

# Package ‘CoxPlus’

October 12, 2022

**Type** Package

**Title** Cox Regression (Proportional Hazards Model) with Multiple Causes and Mixed Effects

**Version** 1.1.1

**Date** 2015-10-24

**Author** Jing Peng

**Maintainer** Jing Peng <pengjing@live.com>

**Description** A high performance package estimating Cox Model when an event has more than one causes. It also supports random and fixed effects, tied events, and time-varying variables.

**License** GPL (>= 3)

**LazyData** TRUE

**Depends** R (>= 3.1.0), Rcpp (>= 0.12.0)

**Imports** methods

**LinkingTo** Rcpp, RcppArmadillo

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2015-10-24 09:32:41

## R topics documented:

CoxPlus . . . . .	2
<b>Index</b>	<b>3</b>

CoxPlus

*Cox Regression (Proportional Hazards Model) with Multiple Causes and Mixed Effects*

## Description

A high performance package estimating Proportional Hazards Model when an even can have more than one causes, including support for random and fixed effects, tied events, and time-varying variables.

## Usage

```
fastCox(head, formula, par = list(), data = NULL)
```

## Arguments

head	A data frame with 4~5 columns: start, stop, event, weight, strata (optional).
formula	A formula specifying the independent variables
par	A optional list of parameters controlling the estimation process
data	The dataset, a data frame containing observations on the independent variables

## Value

A list containing the estimated parameters

## References

1. Jing Peng, Ashish Agarwal, Kartik Hosanagar, and Raghuram Iyengar. Towards Effective Information Diffusion on Social Media Platforms: A Dyadic Analysis of Network Embeddedness. Working Paper.
2. Jing Peng, Ashish Agarwal, Kartik Hosanagar, and Raghuram Iyengar. Toward Effective Social Contagion: A Micro Level Analysis of the Impact of Dyadic Network Relationship. In Proceedings of the 2014 International Conference on Information Systems.

## Examples

```
# Simulate a dataset. lam=exp(x), suvtime depends on lam
x = rnorm(5000)
suvtime = -log(runif(length(x)))/exp(x)
# Censor 80% of events
thd = quantile(suvtime, 0.2)
event = as.numeric(suvtime <= thd)
suvtime[suvtime>thd] = thd

# The estimates of beta should be very close to 1, the true value
head = cbind(start=0, stop=suvtime, event=event, weight=1)
est = fastCox(head, ~x)
print(est$result)
```

# Index

CoxPlus, [2](#)

CoxPlus-package (CoxPlus), [2](#)

fastCox (CoxPlus), [2](#)