

Package ‘JMcmprsk’

May 17, 2019

Type Package

Title Joint Models for Longitudinal Measurements and Competing Risks
Failure Time Data

Version 0.9.4

Date 2019-5-8

Author

Hong Wang<wh@csu.edu.cn>, Ning Li<nli@biomath.ucla.edu> and Gang Li<vli@ucla.edu>

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 (200420.2007.00952.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 6.1.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

Repository CRAN

Date/Publication 2019-05-17 08:30:03 UTC

R topics documented:

anova.JMcmprsk	2
coef.JMcmprsk	2
jmc	3
jmo	6
print.JMcmprsk	8

Index**9**

anova.JMcmprsk	<i>ANOVA of longitudinal model</i>
----------------	------------------------------------

Description

Joint modelling for longitudinal 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 available 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 longitudinal part, i.e. β in jmo or jmc output.
gamma	The Wald test for fixed effects for the survival part, i.e. γ in jmo or jmc output. "gamma1" stands for test for competing risks.
alpha	The Wald test for non-proportional odds covariates, i.e. α in jmo output.

coef.JMcmprsk	<i>Coefficients of longitudinal model</i>
---------------	---

Description

Joint modelling for longitudinal and censored data with competing risks

Usage

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


Value

Object of class JMcmpsk with elements

vcmatrix	The variance-covariance matrix for all the parameters. The parameters are in the order: β , σ^2 , γ , ν , and Σ .
betas	The point estimates of β .
se_betas	The standard error estimate of β .
gamma_matrix	The point estimate of γ .
se_gamma_matrix	The standard error estimate of γ .
v_estimate	The point estimate of ν .
se_v_estimate	The standard error estimate of ν .
sigma2_val	The point estimate of σ^2 .
se_sigma2_val	The standard error estimate of σ^2 .
sigma_matrix	The point estimate of Σ (only the upper triangle portion of the matrix is output).
se_sigma	The standard error estimate of Σ . The standard errors are given in this order: main diagonal, the second main diagonal, the third main diagonal, and so on.
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)
anova(res2,coeff="beta")
anova(res2,coeff="gamma")
#testing the function on real data with trace on
## Not run:
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=8,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=8,ydata,cdata,mdata, do.trace = TRUE,type_file = F)
coef(res1)
anova(res1,coeff="beta")
anova(res1,coeff="gamma")
```

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

Value

Object of class JMmprsk with elements

vcmatrix	The variance-covariance matrix for all the parameters. The parameters are in the order: β , α , θ , γ , ν , and Σ .
betas	The point estimates of β .
se_betas	The standard error estimate of β .
alphamatrix	The point estimates of α .
se_alphas	The standard error estimate of α .
theta	The point estimates of θ .
se_theta	The standard error estimate of θ .
gamma_matrix	The point estimate of γ .
se_gamma_matrix	The standard error estimate of γ .
v_estimate	The point estimate of ν .
se_v_estimate	The standard error estimate of ν .
sigma_matrix	The point estimate of Σ (only the upper triangle portion of the matrix is output).
se_sigma	The standard error estimate of Σ . The standard errors are given in this order: main diagonal, the second main diagonal, and the off-diagonal elements.
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(JMmprsk)
set.seed(123)
yfile=system.file("extdata", "jmosimy.txt", package = "JMmprsk")
cfile=system.file("extdata", "jmosimc.txt", package = "JMmprsk")
mfile=system.file("extdata", "jmosimm.txt", package = "JMmprsk")
res3=jmo(p=3,s=1, yfile,cfile,mfile,point=6)
coef(res3)
anova(res3,coeff="beta")
anova(res3,coeff="gamma")
anova(res3,coeff="alpha")
#testing the function on real data with trace on
## Not run:
require(JMmprsk)
set.seed(123)
yfile=system.file("extdata", "ninds_nrank_y.txt", package = "JMmprsk")
cfile=system.file("extdata", "ninds_nrank_c.txt", package = "JMmprsk")
mfile=system.file("extdata", "ninds_nrank_m.txt", package = "JMmprsk")
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")

## End(Not run)
```

`print.JMcmprsk` *Print JMcmprsk*

Description

Print contents of JMcmprsk object.

Usage

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

Arguments

<code>x</code>	Object of class 'JMcmprsk'.
<code>...</code>	Further arguments passed to or from other methods.

Author(s)

Hong Wang

See Also

[jmc](#)

Index

`anova.JMcmprsk`, 2

`coef.JMcmprsk`, 2

`jmc`, 3, 7, 8

`jmo`, 5, 6

`print.JMcmprsk`, 8