Package: parTimeROC (via r-universe)

October 30, 2026

Title Parametric Time-Dependent Receiver Operating Characteristic

Version 0.2.0

Maintainer Faiz Azhar <faiz.azhar241@gmail.com>

Description Producing the time-dependent receiver operating characteristic (ROC) curve through parametric approaches. Tools for generating random data, fitting, predicting and check goodness of fit are prepared. The methods are developed from the theoretical framework of proportional hazard model and copula functions. Using this package, users can now simulate parametric time-dependent ROC and run experiment to understand the behavior of the curve under different scenario.

License MIT + file LICENSE

URL https://github.com/FaizAzhar/parTimeROC

BugReports https://github.com/FaizAzhar/parTimeROC/issues

Encoding UTF-8

RoxygenNote 7.3.1

Biarch true

Depends R (>= 3.4.0)

Imports cubature, DescTools, flexsurv (>= 2.2.2), GofCens, graphics, Matrix, methods, moments, mvtnorm, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), rstantools (>= 2.4.0), sn (>= 2.1.1), stats, survival (>= 3.5.5), VineCopula (>= 2.4.5)

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

SystemRequirements GNU make

Suggests spelling, testthat (>= 3.0.0)

Config/testthat/edition 3

Language en-US

NeedsCompilation yes

Author Faiz Azhar [aut, cre, cph] (ORCID: <https://orcid.org/0009-0003-8017-2565>), Adina Najwa Kamarudin [cph, ths] (ORCID: <https://orcid.org/0000-0001-7820-3922>), Universiti Teknologi Malaysia, UTM [cph], Universiti Malaya, UM [cph]

Config/pak/sysreqs make

Repository https://cran.r-universe.dev

Date/Publication 2025-07-11 19:00:02 UTC

RemoteUrl https://github.com/cran/parTimeROC

RemoteRef HEAD

RemoteSha 8f3b2ce1602ce90d3ffc22f2b4099ed07868ff56

Contents

parTimeROC-package 2
get.copula 3
get.distributions 3
print.fitTROC 4
print.TimeROC 4
rate_change 5
rtimeroc 7
timeroc_auc 8
timeroc_fit 9
timeroc_gof 11
timeroc_obj 12
timeroc_predict 13

Index **16**

`parTimeROC-package` *The 'parTimeROC' package.*

Description

The goal of parTimeROC is to store methods and procedures needed to run the time-dependent ROC analysis parametrically. This package adopts two different theoretical framework to produce the ROC curve which are from the proportional hazard model and copula function. Currently, this package only able to run analysis for single covariate/biomarker with survival time. The future direction for this work is to be able to include analysis for multiple biomarkers with longitudinal measurements.

_PACKAGE

References

Stan Development Team (NA). RStan: the R interface to Stan. R package version 2.32.3. https://mc-stan.org

---

`get.copula` *get.copula*

Description

Storing list of copula.

Usage

```
get.copula
```

Format

An object of class `list` of length 5.

Value

A list of copula. #' @examples get.copula #> "gaussian" "clayton90" "gumbel90" "gumbel" "joe90"

`get.distributions` *get.distributions*

Description

Storing list of distributions for Biomarker, X and Time-to-event, T.

Usage

```
get.distributions
```

Format

An object of class `list` of length 9.

Value

A list of distributions.

Examples

```
get.distributions
#> "exponential" "weibull"     "gaussian"    "normal"      "lognormal"
#> "gompertz"    "skewnormal"
```

print.fitTROC *print.fitTROC*

Description

print.fitTROC

Usage

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

Arguments

x	A fitTROC object
...	Additional argument (not use)

Value

Summarize estimated parameters in console

Examples

```
test <- timeroc_obj(dist = 'lognormal-weibull-PH')
rr <- rtimeroc(test, n=300, params.x=c(meanlog=1,sdlog=0.8),
               params.t = c(shape=1.6,scale=1.2),
               params.ph = 1.1)
cc <- timeroc_fit(test, rr$x, rr$t, rr$event)
print(cc)
```

print.TimeROC *print.TimeROC*

Description

print.TimeROC

Usage

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

Arguments

x	A TimeROC object
...	Additional arguments (not use)

Value

Summarize model's info in console

Examples

```
test <- timeroc_obj(dist = 'lognormal-weibull-PH')
print(test)
```

`rate_change` *rate_change*

Description

Function to compute the rate change of the ROC with respect to dx and dt.

Usage

```
rate_change(
  obj,
  t,
  n = 3,
  type = "standard",
  params.x,
  params.t,
  copula,
  definition = "c/d",
  params.copula,
  params.ph,
  seed,
  cutoff = 100,
  newx
)
```

Arguments

`obj`	A 'fitROC' or 'TimeROC' object.
`t`	A numeric/vector specifying the time of interest.
`n`	Number of point on the ROC curve which will be used to check the rate of change.
`type`	A string specifying the type of analysis used. (Default: 'standard')

params.x	A named vector for the biomarker's parameter.
params.t	A named vector for the time-to-event parameter.
copula	A string indicating the type of copula to be used.
definition	A string indicitaing the definition of ROC to use.
params.copula	A numeric for copula's parameter.
params.ph	A numeric for proportional hazard model's parameter.
seed	A numeric to pass in set.seed.
cutoff	A numeric to generate total biomarker used.
newx	A numeric.

Value

A list of rate change dt, dx and the angle between these rate of change.

Examples

```
## Not run:
## Using 'fitROC' object
test <- timeroc_obj("normal-weibull-copula", copula = "gaussian")
rr <- rtimeroc(test, n=500,
               params.x = c(mean=5, sd=0.8),
               params.t = c(shape=1.6, scale=5),
              params.copula = -0.3)
cc <- timeroc_fit(test, x = rr$x, t = rr$t, event = rr$event)
jj <- rate_change(cc, t = c(1,2,10,11))

## Using 'TimeROC' object
test <- timeroc_obj("normal-weibull-PH",
params.x = c(mean=5, sd=0.8),
params.t = c(shape=1.6, scale=5),
params.ph = 1)
ee <- rate_change(test, t = c(.1,.2))

## End(Not run)
```

rtimeroc *rtimeroc*

Description

Function to generate bivariate data from PH or copula model.

Usage

```
rtimeroc(obj, n, censor.rate = 0, params.x, params.t, params.copula, params.ph)
```

Arguments

obj	An initialized 'TimeROC' object.
n	An integer of sample size.
censor.rate	An integer between 0 to 1 that is used for randomized censoring.
params.x	Vector of biomarker parameter.
params.t	Vector of time-to-event parameter.
params.copula	An integer for copula parameter.
params.ph	An integer for association parameter.

Value

A dataframe with 3 columns (x = biomarker value, t = observable time-to-event, status = censored/not censor (0 or 1))

Examples

```
## Copula model
test <- timeroc_obj(dist = 'gompertz-gompertz-copula', copula = "gumbel90")
set.seed(23456)
rr <- rtimeroc(obj = test, censor.rate = 0, n=500,
               params.t = c(shape=3,rate=1),
               params.x = c(shape=1,rate=2),
               params.copula=-5)
plot(t~x, rr)

## PH model
test <- timeroc_obj(dist = 'weibull-gompertz-PH')
set.seed(23456)
rr <- rtimeroc(obj = test, censor.rate = 0, n=100,
```

```
                params.t = c(shape=2, rate=1),
                params.x = c(shape=2, scale=1),
                params.ph=0.5)
plot(t~x, rr)
```

`timeroc_auc` *timeroc_auc*

Description

Function to compute the area under the ROC curve

Usage

```
timeroc_auc(obj)
```

Arguments

`obj`	A predictTROC object.

Value

A dataframe of the area under the ROC curve

Examples

```
## Not run:
test <- timeroc_obj('normal-weibull-copula', copula = 'clayton90')
print(test)

set.seed(23456)
rr <- rtimeroc(obj = test, censor.rate = 0.1, n=500,
                params.t = c(shape=1, scale=5),
                params.x = c(mean=5, sd=1),
                params.copula=-2)

cc <- timeroc_fit(x=rr$x, t=rr$t, event=rr$event, obj = test)

jj <- timeroc_predict(cc, t = quantile(rr$t, probs = c(0.25,0.5,0.75)))

timeroc_auc(jj)

## End(Not run)
```

timeroc_fit *timeroc_fit*

Description

Fit TimeROC using Maximum Likelihood Estimator.

Usage

```
timeroc_fit(
  obj,
  x,
  t,
  event,
  init.param.x = NULL,
  init.param.t = NULL,
  init.param.copula = NULL,
  init.param.ph = NULL,
  ci = 0.95,
  method = "mle",
  weights = NULL,
  breakpoints = NULL,
  init_bayes = NULL
)
```

Arguments

obj	An initialized 'TimeROC' object.
x	A numeric vector of single biomarker or covariate.
t	A numeric vector of time-to-event.
event	A numeric vector of event status (0=dead, 1=alive).
init.param.x	Vector of starting value for biomarker parameter.
init.param.t	Vector of starting value for time-to-event parameter.
init.param.copula	An integer of starting value for copula parameter.
init.param.ph	An integer of starting value for association parameter.

ci	An integer 0 to 1 for confidence level.
method	A string specifying method of estimation. (Default = 'mle')
weights	Weights to handle Inverse Probability Censoring Weights.
breakpoints	Break points to specify intervals for Piecewise Hazard Model.
init_bayes	Starting value when running the bayesian estimation.

Value

return a list of frequentist or bayesian estimator.

Examples

```
## Not run:
## fitting copula model
test <- timeroc_obj(dist = 'gompertz-gompertz-copula', copula = "gumbel90")
set.seed(23456)
rr <- rtimeroc(obj = test, censor.rate = 0, n=500,
               params.t = c(shape=3,rate=1),
               params.x = c(shape=1,rate=2),
               params.copula=-5) # name of parameter must follow standard

plot(t~x, rr)
start.t <- Sys.time()
cc <- timeroc_fit(rr$x, rr$t, rr$event, obj = test)
print(Sys.time()-start.t)

## fitting PH model
test <- timeroc_obj(dist = 'weibull-lognormal-PH')
set.seed(23456)
rr <- rtimeroc(obj = test, censor.rate = 0, n=100,
               params.t = c(meanlog=0, sdlog=1),
               params.x = c(shape=2, scale=1),
               params.ph=0.5) # name of parameter must follow standard
plot(t~x, rr)
start.t <- Sys.time()
cc <- timeroc_fit(rr$x, rr$t, rr$event, obj = test)
print(Sys.time()-start.t)

## End(Not run)
```

timeroc_gof *timeroc_gof*

Description

Function to compute goodness-of-fit for the Proportional Hazard (PH) or copula model. For PH model, the Cox-Snell residuals are computed and compared with Exponential(rate=1). For copula model, the Rosenblatt transformation is applied before performing independence testing. Kolmogorov-Smirnov test is performed to check the goodness-of-fit of the biomarker and time-to-event.

Usage

```
timeroc_gof(obj)
```

Arguments

obj	A 'fitTROC' object returned from fitting procedure.

Value

A list of test statistics and p-values. Automatically plot residuals for biomarker and time-to-event.

Examples

```
## Not run:
# Copula model
rt <- timeroc_obj("normal-weibull-copula",copula="clayton90")
set.seed(1)
rr <- rtimeroc(rt, n=50, censor.rate = 0,
               params.x = c(mean=5, sd=1),
               params.t = c(shape=1, scale=5),
               params.copula = -2.5)
plot(t~x, data=rr)
test <- timeroc_obj("normal-weibull-copula",copula="gumbel90")
jj <- timeroc_fit(test, rr$x, rr$t, rr$event)

cc <- timeroc_gof(jj)

test <- timeroc_obj("normal-weibull-copula",copula="clayton90")
jj <- timeroc_fit(test, rr$x, rr$t, rr$event)

cc <- timeroc_gof(jj)

## End(Not run)
```

timeroc_obj *timeroc_obj*

Description

Function to initialized time-dependent ROC object.

Usage

```
timeroc_obj(
  dist,
  params.x = NA,
  params.t = NA,
  copula = NA,
  params.copula = NA,
  params.ph = NA
)
```

Arguments

dist	A string emphasizing the distribution assumption for biomarker-time-model.
params.x	Vector of biomarker parameter.
params.t	Vector of time-to-event parameter.
copula	A string emphasizing on the copula to be used.
params.copula	An integer for copula parameter.
params.ph	An integer for association parameter.

Value

A 'TimeROC' object.

Examples

```
## Copula model
test <- timeroc_obj(dist = 'gompertz-gompertz-copula', copula = "gumbel90")

## PH model
test <- timeroc_obj(dist = 'weibull-gompertz-PH')
```

timeroc_predict *timeroc_predict*

Description

Predict time-dependent ROC from fitted model.

Usage

```
timeroc_predict(
  obj,
  t,
  newx,
  cutoff = 100,
  B = 1,
  type = "standard",
  params.x,
  params.t,
  copula,
  method = "mle",
  definition = "c/d",
  seed,
  params.copula,
  params.ph,
  ci = 0.95,
  h = -1e-04
)
```

Arguments

obj	A 'fitTROC' or 'TimeROC' object.
t	A numeric/vector specifying time point of interest. (Default: Time-to-event at 50th quantile points)
newx	A numeric/vector specifying biomarker of interest.
cutoff	A numeric specifying total cutoff point on ROC curve.
B	An integer specifying bootstrap iteration. If B > 1, will also return confidence interval.
type	A string indicate type of analysis to run. (Default = 'standard')
params.x	A named vector for biomarker's parameter.

params.t	A named vector for time-to-event's parameter.
copula	A string indicating the type of copula to be used.
method	A string specifying method of estimation. (Default = 'mle')
definition	A string indicating ROC definition to use. (Default = 'c/d')
seed	A numeric to pass in set.seed.
params.copula	An integer specifying the copula's parameter.
params.ph	An integer specifying the PH parameter.
ci	An integer 0 to 1 for confidence level.
h	An integer specifying small change of time (To compute density from S(t|x))

Value

A list of ROC dataframe for each time-to-event.

Examples

```
## Not run:
# PH model
test <- timeroc_obj('normal-weibull-PH')
set.seed(23456)
rr <- rtimeroc(obj = test, censor.rate = 0.1, n=500,
               params.t = c(shape=1, scale=5),
               params.x = c(mean=5, sd=1),
               params.ph=0.5)
cc <- timeroc_fit(x=rr$x, t=rr$t, event=rr$event, obj = test)
start.t <- Sys.time()
jj <- timeroc_predict(cc)
print(Sys.time()-start.t)


# Copula model
test <- timeroc_obj(dist = 'gompertz-gompertz-copula', copula='clayton90',
params.t = c(shape=3,rate=1),
params.x = c(shape=1,rate=2),
params.copula=-5)
set.seed(23456)
rr <- rtimeroc(obj = test, censor.rate = 0.2, n=500)
cc <- timeroc_fit(x=rr$x, t=rr$t, event=rr$event, obj = test)
start.t <- Sys.time()
```

```
jj <- timeroc_predict(cc)
print(Sys.time()-start.t)

## End(Not run)
```

Index

∗ **datasets**
    get.copula, 3
    get.distributions, 3

get.copula, 3
get.distributions, 3

parTimeROC *(*parTimeROC-package*)*, 2
parTimeROC-package, 2
print.fitTROC, 4
print.TimeROC, 4

rate_change, 5
rtimeroc, 7

timeroc_auc, 8
timeroc_fit, 9
timeroc_gof, 11
timeroc_obj, 12
timeroc_predict, 13