# Package ‘oceCens’

October 14, 2022

**Type** Package

**Title** Ordered Composite Endpoints with Censoring

**Version** 0.1.1

**Description** Estimates win ratio or Mann-Whitney parameter for two group comparisons using ordered composite endpoints with right censoring as described in Follmann, Fay, Hamasaki, and Evans (2020)<doi:10.1002/sim.7890>.

**License** GPL-3

**Depends** R (>= 3.5.0), survival

**Suggests** testthat (>= 3.0.0),knitr, rmarkdown

**Encoding** UTF-8

**LazyData** true

**VignetteBuilder** knitr

**RoxygenNote** 7.2.1

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Michael P. Fay [aut, cre],
Dean Follmann [aut]

**Maintainer** Michael P. Fay <mfay@niaid.nih.gov>

**Repository** CRAN

**Date/Publication** 2022-09-27 08:10:05 UTC

## R topics documented:

---

`coxph2WR` *Take coxph object and translate results to win ratios.*

---

### Description

Let `cout` a coxph object, then Using normal approximations and the output from the `cout$coefficients` and `cout$var`. If the cluster argument is used in the coxph call, then `cout$var` is the robust variance (see `coxph`.

### Usage

```
coxph2WR(coutput, conf.level = 0.95)
```

### Arguments

| | |
|---|---|
| `coutput` | a coxph object created by `coxph`. |
| `conf.level` | confidence level. |

### Details

The function takes a beta coefficient and returns the win ratio version: exp(-beta). Confidence intervals are calculated by exp(-beta -/+ qnorm(1-(1-conf.level)/2)*sqrt(coutput$var)). P-values are two-sided.

### Value

A vector or matrix with 4 elements (or columns) giving the win ratio, the lower and upper confidence limits, and the two-sided p-value.

### References

Follmann, D., Fay, M. P., Hamasaki, T., and Evans, S. (2020). Analysis of ordered composite endpoints. Statistics in Medicine, 39(5), 602-616.

### Examples

```
data(simScenario5)
xform<-oceFormat(data=simScenario5,oceTime=c("T1","T2","T3"),
   oceStatus=c("I1","I2","I3"),
   group="Z",outputDataFrame=TRUE)
# perform cox regression using time varying treatment efects, IZ1,IZ2, IZ3
# associated with the 3 prioritized endpoints
```

```
cout<- coxph(Surv(START, STOP, status) ~ IZ1+IZ2+IZ3, data=xform$data)
coxph2WR(cout)
```

---

oceCens *oceCens: A package for ordered composite endpoints with censoring.*

---

## Description

An ordered composite endpoint combines several time-to-event endpoints into one score. The package compares two groups with two parameters, the win ratio, P[Y1>Y0]/P[Y0>Y1], and the Mann-Whitney parameter, P[Y1>Y0]+(1/2)P[Y1=Y0], where Y1 and Y0 are the oce scores in the two groups. The main function is `oceTest`, which calls many of the other functions and has several different methods for estimation. Statistical details are in Follmann, et al 2020.

## References

Follmann, D., Fay, M. P., Hamasaki, T., and Evans, S. (2020). Analysis of ordered composite endpoints. Statistics in Medicine, 39(5), 602-616.

---

oceCoxph *Estimate win ratio or Mann-Whitney parameter using Cox Proportional Hazards*

---

## Description

Usually called from within `oceTest`, but useful for getting `coxph` output details or customizing graphics. Estimation done using coxph (partial likelihood methods).

## Usage

```
oceCoxph(oceData)
```

## Arguments

`oceData` output from `oceFormat`

## Value

An `oceCoxph` object, which is a list with the following elements (where Yg=ordered composite endpoint score for group=g):

**oceNames** long names for each of the ordered endpoints

**TAU** maximum of all the time-to-event variables (even censored ones)

**coxOutput** output from coxph function

**int01** estimate of P[Y0>Y1]

**int10** estimate of P[Y1>Y0]

**WR** win ratio, estimate of P[Y1>Y0]/P[Y0>Y1]

**MW** desirability of outcome ranking, estimate of P[Y1>Y0]+(1/2)P[Y1=Y0]

## See Also

For an example using plotting see plot.oceCoxph. For Cox regression with other covariates, see vignette("Using oceCens",package="oceCens").

---

oceFormat *Format ordered composite endpoint.*

---

## Description

Usually called from within oceTest. Input data.frame with one row for each individual and columns for k time-to-event outcomes, k status variables, and a group variable. Format output so that each individual has several rows representing different intervals at risk. Returns a list with elements used for later calculations.

## Usage

```
oceFormat(
  data,
  oceTime,
  oceStatus,
  group,
  id = NULL,
  oceNames = NULL,
  outputDataFrame = FALSE
)
```

## Arguments

| | |
|---|---|
| data | data.frame name, must have variables with names listed in oceTime, oceStatus, group |
| oceTime | character vector with ordered (primary is first) names of different time-to-event variables. |
| oceStatus | character vector with ordered names of status (0=censored, 1=event) variables. |
| group | name of group variable. |
| id | name of ID variable, NULL creates integer IDs. |
| oceNames | long names of ordered endpoints, NULL uses oceTime. |
| outputDataFrame | logical, output a data.frame in the list, defaults to FALSE for speed in the bootstrap. |

### Value

A list with the following elements:

**timeMatrix** n by k matrix with input values for k time-to-event values for each of n individuals

**statusMatrix** n by k matrix of status values

**Z** n vector of group variable with elements either 0 or 1

**oceNames** k vector of long oceNames (for plotting labels)

**id** m vector of individual ids, one element for each interval, so m>n

**group** m vector of group values, either 0 or 1

**status** m vector of status for each interval

**START** m vector, START of interval

**STOP** m vector, end of interval

**TAU** maximum of the time-to-event outcomes

**IZMatrix** m by k matrix, with jth column an indicator of representing ordering score 'time' for the jth endpoint

**data** a data.frame output if outputDataFrame=TRUE, with variables: id, group, status, START, STOP, IZ1,...,IZk (columns of IZMatrix)

### Examples

```
d.temp<-data.frame(T1=c(1,4,3,6),s1=c(0,0,1,0),T2=c(4,1,5,3),
  s2=c(1,0,0,1),z=c(0,0,1,1))
d.temp
x<-oceFormat(data=d.temp,oceTime=c("T1","T2"),oceStatus=c("s1","s2"),
  group="z",outputDataFrame=TRUE)
# put time to second event starting at TAU
x$TAU
x$data
```

---

`oceNPMLE` *Estimate win ratio or Mann-Whitney parameter using NPMLE*

---

### Description

Estimation done using NPMLE (nonparametric maximum likelihood estimators of survival).

### Usage

```
oceNPMLE(oceData)
```

### Arguments

| | |
|---|---|
| `oceData` | output from `oceFormat` |

### Value

An object of class 'oceNPMLE', which is a list with the following elements (where Yg=ordered composite endpoint score for group=g):

**oceNames** long names for each of the ordered endpoints

**TAU** maximum of all the time-to-event variables (even censored ones)

**KM0** survfit output for group=0 subset

**KM1** survfit output for group=1 subset

**WR** win ratio, estimate of P[Y1>Y0]/P[Y0>Y1]

**MW** desirability of outcome ranking, estimate of P[Y1>Y0]+(1/2)P[Y1=Y0]

### See Also

See `plot.oceNPMLE` for an example with plotting.

---

## `oceSimple` *Estimate win ratio or Mann-Whitney parameter using Simple Method*

---

### Description

Usually called from within `oceTest`. Estimation done using simple method and output from `oceCoxph`.

### Usage

```
oceSimple(oceData, oceCoxOutput = NULL)
```

### Arguments

| | |
|---|---|
| `oceData` | output from `oceFormat`. |
| `oceCoxOutput` | output from `oceCoxph`, if NULL recalculates using `oceData` and `oceCoxph`. |

### Value

A list with the following elements (where Yg=ordered composite endpoint score for group=g):

**int01** estimate of P[Y0>Y1] (calculated from `oceCoxph`)

**int10** estimate of P[Y1>Y0] (calculated from `oceCoxph`)

**WR** win ratio, estimate of P[Y1>Y0]/P[Y0>Y1]

**MW** desirability of outcome ranking, estimate of P[Y1>Y0]+(1/2)P[Y1=Y0] oceTest *Tests for ordered composite endpoints with censoring.*

## Description

An ordered composite endpoint (oce) is a way of ranking responses by ordering several types of responses by order of importance. Rank by the most important response, then break ties with the next most important, and so on. The tests here are based on two sample tests. Let Y0 and Y1 be the oce score in the control arm and treatment arm, respectively. Then here we estimate both the win ratio (WR), P[Y1>Y0]/P[Y0>Y1], or the Mann-Whitney parameter, P[Y1>Y0] + (1/2) Pr[Y1=Y0]. Different methods are used to estimate those parameters, and inferences are done by bootstrap percentile methods.

## Usage

```
oceTest(
  data,
  oceTime,
  oceStatus,
  group,
  id = NULL,
  oceNames = NULL,
  method = c("all", "npmle", "coxph", "simple"),
  ciMethod = c("WLW", "bootstrap"),
  conf.int = FALSE,
  conf.level = 0.95,
  nBoot = 2000,
  plot = FALSE,
  ...
)
```

## Arguments

| | |
|---|---|
| `data` | data.frame name, must have variables with names listed in `oceTime`, `oceStatus`, `group` |
| `oceTime` | character vector with ordered (primary is first) names of different time-to-event variables. |
| `oceStatus` | character vector with ordered names of status (0=censored, 1=event) variables. |
| `group` | name of group variable. |
| `id` | name of ID variable, NULL creates integer IDs. |
| `oceNames` | long names of ordered endpoints, NULL uses `oceTime`. |
| `method` | Estimation method, one of 'all', 'npmle', 'coxph' or 'simple'. Default is 'all' which calculates all of the three methods. See details. |
| `ciMethod` | confidence interval method, default is 'bootstrap' |
| `conf.int` | Logical, should confidence intervals be calculated. |

| | |
|---|---|
| `conf.level` | confidence level. |
| `nBoot` | number of bootstrap replicates (ignored if conf.int=FALSE). |
| `plot` | logical, plot oce score by group as survival functions (NPMLE version, except if method='coxph'). For more control over those plots see either `plot.oceNPMLE` or `plot.oceCoxph`. |
| `...` | holder space for future arguments. |

## Details

This idea is to stack the time to first event for the k different types of events. So if TAU is the maximum time that any individual is in the study, then the primary type of event has scores that fall into (0,TAU], the secondary type has scores that fall into (TAU,2*TAU], and so on. Then we rank by the primary type (e.g., death), but if there are many ties in the primary type (e.g., many that did not die during the study), then we break ties by the secondary type of event, and so on.

The difficulty is when there is censoring in time, because that imposes interval censoring on the score scale. This can be handled with interval censoring methods (although in a non-standard way). The 'npmle' method calculates a nonparametric maximum likelihood estimate of the 'survival' distribution of the ordering score for each arm, then gets the estimates by numeric integration. The 'coxph' method uses an interval censored proportional hazards model treating the oce scores as time using `coxph` from the survival R package. The 'simple' method uses part of the 'coxph' method together with a more simple estimator. Each method produces a win ratio (P[Y1>Y0]/P[Y0>Y1]) and a Mann-Whitney (P[Y1>Y0] + (1/2) Pr[Y1=Y0]) estimate. Details are given in Follmann, et al (2020).

When `ciMethod="bootstrap"` inferences are done by nonparametric bootstrap percentile method (see `percci`) in order to account for the correlation among the different types of responses. When `ciMethod="WLW"` and `method="coxph"`, then the win ratio is calculated by the Cox model with the standard errors of the log(HR) or log(WR) calculated by the robust sandwich method suggested by Wei, Lin, and Weissfeld (1989). P-values are all two-sided and test the null hypothesis of no difference between the arms (for the win ratio, the null value is 1, while for the MW the null value is 0).

For access to the `coxph` output see `oceCoxph`, or for the NPMLE output see `oceNPMLE`.

## Value

If `conf.int=FALSE` then a vector of estimates determined by `method` results. If `conf.int=TRUE` then a matrix is returned with a row for each estimate, and 4 columns for the Estimate, lower confidence limit, upper confidence limit, and two-sided p-value.

## References

Follmann, D., Fay, M. P., Hamasaki, T., and Evans, S. (2020). Analysis of ordered composite endpoints. Statistics in Medicine, 39(5), 602-616.

Wei, L. J., Lin, D. Y., & Weissfeld, L. (1989). Regression analysis of multivariate incomplete failure time data by modeling marginal distributions. Journal of the American statistical association, 84(408), 1065-1073.

## Examples

```
data(simScenario5)
oceTest(data=simScenario5, oceTime=c("T1","T2","T3"),
 oceStatus=c("I1","I2","I3"), group=c("Z"), id = "PATID",
 oceNames = c("Death","Stroke/MI","Bleed"), method=c("all"))
```

---

# percci *Percentile Bootstrap Two-sided Confidence Intervals and p-values*

---

## Description

Input vector of bootstrap replicates and get either the two-sided percentile confidence interval or the compatible two-sided p-value.

## Usage

```
percci(Ti, conf.level = 0.95)

percpval(Ti, theta0 = 0)
```

## Arguments

| | |
|---|---|
| `Ti` | A numeric vector of bootstrap replicates of an estimate. |
| `conf.level` | Confidence level. |
| `theta0` | Null hypothesis value of estimand. |

## Details

Simple functions, where `percci` gives two-sided confidence intevals and `percpval` gives two-sided p-values.

We get a two-sided p-value by inverting the percentile Bootstrap confidence interval. This is not straightforward if there are not enough bootstrap samples and/or if the minimum and maximum of the replicates do not cover the null value. If there are B bootstrap resamples, then the interval from the minimum to the maximum has confidence level =1- 2/(B+1). We can see this because the percentile interval (see Efron and Tibshirani, 1993, p. 160 bottom) is T[k], T[B+1-k] where k=floor( (B+1)*(1-conf.level)/2), where T is an ordered vector of B test statistics calculated from B bootstrap replicates (T=Ti[order(Ti)]). Therefore, if conf.level > 1 - 2/(B+1) then we cannot get a percentile interval, so if the min and max of T do not surround theta0, then a two-sided p-value can be stated to be p<= 2/(B+1). If the p-value is 2/(B+1), then it is the lowest possible for that B, and increasing B may produce a lower p-value.

## Value

`percci` returns only a two-sided confidence interval and `percpval` returns only a two-sided p-value.

### Functions

- `percpval()`: Bootstrap percentile p-values

### References

Efron, B and Tibshirani, RJ (1993) An Introduction to the Bootstrap. Chapman and Hall.

### Examples

```
set.seed(123)
y<- rnorm(100)+0.1
nB<- 1e5
Tstat<- rep(NA,nB)
for (i in 1:nB){
  Tstat[i]<-mean( sample(y,replace=TRUE) )
 }
 # two-sided bootstrap percentile p-value
 # that mean is different from 0
 percpval(Tstat,theta0=0)
 # 95% percentile interval
 percci(Tstat)
 # compare to t-test
 t.test(y)

 # to show that the functions are close to compatiable
 # set confidence level to 1-pvalue
 pval<-percpval(Tstat,theta0=0)
 confLevel<- 1-pval
 pval
 # then lower limit should be close to 0
 percci(Tstat, conf.level=confLevel)
```

---

## plot.oceCoxph *Plot* oceNPMLE *object.*

---

### Description

Plot oceNPMLE object.

### Usage

```
## S3 method for class 'oceCoxph'
plot(
  x,
  linesonly = FALSE,
  xlab = "Ordering Score",
  ylab = "Proportion with a larger ordering score",
```

```
  col = c("red", "blue"),
  ...
)
```

## Arguments

| | |
|---|---|
| `x` | oceCoxph object (see `oceCoxph`). |
| `linesonly` | logical, add lines to an existing plot? |
| `xlab` | x label |
| `ylab` | y label |
| `col` | color vector, col[1] for group=0 and col[2] for group=1. |
| `...` | Extra arguments (e.g., lwd=3) added to both `lines` functions. |

## Value

The function invisibly (see `invisible`) returns a list with 4 elements: (time0, surv0, time1, and surv1)

## See Also

Example in `plot.oceNPMLE` shows adding lines from the coxph output to an existing plot.

## Examples

```
# need to first run oceFormat and oceCoxph
data(simScenario5)
dataFormt<-oceFormat(data=simScenario5, oceTime=c("T1","T2","T3"),
   oceStatus=c("I1","I2","I3"), group=c("Z"),
   oceNames = c("Death","Stroke/MI","Bleed"))
coxOutput<- oceCoxph(dataFormt)
plot(coxOutput, xlab="Custom x label")
```

---

`plot.oceNPMLE` *Plot* `oceNPMLE` *object.*

---

## Description

Plot `oceNPMLE` object.

### Usage

```
## S3 method for class 'oceNPMLE'
plot(
  x,
  xlab = "Ordering Score",
  ylab = "Proportion with a larger ordering score",
  ylim = c(0, 1),
  col = c("red", "blue"),
  mark.time = TRUE,
  ...
)
```

### Arguments

| | |
|---|---|
| x | oceNPMLE object (see `oceNPMLE`). |
| xlab | x label |
| ylab | y label |
| ylim | limits for the y axis, defaults to c(0,1) |
| col | color vector, col[1] for group=0 and col[2] for group=1. |
| mark.time | logical, should censored values be plotted? |
| ... | Extra arguments (e.g., lwd=2) added to `lines` functions. |

### Value

No return value, called for side effects.

### Examples

```
data(simScenario5)
dataFormt<-oceFormat(data=simScenario5, oceTime=c("T1","T2","T3"),
   oceStatus=c("I1","I2","I3"), group=c("Z"),
   oceNames = c("Death","Stroke/MI","Bleed"))
npmleOutput<- oceNPMLE(dataFormt)
plot(npmleOutput, xlab="Custom x label", mark.time=FALSE, lwd=2)
# can add lines from coxph output
coxOutput<- oceCoxph(dataFormt)
plot(coxOutput,linesonly=TRUE, col=c("orange","purple"),lwd=2)
legend("bottomleft",
   legend=c("grp=0, NPMLE","grp=1, NPMLE","grp=0, coxph","grp=1, coxph"),
   col=c("red","blue","orange","purple"),lty=c(1,1,1,1),lwd=2)
```

`simScenario5` *Simulated data from simulation scenario 5*

## Description

Simulated data in the supplement to Follmann, et al (2020). T1,T2, and T3 are the time to the first event for three different types of events (e.g., Death, Stroke/MI, Bleed). I1,I2, and I3 are the associated status variables (0=censored, 1=event). Other variables are PATID (patient ID) and Z (0=control arm, 1=treatment arm).

## Usage

```
data(simScenario5)
```

## Format

A data frame with 400 obs. and 8 variables.

## References

Follmann, D., Fay, M. P., Hamasaki, T., and Evans, S. (2020). Analysis of ordered composite endpoints. Statistics in Medicine, 39(5), 602-616.

# Index