

# Package ‘AHR’

December 5, 2018

**Type** Package

**Title** Estimation and Testing of Average Hazard Ratios

**Version** 1.5.0

**Date** 2018-12-04

**Author** Matthias Brueckner [aut, cre]

**Maintainer** Matthias Brueckner <matthias.brueckner@posteo.de>

**Description** Methods for estimation of multivariate average hazard ratios as defined by Kalbfleisch and Prentice. The underlying survival functions of the event of interest in each group can be estimated using either the (weighted) Kaplan-Meier estimator or the Aalen-Johansen estimator for the transition probabilities in Markov multi-state models. Right-censored and left-truncated data is supported. Moreover, the difference in restricted mean survival can be estimated.

**License** GPL-3 | file LICENSE

**Depends** survival

**Imports** stats, MASS, etm, Rcpp

**LinkingTo** Rcpp, RcppArmadillo

**Suggests** testthat

**RoxygenNote** 6.1.1

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2018-12-05 05:10:03 UTC

## R topics documented:

AHR . . . . .	2
ahrAJ . . . . .	3
ahrKM . . . . .	5
ahrUser . . . . .	6
ahrWKM . . . . .	8
aj . . . . .	9

avgHR . . . . .	10
fastkm . . . . .	11
print.ahr . . . . .	12
print.rmd . . . . .	13
print.survQuantile . . . . .	13
rmeanDiff . . . . .	14
rmeanDiff.ahr . . . . .	15
wkm . . . . .	16
wkmCompareQuantiles . . . . .	17
wkmQuantile . . . . .	18
<b>Index</b>	<b>19</b>

AHR

AHR

## Description

This package provides methods for estimation of multivariate average hazard ratios as defined by Kalbfleisch and Prentice. The underlying survival functions of the event of interest in each group can be estimated using either the (weighted) Kaplan-Meier estimator or the Aalen-Johansen estimator for the transition probabilities in Markov multi-state models. Right-censored and left-truncated data is supported. Moreover, the difference in restricted mean survival can be estimated. Currently variance estimation for the average hazard ratio based on the Aalen-Johansen estimator is only supported for competing risks models, i.e. for estimation of the average sub-distribution hazard ratio (Average cause-specific hazard ratios can be estimated by using the Kaplan-Meier estimator with competing risks data).

## Details

Furthermore estimation of quantiles, ratios and differences of quantiles and corresponding p-values and confidence intervals of survival times based on the (weighted) Kaplan-Meier estimator and the Aalen-Johansen estimator is also supported.

## Author(s)

Matthias Brueckner <matthias.brueckner@posteo.de>

## References

- J.-D. Kalbfleisch and R.-L. Prentice. Estimation of the average hazard ratio. *Biometrika*, 68(1):105–112, Apr. 1981.
- S.-Murray and A.-A. Tsiatis. Nonparametric survival estimation using prognostic longitudinal covariates. *Biometrics*, 52(1):137–151, Mar. 1996.
- C.-A. Struthers and J.-D. Kalbfleisch. Misspecified proportional hazard models. *Biometrika*, 73(2):363–369, Aug. 1986.

**Examples**

```

T <- c(rexp(100, 1), rexp(100, 2))
C <- c(rexp(100, 1), rexp(100, 2))
Y <- pmin(T, C)
D <- T <= C
Z <- rep(c(0,1), c(100, 100))

## uses Kaplan-Meier estimator by default
fit <- avgHR(2, data.frame(Y=Y, D=D, Z=Z), formula=Surv(Y, D) ~ Z)
fit

## same as
## Not run: fit <- avgWKM(2, data.frame(Y=Y, D=D, Z=Z), formula=Surv(Y, D) ~ Z)

## use bootstrap to estimate covariance matrix
## Not run: fit <- avgWKM(2, data.frame(Y=Y, D=D, Z=Z), formula=Surv(Y, D) ~ Z, cov=FALSE,
                    bootstrap=10000)

## End(Not run)

## calculate restricted mean difference
rdm <- rmeanDiff.ahr(fit)
rdm

## ventilation status in intensive care unit patients dataset from etm package
library(etm)
data(sir.cont)
df <- sir.cont
df$Trt <- factor(rep(0, nrow(df)), levels=c(0, 1))
ids <- unique(df$id)
df$Trt[df$id %in% sample(ids, floor(length(ids)/2), FALSE)] <- 1

# transition matrix
tra <- matrix(FALSE, nrow=3, ncol=3)
tra[1, 2:3] <- TRUE
tra[2, c(1, 3)] <- TRUE

# NOTE: variance estimation not yet supported for Aalen-Johansen based avg. HR
sc.fit <- avgHR(2, method="aj", data=df, target="0 2", states=c("0", "1", "2"), transitions=tra,
               censoring="cens", cov=FALSE)
sc.fit

```

---

ahrAJ

---

*ahrAJ*


---

**Description**

Estimate average hazard ratios from  $k$  independent samples based on the Aalen-Johansen estimator of the empirical transition probabilities (NOTE: variance estimation not yet implemented)

**Usage**

```
ahrAJ(L, target, states, transitions, censoring, data, null.theta = NULL,
      contrast = NULL, multi.test = FALSE, cov = FALSE, bootstrap = 0)
```

**Arguments**

L	time-limit specifying time-interval [0,L] over which average hazard ratios will be calculated
target	string specifying the target transition, for which the Aalen-Johansen estimator is to be calculated
states	list of state names
transitions	matrix of possible transitions
censoring	name of censoring 'state'
data	data frame containing variables id, time, from, to (see <a href="#">etm</a> ) and Trt (factor giving treatment groups)
null.theta	vector specifying the null hypothesis for the average hazard ratios
contrast	vector of contrasts to test $H_0: \text{contrast} * (\text{theta} - \text{null.theta}) = 0$
multi.test	calculate multivariate test statistic if TRUE
cov	if TRUE calculate covariance matrix estimator (direct)
bootstrap	number of bootstrap samples to draw for variance estimation (default: 0 = no bootstrap, direct variance estimation). This parameter is ignored if cov=TRUE

**Value**

An object of class "ahr"

**References**

J.-D. Kalbfleisch and R.-L. Prentice. Estimation of the average hazard ratio. *Biometrika*, 68(1):105–112, Apr. 1981.

**See Also**

[aj](#)

**Examples**

```
## competing risks
Trt <- factor(rep(c(0,1), c(100, 100)))
T <- c(rexp(100, 1), rexp(100, 2))
C <- c(rexp(100, 1), rexp(100, 2))
r <- c(rbinom(100, 2, 0.5), rbinom(100, 2, 0.4))
r[(r == 0) | (T > C)] <- "cens"
data <- data.frame(id=1:200, time=pmin(T,C), from=rep(0, 200), to=r, Trt=Trt)
tra <- matrix(FALSE, nrow=3, ncol=3)
tra[1, 2:3] <- TRUE
# estimate average subdistribution hazard ratio up to L=2 for event type 1
```

```
fit <- ahrAJ(2, target="0 1", states=c("0", "1", "2"), transitions=tra, censoring="cens",
            data=data, cov=TRUE)
fit
```

ahrKM

*ahrKM***Description**

Estimate average hazard ratios from k independent samples based on the Kaplan-Meier estimator

**Usage**

```
ahrKM(L, formula, data, null.theta = NULL, contrast = NULL,
      multi.test = FALSE, cov = TRUE, bootstrap = 0,
      left.limit = FALSE)
```

**Arguments**

L	time-limit specifying time-interval [0,L] over which average hazard ratios will be calculated
formula	an object of class "formula" specifying the conditional survival model
data	data frame containing the variables in formula
null.theta	vector specifying the null hypothesis for the average hazard ratios (H <sub>0</sub> : theta = null.theta)
contrast	vector of contrasts to test H <sub>0</sub> : contrast * (theta - null.theta) = 0
multi.test	calculate multivariate test statistic if TRUE
cov	if TRUE calculate covariance matrix estimator (direct)
bootstrap	if > 0 then use bootstrap to estimate covariance matrix (ignore if cov is TRUE)
left.limit	if TRUE use left-continuous interpolation of WKM estimates

**Value**

An object of class "ahr"

**References**

J.-D. Kalbfleisch and R.-L. Prentice. Estimation of the average hazard ratio. *Biometrika*, 68(1):105–112, Apr. 1981.

**See Also**

[survfit](#)

**Examples**

```

T <- c(rexp(100, 1), rexp(100, 2))
C <- c(rexp(100, 1), rexp(100, 2))
Y <- pmin(T, C)
D <- T <= C
Z <- rep(c(0,1), c(100, 100)) # treatment indicator
fit <- ahrKM(2, Surv(Y, D) ~ Z, data.frame(Y=Y, D=D, Z=Z))
fit

## the same as above, but estimate covariance matrix using bootstrap
## Not run: fitBS <- ahrKM(2, Surv(Y, D) ~ Z, data.frame(Y=Y, D=D, Z=Z), cov=FALSE,
                    bootstrap=1000)

## End(Not run)

```

ahrUser

*ahrUser***Description**

Estimate average hazard ratios from k independent samples based on user-supplied survival function estimator

**Usage**

```

ahrUser(L, formula, data, null.theta = NULL, contrast = NULL,
        multi.test = FALSE, cov = FALSE, bootstrap = 0,
        user.survfit = wkm, user.param = list(alpha = 1, var = FALSE, cov =
        FALSE, left.limit = FALSE, rr.subset = rep(TRUE, nrow(data))))

```

**Arguments**

L	time-limit specifying time-interval [0,L] over which average hazard ratios will be calculated
formula	an object of class "formula" specifying the conditional survival model
data	data frame containing the variables in formula
null.theta	vector specifying the null hypothesis for the average hazard ratios (H <sub>0</sub> : theta = null.theta)
contrast	vector of contrasts to test H <sub>0</sub> : contrast * (theta - null.theta) = 0
multi.test	calculate multivariate test statistic if TRUE
cov	if TRUE calculate covariance matrix estimator (direct)
bootstrap	if > 0 then use bootstrap to estimate covariance matrix (ignore if cov is TRUE)
user.survfit	user defined function taking vector of times, data.frame and list of parameters returning survival function estimate
user.param	list of parameters passed to function user.survfit

**Details**

user.survfit must return logV (if user supplied survival function estimator has independent increments property) or logCOV

**Value**

An object of class "ahr"

**References**

J.-D. Kalbfleisch and R.-L. Prentice. Estimation of the average hazard ratio. *Biometrika*, 68(1):105–112, Apr. 1981.

**See Also**

[wkm](#)

**Examples**

```
## User supplied survival function estimator (should be exactly the same as
T <- c(rexp(100, 1), rexp(100, 2))
C <- c(rexp(100, 1), rexp(100, 2))
time <- pmin(T, C)
status <- T <= C
trt <- rep(c(0,1), c(100, 100)) # treatment indicator

sfit <- function(times, data, param) {
  fit <- survfit(Surv(Y, D) ~ 1, data=data)
  f <- approxfun(fit$time, fit$surv, method="constant", f=0, yleft=1, rule=2)
  fv <- approxfun(fit$time, fit$std.err^2, method="constant", f=0, yleft=1, rule=2)

  S <- f(times)
  logV <- fv(times) * nrow(data)
  V <- S^2 * logV

  list(times=times, S=S, V=V, logV=logV)
}
fit1 <- ahrUser(2, Surv(time, status) ~ trt,
              data.frame(time=time, status=status, trt=trt), user.survfit=sfit, user.param=list())
fit1
fit2 <- ahrKM(2, Surv(time, status) ~ trt, data.frame(time=time, status=status, trt=trt), cov=FALSE)
fit2
```

ahrWKM

*ahrWKM***Description**

Estimate average hazard ratios from k independent samples based on the weighted Kaplan-Meier (WKM) estimator

**Usage**

```
ahrWKM(L, formula, data, null.theta = NULL, contrast = NULL,
       multi.test = FALSE, cov = TRUE, bootstrap = 0, alpha = 1,
       left.limit = FALSE, rr.subset = rep(TRUE, nrow(data)))
```

**Arguments**

L	time-limit specifying time-interval [0,L] over which average hazard ratios will be calculated
formula	an object of class "formula" specifying the conditional survival model
data	data frame containing the variables in formula
null.theta	vector specifying the null hypothesis for the average hazard ratios (H <sub>0</sub> : theta = null.theta)
contrast	vector of contrasts to test H <sub>0</sub> : contrast * (theta - null.theta) = 0
multi.test	calculate multivariate test statistic if TRUE
cov	if TRUE calculate covariance matrix estimator (direct)
bootstrap	if > 0 then use bootstrap to estimate covariance matrix (ignore if cov is TRUE)
alpha	exponent of the weight function
left.limit	if TRUE use left-continuous interpolation of WKM estimates instead of right-continuous interpolation
rr.subset	logical vector defining subset of observations to use for response rate estimation (default: use all observations)

**Value**

An object of class "ahr"

**References**

J.-D. Kalbfleisch and R.-L. Prentice. Estimation of the average hazard ratio. *Biometrika*, 68(1):105–112, Apr. 1981.

**See Also**

[wkm](#)



**Examples**

```

T <- c(rexp(100, 1), rexp(100, 2))
C <- c(rexp(100, 1), rexp(100, 2))
Y <- pmin(T, C)
D <- T <= C
Z <- rep(c(0,1), c(100, 100)) # treatment indicator
fit <- ahrWKM(2, Surv(Y, D) ~ Z, data.frame(Y=Y, D=D, Z=Z))
fit

## the same as above, but estimate covariance matrix using bootstrap
## Not run: fitBS <- ahrWKM(2, Surv(Y, D) ~ Z, data.frame(Y=Y, D=D, Z=Z), cov=FALSE,
                    bootstrap=1000)

## End(Not run)

```

aj

aj

**Description**

Aalen-Johansen estimator (empirical transition matrix)

**Usage**

```
aj(times, data, param)
```

**Arguments**

times	a vector of evaluation times
data	data frame (see <a href="#">etm</a> function documentation)
param	list of parameters (target, states, transitions, censoring, s, t, cov) (see <a href="#">etm</a> documentation)

**Details**

Wrapper for 'etm' function from the 'etm' package to be used with [ahrAJ](#)

For a description of the parameters in the list param see the documentation of the [etm](#) function in package etm.

**Value**

a list containing

times	the argument times passed to the function
S	vector of 1 minus transition probabilities at times (one for each element of times)
V	vector of variances at 'times' (only if param\$cov is TRUE)

logCOV            matrix containing estimated values of the log-covariance function evaluated for all pairs of elements of the vector times (only if param\$cov is TRUE and model is a competing risks model)

### See Also

[etm](#)

### Examples

```
## competing risks
T <- rexp(100)
C <- rexp(100)
r <- rbinom(100, 2, 0.5)
r[(r == 0) | (T > C)] <- "cens"
data <- data.frame(id=1:100, time=pmin(T,C), from=rep(0, 100), to=r)
data <- data[order(data$time),]
tra <- matrix(FALSE, nrow=3, ncol=3)
tra[1, 2:3] <- TRUE
# estimate cumulative incidence function for event type 1
fit <- aj(sort(data$time), data, list(target="0 1", states=c("0", "1", "2"), transitions=tra,
  censoring="cens", s=0, t="last", covariance=TRUE))
```

---

avgHR

*avgHR*

---

### Description

Estimate average hazard ratios from k independent samples

### Usage

```
avgHR(L, data, method = "km", ...)
```

### Arguments

L	time-limit specifying time-interval [0,L] on which average hazard ratios will be calculated
data	data frame (see data argument to <a href="#">ahrWKM</a> (if method == "wkm"    "km") or <a href="#">ahrAJ</a> (if method == "aj"))
method	method used for estimating survival functions (default: Kaplan-Meier estimator)
...	additional arguments passed to <a href="#">ahrWKM</a> or <a href="#">ahrAJ</a>

### Details

This function is a simple wrapper for [ahrWKM](#) and [ahrAJ](#).

**Value**

An object of class `"ahr"`

**References**

J.-D. Kalbfleisch and R.-L. Prentice. Estimation of the average hazard ratio. *Biometrika*, 68(1):105–112, Apr. 1981.

**See Also**

[ahrWKM](#), [ahrAJ](#)

**Examples**

```
T <- c(rexp(100, 1), rexp(100, 2))
C <- c(rexp(100, 1), rexp(100, 2))
Y <- pmin(T, C)
D <- T <= C
Z <- rep(c(0,1), c(100, 100))
fit <- avgHR(2, data.frame(Y=Y, D=D, Z=Z), formula=Surv(Y, D) ~ Z)
```

---

fastkm

*fastkm*

---

**Description**

Fast Kaplan-Meier estimator

**Usage**

```
fastkm(time, status, ltrunc = rep.int(0, length(time)),
       left.limit = FALSE, eval = time)
```

**Arguments**

<code>time</code>	vector of right-censored survival times
<code>status</code>	censoring indicator for each element of <code>time</code> (0 = right-censored, 1 = event)
<code>ltrunc</code>	vector of left-truncation times
<code>left.limit</code>	indicates whether estimated survival function is left continuous
<code>eval</code>	points at which the estimated survival function should be evaluated

**Details**

This function calculates the Kaplan-Meier estimator for right-censored survival data, at arbitrary time points. It can handle left-truncated and/or right-censored data with ties. Avoids the overhead of the `survfit` or `prodlim` functions by stripping away most of the features not needed here.

**Value**

A list containing the vectors `time`, `surv` and `variance`, and `n.atrisk`

**See Also**

[survfit](#) and [prodlim](#)

**Examples**

```
T <- rexp(100)
C <- rexp(100)
Y <- pmin(T, C)
D <- T <= C
sort(fastkm(Y, D)$surv, decreasing=TRUE)
# should be exactly the same as
fit <- survfit(Surv(Y, D) ~ 1)
f <- approxfun(fit$time, fit$surv, f=0, rule=2, yleft=1)
f(fit$time)
```

---

`print.ahr`

*print.ahr*

---

**Description**

Print ahr object

**Usage**

```
## S3 method for class 'ahr'
print(x, digits = 3, ...)
```

**Arguments**

<code>x</code>	an object of class <code>"ahr"</code> .
<code>digits</code>	minimal number of significant digits.
<code>...</code>	further arguments passed to or from other methods.

---

`print.rmd`                      *print.rmd*

---

**Description**

Print rmd object

**Usage**

```
## S3 method for class 'rmd'  
print(x, digits = 3, ...)
```

**Arguments**

`x`                      an object of class `"rmd"`.  
`digits`                minimal number of significant digits.  
`...`                   further arguments passed to or from other methods.

---

`print.survQuantile`        *print.wkmQuantile*

---

**Description**

Print survQuantile object

**Usage**

```
## S3 method for class 'survQuantile'  
print(x, digits = 3, ...)
```

**Arguments**

`x`                      an object of class `"survQuantile"`.  
`digits`                minimal number of significant digits.  
`...`                   further arguments passed to or from other methods.

rmeanDiff

*rmeanDiff***Description**

Estimate difference of restricted mean survival based on (weighted) Kaplan-Meier estimates of the survival functions in each group.

**Usage**

```
rmeanDiff(L, formula, data, rr.subset = rep(TRUE, nrow(data)))
```

**Arguments**

L	time-limit specifying up to which time restricted mean will be calculated
formula	an object of class "formula" specifying the conditional survival model
data	data frame containing the variables in formula
rr.subset	logical vector defining subset of observations to use for response rate estimation (default: use all observations)

**Value**

An object of class "rmd", i.e. a list containing:

L	time limit, i.e. restricted mean up to time L is calculated
rmean1	restricted mean in group 1
rmean2	restricted mean in group 2
rmean.diff	estimated restricted mean difference
var.rmean1	an estimate of the asymptotic variance of the restricted mean in group 1
var.rmean2	an estimate of the asymptotic variance of the restricted mean in group 2
var.rmean.diff	an estimate of the asymptotic variance of the restricted mean difference
Z.rmean	the standardized test statistic for testing rmean.diff=0
p.value	p-value corresponding to Z.rmean

**Examples**

```
T <- c(rexp(100, 1), rexp(100, 2))
C <- c(rexp(100, 1), rexp(100, 2))
Y <- pmin(T, C)
D <- T <= C
Z <- rep(c(0,1), c(100, 100))
fit <- rmeanDiff(2, formula=Surv(Y, D) ~ Z, data.frame(Y=Y, D=D, Z=Z))
```

---

<code>rmeanDiff.ahr</code>	<i>rmeanDiff.ahr</i>
----------------------------	----------------------

---

**Description**

Estimate difference of restricted mean survival (based on ahr object as returned by ahr)

**Usage**

```
rmeanDiff.ahr(ahr.obj)
```

**Arguments**

`ahr.obj`            object of class "ahr"

**Details**

This function is usefull if the function 'ahr' has already been called, since the survival estimates in the object returned by 'ahr' can be reused.

**Value**

An object of class "rmd", i.e. a list containing:

<code>L</code>	time limit, i.e. restricted mean up to time L is calculated
<code>rmean1</code>	restricted mean in group 1
<code>rmean2</code>	restricted mean in group 2
<code>rmean.diff</code>	estimated restricted mean difference
<code>var.rmean1</code>	an estimate of the asymptotic variance of the restricted mean in group 1
<code>var.rmean2</code>	an estimate of the asymptotic variance of the restricted mean in group 2
<code>var.rmean.diff</code>	an estimate of the asymptotic variance of the restricted mean difference
<code>Z.rmean</code>	the standardized test statistic for testing <code>rmean.diff=0</code>
<code>p.value</code>	p-value corresponding to <code>Z.rmean</code>

**Examples**

```
T <- c(rexp(100, 1), rexp(100, 2))
C <- c(rexp(100, 1), rexp(100, 2))
Y <- pmin(T, C)
D <- T <= C
Z <- rep(c(0,1), c(100, 100))
fit <- avgHR(2, data.frame(Y=Y, D=D, Z=Z), formula=Surv(Y, D) ~ Z)
rmd <- rmeanDiff.ahr(fit)
```

---

wkm	<i>wkm</i>
-----	------------

---

### Description

Weighted Kaplan-Meier estimator with discrete time-independent covariate

### Usage

```
wkm(times, data, param = list(alpha = 1, var = TRUE, cov = FALSE,
  left.limit = FALSE, rr.subset = rep(TRUE, nrow(data))), formula = NULL)
```

### Arguments

times	a vector of evaluation times
data	data frame containing the variables in formula (if is.null(formula) expected column names are: Y (time), D (status), W (strat. factor), V (left-truncation times))
param	list of parameters containing: alpha: fractional parameter (default=1) var: if TRUE (default) calculate variance estimate cov: if FALSE (default) do not calculate covariance matrix estimate left.limit: if TRUE calculate left-continuous estimates, else calculate right-continuous estimates rr.subset: logical vector defining subset of observations to use for response rate estimation (default: use all observations)
formula	an object of class "formula" specifying the conditional survival model (only discrete covariates supported)

### Details

This function calculates the weighted Kaplan-Meier estimator for the survival function with weights based on a discrete time-independent covariate as described in Murray/Tsiatis (1996). The survival probabilities are evaluated at each entry in the vector `times`. The data frame `data` must either contain the variable in `formula` or, if `formula` is `NULL`, the variables `V` (left-truncation time), `Y` (censored failure time), `D` (censoring indicator), `W` (stratification variable). If `var` is `TRUE` then an estimate of the asymptotic variance is calculated for each entry in vector `times`. If `cov` is `TRUE` then the  $n \times n$  asymptotic covariance matrix is estimated, where  $n$  is the length of vector `times`. If `left.limit` is `TRUE` then a left-continuous estimate of the survival function is calculated instead of a right-continuous estimate (default). If a logical vector `rr.subset` is supplied, then only a subset of observations is used to estimate the response rates.

### Value

an object of class "wkm"

### References

S.~Murray and A.~A. Tsiatis. Nonparametric survival estimation using prognostic longitudinal covariates. *Biometrics*, 52(1):137–151, Mar. 1996.



---

wkmCompareQuantiles    *wkmCompareQuantiles*

---

### Description

Compare quantiles of two independent samples (ratio or difference) based on (weighted-) Kaplan-Meier estimator

### Usage

```
wkmCompareQuantiles(tau, formula, data, conf.level = 0.95,  
  null.value = 1, method = "ratio", p.value = FALSE)
```

### Arguments

tau	number between 0 and 1 specifying the quantile
formula	an object of class "formula" specifying the conditional survival model
data	data frame containing the variables in formula
conf.level	confidence level (or NULL if no confidence interval should be calculated)
null.value	true value of quantile ratio or difference
method	either "ratio" or "difference"
p.value	if TRUE p.value will be calculated (requires null.value)

### Value

An object of class "survQuantile", i.e. a list containing the estimated quantiles, confidence interval and p.value (if p.value = TRUE)

### References

su\_nonparametric\_1993

### Examples

```
T <- c(rexp(100, 1), rexp(100, 2))  
C <- c(rexp(100, 1), rexp(100, 2))  
Y <- pmin(T, C)  
D <- T <= C  
Z <- rep(c(0,1), c(100, 100)) # treatment indicator  
wkmCompareQuantiles(0.5, Surv(Y, D) ~ Z, data.frame(Y=Y, D=D, Z=Z))
```

---

wkmQuantile

*wkmQuantile*


---

### Description

Estimate arbitrary quantiles of a survival distribution based on the (weighted) Kaplan-Meier

### Usage

```
wkmQuantile(tau, formula, data, conf.level = 0.95, null.value = NULL,
  rr.subset = rep(TRUE, nrow(data)))
```

### Arguments

tau	number between 0 and 1 specifying the quantile to estimate
formula	an object of class "formula" specifying the conditional survival model
data	data frame containing the variables in formula
conf.level	confidence level (or NULL if no confidence interval should be calculated)
null.value	true value of quantile or NULL if no p-value should be calculated
rr.subset	logical vector defining subset of observations to use for response rate estimation (default: use all observations)

### Value

An object of class "survQuantile"

### References

brookmeyer\_confidence\_1982

### Examples

```
T <- c(rexp(100, 1), rexp(100, 2))
C <- c(rexp(100, 1), rexp(100, 2))
Y <- pmin(T, C)
D <- T <= C
Z <- rep(c(0,1), c(100, 100))
wkmQuantile(0.5, Surv(Y, D) ~ strata(Z), data.frame(Y=Y, D=D, Z=Z))
```

# Index

AHR, [2](#)  
AHR-package (AHR), [2](#)  
ahrAJ, [3](#), [9–11](#)  
ahrKM, [5](#)  
ahrUser, [6](#)  
ahrWKM, [8](#), [10](#), [11](#)  
aj, [4](#), [9](#)  
avgHR, [10](#)

etm, [4](#), [9](#), [10](#)

fastkm, [11](#)

print.ahr, [12](#)  
print.rmd, [13](#)  
print.survQuantile, [13](#)  
prodlm, [12](#)

rmeanDiff, [14](#)  
rmeanDiff.ahr, [15](#)

survfit, [5](#), [12](#)

wkm, [7](#), [8](#), [16](#)  
wkmCompareQuantiles, [17](#)  
wkmQuantile, [18](#)