

# Guidelines for GORIC(A) input

Rebecca M. Kuiper and Leonard Vanbrabant

19 februari 2026



# Contents

<b>1</b>	<b>Types of input</b>	<b>5</b>
<b>2</b>	<b>Generate fictional data</b>	<b>7</b>
<b>3</b>	<b>Obtain input &amp; output for GORICA</b>	<b>9</b>
3.1	lm . . . . .	9
3.2	lavaan . . . . .	11
3.3	lmer / lme4 . . . . .	12



# Chapter 1

## Types of input

The `goric` function of `restriktor` takes different forms of input:

\* enter fitted unconstrained object:

- lm,
- rlm,
- glm,
- glmer(Mod),
- nlmer(Mod),
- lmer(Mod),
- CTmeta,
- rma,
- lavaan; \* enter the (structural) parameter estimates (possibly, standardized) and their covariance matrix.

Functions that can often be used for extraction:

- `coef()` & `vcov()`
- `fixef()` & `vcov()`

Please note that:

\* The GORIC can only be calculated for lm objects.

\* The GORIC (type = 'goric') is the default for lm objects, while GORICA (type = 'gorica') is for the other input options.

\* In the GORICA, the estimates are assumed to be normally distributed. So, for some models when sample size is low, this assumption may not hold. In that case, it is often not clear how well the GORICA performs. See Altinisik et al 2021, for some simulations regarding logistic regression and SEM models (for which the GORICA performs well).

Next, I will generate (regression) data. Afterwards, I will show some example for how one can extract the input needed and evaluate the hypothesis of interest based on that.



# Chapter 2

## Generate fictional data

```
# Population values regression coefficients (for 3 predictors)
coeffs <- c(2, 1.8, 1.6)

# Sample size
n <- 600

# Generate X (predictors)
set.seed(123)
x1 <- rnorm(n, 0, 1)
x2 <- rnorm(n, 0, 1)
x3 <- rnorm(n, 0, 1)
data_unstand <- cbind(x1, x2, x3)
# Standardize data - since parameters for continuous variables will be compared
data <- as.data.frame(scale(data_unstand)) # Standardized!

# Generate y (outcome)
y <- coeffs[1] * data$x1 + coeffs[2] * data$x2 + coeffs[3] * data$x3 + rnorm(n)
data$y <- y

# To also use lmer / lme4, we also need:
data_ID <- data
data_ID$ID <- 1:(dim(data)[1]/10)
```



# Chapter 3

## Obtain input & output for GORICA

Next, I will show how one can extract the input needed and evaluate the hypothesis of interest based on that.

```
# Hypothesis of interest using the default R labeling when predictors are
# called x1, x2, and x3
H1 <- "x1 > x2 > x3"
```

### 3.1 lm

```
fit.lm <- lm(y ~ 1 + x1 + x2 + x3, data = data)
```

#### 3.1.1 fit object

```
# Apply GORIC #
set.seed(123)
goric_lm <- goric(fit.lm, hypotheses = list(H1))
goric_lm
```

restriktor (0.6-30): generalized order-restricted information criterion:

Results:

	model	loglik	penalty	goric	loglik.weights	penalty.weights	goric.weights
1	H1	-834.834	3.815	1677.297	0.989	0.705	0.995
2	complement	-839.346	4.685	1688.063	0.011	0.295	0.005

Conclusion:

The order-restricted hypothesis 'H1' has 217.69 times more support than its complement

```
# Apply GORICA # When using an lm object, then by default goric; so overrule:
set.seed(123)
gorica_lm <- goric(fit.lm, hypotheses = list(H1), type = "gorica")
gorica_lm
```

restriktor (0.6-30): generalized order-restricted information criterion approximation:

Results:

	model	loglik	penalty	gorica	loglik.weights	penalty.weights	gorica.weights
1	H1	9.222	2.815	-12.815	0.990	0.705	0.99
2	complement	4.676	3.685	-1.980	0.010	0.295	0.00

Conclusion:

The order-restricted hypothesis 'H1' has 225.25 times more support than its complement

```
# Note that this is currently based on a covariance matrix using N-k instead of
# N (see also below). TO DO
```

### 3.1.2 extract estimates

```
est_lm <- coef(fit.lm)
VCOV_lm <- vcov(fit.lm)
# Since lm and lmer use N-k instead of N, with k the number of (regression)
# coefficients to be estimated:
N <- dim(data)[1] # = n = 600
k <- dim(data)[2] # = 1+3, namely: 1 intercept + 3 regression slopes
VCOV_lm <- vcov(fit.lm) * (N - k)/N
# VCOV_lm <- vcov(fit.lm)*(N-1)/N TO DO

# Apply GORICA #
set.seed(123)
gorica_lm_est <- goric(est_lm, VCOV = VCOV_lm, hypotheses = list(H1))
gorica_lm_est
```

restriktor (0.6-30): generalized order-restricted information criterion approximation:

Results:

	model	loglik	penalty	gorica	loglik.weights	penalty.weights	gorica.weights
1	H1	9.222	2.815	-12.815	0.990	0.705	0.99
2	complement	4.676	3.685	-1.980	0.010	0.295	0.00

Conclusion:

The order-restricted hypothesis 'H1' has 225.25 times more support than its complement

## 3.2 lavaan

```
library(lavaan)

This is lavaan 0.6-21
lavaan is FREE software! Please report any bugs.

fit.sem <- sem("y ~ 1 + x1 + x2 + x3", data = data)

# The default labeling of lavaan / sem() is:
names(coef(fit.sem))

[1] "y~1"  "y~x1" "y~x2" "y~x3" "y~~y"

# The ~ can (currently) not be used in the hypothesis, therefore, you want to
# label the parameters (of interest) yourself:
fit.sem <- sem("y ~ 1 + beta1*x1 + beta2*x2 + beta3*x3", data = data)

# Now, specify the hypothesis using this labeling:
H1_sem <- "beta1 > beta2 > beta3"
```

### 3.2.1 fit object

```
# Apply GORICA #
set.seed(123)
gorica_sem <- goric(fit.sem, hypotheses = list(H1 = H1_sem))
```

restriktor Message: The covariance matrix of the estimates was obtained via 'vcov()'. This is the  
gorica\_sem

restriktor (0.6-30): generalized order-restricted information criterion approximation:

Results:

	model	loglik	penalty	gorica	loglik.weights	penalty.weights	gorica.weights
1	H1	6.915	1.815	-10.201	0.990	0.705	0.996
2	complement	2.368	2.685	0.634	0.010	0.295	0.004

Conclusion:

The order-restricted hypothesis 'H1' has 225.25 times more support than its complement.

### 3.2.2 extract estimates

```
est_sem <- coef(fit.sem)
VCOV_sem <- vcov(fit.sem)
```

```
# Apply GORICA #
set.seed(123)
gorica_sem_est <- goric(est_sem, VCOV = VCov_sem, hypotheses = list(H1 = H1_sem))
gorica_sem_est

restriktor (0.6-30): generalized order-restricted information criterion approximation:

Results:
      model  loglik  penalty   gorica loglik.weights  penalty.weights   gorica.weights
1       H1    11.210    3.815 -14.791           0.990          0.705        0.990
2 complement     6.664    4.685   -3.957           0.010          0.295        0.000

Conclusion:
The order-restricted hypothesis 'H1' has 225.25 times more support than its complement
```

### 3.3 lmer / lme4

```
library(lme4)

Loading required package: Matrix
fit.lmer <- lmer("y ~ 1 + x1 + x2 + x3 + (1 | ID)", data = data_ID)

boundary (singular) fit: see help('isSingular')
```

#### 3.3.1 fit object

```
# Apply GORICA #
set.seed(123)
gorica_lmer <- goric(fit.lmer, hypotheses = list(H1))
```

```
restriktor Message: The covariance matrix of the estimates was obtained via 'vcov()'.

gorica_lmer

restriktor (0.6-30): generalized order-restricted information criterion approximation:

Results:
      model  loglik  penalty   gorica loglik.weights  penalty.weights   gorica.weights
1       H1    9.209    2.815 -12.788           0.989          0.705        0.990
2 complement     4.692    3.685   -2.014           0.011          0.295        0.000

Conclusion:
The order-restricted hypothesis 'H1' has 218.52 times more support than its complement
```

```
# Note that this is currently based on a covariance matrix using N-k instead of
# N (see also below).
```

### 3.3.2 extract estimates

```
est_lmer <- fixef(fit.lmer)
VCOV_lmer <- vcov(fit.lmer)
# Since lm and lmer use N-k instead of N, with k the number of (regression)
# coefficients to be estimated:
N <- dim(data_ID)[1] # = n = 600
# k <- dim(data_ID)[2]-1 # without ID, only fixed effects, so 1+3, namely: 1
# intercept + 3 regression slopes; or use:
k <- dim(VCOV_lmer)[2]
VCOV_lmer <- vcov(fit.lmer) * (N - k)/N

# Apply GORICA #
set.seed(123)
gorica_lmer_est <- goric(est_lmer, VCOV = VCov_lmer, hypotheses = list(H1))
gorica_lmer_est
```

restriktor (0.6-30): generalized order-restricted information criterion approximation:

Results:

	model	loglik	penalty	gorica	loglik.weights	penalty.weights	gorica.weights
1	H1	9.222	2.815	-12.815	0.990	0.705	0.996
2	complement	4.676	3.685	-1.980	0.010	0.295	0.004

Conclusion:

The order-restricted hypothesis 'H1' has 225.25 times more support than its complement.