

# Evaluate a set of hypotheses with GORIC or GORICA: ANOVA Example

Rebecca M. Kuiper

07 September 2023

## Contents

<b>Example 1: Lucas Data</b>	<b>1</b>
Data preparation . . . . .	1
Preparation for GORIC(A) . . . . .	3
GORIC examples . . . . .	4
GORICA examples . . . . .	6
<b>Example 2: Berzonsky et al.</b>	<b>10</b>
Data preparation . . . . .	10
Preparation for GORIC(A) . . . . .	10
GORIC examples . . . . .	11
GORICA examples . . . . .	13
<b>Example 3: Holubar</b>	<b>16</b>
Data preparation . . . . .	16
Preparation for GORIC(A) . . . . .	16
Model selection using GORIC . . . . .	16
<b>Example 4: Sesame</b>	<b>17</b>
Calculate GORIC values and weights . . . . .	18
Calculate GORIC values and weights for H1 and its complement . . . . .	19
Calculate GORIC values and weights for H2 and its complement . . . . .	19
<b>Notes on weights</b>	<b>20</b>
Note 1: GORIC weights versus GORICA weights . . . . .	20
Note 2: complement in case of two hypotheses of interest . . . . .	20
Note 3: weights when using complement not per se higher . . . . .	25
<b>Extra: Two methods to calculate the penalty</b>	<b>27</b>
Note on not full row-rank . . . . .	29

## Example 1: Lucas Data

### Data preparation

First, load the required libraries (after they have been installed). These libraries contain functions, such as `goric`, that will be used in the R code below. Each time you reopen R, you will have to load the required libraries.

```
## First, install the packages, if you have not done this already:
if (!require("psych")) install.packages("psych")
if (!require("restriktor")) install.packages("restriktor")

## Then, load the packages:
library(psych) # for the function describeBy
library(restriktor) # for the goric function

# If you want to use restriktor from github:
# if (!require("devtools")) install.packages("devtools")
# library(devtools)
# install_github("LeonardV/restriktor")
# library(restriktor) # for goric function
```

Second, it is necessary to load the data.

Notably, it is only possible to load the data if you are using the correct working directory (with both your R script and data file). The command `getwd()` shows you your current working directory. You can change the working directory to the one you prefer using the function `setwd()` by specifying the correct location between parentheses. Alternatively, in Rstudio, you can use the “Session” tab (on top) or you can use the “Files”-pane (on top of probably the right lower box of your Rstudio-screen, this pane is located next to the panes for “Plots”, “Packages”, “Help” and “Viewer”).

If you open the data file `Data_Lucas.txt` in a text editor, you can see that the variable labels have been inserted (using quotes; i.e., “...”) in the first line of the file, which is called a header. Therefore, you have to specify ‘header = TRUE’ when loading the data:

```
# Load the data
Lucas <- read.table("data/Data_Lucas.txt", header = TRUE)
```

Since a .txt file was loaded, R does not know the measurement levels of the variables and assumes all of them to be continuous, meaning that they are of interval or ratio type. Hence, especially when there are more than two groups, one has to tell R that the variable `group` is a factor by using the `factor()` function on the `group` variable (i.e., a grouping / categorical / nominal variable):

```
# Make the variable group a factor
Lucas$group <- factor(Lucas$group)
```

To inspect the first 6 rows of the dataset, use the `head()` function:

```
head(Lucas) # Look at first (6) rows of the data
```

```
  group Influence
1     1      3.58
2     1     -0.15
3     1      0.67
4     1      2.22
5     1      2.56
6     1      1.70
```

To see a more detailed overview of the data via descriptive statistics split by `group` variable, use the `describeBy()` function with `Lucas$group` set to be a grouping variable, as follows:

```
descrstat <- describeBy(Lucas$Influence, Lucas$group, mat = TRUE, digits = 3)
descrstat
```

	item	group1	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X11	1	1	1	30	2.329	1.860	2.330	2.243	2.009	-0.45	6.74	7.19	0.371	-0.719	0.340
X12	2	2	1	30	1.328	1.149	1.320	1.272	1.231	-0.38	3.87	4.25	0.318	-0.943	0.210

X13	3	3	1 30	3.200	1.790	3.475	3.243	1.824	-0.31	6.84	7.15	-0.149	-0.556	0.327
X14	4	4	1 30	2.231	1.450	1.665	2.099	1.171	0.47	5.12	4.65	0.724	-0.867	0.265
X15	5	5	1 30	3.229	1.500	3.730	3.358	1.460	-0.46	5.67	6.13	-0.630	-0.563	0.274

## Preparation for GORIC(A)

**ANOVA model: R-object** First, an R-object with unconstrained estimates is needed, that is, in this example, the five group means and one residual variance. The linear regression model using `lm()` function is specified as follows:

```
lm_fit_Lucas <- lm(Influence ~ group - 1, data = Lucas)
```

Note that:

1. `y ~ group - 1` instructs the function `lm` (linear model) to regress the variable `y` on the variable `group`.
2. The `- 1` instructs the function `lm` to drop the intercept and, therefore, estimate the means of each group, resulting, here, in five group means. On the other hand, if the intercept is not dropped, '`y ~ group`' would estimate an intercept, representing the mean of the reference group, and the mean differences between the other (here, four) groups and the reference group.
3. The results are collected in, what is called, an R-object, named `lm_fit_Lucas`.

It can be helpful to check the names used in this model, because these are needed when specifying the hypotheses:

```
names(coef(lm_fit_Lucas))
```

```
[1] "group1" "group2" "group3" "group4" "group5"
```

**ANOVA model: `coef()` and `vcov()`** Note that `goric()` can use `lm` or `glm` objects (even most `lavaan` objects) as input. Since the GORICA can be applied to a broad range of models, the GORICA has another input option as well: One can enter the (structural) parameter estimates and their covariance matrix. For most R-objects, these can be obtained using `coef()` and `vcov()`.

E.g.,

```
est <- coef(lm_fit_Lucas)
VCOV <- vcov(lm_fit_Lucas)
```

This will be illustrated in an GORICA example below.

**Set of hypotheses** To evaluate the hypotheses of interest, it is necessary to specify the restrictions in these hypotheses correctly:

- Within the `restriktor()` and `goric()` functions, it is possible to use the following operators: `>`, `<`, `=`, `<=`, `>=`, `==` (where the last three denote the same constraint as the first three).
- The `goric()` and the `restriktor()` functions can deal with:
  - pairwise restrictions separated by a semicolon ; (e.g., "`beta1 > beta2; beta2 = beta3`").
  - combined restrictions consisting of more than one operator (e.g., "`beta1 > beta2 = beta3`").

Note that one should use the labels of the parameter estimates (in the example above: `group1-group5`).

- One can also define hypothesis in terms of linear functions of parameters (For more details, see 'Extra possibility specification hypotheses' near the end of the `goric()` tutorial called 'Tutorial\_GORIC\_restriktor\_General').

Let us specify the following hypotheses:

```
H0 <- 'group1 = group2 = group3 = group4 = group5'
H1 <- 'group5 = group3 > group1 > group2; group3 > group4 > group2'
# Note: H1 is not full row-rank (see below and the goric() tutorial for more details).
H2 <- 'group3 > group1 > group4 = group5 > group2'
```

To prevent from selecting a weak hypothesis, that is, a hypothesis not supported by the data, one should include a failsafe/safeguard hypothesis. This can be:

- the unconstrained hypothesis (which includes all possible hypotheses, thus including the one(s) of interest);
- the complement (which includes all other possible hypotheses, thus excluding the one(s) of interest),

where the first option is the default. Notably, currently, the complement can only be used for one hypothesis of interest. Therefore, the examples w.r.t the use of the complement only evaluate H1 (and not the whole set).

**Seed values** In the calculation of the GORIC, an iterative process is used to calculate the penalty / complexity part. Therefore, one needs to set a seed value using the `set.seed()`. This has two advantages:

1. Using the same seed value leads to the same penalty value every time this code is run.
2. Using different seed values, allows for sensitivity check on the penalty value. If it is sensitive, then increase number of iterations used in calculation of the penalty (see below).

## GORIC examples

The GORIC can be used for normal linear models, like ANOVA and regression models. The `goric()` function calculates the *GORIC* value by default (type = “goric”). To calculate the *GORICA* values, the argument type has to be set to `gorica` (type = “gorica”, see example below).

### Example 1.1a: Using the unconstrained as failsafe

```
set.seed(123) # Set seed value
output <- goric(lm_fit_Lucas, hypotheses = list(H0 = H0, H1 = H1, H2 = H2))
summary(output)
```

restriktor (0.5-20): generalized order-restricted information criterion:

Level probabilities:

Number of requested bootstrap draws 99999

Number of successful bootstrap draws for H1: 99999

Results:

	model	loglik	penalty	goric	loglik.weights	penalty.weights	goric.weights
1	H0	-292.268	2.000	588.536	0.000	0.609	0.000
2	H1	-278.051	3.195	562.493	0.493	0.184	0.920
3	H2	-281.761	3.136	569.793	0.012	0.196	0.024
4	unconstrained	-278.048	6.000	568.097	0.495	0.011	0.056

---

Ratio GORIC-weights:

	vs. H0	vs. H1	vs. H2	vs. unconstrained
H0	1.00e+00	0.00e+00	0.00e+00	0.00e+00
H1	4.52e+05	1.00e+00	3.85e+01	1.65e+01
H2	1.17e+04	2.60e-02	1.00e+00	4.28e-01

```
unconstrained  2.74e+04  6.10e-02  2.33e+00  1.00e+00
---
```

Ratio loglik-weights:

	vs. H0	vs. H1	vs. H2	vs. unconstrained
H0	1.00e+00	0.00e+00	0.00e+00	0.00e+00
H1	1.49e+06	1.00e+00	4.08e+01	9.97e-01
H2	3.66e+04	2.40e-02	1.00e+00	2.40e-02
unconstrained	1.50e+06	1.00e+00	4.10e+01	1.00e+00

---

Ratio penalty-weights:

	vs. H0	vs. H1	vs. H2	vs. unconstrained
H0	1.000	3.304	3.113	54.598
H1	0.303	1.000	0.942	16.523
H2	0.321	1.061	1.000	17.537
unconstrained	0.018	0.061	0.057	1.000

---

order-restricted hypotheses:

H0:

group1 = group2 = group3 = group4 = group5

H1:

group5 = group3 > group1 > group2; group3 > group4 > group2

H2:

group3 > group1 > group4 = group5 > group2

Restriktor message: Since the constraint matrix is not full row-rank, the level probabilities are calculated using `mix.weights = "boot"` (the default is `mix.weights = "pmvnorm"`). For more information see `?restriktor`.

It can be seen that the order-restricted hypothesis  $H_1$  has 16.5 times more support than  $H_u$  (the unconstrained hypothesis). Hence,  $H_1$  is not a weak hypotheses and can thus be compared to the other (weak and non-weak) competing hypotheses:  $H_1$  is much more ( $4.52e+05$  times more) supported than  $H_0$  and 38.5 times more likely than  $H_2$ .

### Example 1.1b: Using the complement as failsafe

```
set.seed(123) # Set seed value
output_c <- goric(lm_fit_Lucas, hypotheses = list(H1), comparison = "complement")
summary(output_c)
```

restriktor (0.5-20): generalized order-restricted information criterion:

Level probabilities:

Number of requested bootstrap draws 99999

Number of successful bootstrap draws for H1: 99999

Results:

	model	loglik	penalty	goric	loglik.weights	penalty.weights	goric.weights
1	H1	-278.051	3.195	562.493	0.499	0.931	0.931
2	complement	-278.048	5.795	567.688	0.501	0.069	0.069

```

---
The order-restricted hypothesis 'H1' has 13.431 times more support than its complement.

```

```

Ratio GORIC-weights:
      vs. H1    vs. complement
H1      1.000    13.431
complement 0.074    1.000
---

```

```

Ratio loglik-weights:
      vs. H1    vs. complement
H1      1.000    0.997
complement 1.003    1.000
---

```

```

Ratio penalty-weights:
      vs. H1    vs. complement
H1      1.000    13.466
complement 0.074    1.000
---

```

order-restricted hypotheses:

```

H1:
group5 = group3 > group1 > group2; group3 > group4 > group2

```

Restriktor message: Since the constraint matrix is not full row-rank, the level probabilities are calculated using `mix.weights = "boot"` (the default is `mix.weights = "pmvnorm"`). For more information see `?restriktor`.

The order-restricted hypothesis  $H_1$  has 13.4 times more support than its complement.

## GORICA examples

The *GORICA* can be used for a broad range of models. Besides normal linear models (e.g., ANOVA and regression models) it can be applied also to logistic regression and SEM models and much more. To calculate the *GORICA* values, one should use `type = "gorica"`.

### Example 1.2a: Using the unconstrained as failsafe

```

set.seed(123) # Set seed value
output_gorica <- goric(lm_fit_Lucas, hypotheses = list(H0 = H0, H1 = H1, H2 = H2), type = "gorica")
summary(output_gorica)

```

restriktor (0.5-20): generalized order-restricted information criterion approximation:

Level probabilities:

```

Number of requested bootstrap draws 99999
Number of successful bootstrap draws for H1: 99999

```

Results:

	model	loglik	penalty	gorica	loglik.weights	penalty.weights	gorica.weights
1	H0	-13.485	1.000	28.970	0.000	0.609	0.000
2	H1	1.647	2.195	1.096	0.493	0.184	0.920

3	H2	-2.029	2.136	8.329	0.012	0.196	0.025
4	unconstrained	1.650	5.000	6.700	0.494	0.011	0.056

---

Ratio GORICA-weights:

	vs. H0	vs. H1	vs. H2	vs. unconstrained
H0	1.00e+00	0.00e+00	0.00e+00	0.00e+00
H1	1.13e+06	1.00e+00	3.72e+01	1.65e+01
H2	3.03e+04	2.70e-02	1.00e+00	4.43e-01
unconstrained	6.85e+04	6.10e-02	2.26e+00	1.00e+00

---

Ratio loglik-weights:

	vs. H0	vs. H1	vs. H2	vs. unconstrained
H0	1.00e+00	0.00e+00	0.00e+00	0.00e+00
H1	3.73e+06	1.00e+00	3.95e+01	9.97e-01
H2	9.44e+04	2.50e-02	1.00e+00	2.50e-02
unconstrained	3.74e+06	1.00e+00	3.96e+01	1.00e+00

---

Ratio penalty-weights:

	vs. H0	vs. H1	vs. H2	vs. unconstrained
H0	1.000	3.304	3.113	54.598
H1	0.303	1.000	0.942	16.523
H2	0.321	1.061	1.000	17.537
unconstrained	0.018	0.061	0.057	1.000

---

order-restricted hypotheses:

H0:

group1 = group2 = group3 = group4 = group5

H1:

group5 = group3 > group1 > group2; group3 > group4 > group2

H2:

group3 > group1 > group4 = group5 > group2

Restriktor message: Since the constraint matrix is not full row-rank, the level probabilities are calculated using `mix.weights = "boot"` (the default is `mix.weights = "pmvnorm"`). For more information see `?restriktor`.

In Example 1, the same analysis is done with the GORIC, you can see that the (relative) weights are (about) the same for the GORIC and GORICA.

From this output, it can be seen that the order-restricted hypothesis  $H_1$  has 16.5 times more support than  $H_u$  (the unconstrained hypothesis). Hence,  $H_1$  is not a weak hypotheses and can thus be compared to the other (weak and non-weak) competing hypotheses:  $H_1$  is much more (1.13e+06 times more) supported than  $H_0$  and 37.2 times more likely than  $H_2$ .

```
set.seed(123) # Set seed value
est <- coef(lm_fit_Lucas)
VCOV <- vcov(lm_fit_Lucas)
```

```
output_gorica_alt <- goric(est, VCOV = VCOV, hypotheses = list(H0 = H0, H1 = H1, H2 = H2), type = "gori
summary(output_gorica_alt)
```

## Alternative input

restriktor (0.5-20): generalized order-restricted information criterion approximation:

Level probabilities:

Number of requested bootstrap draws 99999

Number of successful bootstrap draws for H1: 99999

Results:

	model	loglik	penalty	gorica	loglik.weights	penalty.weights	gorica.weights
1	H0	-13.485	1.000	28.970	0.000	0.609	0.000
2	H1	1.647	2.195	1.096	0.493	0.184	0.920
3	H2	-2.029	2.136	8.329	0.012	0.196	0.025
4	unconstrained	1.650	5.000	6.700	0.494	0.011	0.056

---

Ratio GORICA-weights:

	vs. H0	vs. H1	vs. H2	vs. unconstrained
H0	1.00e+00	0.00e+00	0.00e+00	0.00e+00
H1	1.13e+06	1.00e+00	3.72e+01	1.65e+01
H2	3.03e+04	2.70e-02	1.00e+00	4.43e-01
unconstrained	6.85e+04	6.10e-02	2.26e+00	1.00e+00

---

Ratio loglik-weights:

	vs. H0	vs. H1	vs. H2	vs. unconstrained
H0	1.00e+00	0.00e+00	0.00e+00	0.00e+00
H1	3.73e+06	1.00e+00	3.95e+01	9.97e-01
H2	9.44e+04	2.50e-02	1.00e+00	2.50e-02
unconstrained	3.74e+06	1.00e+00	3.96e+01	1.00e+00

---

Ratio penalty-weights:

	vs. H0	vs. H1	vs. H2	vs. unconstrained
H0	1.000	3.304	3.113	54.598
H1	0.303	1.000	0.942	16.523
H2	0.321	1.061	1.000	17.537
unconstrained	0.018	0.061	0.057	1.000

---

order-restricted hypotheses:

H0:

group1 = group2 = group3 = group4 = group5

H1:

group5 > group3 > group1 > group2; group3 > group4 > group2

H2:

group3 > group1 > group4 = group5 > group2

Restriktor message: Since the constraint matrix is not full row-rank, the level probabilities



are calculated using `mix.weights = "boot"` (the default is `mix.weights = "pmvnorm"`).  
For more information see `?restriktor`.

### Example 1.2b: Using the complement as failsafe

```
H1 <- 'group5 = group3 > group1 > group2; group3 > group4 > group2'
# Note: H1 is not full row-rank;
# for more details, see below and/or the gorica tutorial.

set.seed(123) # Set seed value
output_gorica_c <- gorica(lm_fit_Lucas, hypotheses = list(H1), comparison = "complement",
                          type = "gorica")
summary(output_gorica_c)
```

restriktor (0.5-20): generalized order-restricted information criterion approximation:

Level probabilities:

Number of requested bootstrap draws 99999  
Number of successful bootstrap draws for H1: 99999

Results:

	model	loglik	penalty	gorica	loglik.weights	penalty.weights	gorica.weights
1	H1	1.647	2.195	1.096	0.499	0.931	0.931
2	complement	1.650	4.795	6.291	0.501	0.069	0.069

---

The order-restricted hypothesis 'H1' has 13.432 times more support than its complement.

Ratio GORICA-weights:

	vs. H1	vs. complement
H1	1.000	13.432
complement	0.074	1.000

---

Ratio loglik-weights:

	vs. H1	vs. complement
H1	1.000	0.997
complement	1.003	1.000

---

Ratio penalty-weights:

	vs. H1	vs. complement
H1	1.000	13.466
complement	0.074	1.000

---

order-restricted hypotheses:

H1:

group5 = group3 > group1 > group2; group3 > group4 > group2

Restriktor message: Since the constraint matrix is not full row-rank, the level probabilities are calculated using `mix.weights = "boot"` (the default is `mix.weights = "pmvnorm"`).  
For more information see `?restriktor`.

The order-restricted hypothesis  $H_1$  has 13.4 times more support than its complement. Notably, the weights are also now the same as for the GORIC (Example 2).

## Example 2: Berzonsky et al.

Next, the R code to apply the GORIC(A) to the data of Berzonsky et al. is shown. The instruction on loading and preparing the data are similar to the ones from Lucas example (Example 1). Hence, for more comments and details, see that example.

### Data preparation

```
# Read Data.
BerzEtAl <- read.table("data/Data_BerzEtAl.txt", header = TRUE)
BerzEtAl$group <- factor(BerzEtAl$group)
# this command tells R that group is a factor and not a continuous variable

# Inspect data
head(BerzEtAl)
```

	group	Influence
1	1	39.79709
2	1	26.58804
3	1	36.04999
4	1	35.92915
5	1	27.02636
6	1	31.08900

```
# Compute descriptive statistics for each group
descrip <- describeBy(BerzEtAl$Influence, BerzEtAl$group, mat = TRUE, digits = 3)
descrip
```

	item	group1	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X11	1	1	1	15	32.00	4.461	33.508	31.934	3.768	25.055	39.797	14.742	-0.114	-1.353	1.152
X12	2	2	1	15	21.00	4.627	21.426	20.976	3.795	13.179	29.138	15.959	-0.081	-1.007	1.195
X13	3	3	1	15	7.00	4.192	6.364	6.853	3.498	1.456	14.453	12.996	0.440	-1.100	1.082
X14	4	4	1	15	14.00	6.242	14.708	14.259	7.264	0.906	23.724	22.818	-0.349	-0.809	1.612
X15	5	5	1	15	14.00	4.368	14.068	14.195	3.532	5.194	20.266	15.072	-0.465	-0.746	1.128
X16	6	6	1	15	0.01	0.010	0.011	0.009	0.010	-0.004	0.038	0.042	1.081	1.119	0.003
X17	7	7	1	15	0.10	0.093	0.083	0.096	0.129	-0.026	0.284	0.310	0.179	-1.232	0.024
X18	8	8	1	15	0.22	0.217	0.263	0.225	0.218	-0.150	0.528	0.677	-0.491	-1.087	0.056

### Preparation for GORIC(A)

```
# Using the R package lm
lm_fit_BerzEtAl <- lm(Influence ~ group-1, data=BerzEtAl)

# Check names used in model
names(coef(lm_fit_BerzEtAl))
```

```
[1] "group1" "group2" "group3" "group4" "group5" "group6" "group7" "group8"
```

```
# Specify restrictions using those names
```

```
H0 <- 'group1 = group2 = group3 = group4 = group5 = group6 = group7 = group8'
```

```
H1 <- 'group1 > group2; group1 > group3; group1 > group4;
group5 > group6; group5 > group7; group5 > group8;
group1 > group5; group2 > group6; group3 > group7; group4 > group8;
group1 - group5 > group2 - group6;
group1 - group5 > group3 - group7;
group1 - group5 > group4 - group8;'
```

```
H2 <- 'group1 > group2; group1 > group3; group1 > group4;
group5 > group6; group5 > group7; group5 > group8;
group1 > group5; group2 > group6; group3 > group7; group4 > group8;
group1 - group5 > 2*(group2 - group6);
group1 - group5 > 2*(group3 - group7);
group1 - group5 > 2*(group4 - group8);'
```

Set of hypotheses

## GORIC examples

Example 2.1a: Using the unconstrained as failsafe

```
set.seed(123) # Set seed value
output_B <- goric(lm_fit_BerzEtAl, hypotheses = list(H0 = H0, H1 = H1, H2 = H2) )
summary(output_B)
```

restriktor (0.5-20): generalized order-restricted information criterion:

Level probabilities:

```
Number of requested bootstrap draws 99999
Number of successful bootstrap draws for H1: 99999
Number of successful bootstrap draws for H2: 99999
```

Results:

	model	loglik	penalty	goric	loglik.weights	penalty.weights	goric.weights
1	H0	-462.323	2.000	928.645	0.000	0.885	0.000
2	H1	-328.302	4.929	666.463	0.229	0.047	0.946
3	H2	-361.183	4.577	731.522	0.000	0.067	0.000
4	unconstrained	-327.087	9.000	672.173	0.771	0.001	0.054

---

Ratio GORIC-weights:

	vs. H0	vs. H1	vs. H2	vs. unconstrained
H0	1.00e+00	0.00e+00	0.00e+00	0.00e+00
H1	8.55e+56	1.00e+00	1.34e+14	1.74e+01
H2	6.38e+42	0.00e+00	1.00e+00	0.00e+00
unconstrained	4.92e+55	5.80e-02	7.72e+12	1.00e+00

---

Ratio loglik-weights:

	vs. H0	vs. H1	vs. H2	vs. unconstrained
H0	1.00e+00	0.00e+00	0.00e+00	0.00e+00
H1	1.60e+58	1.00e+00	1.91e+14	2.96e-01

```

H2          8.40e+43  0.00e+00  1.00e+00  0.00e+00
unconstrained 5.40e+58  3.37e+00  6.43e+14  1.00e+00
---
```

Ratio penalty-weights:

```

          vs. H0  vs. H1  vs. H2  vs. unconstrained
H0          1.000  18.715  13.162  1096.633
H1          0.053   1.000   0.703   58.598
H2          0.076   1.422   1.000   83.318
unconstrained 0.001   0.017   0.012   1.000
---
```

order-restricted hypotheses:

H0:

```
group1 = group2 = group3 = group4 = group5 = group6 = group7 = group8
```

H1:

```

group1 > group2; group1 > group3; group1 > group4;
group5 > group6; group5 > group7; group5 > group8;
group1 > group5; group2 > group6; group3 > group7; group4 > group8;
group1 - group5 > group2 - group6;
group1 - group5 > group3 - group7;
group1 - group5 > group4 - group8;
```

H2:

```

group1 > group2; group1 > group3; group1 > group4;
group5 > group6; group5 > group7; group5 > group8;
group1 > group5; group2 > group6; group3 > group7; group4 > group8;
group1 - group5 > 2*(group2 - group6);
group1 - group5 > 2*(group3 - group7);
group1 - group5 > 2*(group4 - group8);
```

Restriktor message: Since the constraint matrix is not full row-rank, the level probabilities are calculated using `mix.weights = "boot"` (the default is `mix.weights = "pmvnorm"`). For more information see `?restriktor`.

The order-restricted hypothesis  $H_1$  has 17.4 times more support than  $H_u$  (unconstrained). Hence,  $H_1$  is not a weak hypotheses and can thus be compared to the other (weak and non-weak) competing hypotheses:  $H_1$  is much more ( $8.55e+56$  and  $4.73e+10$  times more) supported than  $H_0$  and  $H_2$  (respectively).

### Example 2.1b: Using the complement as failsafe

```

set.seed(123)
output_B_c <- goric(lm_fit_BerzEtAl, hypotheses = list(H1), comparison = "complement")
summary(output_B_c)
```

restriktor (0.5-20): generalized order-restricted information criterion:

Level probabilities:

```

Number of requested bootstrap draws 99999
Number of successful bootstrap draws for H1: 99999
```

Results:

```

      model    loglik  penalty    goric  loglik.weights  penalty.weights  goric.weights
```

1	H1	-328.302	4.929	666.463	0.229	0.983	0.945
2	complement	-327.087	8.991	672.156	0.771	0.017	0.055

---

The order-restricted hypothesis 'H1' has 17.221 times more support than its complement.

Ratio GORIC-weights:

	vs. H1	vs. complement
H1	1.000	17.221
complement	0.058	1.000

---

Ratio loglik-weights:

	vs. H1	vs. complement
H1	1.000	0.296
complement	3.373	1.000

---

Ratio penalty-weights:

	vs. H1	vs. complement
H1	1.000	58.088
complement	0.017	1.000

---

order-restricted hypotheses:

H1:

```
group1 > group2; group1 > group3; group1 > group4;
group5 > group6; group5 > group7; group5 > group8;
group1 > group5; group2 > group6; group3 > group7; group4 > group8;
group1 - group5 > group2 - group6;
group1 - group5 > group3 - group7;
group1 - group5 > group4 - group8;
```

Restriktor message: Since the constraint matrix is not full row-rank, the level probabilities are calculated using `mix.weights = "boot"` (the default is `mix.weights = "pmvnorm"`).

For more information see `?restriktor`.

The order-restricted hypothesis  $H_1$  has 17.2 times more support than its complement.

## GORICA examples

### Example 2.2a: Using the unconstrained as failsafe

```
set.seed(123) # Set seed value
output_B_gorica <- gorica(lm_fit_BerzEtAl, hypotheses = list(H0 = H0, H1 = H1, H2 = H2), type = "gorica")
summary(output_B_gorica)
```

restriktor (0.5-20): generalized order-restricted information criterion approximation:

Level probabilities:

```
Number of requested bootstrap draws 99999
Number of successful bootstrap draws for H1: 99999
Number of successful bootstrap draws for H2: 99999
```

Results:

	model	loglik	penalty	gorica	loglik.weights	penalty.weights	gorica.weights
1	H0	-484.658	1.000	971.316	0.000	0.885	0.000
2	H1	-8.396	3.929	24.650	0.241	0.047	0.949
3	H2	-50.102	3.577	107.360	0.000	0.067	0.000
4	unconstrained	-7.250	8.000	30.499	0.759	0.001	0.051

---

Ratio GORICA-weights:

	vs. H0	vs. H1	vs. H2	vs. unconstrained
H0	1.00e+00	0.00e+00	0.00e+00	0.00e+00
H1	3.68e+205	1.00e+00	9.12e+17	1.86e+01
H2	4.03e+187	0.00e+00	1.00e+00	0.00e+00
unconstrained	1.98e+204	5.40e-02	4.90e+16	1.00e+00

---

Ratio loglik-weights:

	vs. H0	vs. H1	vs. H2	vs. unconstrained
H0	1.00e+00	0.00e+00	0.00e+00	0.00e+00
H1	6.88e+206	1.00e+00	1.30e+18	3.18e-01
H2	5.31e+188	0.00e+00	1.00e+00	0.00e+00
unconstrained	2.17e+207	3.15e+00	4.08e+18	1.00e+00

---

Ratio penalty-weights:

	vs. H0	vs. H1	vs. H2	vs. unconstrained
H0	1.000	18.715	13.162	1096.633
H1	0.053	1.000	0.703	58.598
H2	0.076	1.422	1.000	83.318
unconstrained	0.001	0.017	0.012	1.000

---

order-restricted hypotheses:

H0:

group1 = group2 = group3 = group4 = group5 = group6 = group7 = group8

H1:

group1 > group2; group1 > group3; group1 > group4;  
group5 > group6; group5 > group7; group5 > group8;  
group1 > group5; group2 > group6; group3 > group7; group4 > group8;  
group1 - group5 > group2 - group6;  
group1 - group5 > group3 - group7;  
group1 - group5 > group4 - group8;

H2:

group1 > group2; group1 > group3; group1 > group4;  
group5 > group6; group5 > group7; group5 > group8;  
group1 > group5; group2 > group6; group3 > group7; group4 > group8;  
group1 - group5 > 2\*(group2 - group6);  
group1 - group5 > 2\*(group3 - group7);  
group1 - group5 > 2\*(group4 - group8);

Restriktor message: Since the constraint matrix is not full row-rank, the level probabilities

are calculated using `mix.weights = "boot"` (the default is `mix.weights = "pmvnorm"`).  
For more information see `?restriktor`.

The order-restricted hypothesis  $H_1$  has 18.6 times more support than  $H_u$  (unconstrained). Hence,  $H_1$  is not a weak hypotheses and can thus be compared to the other (weak and non-weak) competing hypotheses:  $H_1$  is much more ( $3.68e+205$  and  $4.46e+12$  times more) supported than  $H_0$  and  $H_2$  (respectively).

### Example 2.2b: Using the unconstrained as failsafe

```
set.seed(123)
output_B_gorica_c <- gorica(lm_fit_BerzEtAl, hypotheses = list(H1), comparison = "complement", type = "g")
summary(output_B_gorica_c)
```

restriktor (0.5-20): generalized order-restricted information criterion approximation:

Level probabilities:

Number of requested bootstrap draws 99999  
Number of successful bootstrap draws for H1: 99999

Results:

	model	loglik	penalty	gorica	loglik.weights	penalty.weights	gorica.weights
1	H1	-8.396	3.929	24.650	0.241	0.983	0.949
2	complement	-7.250	7.991	30.482	0.759	0.017	0.051

---

The order-restricted hypothesis 'H1' has 18.460 times more support than its complement.

Ratio GORICA-weights:

	vs. H1	vs. complement
H1	1.000	18.460
complement	0.054	1.000

---

Ratio loglik-weights:

	vs. H1	vs. complement
H1	1.000	0.318
complement	3.147	1.000

---

Ratio penalty-weights:

	vs. H1	vs. complement
H1	1.000	58.088
complement	0.017	1.000

---

order-restricted hypotheses:

H1:

```
group1 > group2; group1 > group3; group1 > group4;
group5 > group6; group5 > group7; group5 > group8;
group1 > group5; group2 > group6; group3 > group7; group4 > group8;
group1 - group5 > group2 - group6;
group1 - group5 > group3 - group7;
group1 - group5 > group4 - group8;
```

Restriktor message: Since the constraint matrix is not full row-rank, the level probabilities are calculated using `mix.weights = "boot"` (the default is `mix.weights = "pmvnorm"`). For more information see `?restriktor`.

The order-restricted hypothesis  $H_1$  has 18.5 times more support than its complement.

## Example 3: Holubar

### Data preparation

First, read in the Holubar dataset, and tell R that the variable `gr` (group) is a factor instead of a continuous variable (although it is not necessary because it consists of only two groups).

```
Holubar <- read.table("data/Data_Holubar.txt", header = TRUE) # load the data
Holubar$gr <- factor(Holubar$gr) # tell R that gr is a factor
```

If you want a more detailed overview of the data, also by means of descriptive statistics splitted by `group`, use

```
head(Holubar)
```

```
      at gr
1 0.5549239 1
2 3.6167880 1
3 0.8071903 1
4 1.2733173 1
5 2.3898220 1
6 0.1910118 1
```

```
descrstat <- describeBy(Holubar$at, Holubar$gr, mat = TRUE, digits = 3)
descrstat
```

	item	group	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X11	1	1	1	20	0.98	1.20	1.177	0.930	1.118	-0.719	3.617	4.335	0.190	-0.778	0.268
X12	2	2	1	27	0.02	1.88	0.169	0.111	1.951	-4.961	2.921	7.883	-0.517	-0.104	0.362
X13	3	3	1	28	0.27	1.72	-0.099	0.203	2.435	-2.276	3.796	6.073	0.292	-1.025	0.325

### Preparation for GORIC(A)

#### ANOVA model: R-object

Then, fit an ANOVA-model by means of the `lm()` function (linear model) and directly check the names that are used in this model:

```
lm_fit_Holubar <- lm(at ~ gr - 1, data = Holubar)
names(coef(lm_fit_Holubar))
```

```
[1] "gr1" "gr2" "gr3"
```

#### Set of hypotheses

The following hypothesis will be evaluated:

```
H1 <- 'gr2 > gr1 > gr3'
```

### Model selection using GORIC

Calculate the GORIC values and weights:



```
set.seed(123) # Set seed value
output_Hol <- goric(lm_fit_Holubar, hypotheses = list(H1), comparison = "complement")
summary(output_Hol)
```

restriktor (0.5-20): generalized order-restricted information criterion:

Results:

	model	loglik	penalty	goric	loglik.weights	penalty.weights	goric.weights
1	H1	-144.981	2.803	295.569	0.125	0.710	0.259
2	complement	-143.038	3.697	293.469	0.875	0.290	0.741

---

The order-restricted hypothesis 'H1' has 0.350 times more support than its complement.

Ratio GORIC-weights:

	vs. H1	vs. complement
H1	1.00	0.35
complement	2.86	1.00

---

Ratio loglik-weights:

	vs. H1	vs. complement
H1	1.000	0.143
complement	6.978	1.000

---

Ratio penalty-weights:

	vs. H1	vs. complement
H1	1.000	2.443
complement	0.409	1.000

---

order-restricted hypotheses:

H1:

gr2 > gr1 > gr3

## Example 4: Sesame

```
# read in the sesame data from a text file
Sesame <- read.table("data/sesamesimANOVA.txt", header=TRUE)

# make viewcat a factor, that is, a categorical variable
Sesame$viewcat <- factor(Sesame$viewcat)

# Inspect data
head(Sesame)
```

	viewcat	postnumb
1	1	14.51050
2	3	35.30549
3	3	22.68280

```

4      1 39.82436
5      4 40.42370
6      3 22.87055

# estimate the parameters of the statistical model at hand
fit_Sesame <- lm(postnumb ~ viewcat - 1, data = Sesame)

# Check names used in model
names(coef(fit_Sesame))

[1] "viewcat1" "viewcat2" "viewcat3" "viewcat4"

# Specify restrictions using those names

# Example hypotheses:
H1 <- 'viewcat1 = viewcat2 < viewcat3 < viewcat4'
H2 <- 'viewcat1 < viewcat2 < viewcat3 < viewcat4'
H3 <- 'viewcat1 = viewcat2 < viewcat3 = viewcat4'

```

## Calculate GORIC values and weights

Here, we assume that the interest lies in H1 to H3. Since these do not cover all possibilities, the unconstrained hypotheses is included in the set.

```

set.seed(123)
goric(fit_Sesame, hypotheses = list(H1, H2, H3))

```

restriktor (0.5-20): generalized order-restricted information criterion:

Results:

	model	loglik	penalty	goric	loglik.weights	penalty.weights	goric.weights
1	H1	-918.485	2.810	1842.590	0.065	0.310	0.134
2	H2	-916.540	3.094	1839.269	0.457	0.233	0.704
3	H3	-919.644	2.500	1844.287	0.021	0.422	0.057
4	unconstrained	-916.540	5.000	1843.080	0.457	0.035	0.105

---

From the output, it is concluded that H2 is not a weak hypothesis (nor is H1). Thus, its support can be compared to that of the other hypotheses: H2 is 0.704/0.134 times more supported than H1 and 0.704/0.057 times more than H3. Hence, H2 is the preferred hypothesis and has quite some evidence.

## Calculate GORICA values and weights

```

set.seed(123)
goric(fit_Sesame, hypotheses = list(H1, H2, H3), type = "gorica")

```

restriktor (0.5-20): generalized order-restricted information criterion approximation:

Results:

	model	loglik	penalty	gorica	loglik.weights	penalty.weights	gorica.weights
1	H1	-7.052	1.810	17.725	0.066	0.310	0.136
2	H2	-5.125	2.094	14.438	0.456	0.233	0.702
3	H3	-8.216	1.500	19.432	0.021	0.422	0.058
4	unconstrained	-5.125	4.000	18.249	0.456	0.035	0.104

---

## Calculate GORIC values and weights for H1 and its complement

Here, we assume that the interest lies only in H1. In that case, it should be evaluated against its complement.

```
set.seed(123)
goric(fit_Sesame, hypotheses = list(H1), comparison = "complement")
```

restriktor (0.5-20): generalized order-restricted information criterion:

Results:

	model	loglik	penalty	goric	loglik.weights	penalty.weights	goric.weights
1	H1	-918.485	2.810	1842.590	0.125	0.868	0.484
2	complement	-916.540	4.690	1842.460	0.875	0.132	0.516

---

The order-restricted hypothesis 'H1' has 0.937 times more support than its complement.

From the output, it is concluded that the support for H1 is comparable to that of its complement, with a slight preference for the complement. Note that the complement has the highest (log) likelihood but also the highest complexity/penalty, when balancing fit and complexity this results in a slight preference for the complement of H1.

## Calculate GORICA values and weights for H1 and its complement

```
set.seed(123)
goric(fit_Sesame, hypotheses = list(H1), type = "gorica", comparison = "complement")
```

restriktor (0.5-20): generalized order-restricted information criterion approximation:

Results:

	model	loglik	penalty	gorica	loglik.weights	penalty.weights	gorica.weights
1	H1	-7.052	1.810	17.725	0.127	0.868	0.488
2	complement	-5.125	3.690	17.629	0.873	0.132	0.512

---

The order-restricted hypothesis 'H1' has 0.953 times more support than its complement.

## Calculate GORIC values and weights for H2 and its complement

Here, we assume that the interest lies only in H2. In that case, it should be evaluated against its complement.

```
set.seed(123)
goric(fit_Sesame, hypotheses = list(H2), comparison = "complement")
```

restriktor (0.5-20): generalized order-restricted information criterion:

Results:

	model	loglik	penalty	goric	loglik.weights	penalty.weights	goric.weights
1	H1	-916.540	3.095	1839.270	0.765	0.855	0.950
2	complement	-917.718	4.870	1845.176	0.235	0.145	0.050

---

The order-restricted hypothesis 'H1' has 19.166 times more support than its complement.

From the output, it is concluded that H1 is about 19 times more likely than its complement, showing quite some support for H2.

### Calculate GORICA values and weights for H2 and its complement

```
set.seed(123)
goric(fit_Sesame, hypotheses = list(H2), type = "gorica", comparison = "complement")
```

restriktor (0.5-20): generalized order-restricted information criterion approximation:

Results:

	model	loglik	penalty	gorica	loglik.weights	penalty.weights	gorica.weights
1	H1	-5.125	2.095	14.438	0.762	0.855	0.950
2	complement	-6.288	3.870	20.317	0.238	0.145	0.050

---

The order-restricted hypothesis 'H1' has 18.901 times more support than its complement.

## Notes on weights

Here, the example of Lucas is used again, consequently, using the following hypotheses of interest:

```
H0 <- 'group1 = group2 = group3 = group4 = group5'
H1 <- 'group5 = group3 > group1 > group2; group3 > group4 > group2'
# Note: H1 is not full row-rank;
# for more details, see below and/or the goric tutorial.
H2 <- 'group3 > group1 > group4 = group5 > group2'
```

### Note 1: GORIC weights versus GORICA weights

The GORICA weights (asymptotically) equal the GORIC weights. The differences are minor and often not notable with 2 decimals. Because of these minor differences, the relative weights (i.e., ratio of weights) can differ. Those differences in relative weights can even be large (as is in the Berzonsky et al Example), when dividing a very large number by a very small number with minor differences in these values.

### Note 2: complement in case of two hypotheses of interest

One cannot compare the support of hypotheses when comparing them to their complements. Instead, one should evaluate them simultaneously in one set to get desirable outcomes (like done above):

```
# Calculate goric for H1 and its complement
set.seed(123) # Set seed value
output_c_H1 <- goric(lm_fit_Lucas, hypotheses = list(H1), comparison = "complement")
summary(output_c_H1)
```

restriktor (0.5-20): generalized order-restricted information criterion:

Level probabilities:

Number of requested bootstrap draws 99999  
Number of successful bootstrap draws for H1: 99999

Results:

	model	loglik	penalty	goric	loglik.weights	penalty.weights	goric.weights
--	-------	--------	---------	-------	----------------	-----------------	---------------

1	H1	-278.051	3.195	562.493	0.499	0.931	0.931
2	complement	-278.048	5.795	567.688	0.501	0.069	0.069

---

The order-restricted hypothesis 'H1' has 13.431 times more support than its complement.

Ratio GORIC-weights:

	vs. H1	vs. complement
H1	1.000	13.431
complement	0.074	1.000

---

Ratio loglik-weights:

	vs. H1	vs. complement
H1	1.000	0.997
complement	1.003	1.000

---

Ratio penalty-weights:

	vs. H1	vs. complement
H1	1.000	13.466
complement	0.074	1.000

---

order-restricted hypotheses:

H1:

group5 = group3 > group1 > group2; group3 > group4 > group2

Restriktor message: Since the constraint matrix is not full row-rank, the level probabilities are calculated using mix.weights = "boot" (the default is mix.weights = "pmvnorm").

For more information see ?restriktor.

```
# The order-restricted hypothesis H1 has 13.4 times more support than
# its complement.
```

```
# Calculate goric for H2 and its complement
set.seed(123) # Set seed value
output_c_H2 <- goric(lm_fit_Lucas, hypotheses = list(H2 = H2), comparison = "complement")
summary(output_c_H2)
```

restriktor (0.5-20): generalized order-restricted information criterion:

Results:

	model	loglik	penalty	goric	loglik.weights	penalty.weights	goric.weights
1	H2	-281.761	3.136	569.794	0.024	0.938	0.270
2	complement	-278.048	5.853	567.803	0.976	0.062	0.730

---

The order-restricted hypothesis 'H2' has 0.370 times more support than its complement.

Ratio GORIC-weights:

	vs. H2	vs. complement
H2	1.00	0.37
complement	2.71	1.00

---

```
Ratio loglik-weights:
      vs. H2  vs. complement
H2      1.000  0.024
complement 40.952 1.000
---
```

```
Ratio penalty-weights:
      vs. H2  vs. complement
H2      1.000  15.136
complement 0.066 1.000
---
```

order-restricted hypotheses:

```
H2:
group3 > group1 > group4 = group5 > group2
```

```
#The order-restricted hypothesis H1 has 0.37 times more support than
# its complement.
```

```
# Calculate goric for H1 and H2 (and Hu):
set.seed(123) # Set seed value
output_H1H2 <- goric(lm_fit_Lucas, hypotheses = list(H1, H2)) # Note: by default,
# against the unconstrained
summary(output_H1H2)
```

restriktor (0.5-20): generalized order-restricted information criterion:

Level probabilities:

```
Number of requested bootstrap draws 99999
Number of successful bootstrap draws for H1: 99999
```

Results:

	model	loglik	penalty	goric	loglik.weights	penalty.weights	goric.weights
1	H1	-278.051	3.195	562.493	0.493	0.471	0.920
2	H2	-281.761	3.136	569.794	0.012	0.500	0.024
3	unconstrained	-278.048	6.000	568.097	0.495	0.029	0.056

---

```
Ratio GORIC-weights:
      vs. H1  vs. H2  vs. unconstrained
H1      1.000  38.497  16.481
H2      0.026  1.000  0.428
unconstrained 0.061  2.336  1.000
---
```

```
Ratio loglik-weights:
      vs. H1  vs. H2  vs. unconstrained
H1      1.000  40.844  0.997
H2      0.024  1.000  0.024
unconstrained 1.003  40.952  1.000
---
```

Ratio penalty-weights:

	vs. H1	vs. H2	vs. unconstrained
H1	1.000	0.943	16.524
H2	1.061	1.000	17.532
unconstrained	0.061	0.057	1.000

---

order-restricted hypotheses:

H1:

group5 = group3 > group1 > group2; group3 > group4 > group2

H2:

group3 > group1 > group4 = group5 > group2

Restriktor message: Since the constraint matrix is not full row-rank, the level probabilities are calculated using `mix.weights = "boot"` (the default is `mix.weights = "pmvnorm"`). For more information see `?restriktor`.

```
# H1 has 38.5 times more support than H2.
```

```
#
```

```
# This is not equal to:
```

```
output_c_H1$ratio.gw[1,2] / output_c_H2$ratio.gw[1,2]
```

```
[1] 36.3384
```

```
# approx 13.4 / 0.37
```

```
# 36.34
```

```
# Notably, you could derive the support from H1 vs H2 from their support versus  
# that of Hu:
```

```
set.seed(123) # Set seed value
```

```
output_u_H1 <- goric(lm_fit_Lucas, hypotheses = list(H1))
```

```
summary(output_u_H1)
```

restriktor (0.5-20): generalized order-restricted information criterion:

Level probabilities:

Number of requested bootstrap draws 99999

Number of successful bootstrap draws for H1: 99999

Results:

	model	loglik	penalty	goric	loglik.weights	penalty.weights	goric.weights
1	H1	-278.051	3.195	562.493	0.499	0.943	0.943
2	unconstrained	-278.048	6.000	568.097	0.501	0.057	0.057

---

Ratio GORIC-weights:

	vs. H1	vs. unconstrained
H1	1.000	16.481
unconstrained	0.061	1.000

---

Ratio loglik-weights:

	vs. H1	vs. unconstrained
--	--------	-------------------

H1	1.000	0.997
unconstrained	1.003	1.000

---

Ratio penalty-weights:

	vs. H1	vs. unconstrained
H1	1.000	16.524
unconstrained	0.061	1.000

---

order-restricted hypotheses:

H1:

group5 = group3 > group1 > group2; group3 > group4 > group2

Restriktor message: Since the constraint matrix is not full row-rank, the level probabilities are calculated using mix.weights = "boot" (the default is mix.weights = "pmvnorm"). For more information see ?restriktor.

```
#
set.seed(123) # Set seed value
output_u_H2 <- goric(lm_fit_Lucas, hypotheses = list(H2 = H2))
summary(output_u_H2)
```

restriktor (0.5-20): generalized order-restricted information criterion:

Results:

	model	loglik	penalty	goric	loglik.weights	penalty.weights	goric.weights
1	H2	-281.761	3.136	569.794	0.024	0.946	0.300
2	unconstrained	-278.048	6.000	568.097	0.976	0.054	0.700

---

Ratio GORIC-weights:

	vs. H2	vs. unconstrained
H2	1.000	0.428
unconstrained	2.336	1.000

---

Ratio loglik-weights:

	vs. H2	vs. unconstrained
H2	1.000	0.024
unconstrained	40.952	1.000

---

Ratio penalty-weights:

	vs. H2	vs. unconstrained
H2	1.000	17.532
unconstrained	0.057	1.000

---

order-restricted hypotheses:

H2:

group3 > group1 > group4 = group5 > group2



```
#
output_u_H1$ratio.gw[1,2] / output_u_H2$ratio.gw[1,2]

[1] 38.49661
# approx (0.943/0.057) / (0.3/0.7) = 16.481 / 0.428 =
# 38.49661
# which (approximately) equals:
output_H1H2$ratio.gw[1,2]

[1] 38.49665
```

### Note 3: weights when using complement not per se higher

In the Lucas example:

- The order-restricted hypothesis  $H_1$  has 16.5 times more support than  $H_u$  (unconstrained).
- The order-restricted hypothesis  $H_1$  has 13.4 times more support than its complement  $H_m$ .

Now, the complement does not render higher weight than if  $H_u$  was used.

If  $H_m$  is not in agreement with the data, the complement does not always render a higher weight. This is actually a good thing: Against  $H_u$ ,  $H_m$  might obtain too much support then.

### Some more explanation

This is because for this particular example the mean of group 3 and 5 are close (and they are compared in Hypothesis  $H_1$ ). Notably, it will hold for all of the following three hypothesis:

```
H1_gr <- 'group5 > group3 > group1 > group2; group3 > group4 > group2'
# not full row-rank
H1_sm <- 'group5 < group3 > group1 > group2; group3 > group4 > group2'
# not full row-rank
H1 <- 'group5 = group3 > group1 > group2; group3 > group4 > group2'
# not full row-rank
```

Because the means of groups 3&5 are similar (and the other restrictions are in agreement with the data), the log likelihood values (LL's) of  $H_1$ ,  $H_u$  and  $H_c$  (i.e., the complement of  $H_1$ ) will be close - since the restricted/bounded solution (which is in agreement with  $H_1$ ) is near the (unconstrained) maximum likelihood estimate (mle). Since the penalty for the complement is (always) lower than for  $H_u$ , it will receive more support than  $H_u$  - thus  $H_1$  receives less. Notably, when  $H_1$  is very specific (as in Berzonsky et al Example below), the penalty of  $H_u$  and  $H_{\text{complement}}$  are almost the same.

When the means of groups 3 and 5 differ more, then evaluating against the complement does render a higher weight (than if  $H_u$  were used): Let us increase the mean of group 5 with 0.5 points (and change the name of the data & the analysis):

```
Lucas2 <- Lucas
Lucas2$Influence[Lucas$group == 5] <- Lucas$Influence[Lucas$group == 5] + 0.5
#describeBy(Lucas$Influence, Lucas$group, mat = TRUE)
#describeBy(Lucas2$Influence, Lucas$group, mat = TRUE)
lm_fit_Lucas2 <- lm(Influence ~ group-1, data = Lucas2)
# Now, from the three hypotheses H1_gr, H1_sm, and H1, # the first (H_gr) is
# correct. Hence, that one is used to illustrate the case where # a correct
# hypothesis obtains more support when it is evaluated against its complement:

H1_gr <- 'group5 > group3 > group1 > group2; group3 > group4 > group2'
# not full row-rank
```

```
set.seed(123) # Set seed value
output_u_gr <- goric(lm_fit_Lucas2, hypotheses = list(H1_gr = H1_gr))
summary(output_u_gr)
```

restriktor (0.5-20): generalized order-restricted information criterion:

Level probabilities:

Number of requested bootstrap draws 99999

Number of successful bootstrap draws for H1\_gr: 99999

Results:

	model	loglik	penalty	goric	loglik.weights	penalty.weights	goric.weights
1	H1_gr	-278.048	3.431	562.959	0.500	0.929	0.929
2	unconstrained	-278.048	6.000	568.097	0.500	0.071	0.071

---

Note: If log-likelihood values are equal or close to each other, the goric ratio weights are determined

Ratio GORIC-weights:

	vs. H1_gr	vs. unconstrained
H1_gr	1.000	13.053
unconstrained	0.077	1.000

---

Ratio loglik-weights:

	vs. H1_gr	vs. unconstrained
H1_gr	1	1
unconstrained	1	1

---

Ratio penalty-weights:

	vs. H1_gr	vs. unconstrained
H1_gr	1.000	13.053
unconstrained	0.077	1.000

---

order-restricted hypotheses:

H1\_gr:

group5 > group3 > group1 > group2; group3 > group4 > group2

Restriktor message: Since the constraint matrix is not full row-rank, the level probabilities are calculated using mix.weights = "boot" (the default is mix.weights = "pmvnorm").

For more information see ?restriktor.

*# The order-restricted hypothesis 'H1' has 13 times more support than Hu  
# (the unconstrained hypothesis). Notably, this is the maximum support H1 can  
# receive versus Hu (because the log likelihoods (LLs) are the same).*

```
set.seed(123) # Set seed value
output_c_gr <- goric(lm_fit_Lucas2, hypotheses = list(H1_gr = H1_gr), comparison = "complement")
summary(output_c_gr)
```

restriktor (0.5-20): generalized order-restricted information criterion:

Level probabilities:

Number of requested bootstrap draws 99999

Number of successful bootstrap draws for H1\_gr: 99999

Results:

	model	loglik	penalty	goric	loglik.weights	penalty.weights	goric.weights
1	H1_gr	-278.048	3.431	562.959	0.706	0.924	0.967
2	complement	-278.923	5.933	569.712	0.294	0.076	0.033

---

The order-restricted hypothesis 'H1\_gr' has 29.273 times more support than its complement.

Ratio GORIC-weights:

	vs. H1_gr	vs. complement
H1_gr	1.000	29.273
complement	0.034	1.000

---

Ratio loglik-weights:

	vs. H1_gr	vs. complement
H1_gr	1.000	2.398
complement	0.417	1.000

---

Ratio penalty-weights:

	vs. H1_gr	vs. complement
H1_gr	1.000	12.207
complement	0.082	1.000

---

order-restricted hypotheses:

H1\_gr:

group5 > group3 > group1 > group2; group3 > group4 > group2

Restriktor message: Since the constraint matrix is not full row-rank, the level probabilities are calculated using mix.weights = "boot" (the default is mix.weights = "pmvnorm").

For more information see ?restriktor.

```
#The order-restricted hypothesis 'H1' has 29 times more support than  
#                               its complement.
```

Now, when the means of groups 3 and 5 differ more, then evaluating against the complement does render a higher weight (29) than if H<sub>0</sub> were used (13).

## Extra: Two methods to calculate the penalty

There are two methods that can be used in calculating the penalty. The default method is often much faster (if the number of parameters is not too high) and needs less input specification. It can, however, not deal with hypotheses that are not of full row-rank (like  $H_1$  above). In that case, **restriktor** uses automatically the other (bootstrap) method. In this bootstrap method, one can also more easily change the number of iterations on which the penalty is based (mix.bootstrap). The computation time of this bootstrap method can be reduced by using multiple cores. For a windows device, you then have to use 'parallel = "snow"' (see the goric tutorial for more options). It is worth noting that setting up a local cluster can take some time if one is not provided. Therefore, in the example given below, using no parallel computing may actually be

faster. However, the advantage of using parallel computing becomes apparent when dealing with a much larger number of bootstrap samples. To use this bootstrap method (on a windows machine), use:

```
if (!require("parallel")) install.packages("parallel")
library(parallel)
nrCPUcores <- detectCores(all.tests = FALSE, logical = TRUE)

set.seed(123) # Set seed value
output_b <- goric(lm_fit_Lucas, hypotheses = list(H0 = H0, H1 = H1, H2 = H2),
                  mix.weights = "boot", parallel = "no", ncpus = nrCPUcores, mix.bootstrap = 99999)
summary(output_b)
```

restriktor (0.5-20): generalized order-restricted information criterion:

Level probabilities:

Number of requested bootstrap draws 99999  
 Number of successful bootstrap draws for H1: 99999  
 Number of successful bootstrap draws for H2: 99999

Results:

	model	loglik	penalty	goric	loglik.weights	penalty.weights	goric.weights
1	H0	-292.268	2.000	588.536	0.000	0.609	0.000
2	H1	-278.051	3.195	562.493	0.493	0.184	0.920
3	H2	-281.761	3.136	569.795	0.012	0.195	0.024
4	unconstrained	-278.048	6.000	568.097	0.495	0.011	0.056

---

Ratio GORIC-weights:

	vs. H0	vs. H1	vs. H2	vs. unconstrained
H0	1.00e+00	0.00e+00	0.00e+00	0.00e+00
H1	4.52e+05	1.00e+00	3.85e+01	1.65e+01
H2	1.17e+04	2.60e-02	1.00e+00	4.28e-01
unconstrained	2.74e+04	6.10e-02	2.34e+00	1.00e+00

---

Ratio loglik-weights:

	vs. H0	vs. H1	vs. H2	vs. unconstrained
H0	1.00e+00	0.00e+00	0.00e+00	0.00e+00
H1	1.49e+06	1.00e+00	4.08e+01	9.97e-01
H2	3.66e+04	2.40e-02	1.00e+00	2.40e-02
unconstrained	1.50e+06	1.00e+00	4.10e+01	1.00e+00

---

Ratio penalty-weights:

	vs. H0	vs. H1	vs. H2	vs. unconstrained
H0	1.000	3.304	3.116	54.598
H1	0.303	1.000	0.943	16.523
H2	0.321	1.061	1.000	17.524
unconstrained	0.018	0.061	0.057	1.000

---

order-restricted hypotheses:

H0:

group1 = group2 = group3 = group4 = group5

H1:

```
group5 = group3 > group1 > group2; group3 > group4 > group2
```

H2:

```
group3 > group1 > group4 = group5 > group2
```

This, of course, renders the same results as above (if there is a difference, it is in the second decimal of the penalty).

### **Note on not full row-rank**

If the restriction matrix is not of full row-rank, this means one of the following:

a) There is at least one redundant restriction.

Then, either a.1) leave the redundant one out or a.2) use another (more time-consuming) way of obtaining the level probabilities for the penalty term (goric function does this by default): Bootstrapping, as discussed above.

b) There is at least one range restriction (e.g.,  $-2 < \text{group1} < 2$ ).

Such a restriction can be evaluated but there is a sensitivity (of a scaling factor in the covariance matrix, like with a prior in a Bayes factor) which currently cannot be checked for.

c) There is at least one conflicting restriction (e.g.,  $2 < \text{group1} < -2$ ).

Such a restriction can evidently never hold and is thus impossible to evaluate. To prevent this type of error delete the one that is incorrect and apply the goric() again.