# Package ‘JMcmprsk’

June 19, 2020

**Type** Package

**Title** Joint Models for Longitudinal and Competing Risks Data

**Version** 0.9.8

**Date** 2020-6-13

**Author** Hong Wang [aut, cre], Ning Li [ctb] and Gang Li [ctb]

**Maintainer** Hong Wang <wh@csu.edu.cn>

**Description** Fit joint models of continuous or ordinal longitudinal data and time-to-event data with competing risks. For a detailed information, see Robert Elashoff, Gang Li and Ning Li (2016, ISBN:9781439807828); Robert M. Elashoff,Gang Li and Ning Li (200
0420.2007.00952.x> ; Ning Li, Robert Elashoff, Gang Li and Jeffrey Saver (2010) <doi:10.1002/sim.3798> .

**License** GPL (>= 2)

**Imports** Rcpp,MASS,stats,utils,statmod

**LinkingTo** Rcpp

**SystemRequirements** GNU GSL

**NeedsCompilation** yes

**RoxygenNote** 7.1.0

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**Encoding** UTF-8

**Repository** CRAN

**Date/Publication** 2020-06-19 05:50:15 UTC

## R topics documented:

## `anova.JMcmprsk` *ANOVA of longitudinal model*

### Description

Joint modelling for longitutal and censored data with competing risks

### Usage

```
## S3 method for class 'JMcmprsk'
anova(object, coeff = c("beta", "gamma", "alpha"), ...)
```

### Arguments

`object` The JMcmprsk object returned by either jmo or jmc function.

`coeff` Types of coefficients selected for anova. Note "alpha" is only avaiable to jmo type JMcmprsk object.

`...` further arguments passed to or from other methods.

### Value

Return a Wald test statistic and the p value

`beta` The Wald test for fixed effects for the longitutal part,i.e. $\beta$ in jmo or jmc output.
`gamma` The Wald test for fixed effects for the survival part,i.e. $\gamma$ in jmo or jmc output. "gamma1" stands for test for competin
`alpha` The Wald test for non-proportional odds covariates,i.e. $\alpha$ in jmo output.

## `coef.JMcmprsk` *Coefficients of longitudinal model*

### Description

Joint modelling for longitutal and censored data with competing risks

### Usage

```
## S3 method for class 'JMcmprsk'
coef(object, ...)
```

### Arguments

`object` The JMcmprsk object returned by either jmo or jmc function.

`...` further arguments passed to or from other methods.

### Value

Return estimates fixed effects with variable names

## jmc *Joint Modelling for Continuous outcomes*

### Description

Joint modeling of longitudinal continuous data and competing risks

### Usage

```
jmc(
  p,
  yfile,
  cfile,
  mfile,
  point = 20,
  maxiterations = 1e+05,
  do.trace = FALSE,
  type_file = TRUE
)
```

### Arguments

`p` The dimension of fixed effects (not including intercept) in yfile.

`yfile` Y matrix for longitudinal measurements in long format. For example, for a subject with n measurements, there should be n rows for this subject. The # of rows in y matrix is the total number of measurements for all subjects in the study. The columns in Y should start with the longitudinal outcome (column 1), the covariates for the random effects, and then the covariates for the fixed effects.

`cfile` C matrix for competing risks failure time data. Each subject has one data entry, so the number of rows equals to the number of subjects. The survival / censoring time is included in the first column, and the failure type coded as 0 (censored events), 1 (risk 1), or 2 (risk 2) is given in the second column. Two competing risks are assumed. The covariates are included in the third column and on.

`mfile` M vector to indicate the number of longitudinal measurements per subject. The number of rows equals to the number of subjects.

`point` Quadrature points used in the EM procedure.Default is 20.

`maxiterations` Maximum values of iterations. Default is 100000.

`do.trace` Print detailed information of each iteration. Default is false, i.e., not to print the iteration details.

`type_file` Types of inputs. Default is true, i.e. data files with headers. If set to "F", inputs are changed to data matrixes or data.frames (with headers)

### Value

Object of class `JMcmprsk` with elements

`vcmatrix` The variance-covariance matrix for all the parameters. The parameters are in the order: $\beta$, $\sigma^2$, $\gamma$, $\nu$, and $\Sigma$
`betas` The point estimates of $\beta$.
`se_betas` The standard error estimate of $\beta$.
`gamma_matrix` The point estimate of $\gamma$.
`se_gamma_matrix` The standard error estimate of $\gamma$.
`v_estimate` The point estimate of $\nu$.
`se_v_estimate` The standard error estimate of $\nu$.
`sigma2_val` The point estimate of $\sigma^2$.
`se_sigma2_val` The standard error estimate of $\sigma^2$.
`sigma_matrix` The point estimate of $\Sigma$ (only the upper triangle portion of the matrix is output).
`se_sigma` The standard error estimate of $\Sigma$.The standard errors are given in this order: main diagonal, the second m
`loglike` Log Likelihood.

### References

- Elashoff, Robert M., Gang Li, and Ning Li. "A joint model for longitudinal measurements and survival data in the presence of multiple failure types." Biometrics 64.3 (2008): 762-771.

### See Also

`jmo`

### Examples

```
# A toy example on simulated data
require(JMcmprsk)
set.seed(123)
yfile=system.file("extdata", "jmcsimy.txt", package = "JMcmprsk")
cfile=system.file("extdata", "jmcsimc.txt", package = "JMcmprsk")
mfile=system.file("extdata", "jmcsimm.txt", package = "JMcmprsk")
res2=jmc(p=4,yfile,cfile,mfile,point=6)
coef(res2)
#testing the function on real data with trace on

require(JMcmprsk)
set.seed(123)
yfile=system.file("extdata", "fvc621_y.txt", package = "JMcmprsk")
cfile=system.file("extdata", "fvc621_c.txt", package = "JMcmprsk")
mfile=system.file("extdata", "fvc621_m.txt", package = "JMcmprsk")
```

```
res1=jmc(p=7,yfile,cfile,mfile,do.trace = TRUE)
#if the input are not files but matrixes or data.frames,i.e. type_file=F
 ydata=read.table(yfile,header = T)
 cdata=read.table(cfile,header = T)
 mdata=read.table(mfile)
 res1=jmc(p=7,ydata,cdata,mdata, do.trace = TRUE,type_file = F)
coef(res1)
anova(res1,coeff="beta")
anova(res1,coeff="gamma")
```

## jmc_long *An integrated function for reconstructing data and do the joint modelling*

### Description

Reconstruct data into a regular longitudinal format as a refined dataset and do joint modelling for this refined data with continuous outcome.

### Usage

```
jmc_long(
  long_data,
  surv_data,
  out,
  FE,
  RE,
  ID,
  cate = NULL,
  intcpt = 1,
  quad.points = 10,
  max.iter = 10000,
  quiet = TRUE,
  do.trace = FALSE
)
```

### Arguments

| | |
|---|---|
| long_data | Data matrix for longitudinal in long form. The time variable should be labeled 'time'. |
| surv_data | Data matrix for competing risks data. Each subject has one row of observation (as opposed to the long_data). First and second column should be the observed event time and censoring indicator, respectively. The coding for the censoring indicator is as follows: 0 - censored events, 1 - risk 1, 2 - risk 2. Two competing risks are assumed. |
| out | Column name for outcome variable in long_data. |

FE Vector of column names that correspond to the fixed effects in long_data. If missing, then all columns except for the outcome and ID columns will be considered.

RE Types/Vector of random effects in long_data. The available type are "intercept", "linear", "quadratic" (time-related random effect specification) or other covariates in the input dataset. If specify other covariates, then they to be numerical vectors.

ID Column name for subject ID number in long_data.

cate Vector of categorical variables in long_data. Default is NULL.

intcpt Specify either 0 or 1. Default is set as 1. 0 means no intercept in random effect.

quad.points Number of quadrature points used in the EM procedure. Default is 20. Must be an even number. Larger values means higher accuracy but more time-consuming.

max.iter Max iterations. Default is 10000.

quiet Logical. Print progress of function. Default is TRUE.

do.trace Logical. Print the parameter estimates during the iterations. Default is FALSE.

## Value

Object of class JMcmprsk with elements

vcmatrix The variance-covariance matrix for all the parameters. The parameters are in the order: $\beta$, $\sigma^2$, $\gamma$, $\nu$, and

betas The point estimates of $\beta$.

se_betas The standard error estimate of $\beta$.

gamma_matrix The point estimate of $\gamma$.

se_gamma_matrix The standard error estimate of $\gamma$.

v_estimate The point estimate of $\nu$.

se_v_estimate The standard error estimate of $\nu$.

sigma2_val The point estimate of $\sigma^2$.

se_sigma2_val The standard error estimate of $\sigma^2$.

sigma_matrix The point estimate of $\Sigma$ (only the upper triangle portion of the matrix is output).

se_sigma The standard error estimate of $\Sigma$.The standard errors are given in this order: main diagonal, the second

loglike Log Likelihood.

## See Also

jmc

## Examples

```
## Not run:
yfile=system.file("extdata", "rawfvc621_y.txt", package = "JMcmprsk")
cfile=system.file("extdata", "fvc621_c.txt", package = "JMcmprsk")
cread <- read.table(file = "rawfvc621_c.txt", header = T)
yread <- read.table(file = "fvc621_y.txt", header = T)
res <- jmc_long(long_data = yread, surv_data = cread, out = "FVC", cate = NULL,
FE = c("time", "FVC0", "FIB0", "CYC", "FVC0.CYC", "FIB0.CYC", "time.CYC"),
RE = "time", ID = "rowId", intcpt = 1, quad.points = 8, max.iter = 10000, quiet = FALSE)
coef(res)
anova(res,coeff="beta")
anova(res,coeff="gamma")
#make up two categorical variables and add them into yread
require(tidyverse)
mfile=system.file("extdata", "fvc621_m.txt", package = "JMcmprsk")
mread <- read.table(file = "fvc621_m.txt", header = T)
rowId <- c(1:nrow(cread))
sex <- sample(c("Feamle", "Male"), nrow(mread), replace = T)
race <- sample(c("White", "Black", "Asian", "Hispanic"), nrow(mread), replace = T)
cate_var <- data.frame(rowId, sex, race)
yread <- left_join(yread, cate_var, by = "rowId")
# run jmc_long function again for yread file with two added categorical variables
res2 <- jmc_long(long_data = yread, surv_data = cread, out = "FVC", cate = c("sex", "race"),
FE = c("time", "FVC0", "FIB0", "CYC", "FVC0.CYC", "FIB0.CYC", "time.CYC"),
RE = "time", ID = "rowId", intcpt = 1, quad.points = 8, max.iter = 10000, quiet = TRUE)

## End(Not run)
```

## jmo *Joint Modelling for Ordinal outcomes*

### Description

Joint modeling of longitudinal ordinal data and competing risks

### Usage

```
jmo(
  p,
  s,
  yfile,
  cfile,
  mfile,
  point = 20,
  maxiterations = 1e+05,
  do.trace = FALSE,
  type_file = TRUE
)
```

### Arguments

`p` The dimension of proportional odds covariates (not including intercept) in yfile.

`s` The dimension of non-proportional odds covariates in yfile.

`yfile` Y matrix for longitudinal measurements in long format. For example, for a subject with n measurements, there are n rows for this subject. The # of rows in y matrix is the total number of measurements for all subjects. The columns in Y are ordered this way: the longitudinal outcome (column 1), then the covariates for random effects, and lastly, the covariates for fixed effects (no intercept).

`cfile` C matrix for competing risks failure time data. Each subject has one data entry, so the number of rows equals to the number of subjects. The survival / censoring time is included in the first column, and the failure type coded as 0 (censored events), 1 (risk 1), or 2 (risk 2) is given in the second column. Two competing risks are assumed. The covariates are included in the third column and on.

`mfile` M vector to indicate the number of longitudinal measurements per subject. The number of rows equals to the number of subjects.

`point` Quadrature points used in the EM procedure. Default is 20.

`maxiterations` Maximum values of iterations. Default is 100000.

`do.trace` Print detailed information of each iteration. Default is false, not to print the iteration details.

`type_file` Types of inputs. Default is true, i.e. data files with headers. If set to "F", inputs are changed to data matrixes or data.frames (with headers)

## Value

Object of class `JMcmprsk` with elements

`vcmatrix` The variance-covariance matrix for all the parameters. The parameters are in the order: $\beta$, $\alpha$, $\theta$, $\gamma$, $\nu$, and

`betas` The point estimates of $\beta$.

`se_betas` The standard error estimate of $\beta$.

`alphamatrix` The point estimates of $\alpha$.

`se_alphas` The standard error estimate of $\alpha$.

`theta` The point estimates of $\theta$.

`se_theta` The standard error estimate of $\theta$.

`gamma_matrix` The point estimate of $\gamma$.

`se_gamma_matrix` The standard error estimate of $\gamma$.

`v_estimate` The point estimate of $\nu$.

`se_v_estimate` The standard error estimate of $\nu$.

`sigma_matrix` The point estimate of $\Sigma$ (only the upper triangle portion of the matrix is output).

`se_sigma` The standard error estimate of $\Sigma$.The standard errors are given in this order: main diagonal, the second m

`loglike` Log Likelihood.

## References

- Ning Li,Robert M. Elashoff,Gang Li and Jeffrey Saver. "Joint modeling of longitudinal ordinal data and competing risks survival times and analysis of the NINDS rt-PA stroke trial." Statistics in medicine 29.5 (2010): 546-557.

## See Also

`jmc`

## Examples

```
# A toy example on simulated data
 require(JMcmprsk)
 set.seed(123)
 yfile=system.file("extdata", "jmosimy.txt", package = "JMcmprsk")
 cfile=system.file("extdata", "jmosimc.txt", package = "JMcmprsk")
 mfile=system.file("extdata", "jmosimm.txt", package = "JMcmprsk")
 res3=jmo(p=3,s=1, yfile,cfile,mfile,point=6)
 coef(res3)
#testing the function on real data with trace on

require(JMcmprsk)
set.seed(123)
yfile=system.file("extdata", "ninds_nrank_y.txt", package = "JMcmprsk")
cfile=system.file("extdata", "ninds_nrank_c.txt", package = "JMcmprsk")
mfile=system.file("extdata", "ninds_nrank_m.txt", package = "JMcmprsk")
res=jmo(p=9,s=2, yfile,cfile,mfile,point=6,do.trace = TRUE)
#if the input are not files but matrixes or data.frames,i.e. type_file=F
ydata=read.table(yfile,header = T)
cdata=read.table(cfile,header = T)
mdata=read.table(mfile)
res=jmo(p=9,s=2, ydata,cdata,mdata,point=6,do.trace = TRUE,type_file = F)
coef(res)
anova(res,coeff="beta")
anova(res,coeff="gamma")
anova(res,coeff="alpha")
```

# jmo_long *An integrated function for reconstructing data and do the joint modelling*

## Description

Reconstruct data into a regular longitudinal format as a refined dataset and do joint modelling for this refined data with ordinal outcome.

## Usage

```
jmo_long(
  long_data,
  surv_data,
  out,
  FE,
  RE,
  NP,
  ID,
  cate = NULL,
  intcpt = 1,
  quad.points = 20,
  max.iter = 10000,
  quiet = TRUE,
  do.trace = FALSE
)
```

## Arguments

`long_data` Data matrix for longitudinal in long format. The time variable should be labeled 'time'.

`surv_data` Data matrix for competing risks data. Each subject has one row of observation (as opposed to the long_data). First and second column should be the observed event time and censoring indicator, respectively. The coding for the censoring indicator is as follows: 0 - censored events, 1 - risk 1, 2 - risk 2. Two competing risks are assumed.

`out` Column name for outcome variable in long_data.

`FE` Vector of column names that correspond to the fixed effects in long_data. If missing, then all columns except for the outcome and ID columns will be considered.

`RE` Types/Vector of random effects in long_data. The available type are "intercept", "linear", "quadratic" (time-related random effect specification) or other covariates in the input dataset.

`NP` Vector of column names that correspond to the non-proportional odds covariates. It won't run the model if NP is not specified.

`ID` Column name for subject ID number in long_data.

`cate` Vector of categorical variables in long_data.

`intcpt` Specify either 0 or 1. Default is set as 1. 0 means no intercept in random effect.

`quad.points` Number of quadrature points used in the EM procedure. Default is 20. Must be an even number. Larger values means higher accuracy but more time-consuming.

`max.iter` Max iterations. Default is 10000.

`quiet` Logical. Print progress of function. Default is TRUE.

`do.trace` Logical. Print the parameter estimates during the iterations. Default is FALSE.

## Value

Object of class `JMcmprsk` with elements

`vcmatrix` The variance-covariance matrix for all the parameters. The parameters are in the order: $\beta$, $\sigma^2$, $\gamma$, $\nu$, and

`betas` The point estimates of $\beta$.

`se_betas` The standard error estimate of $\beta$.

`gamma_matrix` The point estimate of $\gamma$.

`se_gamma_matrix` The standard error estimate of $\gamma$.

`v_estimate` The point estimate of $\nu$.

`se_v_estimate` The standard error estimate of $\nu$.

`sigma2_val` The point estimate of $\sigma^2$.

`se_sigma2_val` The standard error estimate of $\sigma^2$.

`sigma_matrix` The point estimate of $\Sigma$ (only the upper triangle portion of the matrix is output).

`se_sigma` The standard error estimate of $\Sigma$.The standard errors are given in this order: main diagonal, the second m

`loglike` Log Likelihood.

## See Also

jmo

## Examples

```
## Not run:
  require(tidyverse)
  yfile = system.file("extdata", "ninds_nrank_y.txt", package = "JMcmprsk")
  cfile = system.file("extdata", "ninds_nrank_c.txt", package = "JMcmprsk")
  mfile = system.file("extdata", "ninds_nrank_m.txt", package = "JMcmprsk")
  yread = read.table(yfile, header = TRUE)
  cread = read.table(cfile, header = TRUE)
  mread = read.table(mfile)
  # make a raw yread: this is a dataset like what users usually have
  yread <- yread[, -c(2:4)]
  # create an ID column for each subject and add it to yread
  ID <- vector()
  for (i in 1:nrow(mread)) {
  ID <- c(ID, replicate(mread[i, 1], i))
  }
  yread <- data.frame(ID, yread)
  ID <- c(1:nrow(cread))
  #Create two categorical variables and add them into yread
  set.seed(100)
  sex <- sample(c("Feamle", "Male"), nrow(mread), replace = T)
  race <- sample(c("White", "Black", "Asian", "Hispanic"), nrow(mread), replace = T)
  cate_var <- data.frame(ID, sex, race)
  yread <- left_join(yread, cate_var, by = "ID")
  # run jmo_long function again for yread file with two added categorical variables
  res1 <- jmo_long(yread, cread, out = "Y",
  FE = c("group", "time3", "time6", "time12", "mrkprior",
  "smlves", "lvORcs", "smlves.group", "lvORcs.group"), cate = c("sex", "race"),
  RE = "intercept", NP = c("smlves", "lvORcs", "sex", "race"), ID = "ID",intcpt = 1,
  quad.points = 12, max.iter = 1000, quiet = FALSE)

## End(Not run)
```

## print.JMcmprsk *Print JMcmprsk*

## Description

Print contents of JMcmprsk object.

### Usage

```
## S3 method for class 'JMcmprsk'
print(x, ...)
```

### Arguments

| | |
|---|---|
| `x` | Object of class ’JMcmprsk’. |
| `...` | Further arguments passed to or from other methods. |

### Author(s)

Hong Wang

### See Also

`jmc`

## SimDataC *Data simulation of continuous outcomes and competing risks*

### Description

Simulation of continuous longitudinal outcome and competing risks data Currently, only the simulation in Elashoff et al(2008) is implemented.

### Usage

```
SimDataC(
  k_val,
  p1_val,
  p1a_val,
  p2_val,
  g_val,
  truebeta,
  truegamma,
  randeffect,
  yfn,
  cfn,
  mfn
)
```

## Arguments

`k_val` The number of subjects in study.

`p1_val` The dimension of fixed effects in longitudinal measurements.

`p1a_val` The dimension of random effects in longitudinal measurements.

`p2_val` The dimension of fixed effects in competing risks failure time data.

`g_val` The number of type of failure in competing risks data.

`truebeta` True values for beta, the longitudinal coefficients.

`truegamma` True values for gamma, the survival coefficients.

`randeffect` True values for random effects in longitudinal and competing risks parts,namely in the order of $\sigma$,$\sigma_b$,$\nu_2$,$\sigma_u$.

`yfn` Filename of genereated Y matrix for longitudinal measurements in long format.

`cfn` Filename of genereated C matrix for competing risks failure time data.

`mfn` Filename of genereated M vector to indicate the number of longitudinal measurements per subject.

## Value

Files with names yfn, cfn and mfn.

`censoring_rate` Censoring rate of the survival data.

`rate1` Censoring rate of competing risk 1.

`rate2` Censoring rate of competing risk 2.

`yfn` Filename of genereated Y matrix for longitudinal measurements.

`cfn` Filename of genereated C matrix for competing risks failure time data.

`mfn` Filename of genereated M vector to indicate the number of longitudinal measurements per subject.

## References

- Elashoff, Robert M., Gang Li, and Ning Li. "A joint model for longitudinal measurements and survival data in the presence of multiple failure types." Biometrics 64.3 (2008): 762-771.

## See Also

SimDataO

## Examples

```
# A toy example testint data generations
require(JMcmprsk)
set.seed(123)
yfn=tempfile(pattern = "", fileext = ".txt")
cfn=tempfile(pattern = "", fileext = ".txt")
mfn=tempfile(pattern = "", fileext = ".txt")
k_val=30;p1_val=4;p1a_val=1; p2_val=2;g_val=2;
truebeta=c(10,-1,1.5,0.6);truegamma=c(0.8,-1,0.5,-1); randeffect=c(5,0.5,0.5,0.5);
#writing files
```

```
SimDataC(k_val, p1_val, p1a_val, p2_val, g_val,truebeta,
        truegamma, randeffect, yfn,  cfn,  mfn)

jmc(p=4,yfn,cfn,mfn,point=6)
```

## SimDataO *Data simulation of ordinal outcomes and competing risks*

### Description

Simulation of ordinal longitudinal outcome and competing risks data Currently, only the simulation in Li et al(2010) is implemented.

### Usage

```
SimDataO(
  k_val,
  p1_val,
  p1a_val,
  p2_val,
  g_val,
  truebeta,
  truetheta,
  truegamma,
  randeffect,
  yfn,
  cfn,
  mfn
)
```

### Arguments

| | |
|---|---|
| k_val | The number of subjects in study. |
| p1_val | The dimension of fixed effects in longitudinal measurements. |
| p1a_val | The dimension of random effects in longitudinal measurements. |
| p2_val | The dimension of fixed effects in competing risks failure time data. |
| g_val | The number of type of failure in competing risks data. |
| truebeta | True values for beta, the longitudinal coefficients. |
| truetheta | True values for theta, subset of the non-proportional odds longitudinal coefficients. |
| truegamma | True values for gamma, the survival coefficients. |
| randeffect | True values for random effects in longitudinal and competing risks parts,namely in the order of $\sigma_b$,$\nu_2$,$\sigma_u$. |
| yfn | Filename of genereated Y matrix for longitudinal measurements in long format. |

cfn Filename of genereated C matrix for competing risks failure time data.

mfn Filename of genereated M vector to indicate the number of longitudinal measurements per subject.

## Value

Files with names yfn, cfn and mfn.

censoring_rate Censoring rate of the survival data.

rate1 Censoring rate of competing risk 1.

rate2 Censoring rate of competing risk 2.

yfn Filename of genereated Y matrix for longitudinal measurements.

cfn Filename of genereated C matrix for competing risks failure time data.

mfn Filename of genereated M vector to indicate the number of longitudinal measurements per subject.

## References

- Ning Li,Robert M. Elashoff,Gang Li and Jeffrey Saver. "Joint modeling of longitudinal ordinal data and competing risks survival times and analysis of the NINDS rt-PA stroke trial." Statistics in medicine 29.5 (2010): 546-557.

## See Also

SimDataC

## Examples

```
# A toy example testint data generations
require(JMcmprsk)
set.seed(123)
yfn=tempfile(pattern = "", fileext = ".txt")
cfn=tempfile(pattern = "", fileext = ".txt")
mfn=tempfile(pattern = "", fileext = ".txt")
 k_val=50;p1_val=3;p1a_val=1; p2_val=2;g_val=2;
truebeta=c(-1,1.5,0.8);truetheta=c(-0.5,1);truegamma=c(0.8,-1,0.5,-1); randeffect=c(1,0.5,0.5);
#writing files
SimDataO(k_val, p1_val, p1a_val, p2_val, g_val,
      truebeta, truetheta, truegamma, randeffect, yfn,  cfn,  mfn)

jmo(p=3,s=1, yfn,cfn,mfn,point=10,do.trace = TRUE)
```

# Index