*MMMMMM YYYY, Volume VV, Issue II.*

# Exact and Asymptotic Weighted Logrank Tests for Interval Censored Data: The interval R package (Draft Date: April 2, 2009) (Same as submitted except for headers)

**Michael P. Fay**
National Institute of Allergy and Infectious Diseases

**Pamela A. Shaw**
National Institute of Allergy and Infectious Diseases

**Abstract**

For right censored data perhaps the most commonly used test is the logrank test. In this paper we review several of generalizations of the logrank test to interval censored data and present an R package, **interval**, to implement many of them. The **interval** package depends on the **perm** package, which performs exact and asymptotic linear permutation tests. The **perm** package performs many of the tests from the **coin** package, and provides an independent validation of **coin**. We discuss steps that were taken to test and validate both the **interval** and **perm** packages.

*Keywords*: logrank test, wilcoxon test, exact tests, network algorithm, R.

## 1. Introduction

Finkelstein (1986) generalized the logrank test to interval censored data over 20 years ago. Finkelstein's test is appropriate for comparing treatment groups when the response is time to an event and time may only be known to fall into an interval. An example is time to progression-free survival (see e.g., Freidlin, Korn, Hunsberger, Gray, Saxman, and Zujewski (2007)), where patients are monitored intermittently and progression is known to have occurred only to within the time since the last visit. Despite this long history, including several different methods of generalization, these tests are rarely used in the medical literature. This maybe due in part to the lack of widely available software to analyze interval censored data.

In this paper we describe an R package, called **interval**, to perform weighted logrank tests (WLRT) (including a generalization of the Wilcoxon-Mann-Whitney test) for interval censored data. For each type of score (either logrank-type or Wilcoxon-type) the **interval** package allows three methods for creating tests: (1) score tests, (2) permutation tests derived from the score statistics in the grouped continuous model (GCM), both described in Fay (1999), and (3) multiple imputation tests as described in Huang, Lee, and Yu (2008). The p-values from the permutation tests may be calculated either asymptotically using a permutational central limit theorem or exactly using either complete enumeration, a network algorithm (for the two-sample case only), or Monte Carlo resampling. We know of no readily available software to perform these tests (nor other different generalizations of the WLRTs to interval censored data), except for the Splus functions (written by the first author) upon which the **interval** package is based (see **interval.tar.gz** at `http://stat.cmu.edu/S/`).

In Section 2 we give background on the generalizations the of WLRTs for interval censored data. We review different forms of the likelihood as well as the different methods for carrying out the tests. This section reviews what is known about the asymptotic validity of these tests for different interval censoring settings. With this background, we hope to give intuition on the methods for users primarily familiar with the usual right-censored logrank tests and also provide information to guide the applied researcher to make an appropriate choice for the test for their setting of interest.

The mathematical formulation of the WLRTs used in **interval** package are outlined in Section 3. Section 4 provides step-by-step instructions for how to use the **interval** package to perform data analysis. This application section demonstrates the use of two main functions of **interval**: `icfit` which provide survival distribution estimates and `ictest` which performs weighted logrank tests.

A major focus of this paper is the validation of the **interval** package. Since there is no other software available upon which to validate the **interval** software, we divide up the algorithms necessary for the test into several distinct pieces for which there *is* software to validate the program.

In Section 5, we discuss the estimation of the nonparametric maximum likelihood (NPMLE) estimator of the survival function from the full (interval censored) data. Currently, there is an R package, **Icens**, which provides many different algorithms for estimating that NPMLE. We have designed the **interval** package to be able to input NPMLEs from that package, using the class structure designed there (see Section 7). We also provide one algorithm as an independent estimator.

Some of the interval tests in the **interval** package are permutation tests, so we have created a separate package, **perm**, upon which **interval** depends, that performs exact and asymptotic linear permutation tests. The **perm** package is discussed in Section 6. Note that this package is mostly redundant because all of the tests performed by **perm** are available in the package **coin** (Hothorn, Hornik, van de Wiel, and Zeileis (2006)). The redundancy provides independent software to check the results from **perm**. We also discuss an important difference between **perm** and **coin** that arises when there are non-integer ties in response scores, something that can occur with non-negligible probability in interval censored data. We show one such example which details how the **interval** package treats these ties correctly, while the **coin** package (Version 1.0-5 or less) does not. Additionally, the **perm** package provides a network algorithm for the two-sample test (see e.g., Agresti, Mehta, and Patel (1990)), which is not

provided in **coin**, although the algorithms implemented in **coin** are generally faster than the network algorithm in **perm**. In Section 7, we also demonstrate how the **interval** can employ **coin** to carry out a permutation test of the logrank scores.

# 2. Background on Weighted Logrank Tests for Interval Censored Data

## 2.1. Overview for Applied Researchers

For right censored data, the logrank test is a score test on the proportional hazards model, so it is an efficient test to use when there are proportional hazards. There are several different versions of the logrank test that have been developed (see Kalbfleisch and Prentice (1980)). In particular, the likelihood could be the marginal likelihood of the ranks, the partial likelihood, or the grouped continuous model. Further, the variance could be estimated by the Fisher's information from the likelihood, by Martingale methods (see Fleming and Harrington (1991)) or by permutation methods. The differences between the several different versions of the logrank test are often not a focus of applied statisticians; however, in this paper since we are emphasizing validation of software, these slight differences need to be considered to avoid confusion and will be discussed in detail in later sections (see e.g., Callaert (2003)).

In addition to the logrank test , which is a WLRT with constant weight of 1 (or approximately 1), an important WLRT is the one that generalizes the Wilcoxon-Mann-Whitney test. We will call these latter tests Wilcoxon-type tests, but they are known by other names (e.g., Prentice-Wilcoxon test, propotional odds model, Harrington-Fleming $G^\rho$ class with $\rho = 1$). Similar to the logrank test, the Wilcoxon-type tests also have been derived using different likelihoods and using different variances. The important point for the applied researcher is that the Wilcoxon-type tests emphasize early events (when there are more people at risk) more than the later events (when there are fewer people at risk), while the logrank test gives constant weights over time.

One might be tempted, if one has interval censored data, to simply impute the mid-point of the intervals, and perform the usual right censored weighted logrank tests. Law and Brookmeyer (1992) studied the mid-point imputation method, where interval censored observations (Law and Brookmeyer (1992) use the term 'doubly censored observations' to mean interval censored observations) are replaced by the mid-point of the interval, then the data are treated as right-censored responses. They performed some simulations and showed that when the censoring mechanism is different between the two groups, the type I error of a nominal 0.05 test was in one case estimated to be as high as 0.19.

We now summarize the details of the next few sections. Both likelihoods that may be applied to interval censored data (the likelihood under the grouped continuous model (LGCM) and the marginal likelihood of the ranks (MLR)) should give similar answers. The permutation form of the tests are generally preferred over the score test forms when using the LGCM, since permuting allows exact inference when the censoring is not related to the covariate (e.g., treatment), and the permutation results avoid theoretical problems of the score test (see below and Fay (1996)). When the censoring is related to treatment and there are few inspection times compared to the number of subjects, the usual score test is recommended since it is asymptotically valid in this case. Now we give some more details on the different tests for interval censored data.

## 2.2. Which Likelihood?

From the start there is not a clear decision about which likelihood to use. Self and Grosman (1986) used the marginal likelihood of the ranks (MLR). This has the advantage that all the nuisance parameters are eliminated. The disadvantage of the MLR is that it is difficult to calculate. Note that even in the right censored case with ties, the likelihood is usually only approximated (see Kalbfleisch and Prentice (1980) pp. 74-75). Satten (1996) introduces a stochastic approximation to the MLR using Gibbs sampling for the proportional hazards model and it is generalized to proportional odds and other models by Gu, Sun, and Zuo (2005).

Finkelstein (1986) (see also, Fay (1996, 1999)) used the likelihood under the grouped continuous model (LGCM). In the LGCM, there are many nuisance parameters that must be estimated, and the standard likelihood-based tests (i.e., Score test, Wald test, and Likelihood ratio test) do not follow the usual theory unless there is a limited number of observation times which do not grow as the sample size increases (see Fay (1996)). Note however, that, the permutation test formed from the scores of the LGCM is theoretically justified, and is known to be a valid test when the censoring is unrelated to the covariate (see the following section). We discuss the computational issues of the LGCM in the next section. For the non-censored case, Pettitt (1984) studied the two likelihoods and showed that both likelihoods give asymptotically equivalent score tests as long as either the number of categories of response is fixed, or the number of categories does not increase too quickly compared to the total sample size. Pettitt concluded (see Pettitt (1984), section 5.1) that the score test for the MLR was more efficient (i.e., had greater power) than the score test for the LGCM; however, Pettitt did not consider the permutation form of the test using the LGCM.

Finally, when imputation methods are used then Martingale methods may be used (see Huang *et al.* (2008) and below).

## 2.3. Score Test, Permutation Test, or Imputation?

Once the likelihood is chosen, and the scores (i.e., each summand in the first derivative of the loglikelihood with respect to the parameter of interest evaluated under the null) are calculated, then the distribution of those scores under the null must be estimated. There are several methods for doing this, but the three most common are using asymptotic methods with the observed Fisher's information, which is commonly known as the score test, using permutation methods, or using multiple imputation (Huang *et al.* (2008)) (which in this paper we call within subject resampling).

When the censoring mechanism is the same for all treatment groups, then the permutation test is known to be valid for either the MRL or the LGCM. In this case of equal censoring, the score test is only known to be asymptotically valid using the MRL; using the LGCM we require the additional assumption that the number of observation times remains fixed as the sample size goes to infinity (see Fay (1996) for a discussion of this issue).

When there is unequal censoring then the theory of the permutation variance is not met. Thus, many authors have suggested that with unequal censoring the score variance is better (see e.g., Fay (1996), p. 820 for the interval censoring case).

Another strategy to create WLRT for interval censored data is to impute right censored data from the interval censored data and then properly adjust the variance. Huang *et al.* (2008)

improved on some earlier attempts at this variance adjustment after imputation. This appears to be a reasonable strategy, and provides an independent check on the other methods. On each imputation Huang *et al.* (2008) only considered the usual Martingale derived variance (use method="wsr.HLY" in `ictest`), while the **interval** package additionally allows for permutational variance (method="wsr.pclt") and Monte Carlo estimation within each imputation (method="wsr.mc").

## 3. Mathematical Formulation of the Scores for the WLRT

In this section, we provide the general form of rank invariant score test on the grouped continuous model, and for each of the three score tests available within **ictest**, we briefly describe the underlying survival model (or hazard model) and the mathematical form of the individual scores. Further details on the derivation of the tests are given in Fay (1996) and Fay (1999).

Suppose we have $n$ subjects. For the $i$th subject, use the following notation:

$x_i$ is the time to event, $X_i$ is the associated random variable.

$L_i$ is the largest observation time before the event is known to have occurred.

$R_i$ is the smallest observation time at or after the event is known to have occurred. In other words, we know that $x_i \in (L_i, R_i]$. We allow $R_i = \infty$ to denote right censoring.

$z_i$ is a $k \times 1$ vector of covariates.

Let the ordered potential observation times be $0 = t_0 < t_1 < t_2 < \cdots < t_m < t_{m+1} = \infty$. Partition the sample space by creating $(m+1)$ intervals, with the $j$th interval denoted $I_j \equiv (t_{j-1}, t_j]$. For simplicity, we assume that $L_i, R_i \in \{t_0, \ldots, t_{m+1}\}$. Let

$$\alpha_{ij} = \begin{cases} 1 & \text{if } L_i < t_j \leq R_i \\ 0 & \text{otherwise} \end{cases}$$

We write the general model of the survival for the $i$th individual as

$$Pr(X_i > t_j | z_i) = S(t_j | z_i'\beta, \gamma)$$

where $\beta$ is a $k \times 1$ vector of treatment parameters, and $\gamma$ is an $m \times 1$ vector of nuisance parameters for the unknown survival function. In the **interval** package, there are three different ways we model $S(t_j | z_i'\beta, \gamma)$, giving three different tests.

The grouped continuous likelihood for interval censored data is

$$L = \prod_{i=1}^{n} \sum_{j=1}^{m+1} \alpha_{ij} \left[ S(t_{j-1} | z_i'\beta, \gamma) - S(t_j | z_i'\beta, \gamma) \right] = \prod_{i=1}^{n} \left[ S(L_i | z_i'\beta, \gamma) - S(R_i | z_i'\beta, \gamma) \right] \quad (1)$$

To form the score statistic we take the derivative of $\log(L)$ with respect to $\beta$ and evaluate it at $\beta = 0$. The MLE of the nuisance parameters when $\beta = 0$ (in terms of the baseline survival) are the self-consistent estimates of survival, $\hat{S}(t_j), j = 1, \ldots, m$. For convenience, let $\hat{S}(t_0) = 1$ and $\hat{S}(t_{m+1}) = 0$, even though these values are known by assumption.

We can write the efficient score vector for the parameter $\beta$ (see Fay (1996), Fay (1999)) as

$$U = \sum_{i=1}^{n} z_i \left( \frac{\hat{S}'(L_i) - \hat{S}'(R_i)}{\hat{S}(L_i) - \hat{S}(R_i)} \right) \equiv \sum_{i=1}^{n} z_i c_i \tag{2}$$

where $\hat{S}(t)$ is the nonparametric estimate of the survival function at $t$ using the pooled data across all groups, and $\hat{S}'(t)$ is the derivative with respect to $\beta$.

When $z_i$ is an $k \times 1$ vector of indicators of $k$ treatments, we can rewrite the $\ell$th row of $U$ as

$$U_\ell = \sum_{j=1}^{m} w_j \left[ d'_{j\ell} - \frac{n'_{j\ell} d'_j}{n'_j} \right], \tag{3}$$

where

$$w_j = \frac{\hat{S}(t_j)\hat{S}'(t_{j-1}) - \hat{S}(t_{j-1})\hat{S}'(t_j)}{\hat{S}(t_j)\left[\hat{S}(t_{j-1}) - \hat{S}(t_j)\right]},$$

and $d'_{j\ell}$ represents the expected value under the null of the number of deaths in $I_j$ for the $\ell$th treatment group, $d'_j$ represents the expected value under the null of the total number of deaths in $I_j$, similarly $n'_{j\ell}$ and $n'_j$ represent the expected number at risk.

We now give the values for $c_i$ (from equation 2) and $w_i$ (from equation 3) for 3 different survival models provided in **ictest**. Although not developed first, we present the model of Sun (1996) first because it is the generalization of the logrank test most commonly used for right censored data. Sun (1996) modelled the odds of discrete hazards as proportional to $\exp(z'_i\beta)$ (see Fay (1999)), leading to the more complicated survival function:

$$S(t_j|z_i,\gamma) = \prod_{k=1}^{j} \left\{ 1 + \left( \frac{S(t_{k-1}|\gamma) - S(t_k|\gamma)}{S(t_k|\gamma)} \right) \right\}^{-1}.$$

Here and in the other two models, $S(t_j|\gamma)$ is a estimator of survival that does not depend on the covariates $z_i$, and $S(t_j|\gamma)$ is nonparametric because the $\gamma$ is $m \times 1$ and there are only $m$ unique time points observed in the data. Denote its estimator $S(t|\hat{\gamma}) \equiv \hat{S}(t)$, which is the NPMLE of the survival function of all the data ignoring covariates. Under the model of Sun (1996) we get,

$$c_i = \frac{\hat{S}(L_i)\log\tilde{S}(L_i) - \hat{S}(R_i)\log\tilde{S}(R_i)}{\hat{S}(L_i) - \hat{S}(R_i)} \tag{4}$$

where $\tilde{S}(t_j) = \exp\left(-\sum_{\ell=1}^{j}\hat{\lambda}_\ell\right)$, and $\lambda_\ell = \left\{\hat{S}(t_{\ell-1}) - \hat{S}(t_\ell)\right\}/\hat{S}(t_{\ell-1})$, and

$$w_j = 1.$$

This model is called from the **interval** package by the option scores="logrank1".

The second model we consider was actually developed first, it is the grouped proportional hazards model introduced by Finkelstein (1986), where the survival function is modeled as

$S(t_j|z_i'\beta,\gamma) = S(t_j|\gamma)^{\exp(z_i'\beta)}$. Under this grouped propotional hazards model, the $c_i$ values are:

$$c_i = \begin{cases} \frac{\hat{S}(L_i)\log\hat{S}(L_i)-\hat{S}(R_i)\log\hat{S}(R_i)}{\hat{S}(L_i)-\hat{S}(R_i)} & \text{for } R_i < t_{m+1} \\ \log\hat{S}(L_i) & \text{for } R_i = t_{m+1} \equiv \infty \end{cases} \quad (5)$$

and

$$w_j = \frac{\hat{S}(t_{j-1})\left[\log\hat{S}(t_{j-1}) - \log\hat{S}(t_j)\right]}{\hat{S}(t_{j-1}) - \hat{S}(t_j)}.$$

Note that because this model is a proportional hazards one, we call the resulting test a logrank test also and the model is called by scores="logrank2" in the **interval** package. When $\hat{S}(t_{j-1})/\hat{S}(t_j) \approx 1$ then $w_j \approx 1$.

Finally, we consider the model proposed by Fay (1996) giving the Wilcoxon-type test, where the odds are proportional to $\exp(-z_i\beta)$ so that the survival function is

$$S(t_j|z_i,\gamma) = \left\{1 + \left(\frac{1-S(t_j|\gamma)}{S(t_j|\gamma)}\right) exp(z_i\beta)\right\}^{-1}$$

and we get

$$c_i = \hat{S}(L_i) + \hat{S}(R_i) - 1$$

and

$$w_j = \hat{S}(t_{j-1})$$

We now show the form of the scores in the special case of right censoring. For this, we introduce new notation. Suppose that there are $m^*$ observed failure times, $t_1^* < t_2^* < \cdots < t_{m^*}^*$. In other words there are $m^*$ subjects for which $x_i = R_i$ is known, so that $L_i = \lim_{\epsilon\to 0} R_i - \epsilon \equiv R_i - 0$. Let $n_j$ and $d_j$ be the number of subjects who are at risk or fail respectively at $t_j^*$. Then the Kaplan-Meier survival estimate is (see e.g., Kalbfleisch and Prentice (1980))

$$\hat{S}(t) = \prod_{j|t_j^*\le t}\left(\frac{n_j - d_j}{n_j}\right).$$

In the following table we summarize the formulation of the scores for the 3 model (score) choices in **interval**, as described above, for both interval censored and ordinary right censored data.

**Scores For Right Censored Data**

| Test (Model) | Score ($c_i$) for Observed failure at $t_h^*$ | Score ($c_{i'}$) for Right-censored observation at $t_{h'}^*$ |
|---|---|---|
| Logrank1 (Logistic, Sun) | $1 - \sum_{\ell=1}^{h}\frac{d_\ell}{n_\ell}$ | $-\sum_{\ell=1}^{h}\frac{d_\ell}{n_\ell}$ |
| Logrank2 (Group Prop Hazards, Finkelstein) | $\frac{n_h}{d_h}\left\{-\log\left(\frac{n_h-d_h}{n_h}\right)\right\} + \log\hat{S}(t_h^*)$ | $\log\left\{\hat{S}(t_{h'}^*)\right\}$ |
| Generalized WMW (Proportional Odds) | $\hat{S}(t_{h-1}^*) + \hat{S}(t_h^*) - 1$ | $\hat{S}(t_{h'}^*) - 1$ |

# 4. Application

The calls to the **interval** package are designed to be in the same style as in the **survival** package. As noted in the previous section, the `icfit` and `ictest` functions will work on right censored data (see `\interval\demo\right.censored.examples.R\`). We demonstrate the two main functions in **interval**, `icfit` and `ictest`, using the often cited interval censored breast cosmesis data set of Finkelstein and Wolfe (1985).

## 4.1. Survival Estimation

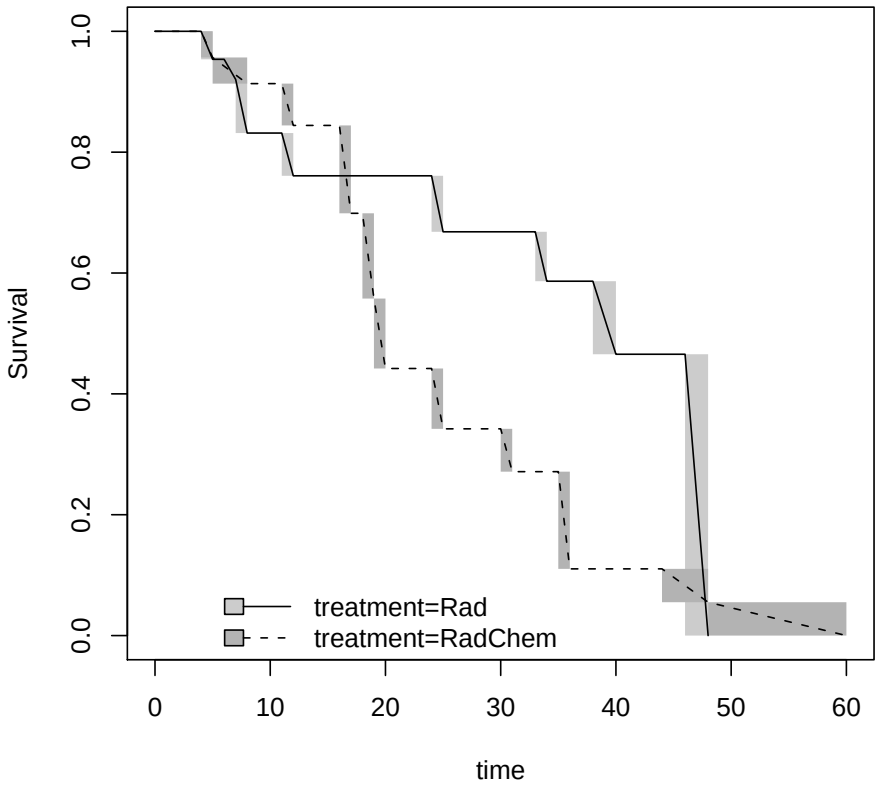
Here we calculate the NPMLE for each treatment group in the breast cosmesis data separately and print them out:

```
> library(interval)
> data(bcos)
> fit1 <- icfit(Surv(left, right, type = "interval2") ~ treatment,
+     data = bcos)
> summary(fit1)

treatment=Rad:
  Interval Probability
1    (4,5]      0.0463
2    (6,7]      0.0334
3    (7,8]      0.0887
4  (11,12]      0.0708
5  (24,25]      0.0926
6  (33,34]      0.0818
7  (38,40]      0.1209
8  (46,48]      0.4656
treatment=RadChem:
   Interval Probability
1     (4,5]      0.0433
2     (5,8]      0.0433
3   (11,12]      0.0692
4   (16,17]      0.1454
5   (18,19]      0.1411
6   (19,20]      0.1157
7   (24,25]      0.0999
8   (30,31]      0.0709
9   (35,36]      0.1608
10  (44,48]      0.0552
11  (48,60]      0.0552
```

These results match those calculated from **Icens**. The `summary` function applied to an `icfit` object gives the intervals which have positive probability in the NPMLE of the survival distribution function, i.e. where the estimated survival distribution drops; however, there are infinitely many functions which drop exactly the same increment within those intervals.

Figure 1: Non-parametric Maximum Likelihood Survival from Breast Cosmesis Data




The NPMLE is only unique outside of the intervals which are listed from the summary of the fit. For example, there are infinitely many survival functions for the treatment=Rad group, that have $S(4) = 1$ and $S(5) = 1 - 0.0463 = 0.9537$. Thus, as has been done in the `Icens` package, when plotting the NPMLEs we denote the areas with the indeterminite drops with grey rectangles. The function which linearly interpolates the survival within these indeterminte regions is also displayed on the graph. We plot the NPMLE for each treatment group using `plot(fit1)` to get Figure 1.

### 4.2. Two-sample Weighted logrank tests

There are three score tests available in `ictest`, which are selected by setting the `scores` argument to be one of "logrank1", "logrank2", or "WMW". As stated in Section 3, the two forms of the logrank scores are the scores associated with Finkelstein (1986) and the scores associated with Sun (1996). Although Finkelstein (1986) are perhaps more natural for interval censored data, we make those of Sun (1996) the default (`scores`="logrank1" or equivalently `rho`=0) since these scores reduce to the usual logrank scores with right censored data. The default method is the permutation test, and since the sample size is sufficiently large we automatically get the version based on the permutational central limit theorem:

```
> icout <- ictest(Surv(left, right, type = "interval2") ~ treatment,
+     data = bcos)
> icout

        Asymptotic Logrank two-sample test (permutation form), Sun's scores

data:  Surv(left, right, type = "interval2") by treatment
Z = -2.6684, p-value = 0.007622
alternative hypothesis: survival distributions not equal

                   n Score Statistic*
treatment=Rad     46        -9.141846
treatment=RadChem 48         9.141846
* like Obs-Exp, positive implies earlier failures than expected
```

Because a major part of the calculation of the test statistic is solving the NPMLE under the null hypothesis (i.e., for the pooled treatment groups), this NPMLE is saved as part of the output so that we can calculate this NPMLE once and reuse it for the calculation of the other two score tests. Here is code for the Finklestein logrank formulation:

```
> ictest(Surv(left, right, type = "interval2") ~ treatment, data = bcos,
+     initfit = icout$fit, scores = "logrank2")

        Asymptotic Logrank two-sample test (permutation form), Finkelstein's
        scores

data:  Surv(left, right, type = "interval2") by treatment
Z = -2.6839, p-value = 0.007277
alternative hypothesis: survival distributions not equal

                   n Score Statistic*
treatment=Rad     46        -9.944182
treatment=RadChem 48         9.944182
* like Obs-Exp, positive implies earlier failures than expected
```

Notice how the two different logrank tests give very similar results. We demonstrate the third score test, the generalization of the Wilcoxon-Mann-Whitney scores to interval censored data, and also demonstrate the `ictest` function in default mode:

```
> L <- bcos$left
> R <- bcos$right
> trt <- bcos$treatment
> ictest(L, R, trt, scores = "wmw", initfit = icout$fit)

        Asymptotic Wilcoxon two-sample test (permutation form)

data:  {L,R} by trt
```

```
Z = -2.1672, p-value = 0.03022
alternative hypothesis: survival distributions not equal

         n Score Statistic*
Rad     46        -5.656724
RadChem 48         5.656724
* like Obs-Exp, positive implies earlier failures than expected
```

### 4.3. K-sample and trend tests

We can perform $k-sample$ tests using the `ictest` function. We create fake treatment assignments with four treatment groups to demonstrate.

```
> set.seed(1232)
> fakeTrtGrps <- sample(letters[1:4], dim(bcos)[[1]], replace = TRUE)
> ictest(L, R, fakeTrtGrps)

        Asymptotic Logrank k-sample test (permutation form), Sun's scores

data:  {L,R} by fakeTrtGrps
Chi Square = 1.3685, p-value = 0.7129
alternative hypothesis: survival distributions not equal

   n Score Statistic*
d 27       -0.7520431
b 24        1.8821766
c 20       -2.8048270
a 23        1.6746934
* like Obs-Exp, positive implies earlier failures than expected
```

When `scores`= "*wmw*" and the responses are all non-overlapping intervals then this reduces to the Kruskal-Wallis test. The function `ictest` performs a trend test when the covariate is numeric. The one-sided test with `alternative`= "*less*" rejects when the correlation between the generalized rank scores (e.g., WMW scores or logrank scores) and the covariate are small.

```
> set.seed(931)
> fakeZ <- rnorm(dim(bcos)[[1]])
> ictest(L, R, fakeZ, alternative = "less")

        Asymptotic Logrank trend test(permutation form), Sun's scores

data:  {L,R} by fakeZ
Z = 0.068, p-value = 0.5271
alternative hypothesis: higher independent variable implies earlier failure times than exp

      n Score Statistic*
[1,] 94        0.4421144
* postive so larger covariate values give earlier failures than expected
```

### 4.4. Exact permutation tests

We can also estimate the exact permutation p-value for any score choice in **ictest** using the `exact` argument. Here the logrank test using Sun (1996) scores is redone as an exact test:

```
> ictest(Surv(left, right, type = "interval2") ~ treatment, data = bcos,
+     exact = TRUE, scores = "logrank1")

        Exact Logrank two-sample test (permutation form), Sun's scores

data:  Surv(left, right, type = "interval2") by treatment
p-value = 0.006
alternative hypothesis: survival distributions not equal

                   n Score Statistic*
treatment=Rad     46        -9.141846
treatment=RadChem 48         9.141846
* like Obs-Exp, positive implies earlier failures than expected
p-value estimated from 999 Monte Carlo replications
99 percent confidence interval on p-value:
 0.0002072893 0.0184986927
```

The exact argument automatically chooses between an exact calculation by network algorithm or an approximation to the exact p-value by Monte Carlo through the `methodRuleIC1` function. In this case the network algorithm was expected to take too long and the Monte Carlo approximation was used. If a more accurate approximation to the exact p-value is needed then more Monte Carlo simulations could be used and these are changed using the `mcontrol` option.

### 4.5. Other test options

All of the above are permutation based tests, but we may use other methods. Here are the results from the usual score test for interval censored data:

```
> ictest(Surv(left, right, type = "interval2") ~ treatment, data = bcos,
+     initfit = icout$fit, method = "scoretest", scores = "logrank2")

        Asymptotic Logrank two-sample test (score form), Finkelstein's scores

data:  Surv(left, right, type = "interval2") by treatment
Chi Square = 7.8749, p-value = 0.005012
alternative hypothesis: survival distributions not equal

                   n Score Statistic*
treatment=Rad     46        -9.944182
treatment=RadChem 48         9.944182
* like Obs-Exp, positive implies earlier failures than expected
```

where in this case the nuisance parameters are defined after calculation of the NPMLE as described in Fay (1996), and the results agree exactly with Fay (1996) and are similar to those in Finkelstein (1986). The very small differences may be due to differing convergence criteria in the NPMLE. The imputation method of Huang *et al.* (2008) may also be employed (note that `scores`="logrank2" are not available for this method):

```
> ictest(Surv(left, right, type = "interval2") ~ treatment, data = bcos,
+     initfit = icout$fit, method = "wsr.HLY", mcontrol = mControl(nwsr = 99),
+     scores = "logrank1")


        Asymptotic Logrank 2-sample test(WSR HLY), Sun's scores

data:  Surv(left, right, type = "interval2") by treatment
Chi Square = 7.1047, p-value = 0.007688
alternative hypothesis: survival distributions not equal

                   n Score Statistic*
treatment=Rad     46        -9.141846
treatment=RadChem 48         9.141846
* like Obs-Exp, positive implies earlier failures than expected
p-value estimated from  Monte Carlo replications
```

These results agree with Huang *et al.* (2008) within the error to be expected from such an imputation method (Huang *et al.* (2008) had $p = 0.0075$).

## 5. Nonparametric Estimator of Survival

For each of the tests in `ictest`, the NPLME survival function must be obtained. There are many algorithms for calculating the NPMLE from interval censored data (i.e., $\hat{S}$), including several options in the **Icens** package. In our **interval** package, we provide an internally calculated estimate and give the user the option for an externally obtained estimate, say from the existing package **Icens**, to be supplied to **ictest**. For the internal (default) calculation, we use a self-consistent algorithm, which is an EM-algorithm applied to interval censored data (see Turnbull (1976)); however, first there is a primary reduction (see Aragón and Eberly (1992)). Also, as recommended in Gentleman and Geyer (1994), to speed up computations, we provisionally set the probability in some intervals to zero if they are below some bound, then check the Kuhn-Tucker conditions to make sure that those values are really very close to zero. If those conditions are not met then the small probability is added back on and the iterations continue. Convergence is defined when the maximum reduced gradient is less than some minimum error, and the Kuhn-Tucker conditions are approximately met (see Gentleman and Geyer (1994)).

We test in `\demo\npmle.R` that the NPMLE from the **Icens** package match with those from the **interval** package. In that file we compare the NPMLE from the cosmesis data set. We additionally simulate 30 other data sets and show that the NPMLE's match for all the simulated data sets (data not shown).

To demonstrate the software we take an artificial example with hypothetical data where we can calculate the NPMLE exactly. We include group membership to validate the logrank tests in the next section. Consider the data set with:

```
> L <- c(2, 5, 1, 1, 9, 8, 10)
> R <- c(3, 6, 7, 7, 12, 10, 13)
> group <- c(0, 0, 1, 1, 0, 1, 0)
> example1 <- data.frame(L, R, group)
> example1
```

```
   L  R group
1  2  3     0
2  5  6     0
3  1  7     1
4  1  7     1
5  9 12     0
6  8 10     1
7 10 13     0
```

The NPMLE using all the data is:

| $(L$ | $R]$ | probability |
|---|---|---|
| 2 | 3 | $\frac{2}{7}$ |
| 5 | 6 | $\frac{2}{7}$ |
| 9 | 10 | $\frac{3}{14}$ |
| 10 | 12 | $\frac{3}{14}$ |

We calculate this with the **interval** package as

```
> library(interval)
> summary(icfit(L, R), digits = 12)
```

```
  Interval    Probability
1    (2,3] 0.285714285714
2    (5,6] 0.285714285714
3   (9,10] 0.214285714286
4  (10,12] 0.214285714286
```

which matches the exact to at least 12 digits:

```
> print(3/14, digits = 12)
```

```
[1] 0.214285714286
```

Usually the fit will not be this close, and the closeness of the fit is determined by the `icfitControl` list (see the help). In Section 7, we provide an example to show how NPMLE survival estimates from other packages can be used by `ictest`.

# 6. Permutation tests for interval censored data

The package **interval** relies on our **perm** package to perform exact and asymptotic linear permutation tests of the logrank statistics. Appendix I provides a detailed description of **perm**, including validation details for a set of standard statistical tests where the results from **perm** are compared to those from the widely used permutation test package **coin**. In this section, we present an example which demonstrates a problem with the **coin** package that can occur with interval censored data and one that can be addressed appropriately with the **perm** package. We consider the `example1` data set from Section 5 to elucidate the issue. The problem relates to the numerical precision of the calculated scores and subsequent permutation p-value when there is a small number of permuations and ties in the scores (for interval censoring, stemming from overlapping intervals). While not unique to interval censored data, this combination of factors may be more common in this setting.

We can calculate the exact scores for the Sun method (Eq (4); i.e. scores="logrank1") these are

$$\left[\frac{5}{7}, \frac{11}{35}, \frac{18}{35}, \frac{18}{35}, -\frac{24}{35}, -\frac{13}{70}, -\frac{83}{70}\right]$$

These scores sum to zero (as do all such scores regardless of the model). There are $\binom{7}{3} = 35$ unique permutations with equal probability. Note that the difference in means of the original scores, (with group=$[0,0,1,1,0,1,0]$), gives equivalent values to the permutation with group=$[1,1,0,0,0,1,0]$ because the sum of the first and second scores equals the sum of the third and fourth scores. Thus, we have a tie in the permutation distribution. We need to make sure the computer treats it as a tie otherwise the p-value will be wrong. Dealing with ties in computer computations can be tricky (see R FAQ 7.31 at http://cran.r-project.org/doc/FAQ/R-FAQ.html). To see the details, we completely enumerate all the sums of the scores in one group. We use the function `chooseMatrix` from **perm** to generate the full list of permuations of the original `group` variable. We print out only the first 9 of the 35 ordered test statistics, placing the difference in means in the 8th column, next to the permutation of the group allocation:

```
> score1 <- wlr_trafo(Surv(L, R, type = "interval2"))
> cm <- chooseMatrix(7, 3)
> T <- ((1 - cm) %*% score1)/4 - (cm %*% score1)/3
> cbind(cm, T)[order(T), ][1:9, ]
```

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7]       [,8]
[1,]    1    0    1    1    0    0    0 -1.0166667
[2,]    1    1    1    0    0    0    0 -0.9000000
[3,]    1    1    0    1    0    0    0 -0.9000000
[4,]    0    1    1    1    0    0    0 -0.7833333
[5,]    1    0    1    0    0    1    0 -0.6083333
[6,]    1    0    0    1    0    1    0 -0.6083333
[7,]    1    1    0    0    0    1    0 -0.4916667
[8,]    0    0    1    1    0    1    0 -0.4916667
[9,]    0    1    1    0    0    1    0 -0.3750000
```

The seventh and eighth largest of the 35 test statistics are tied, and the eighth largest is equal to the original group assignment, so that the one sided p-value is $8/35 = 0.2286$. We see that `ictest` properly calculates this p-value while the **coin** package version 1.0-5 used on the scores does not:

```
> ictest(L, R, group, alternative = "less")

        Exact Logrank two-sample test (permutation form), Sun's scores

data:  {L,R} by group
p-value = 0.2286
alternative hypothesis: 1 has earlier failure times than expected

  n Score Statistic*
0 4       -0.8428571
1 3        0.8428571
* like Obs-Exp, positive implies earlier failures than expected

> library(coin)
> packageDescription("coin")$Version

[1] "1.0-5"

> independence_test(score1 ~ as.factor(group), alternative = "less",
+     distribution = exact())

        Exact General Independence Test

data:  score1 by as.factor(group) (0, 1)
Z = -0.9031, p-value = 0.2
alternative hypothesis: less
```

The way that **perm** can directly address the ties issue is to allow the user to specify numerical precision, i.e. rounding to the nearest `permControl()$digits` significant digits; and **perm** treats values of the permutation distribution that are tied for that many significant digits as true ties.

## 7. Interacting with Other Packages

The focus of the **interval** and **perm** packages has been accuracy rather than speed; however, faster calculations can be performed by incorporating some of the fast algorithms available in other packages. We allow an option for the user to provide the NPMLE for survival as input, both to increase the calculation speed and also the flexibility of the package to use other estimates for the NPMLE, such as those provided by the `Icens` package.

The major computation time for the `ictest` function when the sample size is large and the permutational central limit theorem may be used is the calculation of the NPMLE from all

treatment groups combined. Although the **interval** package (which requires the **perm** and **survival** packages) can calculate the NPMLE fairly quickly, there are other algorithms which may be faster available in the **Icens** package. One algorithm available is the hybrid EM ICM (Iterative convex minorant) estimator of the distribution function proposed by Wellner and Zhan (1997). We now show how the `EMICM` function may be used together with `ictest` by employing the `initfit` opiton and compare this to the `icfit` function used the same way. We again use the `bcos` data, and see the resulting p-value is the same, but the `EMICM` algorithm is faster. Note since we use the `initfit` option within `ictest`, the method that first calls `EMICM` converges is at least as close to the true NPMLE compared to the single call to `ictest`. This noninferiority of convergence is due to the fact that both calls to `ictest` use the same default `icfitControl` option, but the `EMICM` result may have already converged closer to the NPMLE than required by the convergence criteria of the default `icfitControl`.

```
> ictest.alone <- function() {
+     ictest(Surv(left, right, type = "interval2") ~ treatment,
+         data = bcos)$p.value
+ }
> library(Icens)
> emicm.first <- function() {
+     npmle <- EMICM(bcos[, c("left", "right")])
+     ictest(Surv(left, right, type = "interval2") ~ treatment,
+         data = bcos, initfit = npmle)$p.value
+ }
> ictest.alone()

[1] 0.007621637

> emicm.first()

[1] 0.007621638

> system.time(ictest.alone())

   user  system elapsed
   1.30    0.00    1.29

> system.time(emicm.first())

   user  system elapsed
   0.49    0.00    0.48
```

When an exact test is desired, then there are some algorithms in the **coin** package which will likely be faster than the network algorithm in the **perm** package. We take the first 12 subjects in each treatment group from `bcos` in order to compare the exact methods. From **interval**, we use the `wlr_trafo` function to calculate the WLR scores for the interval censored data and use `independence_test` from **coin**, as well as `ictest`, to calculate the permutation p-value from the scores.

```
> c12 <- bcos[c(1:12, 47:58), ]
> ictest(Surv(left, right, type = "interval2") ~ treatment, data = c12,
+     method = "exact.network", alternative = "less")

        Exact Logrank two-sample test (permutation form), Sun's scores

data:  Surv(left, right, type = "interval2") by treatment
p-value = 0.2634
alternative hypothesis: treatment=RadChem has earlier failure times than expected

                   n Score Statistic*
treatment=Rad     12        -1.191741
treatment=RadChem 12         1.191741
* like Obs-Exp, positive implies earlier failures than expected

> independence_test(wlr_trafo(Surv(left, right, type = "interval2")) ~
+     treatment, data = c12, distribution = exact(), alternative = "less")

        Exact General Independence Test

data:  wlr_trafo(Surv(left, right, type = "interval2")) by treatment (Rad, RadChem)
Z = -0.6506, p-value = 0.2634
alternative hypothesis: less

> system.time(ictest(Surv(left, right, type = "interval2") ~ treatment,
+     data = c12, method = "exact.network", alternative = "less"))

   user  system elapsed
   2.86    0.00    2.86

> system.time(independence_test(wlr_trafo(Surv(left, right, type = "interval2")) ~
+     treatment, data = c12, distribution = exact(), alternative = "less"))

   user  system elapsed
   0.20    0.00    0.21
```

We see that in fact, the **coin** package does give the same answer considerably faster. Note that the network algorithm in **perm** was written in R instead of calling faster code in C as was done in **coin** using a different algorithm. Though we leave this choice up to the user, if there are ties in the permutation distribution, the **coin** package should be used with some caution (see Section 6).

## References

Agresti A, Mehta C, Patel N (1990). "Exact inference for contingency tables with ordered categories." *Journal of the American Statistical Association*, **85**, 453–458.

Aragón J, Eberly D (1992). "On Convergence of Convex Minorant Algorithms for Distribution Estimation with Interval-Censored Data." *Journal of Computational and Graphical Statistics*, **1**, 129–140.

Callaert H (2003). "Comparing Statistical Software Packages: The Case of the Logrank Test in StatXact." *American Statistician*, **57**, 214–217.

Fay MP (1996). "Rank invariant tests for interval censored data under the grouped continuous model." *Biometrics*, **52**, 811–822.

Fay MP (1999). "Comparing several score tests for interval censored data (Corr: 1999V18 p2681)." *Statistics in Medicine*, **18**, 273–285.

Finkelstein D, Wolfe R (1985). "A Semiparametric Model for Regression Analysis of Interval-Censored Failure Time Data." *Biometrics*, **41**, 845–854.

Finkelstein DM (1986). "A proportional hazards model for interval-censored failure time data." *Biometrics*, **42**, 845–854.

Fleming T, Harrington D (1991). *Counting Processes and Survival Analysis*. Wiley, New York.

Freidlin B, Korn EL, Hunsberger S, Gray R, Saxman S, Zujewski JA (2007). "Proposal for the use of progression-free survival in unblinded randomized trials." *Journal of Clinical Oncology*, **25**(15), 2122–2126.

Gentleman R, Geyer C (1994). "Maximum Likelihood for Interval Censored Data: Consistency and Computation." *Biometrika*, **81**, 618–623.

Gu M, Sun L, Zuo G (2005). "A Baseline-Free Procedure for Transformation Models Under Interval Censorship." *Lifetime Data Analysis*, **11**, 473–488.

Hothorn T, Hornik K, van de Wiel MA, Zeileis A (2006). "A Lego System for Conditional Inference." *The American Statistician*, **60**(3), 257–263.

Huang J, Lee C, Yu Q (2008). "A generalized log-rank test for interval-censored failure time data via multiple imputation." *Statistics in Medicine*, **DOI: 10.1002/sim.3211**.

Kalbfleisch J, Prentice R (1980). *The Statistical Analysis of Failure Time Data*. Wiley, New York.

Law C, Brookmeyer R (1992). "Effects of mid-point imputation on the analysis of doubly censored data." *Statistics in Medicine*, **11**, 1569–1578.

Pettitt A (1984). "Tied, grouped continuous and ordered categorical data: A comparison of two models." *Biometrika*, **71**, 35–42.

Satten GA (1996). "Rank-based inference in the proportional hazards model for interval censored data." *Biometrika*, **83**(2), 355–370. Cited By (since 1996): 37.

Self SG, Grosman EA (1986). "Linear rank tests for interval-censored data with application to PCB levels in adipose tissue of transformer repair workers." *Biometrics*, **42**, 521–530.

Sen P (1985). "Permutational Central Limit Theorems." In S Kotz, NL Johnson (eds.), "Encyclopedia of Statistics," volume 6. Wiley.

Sun J (1996). "A Non-Parametric Test for Interval-Censored Failure Time Data with Application to AIDS Studies." *Statistics in Medicine*, **15**, 1387–1395.

Turnbull B (1976). "The Empirical Distribution Function with Arbitrarily Grouped, Censored and Truncated Data." *Journal of the Royal Statistical Society, Series B*, **38**, 290–295.

Wellner JA, Zhan Y (1997). "A hybrid algorithm for computation of the nonparametric maximum likelihood estimator from censored data." *Journal of the American Statistical Association*, **92**, 945–959.

## Appendix I: Perm Package

The **perm** package is a stand alone package to perform linear permutation tests, i.e. permutation tests where the test statistic is either of the form,

$$T(\mathbf{y}, \mathbf{x}) = \sum_{i=1}^{n} c_i z_i$$

as in equation 2, or of a quadratic version of $T(\mathbf{y}, \mathbf{x})$ (e.g., see $k$-sample tests below). Currently, there are three permutation tests available in **perm**: `permTS` to perform two sample tests, `permKS` to perform K-sample tests, `permTREND` to perform trend tests on numeric values. In Section 6, we provided an example with interval censored data where the existing permutation **coin** was problematic. In this section, we provide more standard examples of these three classical permutation tests and demonstrate that both **perm** and the existing **coin** package provide identical results.

We consider only the case where $c_i$ is a scalar and $z_i$ is either a scalar or a $k \times 1$ vector (although more general cases are studied in Sen (1985), see also Hothorn *et al.* (2006)). Following Sen (1985), we can write the mean and variance of $T$ under the permutation distribution (i.e., permute indices of $c_1, \ldots, c_n$ and recalculate $T$, where there are $n!$ different permutations with each equally likely) as,

$$U = E_P(T) = n\bar{c}\bar{z}$$

$$V = Var_P(T) = \frac{1}{n-1}\left\{\sum_{i=1}^{n}(c_i - \bar{c})^2\right\}\left\{\sum_{j=1}^{n}(z_i - \bar{z})(z_i - \bar{z})'\right\},$$

where $\bar{c}$ and $\bar{z}$ are the sample means. Sen (1985) reviews the permutational central limit theorem (PCLT) which shows that under the permutation distribution with standard regularity conditions on the $c_i$ and $z_i$, $V^{-1/2}(T - U)$ is asymptotically approximately multivariate normal with mean 0 and variance the identity matrix.

In the **perm** package, if $z_i$ is a scalar we define the one-sided p-value when `alternative`="greater" as

$$p_G = \frac{\sum_{i=1}^{n!} I(T_i \geq T_0)}{n!},$$

where $I(A) = 1$ when $A$ is true and 0 otherwise, $T_i$ is the $i$th of the $n!$ permutations, and $T_0$ is the observed value of $T$. When `alternative`="less" then the p-value, say $p_L$, is given as above except we reverse the direction of the comparison operator in the indicator function. Note that if you add or multiply by constants which do not change throughout all permutations then the p-value does not change. Thus, a permutation test on $T$ can represent a test on the difference in means in the two-sample case, and can represent a test on the correlation when $z_i$ is numeric. When `alternative`="two.sided", then $p_2$ is twice the minimum one-sided p-value (i.e., $p_2 = \min(1, 2\min(p_L, p_G))$), and when `alternative`="two.sidedAbs" then

$$p_{2A} = \frac{\sum_{i=1}^{n!} I(|T_i - U| \geq |T_0 - U|)}{n!}.$$

When $\bar{c} = 0$ (as is the case with the weighted logrank $c_i$ values defined in Section 3) then both two-sided p-values are equivalent. Here is a two-sample permutation t-test in both **perm** and **coin** giving first the asymptotic, then the exact p-values:

```
> independence_test(extra ~ group, data = sleep)

        Asymptotic General Independence Test

data:  extra by group (1, 2)
Z = -1.7508, p-value = 0.07998
alternative hypothesis: two.sided

> permTS(extra ~ group, data = sleep)

        Permutation Test using Asymptotic Approximation

data:  extra by group
Z = -1.7508, p-value = 0.07998
alternative hypothesis: true mean of group=1 minus mean of group=2 is not equal to 0
sample estimates:
mean of group=1 minus mean of group=2
                                -1.58

> independence_test(extra ~ group, data = sleep, distribution = exact())

        Exact General Independence Test

data:  extra by group (1, 2)
Z = -1.7508, p-value = 0.08145
alternative hypothesis: two.sided

> permTS(extra ~ group, data = sleep, method = "exact.network")

        Exact Permutation Test (network algorithm)
```

```
data:  extra by group
p-value = 0.08145
alternative hypothesis: true mean of group=1 minus mean of group=2 is not equal to 0
sample estimates:
mean of group=1 minus mean of group=2
                                -1.58
```

When $z_i$ is a $k \times 1$ vector, we consider only the `alternative`="two.sided", and reject when $Q = (T-U)'V^{-}(T-U)$ is large, where $V^{-}$ is the generalized inverse of $V$. By the PCLT, $Q$ is asymptotically chi-squared with $k-1$ degrees of freedom. In the **perm** package, when the covariate (represented by $z_i$) is a factor, then $Q$ reduces a weighted sum of the squared means of the scores $c_i$ within each group. When $c_i$ is a rank, this gives the usual Kruskal-Wallis test. For example,

```
> kruskal.test(Ozone ~ Month, data = airquality)

        Kruskal-Wallis rank sum test

data:  Ozone by Month
Kruskal-Wallis chi-squared = 29.2666, df = 4, p-value = 6.901e-06

> airq <- airquality[!is.na(airquality$Ozone), ]
> permKS(rank(Ozone) ~ Month, data = airq)

        K-Sample Asymptotic Permutation Test

data:  rank(Ozone) by Month
Chi Square = 29.2666, df = 4, p-value = 6.901e-06
```

(Note care must be taken when using `rank` with some missing responses, see help for `rank`). If we wanted to take into account the ordering of the months and not rank the Ozone responses, we could do a trend test, which is a test on the correlation that matches results from **coin** and gives very similar results to the asymptotic test for Pearson's product moment correlation coefficient in `cor.test` from the **stats** package:

```
> permTREND(Ozone ~ Month, data = airq)

        Permutation Test using Asymptotic Approximation

data:  Ozone by Month
Z = 1.7643, p-value = 0.07769
alternative hypothesis: true correlation of x and y is not equal to 0
sample estimates:
correlation of x and y
             0.1645193

> library(coin)
> independence_test(Ozone ~ Month, data = airq)
```

```
        Asymptotic General Independence Test

data:  Ozone by Month
Z = 1.7643, p-value = 0.07769
alternative hypothesis: two.sided

> cor.test(airq$Ozone, airq$Month)

        Pearson's product-moment correlation

data:  airq$Ozone and airq$Month
t = 1.7809, df = 114, p-value = 0.0776
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.01834762  0.33673567
sample estimates:
      cor
0.1645193
```

---