Left_and_right_and_interval_censoring_with_survfit.R

Hein

Thu Apr 21 10:54:33 2016

```
library(survival)

# The survfit function of the standard survival package can handle left and right
# censored data as well as interval censored data. See the help of Surv, which
# mentions the following:
# The second approach is to think of each observation as a time interval with
# (-infinity, t) for left censored, (t, infinity) for right censored, (t,t) for
# exact and (t1, t2) for an interval. This is the approach used for type = interval2.
# Infinite values can be represented either by actual infinity (Inf) or NA. The
# second form has proven to be the more useful one.

# This is the small example shown during the lecture, and also written up on
# Blackboard. It has probability of event in (2,3] equal to 2/3, and probability
# of event in (4,5] equal to 1/3. Apparently, survfit assigns the probability mass
# in the middle of the intervals, so the survival function goes from 1 to 1/3
# halfway (2,3], and then the final step of 1/3 to 0 halfway (4,5].
test <- data.frame(start=c(0,1,2,4),stop=c(3,5,4,6),status=c(1,1,1,1))
Surv(test$start, test$stop, type='interval2')
```

```
## [1] [0, 3] [1, 5] [2, 4] [4, 6]
```

```
fit  <- survfit(Surv(start, stop, type='interval2') ~ 1,data=test)
summary(fit)
```

```
## Call: survfit(formula = Surv(start, stop, type = "interval2") ~ 1,
##     data = test)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##   2.5   4.00    2.67    0.333   0.236       0.0834            1
##   4.5   1.33    1.33    0.000     NaN           NA           NA
```

```
# Also the larger data set on breast cancer deteration discussed in
# Klein & Moeschberger and the lecture can be fitted with this.
library(KMsurv)
data(bcdeter)
bcdeter1 <- subset(bcdeter, treat==1)
head(bcdeter)
```

```
##   lower upper treat
## 1     0     5     1
## 2     0     7     1
## 3     0     8     1
## 4     4    11     1
## 5     5    11     1
## 6     5    12     1
```

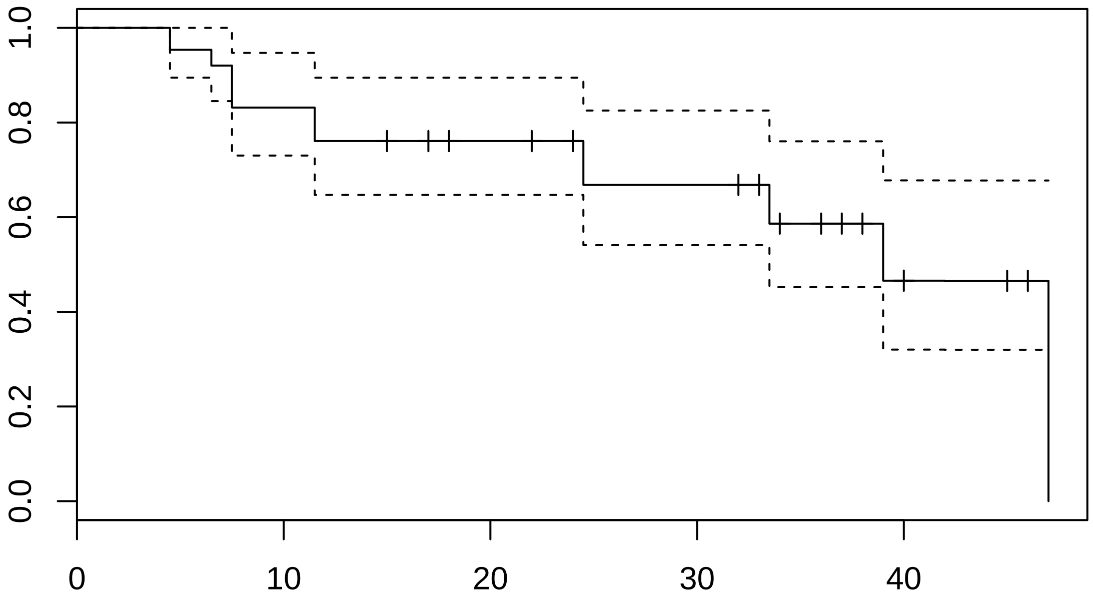
```
Surv(bcdeter$lower, bcdeter$upper, type='interval2')
```

```
##  [1] [ 0,  5] [ 0,  7] [ 0,  8] [ 4, 11] [ 5, 11] [ 5, 12] [ 6, 10]
##  [8] [ 7, 14] [ 7, 16] [11, 15] [11, 18] [17, 25] [17, 25] [18, 26]
## [15] [19, 35] [25, 37] [26, 40] [27, 34] [36, 44] [36, 48] [37, 44]
## [22] [ 0,  5] [ 0, 22] [ 4,  8] [ 4,  9] [ 5,  8] [ 8, 12] [ 8, 21]
## [29] [10, 17] [10, 35] [11, 13] [11, 17] [11, 20] [12, 20] [13, 39]
## [36] [14, 17] [14, 19] [15, 22] [16, 20] [16, 24] [16, 24] [16, 60]
## [43] [17, 23] [17, 26] [17, 27] [18, 24] [18, 25] [19, 32] [22, 32]
## [50] [24, 30] [24, 31] [30, 34] [30, 36] [33, 40] 34       [35, 39]
## [57] [44, 48] 48       15+      17+      18+      22+      24+
## [64] 24+      32+      33+      34+      36+      36+      37+
## [71] 37+      37+      38+      40+      45+      46+      46+
## [78] 46+      46+      46+      46+      46+      46+      11+
## [85] 11+      13+      13+      13+      21+      23+      31+
## [92] 32+      34+      34+      35+
```

```
fit  <- survfit(Surv(lower, upper, type='interval2') ~ 1, data=bcdeter1)
summary(fit)
```

```
## Call: survfit(formula = Surv(lower, upper, type = "interval2") ~ 1,
##     data = bcdeter1)
##
##  time n.risk  n.event survival std.err lower 95% CI upper 95% CI
##   4.5 46.000 2.13e+00    0.954  0.0310        0.895        1.000
##   6.5 43.868 1.54e+00    0.920  0.0399        0.845        1.000
##   7.5 42.332 4.08e+00    0.832  0.0552        0.730        0.947
##  11.5 38.255 3.25e+00    0.761  0.0629        0.647        0.895
##  17.5 33.000 9.07e-04    0.761  0.0629        0.647        0.895
##  24.5 28.999 3.53e+00    0.668  0.0720        0.541        0.825
##  25.5 25.469 3.11e-08    0.668  0.0720        0.541        0.825
##  33.5 23.469 2.87e+00    0.586  0.0777        0.452        0.760
##  34.5 19.596 1.85e-10    0.586  0.0777        0.452        0.760
##  36.5 17.596 1.43e-04    0.586  0.0777        0.452        0.760
##  39.0 13.596 2.79e+00    0.466  0.0891        0.320        0.678
##  42.0  9.801 6.71e-03    0.466  0.0891        0.320        0.678
##  47.0  0.794 7.94e-01    0.000     NaN           NA           NA
```

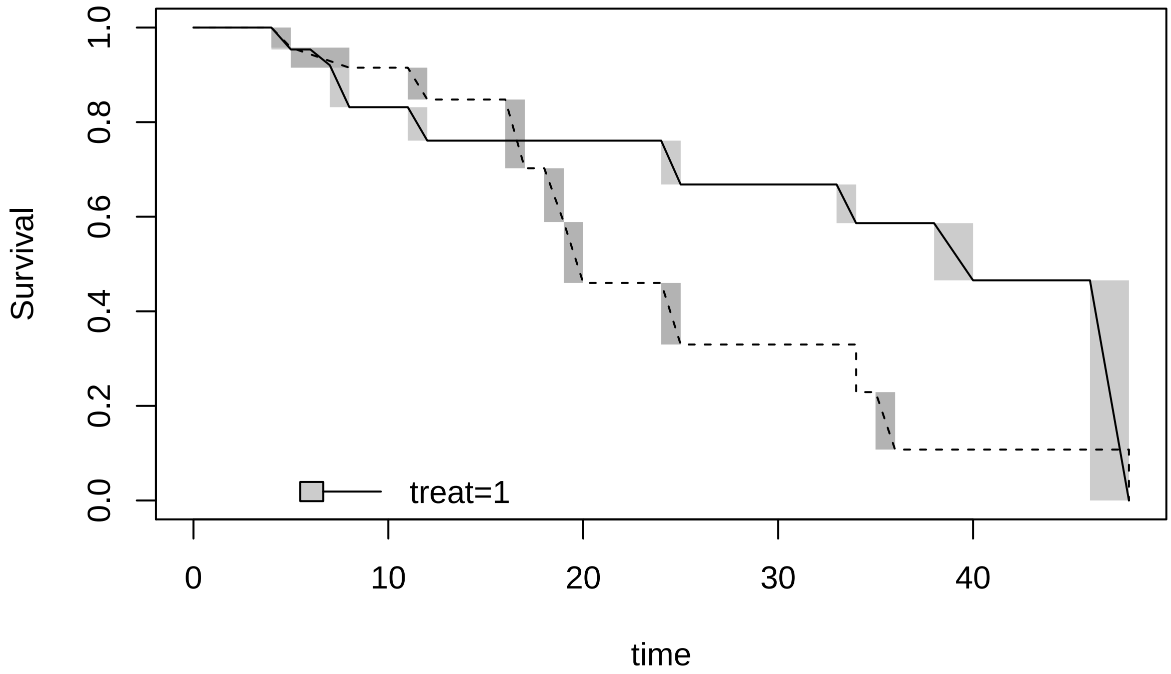
```
plot(fit)
```



```
# A nice package for non-parametric estimation of interval censorerd data is the
# interval package. Here is a similar fit, where the Turnbull intervals are nicely
# shown.
library(interval)
fit1 <- icfit(Surv(lower, upper, type = "interval2") ~ treat, data = bcdeter)
summary(fit1)
```

```
## treat=1:
##   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
## treat=2:
##    Interval Probability
## 1     (4,5]      0.0424
## 2     (5,8]      0.0424
## 3   (11,12]      0.0673
## 4   (16,17]      0.1453
## 5   (18,19]      0.1138
## 6   (19,20]      0.1288
## 7   (24,25]      0.1302
## 8   [34,34]       0.1007
## 9   (35,36]       0.1215
## 10  [48,48]       0.1076
```

```
plot(fit1)
```



```
# Finally, the left and right censoring example of yesterday
# Results are not exactly the same, because of the way that survfit puts
# the time points of interval censored observations in the middle of the
# (Turnbull) intervals
mar<-matrix(c(10:19,4,12,19,24,20,13,3,1,0,4,0,0,2,15,24,18,14,6,0,0,0,0,0,1,2,3,2,3,1,0),ncol=4)
colnames(mar)<-c("Age","N.ExactOb", "N.YetToSmoke","N.StartedToSmoke");
mar<-data.frame(mar)
mar
```

```
##     Age N.ExactOb N.YetToSmoke N.StartedToSmoke
## 1    10         4            0                0
## 2    11        12            0                0
## 3    12        19            2                0
## 4    13        24           15                1
## 5    14        20           24                2
## 6    15        13           18                3
## 7    16         3           14                2
## 8    17         1            6                3
## 9    18         0            0                1
## 10   19         4            0                0
```

```
# In order to use this second approach, we have to make separate lines (with weights)
# to represent all possibilities.
# First the first column of exact observations. They will be represented by the
# intervals (t,t).
marexact <- data.frame(lower=10:19, upper=10:19, n=mar$N.ExactOb)
marexact
```

```
##    lower upper  n
## 1     10    10  4
## 2     11    11 12
## 3     12    12 19
## 4     13    13 24
## 5     14    14 20
## 6     15    15 13
## 7     16    16  3
## 8     17    17  1
## 9     18    18  0
## 10    19    19  4
```

```
# The left censored observations will be represented by (-Inf, t]
marleft <- data.frame(lower=-Inf, upper=10:19, n=mar$N.StartedToSmoke)
marleft
```

```
##    lower upper n
## 1   -Inf    10 0
## 2   -Inf    11 0
## 3   -Inf    12 0
## 4   -Inf    13 1
## 5   -Inf    14 2
## 6   -Inf    15 3
## 7   -Inf    16 2
## 8   -Inf    17 3
## 9   -Inf    18 1
## 10  -Inf    19 0
```

```
# And finally the right censored observations are represented by (t, Inf]
marright <- data.frame(lower=10:19, upper=Inf, n=mar$N.YetToSmoke)
marright
```

```
##    lower upper  n
## 1     10   Inf  0
## 2     11   Inf  0
## 3     12   Inf  2
## 4     13   Inf 15
## 5     14   Inf 24
## 6     15   Inf 18
## 7     16   Inf 14
## 8     17   Inf  6
## 9     18   Inf  0
## 10    19   Inf  0
```

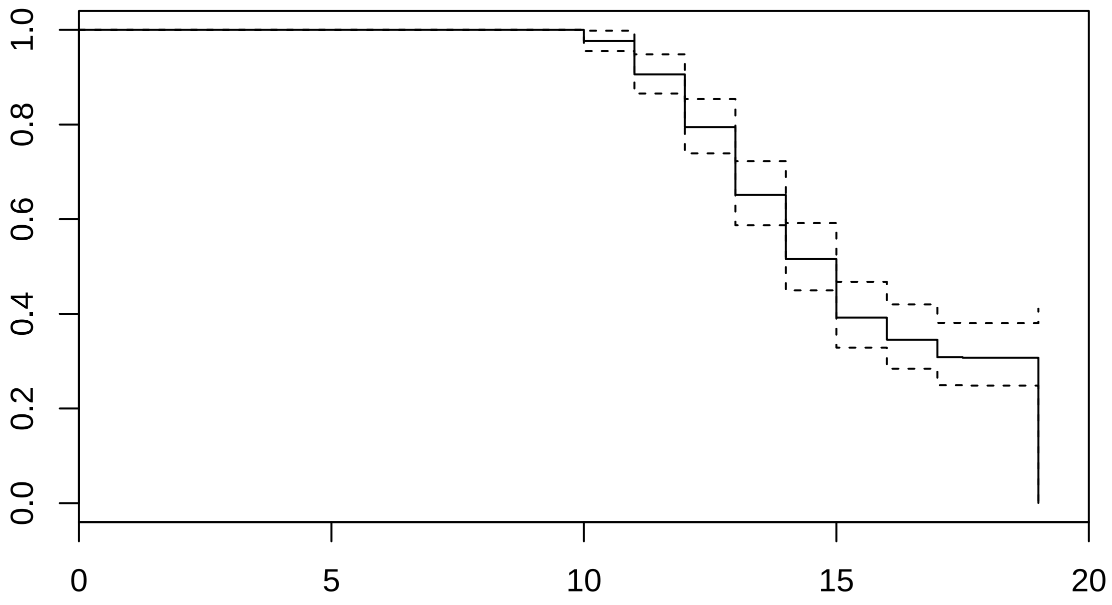
```r
# Gather everything and throw away the empty cells
mardata <- rbind(marexact, marleft, marright)
mardata <- subset(mardata, n>0)
mardata
```

```
##    lower upper  n
## 1     10    10  4
## 2     11    11 12
## 3     12    12 19
## 4     13    13 24
## 5     14    14 20
## 6     15    15 13
## 7     16    16  3
## 8     17    17  1
## 10    19    19  4
## 14  -Inf    13  1
## 15  -Inf    14  2
## 16  -Inf    15  3
## 17  -Inf    16  2
## 18  -Inf    17  3
## 19  -Inf    18  1
## 23    12   Inf  2
## 24    13   Inf 15
## 25    14   Inf 24
## 26    15   Inf 18
## 27    16   Inf 14
## 28    17   Inf  6
```

```r
# And finally, fit with survfit
fit  <- survfit(Surv(lower, upper, type='interval2') ~ 1, data=mardata, weights=n)
summary(fit)
```

```
## Call: survfit(formula = Surv(lower, upper, type = "interval2") ~ 1,
##     data = mardata, weights = n)
##
##  time   n.risk n.event survival std.err lower 95% CI upper 95% CI
##  10.0 191.0000  4.4880 0.976503 0.01096     9.55e-01        0.998
##  11.0 186.5120 13.4640 0.906010 0.02111     8.66e-01        0.948
##  12.0 173.0480 21.3180 0.794398 0.02924     7.39e-01        0.854
##  13.0 151.7300 27.3322 0.651297 0.03448     5.87e-01        0.723
##  14.0 124.3978 25.8894 0.515751 0.03616     4.50e-01        0.592
##  15.0  98.5084 23.6147 0.392114 0.03533     3.29e-01        0.468
##  16.0  74.8937  8.9272 0.345375 0.03441     2.84e-01        0.420
##  17.0  65.9665  7.1124 0.308137 0.03341     2.49e-01        0.381
##  17.5  58.8541  0.1508 0.307347 0.03339     2.48e-01        0.380
##  19.0  58.7033 58.6221 0.000426 0.00149     4.41e-07        0.411
##   Inf   0.0813  0.0813 0.000000     NaN           NA           NA
```

```r
plot(fit, xlim=c(0,20))
```


1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20