*x2
  wx3 =~ 1*x3
  wx4 =~ 1*x4
  wx5 =~ 1*x5
  wy1 =~ 1*y1
  wy2 =~ 1*y2
  wy3 =~ 1*y3
  wy4 =~ 1*y4
  wy5 =~ 1*y5
```

```
  # Estimate lagged effects between within-person centered variables
  wx2 + wy2 ~ wx1 + wy1
  wx3 + wy3 ~ wx2 + wy2
  wx4 + wy4 ~ wx3 + wy3
  wx5 + wy5 ~ wx4 + wy4


  # Estimate covariance between within-person centered variables at first wave
  wx1 ~~ wy1 # Covariance

  # Estimate covariances between residuals of within-person centered variables
  # (i.e., innovations)
  wx2 ~~ wy2
  wx3 ~~ wy3
  wx4 ~~ wy4
  wx5 ~~ wy5

  # Estimate variance and covariance of random intercepts
  RIx ~~ RIx
  RIy ~~ RIy
  RIx ~~ RIy

  # Estimate (residual) variance of within-person centered variables
  wx1 ~~ wx1 # Variances
  wy1 ~~ wy1
  wx2 ~~ wx2 # Residual variances
  wy2 ~~ wy2
  wx3 ~~ wx3
  wy3 ~~ wy3
  wx4 ~~ wx4
  wy4 ~~ wy4
  wx5 ~~ wx5
  wy5 ~~ wy5
'
RICLPM.fit <- lavaan(RICLPM,
                     data = dat,
                     missing = "ML",
                     meanstructure = T,
                     int.ov.free = T
)
summary(RICLPM.fit, standardized = T)
```

```
lavaan 0.6-19 ended normally after 116 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        44

  Number of observations                          1189
  Number of missing patterns                         1

Model Test User Model:

  Test statistic                                25.806
```

```
  Degrees of freedom                                21
  P-value (Chi-square)                           0.214

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Observed
  Observed information based on                Hessian

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  RIx =~
    x1                1.000                               0.096    0.390
    x2                1.000                               0.096    0.473
    x3                1.000                               0.096    0.475
    x4                1.000                               0.096    0.461
    x5                1.000                               0.096    0.465
  RIy =~
    y1                1.000                               0.178    0.569
    y2                1.000                               0.178    0.558
    y3                1.000                               0.178    0.535
    y4                1.000                               0.178    0.525
    y5                1.000                               0.178    0.533
  wx1 =~
    x1                1.000                               0.227    0.921
  wx2 =~
    x2                1.000                               0.179    0.881
  wx3 =~
    x3                1.000                               0.178    0.880
  wx4 =~
    x4                1.000                               0.185    0.887
  wx5 =~
    x5                1.000                               0.183    0.885
  wy1 =~
    y1                1.000                               0.257    0.822
  wy2 =~
    y2                1.000                               0.265    0.830
  wy3 =~
    y3                1.000                               0.281    0.845
  wy4 =~
    y4                1.000                               0.288    0.851
  wy5 =~
    y5                1.000                               0.282    0.846

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  wx2 ~
    wx1               0.232    0.028    8.314    0.000    0.294    0.294
    wy1               0.009    0.026    0.329    0.742    0.012    0.012
  wy2 ~
    wx1               0.174    0.045    3.888    0.000    0.149    0.149
    wy1               0.004    0.046    0.092    0.927    0.004    0.004
  wx3 ~
    wx2               0.241    0.037    6.509    0.000    0.242    0.242
```

```
    wy2               0.026    0.024    1.082    0.279    0.039    0.039
  wy3 ~
    wx2               0.156    0.054    2.871    0.004    0.099    0.099
    wy2               0.262    0.039    6.747    0.000    0.247    0.247
  wx4 ~
    wx3               0.279    0.038    7.267    0.000    0.269    0.269
    wy3               0.010    0.023    0.431    0.666    0.015    0.015
  wy4 ~
    wx3               0.185    0.055    3.367    0.001    0.114    0.114
    wy3               0.296    0.035    8.362    0.000    0.288    0.288
  wx5 ~
    wx4               0.290    0.035    8.244    0.000    0.293    0.293
    wy4              -0.004    0.022   -0.186    0.852   -0.006   -0.006
  wy5 ~
    wx4               0.124    0.048    2.612    0.009    0.082    0.082
    wy4               0.392    0.031   12.644    0.000    0.400    0.400

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  wx1 ~~
    wy1               0.021    0.002    9.372    0.000    0.364    0.364
 .wx2 ~~
   .wy2               0.009    0.002    5.168    0.000    0.196    0.196
 .wx3 ~~
   .wy3               0.013    0.002    7.837    0.000    0.274    0.274
 .wx4 ~~
   .wy4               0.013    0.002    8.177    0.000    0.277    0.277
 .wx5 ~~
   .wy5               0.007    0.001    4.916    0.000    0.160    0.160
  RIx ~~
    RIy               0.010    0.001    7.992    0.000    0.587    0.587

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .x1                0.241    0.007   33.687    0.000    0.241    0.977
   .x2                0.173    0.006   29.331    0.000    0.173    0.851
   .x3                0.186    0.006   31.646    0.000    0.186    0.918
   .x4                0.117    0.006   19.288    0.000    0.117    0.559
   .x5                0.111    0.006   18.427    0.000    0.111    0.534
   .y1                0.336    0.009   37.099    0.000    0.336    1.076
   .y2                0.348    0.009   37.686    0.000    0.348    1.093
   .y3                0.319    0.010   33.098    0.000    0.319    0.960
   .y4                0.384    0.010   39.097    0.000    0.384    1.134
   .y5                0.388    0.010   40.056    0.000    0.388    1.162

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
    RIx               0.009    0.001    8.722    0.000    1.000    1.000
    RIy               0.032    0.003   12.351    0.000    1.000    1.000
    wx1               0.052    0.002   21.067    0.000    1.000    1.000
    wy1               0.066    0.004   17.985    0.000    1.000    1.000
   .wx2               0.029    0.001   20.793    0.000    0.911    0.911
   .wy2               0.068    0.004   17.503    0.000    0.977    0.977
   .wx3               0.030    0.001   20.467    0.000    0.935    0.935
```

```
   .wy3              0.072    0.003   21.324    0.000    0.918    0.918
   .wx4              0.032    0.001   21.445    0.000    0.925    0.925
   .wy4              0.074    0.003   21.876    0.000    0.884    0.884
   .wx5              0.031    0.001   21.680    0.000    0.915    0.915
   .wy5              0.065    0.003   22.446    0.000    0.813    0.813
   .x1               0.000                               0.000    0.000
   .x2               0.000                               0.000    0.000
   .x3               0.000                               0.000    0.000
   .x4               0.000                               0.000    0.000
   .x5               0.000                               0.000    0.000
   .y1               0.000                               0.000    0.000
   .y2               0.000                               0.000    0.000
   .y3               0.000                               0.000    0.000
   .y4               0.000                               0.000    0.000
   .y5               0.000                               0.000    0.000
```

```r
# One could label the parameters, similarly to example with constrained parameters,
# but then using unique names.
# Alternatively, one can extract the standardized cross-lagged estimates
# and their covariance matrix:
#
# Standardize parameter estimates and there covariance matrix
StdEst <- standardizedsolution(RICLPM.fit, type = "std.nox")
vcov_StdEst <- lavInspect(RICLPM.fit, "vcov.std.nox")
#
# Check which are the indices for the parameters of interest:
StdEst
```

```
   lhs op rhs est.std    se      z pvalue ci.lower ci.upper
1  RIx =~  x1   0.390 0.022 17.562  0.000    0.347    0.434
2  RIx =~  x2   0.473 0.026 18.155  0.000    0.422    0.524
3  RIx =~  x3   0.475 0.027 17.625  0.000    0.422    0.528
4  RIx =~  x4   0.461 0.026 17.931  0.000    0.411    0.511
5  RIx =~  x5   0.465 0.025 18.299  0.000    0.415    0.515
6  RIy =~  y1   0.569 0.021 27.060  0.000    0.528    0.611
7  RIy =~  y2   0.558 0.022 25.123  0.000    0.515    0.602
8  RIy =~  y3   0.535 0.021 25.795  0.000    0.495    0.576
9  RIy =~  y4   0.525 0.020 26.393  0.000    0.486    0.564
10 RIy =~  y5   0.533 0.019 27.588  0.000    0.495    0.571
11 wx1 =~  x1   0.921 0.009 97.752  0.000    0.902    0.939
12 wx2 =~  x2   0.881 0.014 62.978  0.000    0.854    0.908
13 wx3 =~  x3   0.880 0.015 60.555  0.000    0.852    0.909
14 wx4 =~  x4   0.887 0.013 66.480  0.000    0.861    0.914
15 wx5 =~  x5   0.885 0.013 66.330  0.000    0.859    0.911
16 wy1 =~  y1   0.822 0.015 56.430  0.000    0.794    0.851
17 wy2 =~  y2   0.830 0.015 55.545  0.000    0.801    0.859
18 wy3 =~  y3   0.845 0.013 64.256  0.000    0.819    0.870
19 wy4 =~  y4   0.851 0.012 69.350  0.000    0.827    0.875
20 wy5 =~  y5   0.846 0.012 69.509  0.000    0.822    0.870
21 wx2  ~ wx1   0.294 0.034  8.661  0.000    0.227    0.360
22 wx2  ~ wy1   0.012 0.038  0.329  0.742   -0.061    0.086
23 wy2  ~ wx1   0.149 0.039  3.865  0.000    0.073    0.225
24 wy2  ~ wy1   0.004 0.045  0.092  0.927   -0.084    0.092
25 wx3  ~ wx2   0.242 0.036  6.710  0.000    0.172    0.313
26 wx3  ~ wy2   0.039 0.036  1.081  0.280   -0.031    0.108
```

```
27 wy3  ~ wx2   0.099 0.035  2.873  0.004    0.032    0.167
28 wy3  ~ wy2   0.247 0.036  6.817  0.000    0.176    0.318
29 wx4  ~ wx3   0.269 0.036  7.388  0.000    0.197    0.340
30 wx4  ~ wy3   0.015 0.035  0.431  0.666   -0.053    0.083
31 wy4  ~ wx3   0.114 0.034  3.366  0.001    0.048    0.181
32 wy4  ~ wy3   0.288 0.033  8.593  0.000    0.222    0.353
33 wx5  ~ wx4   0.293 0.035  8.466  0.000    0.225    0.361
34 wx5  ~ wy4  -0.006 0.034 -0.187  0.852   -0.073    0.060
35 wy5  ~ wx4   0.082 0.031  2.609  0.009    0.020    0.143
36 wy5  ~ wy4   0.400 0.030 13.486  0.000    0.342    0.459
37 wx1 ~~ wy1   0.364 0.031 11.655  0.000    0.303    0.425
38 wx2 ~~ wy2   0.196 0.035  5.577  0.000    0.127    0.265
39 wx3 ~~ wy3   0.274 0.031  8.816  0.000    0.213    0.335
40 wx4 ~~ wy4   0.277 0.030  9.240  0.000    0.218    0.336
41 wx5 ~~ wy5   0.160 0.031  5.146  0.000    0.099    0.220
42 RIx ~~ RIx   1.000 0.000     NA     NA    1.000    1.000
43 RIy ~~ RIy   1.000 0.000     NA     NA    1.000    1.000
44 RIx ~~ RIy   0.587 0.050 11.802  0.000    0.490    0.685
45 wx1 ~~ wx1   1.000 0.000     NA     NA    1.000    1.000
46 wy1 ~~ wy1   1.000 0.000     NA     NA    1.000    1.000
47 wx2 ~~ wx2   0.911 0.019 48.081  0.000    0.874    0.948
48 wy2 ~~ wy2   0.977 0.011 90.944  0.000    0.956    0.998
49 wx3 ~~ wx3   0.935 0.018 50.856  0.000    0.899    0.971
50 wy3 ~~ wy3   0.918 0.021 44.242  0.000    0.877    0.959
51 wx4 ~~ wx4   0.925 0.019 48.074  0.000    0.887    0.963
52 wy4 ~~ wy4   0.884 0.022 40.250  0.000    0.841    0.927
53 wx5 ~~ wx5   0.915 0.019 47.242  0.000    0.877    0.953
54 wy5 ~~ wy5   0.813 0.024 33.316  0.000    0.765    0.861
55  x1 ~~  x1   0.000 0.000     NA     NA    0.000    0.000
56  x2 ~~  x2   0.000 0.000     NA     NA    0.000    0.000
57  x3 ~~  x3   0.000 0.000     NA     NA    0.000    0.000
58  x4 ~~  x4   0.000 0.000     NA     NA    0.000    0.000
59  x5 ~~  x5   0.000 0.000     NA     NA    0.000    0.000
60  y1 ~~  y1   0.000 0.000     NA     NA    0.000    0.000
61  y2 ~~  y2   0.000 0.000     NA     NA    0.000    0.000
62  y3 ~~  y3   0.000 0.000     NA     NA    0.000    0.000
63  y4 ~~  y4   0.000 0.000     NA     NA    0.000    0.000
64  y5 ~~  y5   0.000 0.000     NA     NA    0.000    0.000
65  x1 ~1       0.977 0.035 27.987  0.000    0.909    1.045
66  x2 ~1       0.851 0.034 25.215  0.000    0.784    0.917
67  x3 ~1       0.918 0.034 26.726  0.000    0.850    0.985
68  x4 ~1       0.559 0.031 17.974  0.000    0.498    0.620
69  x5 ~1       0.534 0.031 17.274  0.000    0.474    0.595
70  y1 ~1       1.076 0.036 29.812  0.000    1.005    1.147
71  y2 ~1       1.093 0.036 30.327  0.000    1.022    1.164
72  y3 ~1       0.960 0.035 27.811  0.000    0.892    1.028
73  y4 ~1       1.134 0.037 30.973  0.000    1.062    1.206
74  y5 ~1       1.162 0.037 31.268  0.000    1.089    1.234
75 RIx ~1       0.000 0.000     NA     NA    0.000    0.000
76 RIy ~1       0.000 0.000     NA     NA    0.000    0.000
77 wx1 ~1       0.000 0.000     NA     NA    0.000    0.000
78 wx2 ~1       0.000 0.000     NA     NA    0.000    0.000
79 wx3 ~1       0.000 0.000     NA     NA    0.000    0.000
80 wx4 ~1       0.000 0.000     NA     NA    0.000    0.000
```

```
81 wx5 ~1       0.000 0.000     NA     NA    0.000    0.000
82 wy1 ~1       0.000 0.000     NA     NA    0.000    0.000
83 wy2 ~1       0.000 0.000     NA     NA    0.000    0.000
84 wy3 ~1       0.000 0.000     NA     NA    0.000    0.000
85 wy4 ~1       0.000 0.000     NA     NA    0.000    0.000
86 wy5 ~1       0.000 0.000     NA     NA    0.000    0.000
```

```r
index_StdEst <- c(22,23, 26,27, 30,31, 34,35)
# CHECK: StdEst[index_StdEst, ]
# and what the indices for the corresponding covariance matrix:
vcov_StdEst
```

```
         wx2~wx1 wx2~wy1 wy2~wx1 wy2~wy1 wx3~wx2 wx3~wy2 wy3~wx2 wy3~wy2 wx4~wx3 wx4~wy3 wy4~wx3 wy4~wy
wx2~wx1    0.001
wx2~wy1    0.000   0.001
wy2~wx1    0.000   0.000   0.001
wy2~wy1    0.000   0.000  -0.001   0.002
wx3~wx2    0.000   0.000   0.000   0.000   0.001
wx3~wy2    0.000   0.000   0.000   0.000   0.000   0.001
wy3~wx2    0.000   0.000   0.000   0.000   0.000   0.000   0.001
wy3~wy2    0.000   0.000   0.000   0.001   0.000   0.000   0.000   0.001
wx4~wx3    0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.001
wx4~wy3    0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.001
wy4~wx3    0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.001
wy4~wy3    0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
wx5~wx4    0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
wx5~wy4    0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
wy5~wx4    0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
wy5~wy4    0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
wx1~~wy1   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
wx2~~wy2   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
wx3~~wy3   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
wx4~~wy4   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
wx5~~wy5   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
RIx~~RIx   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
RIy~~RIy   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
RIx~~RIy   0.000   0.000  -0.001   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
wx1~~wx1   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
wy1~~wy1   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
wx2~~wx2  -0.001   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
wy2~~wy2   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
wx3~~wx3   0.000   0.000   0.000   0.000  -0.001   0.000   0.000   0.000   0.000   0.000   0.000   0.00
wy3~~wy3   0.000   0.000   0.000   0.000   0.000   0.000   0.000  -0.001   0.000   0.000   0.000   0.00
wx4~~wx4   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000  -0.001   0.000   0.000   0.00
wy4~~wy4   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000  -0.00
wx5~~wx5   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
wy5~~wy5   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
x1~1       0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
x2~1       0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
x3~1       0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
x4~1       0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
x5~1       0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
y1~1       0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
y2~1       0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
y3~1       0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
```

```
y4~1       0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
y5~1       0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.00
```

```
index_vcov <- c(2,3, 6,7, 10,11, 14,15)
# CHECK: vcov_StdEst[index_vcov, index_vcov]
#
est  <- StdEst[index_StdEst, 4] # Standardize parameter estimates
# label estimates, and these labels should be used in the hypothesis/-es:
names(est) <- c("beta2", "gamma2", "beta3", "gamma3", "beta4", "gamma4", "beta5", "gamma5")
vcov <- vcov_StdEst[index_vcov, index_vcov] # Covariance matrix of standardize parameter estimates
#
# Note: make sure to change the numbers
#       such that they correspond to the correct estimates.

# Compute GORICA values and weights
set.seed(123)
GORICA.Result.ws <- goric(est, VCOV = vcov,
                          hypotheses = list(H1ws = H1ws))
# Defaults: comparison = "complement"
#           type = "gorica"
#
GORICA.Result.ws
```

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

Results:
        model  loglik  penalty   gorica  loglik.weights  penalty.weights  gorica.weights
1        H1ws  19.591    5.966  -27.250           0.684            0.858           0.929
2  complement  18.817    7.767  -22.099           0.316            0.142           0.071

Conclusion:
The order-restricted hypothesis 'H1ws' has 13.14 times more support than its complement.
```

```
#summary(GORICA.Result.ws)
```

This of course gives the same results as when using the lavaan object, that is:
The order-restricted hypothesis $H1ws$ has 13 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.

Note that this function is developed for CLPM estimates. It is not clear yet whether this can also be used for the RI-CLPM estimates. Nevertheless, one needs to bear in mind that also RI-CLPM estimates are time-interval dependent.