Conditional Kendall's tau with perturbation weights

Steven Chiou

Install *tranSurv* package from GitHub using

```
> devtools::install_github("stc04003/tranSurv")
```

Load *tranSurv* and *copula* packages.

```
> library(tranSurv)
> library(copula)
```

Generating correlated data:

```
> set.seed(1)
> rho <- iTau(normalCopula(dim = 2), .5) ## convert kendall's tau to pearson's rho
> u <- rCopula(2000, normalCopula(rho, dim = 2)) ## generate correlated data
> u <- qweibull(u, 2, 1) ## assumes t1 and t2 follows some weibull distribution
> colnames(u) <- c("t1", "t2")
```

This gives a Kendall's tau of 0.5 (between *t1* and *t2*)

```
> cor(u, method = "kendall")
          t1        t2
t1 1.0000000 0.5110835
t2 0.5110835 1.0000000
> kendall(u)
[1] 0.5110835
```

Now apply the truncation

```
> u <- subset(as.data.frame(u), t1 < t2)
> dim(u) ## ~50% truncation rate because t1 and t2 follows the same distribution
[1] 1011    2
> head(u)
          t1        t2
1  0.6038558 0.7992435
2  0.7565791 1.4441495
4  1.2090490 1.2835750
9  1.0044891 1.2819928
15 0.7019466 0.9320237
18 0.2815466 0.4472697
```

Arguments for *wKendall*:

```
> args(wKendall)
function (trun, obs, delta = NULL, weights = NULL)
NULL
```

When there is no perturbation weights, *wKendall* is equivalent to *condKendall*.

```
> attach(u)
> condKendall(t1, t2)$PE
[1] 0.1407727
> wKendall(t1, t2)
[1] 0.1407727
> detach(u)
```

wKendall with perturbation weight, which assumes to be a standard exponential distribution.

```
> with(u, wKendall(t1, t2, NULL, rexp(length(t1))))
[1] 0.1285347
```

Use perturbation weights for standard error estimation:

```
> attach(u)
> set.seed(2)
> sd(replicate(500, wKendall(t1, t2, NULL, rexp(length(t1))))) ## 0.018
[1] 0.01835329
> condKendall(t1, t2)$SE ## 0.208
[1] 0.02084022
> detach(u)
```

Small scale simulation:

```
> do <- function() {
+     u <- rCopula(2000, normalCopula(rho, dim = 2)) ## generate correlated data
+     u <- qweibull(u, 2, 1) ## assumes t1 and t2 follows some weibull distribution
+     colnames(u) <- c("t1", "t2")
+     u <- subset(as.data.frame(u), t1 < t2)
+     out <- with(u, c(sd(replicate(500, wKendall(t1, t2, NULL, rexp(length(t1))))) ,
+                      condKendall(t1, t2)$SE))
+     out
+ }
>
> set.seed(3)
> rowMeans(replicate(100, do()))
[1] 0.01819073 0.02055373
```