Package 'survMS'

April 16, 2021

Title Survival Model Simulation

Version 0.0.1

Depends ggplot2

URL https://github.com/mathildesautreuil/survMS/

BugReports https://github.com/mathildesautreuil/survMS/issues

Maintainer Mathilde Sautreuil <mathilde.sautreuil@gmail.com>

Description Package enables the data simulation from different survival models (Cox, AFT, and AH models). The simulated data will have various levels of complexity according to the survival model considered. The implemented methods for the Cox model are described in Ralf Bender, Thomas Augustin, Maria Blettner (2004) <doi:10.1002/sim.2059>.

License GPL-3

Encoding UTF-8

LazyData false

Suggests knitr, rmarkdown

Imports graphics

VignetteBuilder knitr

RoxygenNote 7.1.1

NeedsCompilation no

Author Mathilde Sautreuil [aut, cre]

Repository CRAN

Date/Publication 2021-04-16 07:00:02 UTC

R topics documented:

`get_param_exp` *Getting parameters of exponential distribution*

Description

Getting parameters of exponential distribution

Usage

```
get_param_exp(int_a = c(1e-06, 110), med, mu)
```

Arguments

`int_a`	: grid of first parameter of the distribution
`med`	: median of real dataset
`mu`	: mean of real dataset

Value

: list with parameter values of the distribution

Examples

```
library(survMS)
```

`get_param_ln` *Getting parameters of log-normal distribution*

Description

Getting parameters of log-normal distribution

Usage

```
get_param_ln(var = 170000, mu = 2325)
```

Arguments

var	: variance of real dataset
mu	: mean of real dataset

Value

list with parameter values of the distribution

Examples

```
library(survMS)
```

---

get_param_ln2 *Getting parameters of log-normal distribution*

Description

Getting parameters of log-normal distribution

Usage

```
get_param_ln2(int_a = c(0.1, 11), med, mu)
```

Arguments

int_a	: grid of first parameter of the distribution
med	: median of real dataset
mu	: mean of real dataset

Value

: list with parameter values of the distribution

Examples

```
library(survMS)
```

get_param_weib *Get Parameters for weibull distribution*

Description

Get Parameters for weibull distribution

Usage

```
get_param_weib(int_a = c(0.1, 11), med, mu)
```

Arguments

int_a	: grid of first parameter of the distribution
med	: median of real dataset
mu	: mean of real dataset

Value

: list with parameter values of the distribution

Examples

```
library(survMS)
```

hist.modSim *Histogram of survival times*

Description

Histogram of survival times

Usage

```
## S3 method for class 'modSim'
hist(x, ...)
```

Arguments

x	output of modelSim function (must be of type modSim)
...	supplementary parameters

Value

hist x

Examples

```
library(survMS)
### Survival data simulated from AH model
res_paramLN = get_param_ln(var=170000, mu=2325)
listAHSim_n500_p1000 <- modelSim(model = "AH", matDistr = "unif", matParam = c(-1,1), n = 500,
                          p = 100, pnonull = 100, betaDistr = 1.5, hazDistr = "log-normal",
                                  hazParams = c(res_paramLN$a*4, res_paramLN$lambda),
                                  Phi = 0, seed = 1, d = 0)

### Histogram of survival times
hist(listAHSim_n500_p1000)
```

`modelSim` *Data simulation from different survival models*

Description

Data simulation from different survival models

Usage

```
modelSim(
  model = "cox",
  matDistr,
  matParam,
  n,
  p,
  pnonull,
  betaDistr,
  hazDistr,
  hazParams,
  seed,
  Phi = NULL,
  d = 0,
  pourc = 0.9
)
```

Arguments

`model`	Survival model: "cox", "AFT", "AFTshift" or "AH"
`matDistr`	Distribution of matrix
`matParam`	Parameters of matrix
`n`	size of sample
`p`	number of parameters
`pnonull`	number of partinent covariates
`betaDistr`	Distribution of beta or vector of beta

hazDistr	distribution of baseline hazard
hazParams	Parameters of baseline hazard
seed	seed
Phi	nonlinearity (not coded)
d	censorship
pourc	pourcents

Details

This function simulates survival data from different models: Cox model, AFT model and AH model. 1. The Cox model is defined as: $\lambda(t|X) = \alpha_0(t)\exp(\beta^T X_{i.})$, with $\alpha_0(t)$ is the baseline risk and β is the vector of coefficients. Two distributions are considered for the baseline risk:

- Weibull: $\alpha_0(t) = \lambda a t^{(a-1)}$;
- Log-normal: $\alpha_0(t) = (1/(\sigma\sqrt{(}2\pi t)\exp[-(\log t - \mu)^2/2\sigma^2]))/(1 - \Phi[(\log t - \mu)/\sigma])$;
- Exponential: $\alpha_0(t) = \lambda$;
- Gompertz: $\alpha_0(t) = \lambda\exp(\alpha t)$.

To Simulate the covariates, two distributions are also proposed:

- Uniform
- Normal

and the choice of parameters The Phi parameter enables to simulate survival data in a linear framework with no interaction, but its future implementation will take into account a non-linear framework with interactions. If the parameter Phi is NULL (to complete...).

2. The AFT model is defined from a linear regression of the interest covariate: $Y_i = X_{i.}\beta + W_i$, with $X_{i.}$ the covariates, β the vector of regression coefficients et ϵ_i the error term AFT model can also be defined from the baseline survival function $S_0(t)$, corresponding distribution tail $\exp(\epsilon_i)$. Survival function of AFT model is written as: $S(t|X_{i.}) = S_0(t\exp(\beta^T X_{i.}))$, and the expression of hazard risk is the form of: $\lambda(t|X_{i.}) = \exp(\beta^T X_{i.})\alpha_0(t\exp(\beta^T X_{i.}))$. with $\alpha_0(t)$ is the baseline risk and β is the vector of coefficients. The advantage of AFT model is that the variables have a multiplicative effect on t rather than on the risk function, as is the case in Cox model. Two distributions are considered for the baseline risk:

- Weibull: $\alpha_0(t) = \lambda a t^{(a-1)}$;
- Log-normal: $\alpha_0(t) = (1/(\sigma\sqrt{(}2\pi t)\exp[-(\log t - \mu)^2/2\sigma^2]))/(1 - \Phi[(\log t - \mu)/\sigma])$

.

To Simulate the covariates, two distributions are also proposed:

- Uniform
- Normal

and the choice of parameters The Phi parameter enables to simulate survival data in a linear framework with no interaction, but its future implementation will take into account a non-linear framework with interactions. If the parameter Phi is NULL (to complete...). 3. The hazard risk of the AH model is defined for an individual i as: $\lambda_{AH}(t|X_{i.}) = \alpha_0(t\exp(\beta^T X_{i.}))$, with α_0 the baseline risk and β the vector of regression parameters. In a model with only one binary variable considered that corresponds to the treatment, the hazard risk is written as follows: $\lambda_1(t) = \alpha_0(\beta t)$. with α_0 the baseline risk and β the vector of regression parameters. In a model with only one binary variable considered that corresponds to the treatment, the hazard risk is written as follows: $\lambda_1(t) = \alpha_0(\beta t)$. The regression vector β characterizes the influence of variables on the survival time of individuals, and $\exp(\beta^T X_{i.})$ is a factor altering the time scale on hazard risk. The positive or negative value of $\beta^T X_{i.}$ will respectively imply an acceleration or deceleration of the risk.The AH model is defined from a linear regression of the interest covariate: Two distributions are considered for the baseline risk:

- Weibull: $\alpha_0(t) = \lambda a t^{(a-1)}$;
- Log-normal: $\alpha_0(t) = (1/(\sigma\sqrt(2\pi t)\exp[-(\log t - \mu)^2/2\sigma^2]))/(1 - \Phi[(\log t - \mu)/\sigma])$.

To Simulate the covariates, two distributions are also proposed:

- Uniform
- Normal

and the choice of parameters The Phi parameter enables to simulate survival data in a linear framework with no interaction, but its future implementation will take into account a non-linear framework with interactions. If the parameter Phi is NULL (to complete...).

sim$model <- model

Value

modelSim returns a list containing:

- model model (Cox, AFT, AFTshift, AH)
- Z Matrix of covariates
- Y random covariates
- TC Vector of survival times
- delta Vector of censorship indicator
- betanorm Vector of normalized regression parameter
- crate Censorship rate
- crate_delta Censorship rate
- vecY Vector of number of individuals at risk at time t_i
- hazParams Vector of parameter distribution of the baseline hazard function
- hazDistr Distribution of the baseline hazard function
- St Matrix of survival functions
- ht Matrix of hazard risk functions
- grilleTi Time grid

Author(s)

Mathilde Sautreuil

See Also

print.modSim,plot.modSim

Examples

```
library(survMS)
### Survival data simulated from Cox model
res_paramW = get_param_weib(med = 2228, mu = 2325)
listCoxSim_n500_p1000 <- modelSim(model = "cox", matDistr = "unif", matParam = c(-1,1), n = 500,
                            p = 1000, pnonull = 20, betaDistr = 1, hazDistr = "weibull",
                           hazParams = c(res_paramW$a, res_paramW$lambda), seed = 1, d = 0)
print(listCoxSim_n500_p1000)
hist(listCoxSim_n500_p1000)
plot(listCoxSim_n500_p1000, ind = sample(1:500, 5))
plot(listCoxSim_n500_p1000, ind = sample(1:500, 5), type = "hazard")

df_p1000_n500 = data.frame(time = listCoxSim_n500_p1000$TC,
                           event = listCoxSim_n500_p1000$delta,
                           listCoxSim_n500_p1000$Z)
df_p1000_n500[1:6,1:10]
dim(df_p1000_n500)
### Survival data simulated from AFT model
res_paramLN = get_param_ln(var = 200000, mu = 1134)
listAFTSim_n500_p1000 <- modelSim(model = "AFT", matDistr = "unif", matParam = c(-1,1), n = 500,
                            p = 100, pnonull = 100, betaDistr = 1, hazDistr = "log-normal",
                                 hazParams = c(res_paramLN$a, res_paramLN$lambda),
                                 Phi = 0, seed = 1, d = 0)
hist(listAFTSim_n500_p1000)
plot(listAFTSim_n500_p1000, ind = sample(1:500, 5))
df_p1000_n500 = data.frame(time = listAFTSim_n500_p1000$TC,
                           event = listAFTSim_n500_p1000$delta,
                           listAFTSim_n500_p1000$Z)
df_p1000_n500[1:6,1:10]
dim(df_p1000_n500)

### Survival data simulated from AH model
res_paramLN = get_param_ln(var=170000, mu=2325)
listAHSim_n500_p1000 <- modelSim(model = "AH", matDistr = "unif", matParam = c(-1,1), n = 500,
                           p = 100, pnonull = 100, betaDistr = 1.5, hazDistr = "log-normal",
                                  hazParams = c(res_paramLN$a*4, res_paramLN$lambda),
                                  Phi = 0, seed = 1, d = 0)

print(listAHSim_n500_p1000)
hist(listAHSim_n500_p1000)
plot(listAHSim_n500_p1000, ind = sample(1:500, 5))
plot(listAHSim_n500_p1000, ind = sample(1:500, 5), type = "hazard")
```

plot.modSim *Survival or hazard curves of simulated data*

Description

Survival or hazard curves of simulated data

Usage

```
## S3 method for class 'modSim'
plot(x, ind, type = "surv", ...)
```

Arguments

x	output of modelSim function (must be of type modSim)
ind	vector (individuals to show)
type	type of plots (survival or hazard curves)
...	supplementary parameters

Value

plot x

Examples

```
library(survMS)
ind = sample(1:500, 5)
### Example with survival data simulated from AH model
res_paramLN = get_param_ln(var=170000, mu=2325)
listAHSim_n500_p1000 <- modelSim(model = "AH", matDistr = "unif", matParam = c(-1,1), n = 500,
                          p = 100, pnonull = 100, betaDistr = 1.5, hazDistr = "log-normal",
                                  hazParams = c(res_paramLN$a*4, res_paramLN$lambda),
                                  Phi = 0, seed = 1, d = 0)
### Two types of plot are available (survival (by default) and hazard curves)
## Survival curves
plot(listAHSim_n500_p1000, ind = ind)
## Hazard curves
plot(listAHSim_n500_p1000, ind = ind, type = "hazard")
```

print.modSim *Print information about data simulation*

Description

Print information about data simulation

Usage

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

Arguments

x	output of modelSim function (must be of type modSim)
...	supplementary parameters

Value

print x

Examples

```
library(survMS)
### Survival data simulated from AH model
res_paramLN = get_param_ln(var=170000, mu=2325)
listAHSim_n500_p1000 <- modelSim(model = "AH", matDistr = "unif", matParam = c(-1,1), n = 500,
                          p = 100, pnonull = 100, betaDistr = 1.5, hazDistr = "log-normal",
                                  hazParams = c(res_paramLN$a*4, res_paramLN$lambda),
                                  Phi = 0, seed = 1, d = 0)

### Information about simulation
print(listAHSim_n500_p1000)
```

survMS *survMS Package*

Description

Package enables simulating data with different levels of complexity from survival models (Cox, AFT and AH models).

Author(s)

Mathilde Sautreuil <mathilde.sautreuil@gmail.com>

SurvTimesAFTshiftWeib *Simulation survival times from Shift AFT/Weibull model*

Description

Simulation survival times from Shift AFT/Weibull model

Usage

```
SurvTimesAFTshiftWeib(Z, beta, beta2, Y, pp, hazParams)
```

Arguments

Z	Matrix of covariates
beta	regression parameter
beta2	vector of regression parameter or distribution of regression parameter
Y	random uniform
pp	number of pertinent covariates
hazParams	distribution parameters of baseline hazard risk

Value

Ts Observed times

Examples

```
library(survMS)
```

SurvTimesAHLN *Simulation survival times from AH/Log-normal model*

Description

Simulation survival times from AH/Log-normal model

Usage

```
SurvTimesAHLN(Z, beta, Y, pp, hazParams)
```

Arguments

Z	Matrix of covariates
beta	regression parameter
Y	random uniform
pp	number of pertinent covariates
hazParams	distribution parameters of baseline hazard risk

Value

Ts Observed times

Examples

```
library(survMS)
```

Index