M215_HW3

```
library(survival)
library(KMsurv)
data(bmt)
help(bmt)
```

Klein and Moeschberger 4.2

- Using the data reported in section 1.3, find the quantities specified below for the AML low risk and AML high risk groups. Note that most of these quantities are worked out in detail in Example 4.2 and its continuations for the ALL group.

a) Estimate the survival functions and their standard errors for the AML low risk and AML high risk groups.

```
# AML Low Risk
fit.low <- survfit(Surv(bmt$t1[bmt$group == 2], bmt$d1[bmt$group == 2]) ~ 1,
conf.type = 'none')
summary(fit.low)

## Call: survfit(formula = Surv(bmt$t1[bmt$group == 2], bmt$d1[bmt$group ==
##     2]) ~ 1, conf.type = "none")
##
##  time n.risk n.event survival std.err
##    10     54       1    0.981  0.0183
##    35     53       1    0.963  0.0257
##    48     52       1    0.944  0.0312
##    53     51       1    0.926  0.0356
##    79     50       1    0.907  0.0394
##    80     49       1    0.889  0.0428
##   105     48       1    0.870  0.0457
##   222     47       1    0.852  0.0483
##   288     46       1    0.833  0.0507
##   390     45       1    0.815  0.0529
##   393     44       1    0.796  0.0548
##   414     43       1    0.778  0.0566
##   431     42       1    0.759  0.0582
##   481     41       1    0.741  0.0596
##   522     40       1    0.722  0.0610
##   583     39       1    0.704  0.0621
##   641     38       1    0.685  0.0632
##   653     37       1    0.667  0.0642
##   704     36       1    0.648  0.0650
##  1063     29       1    0.626  0.0665
##  1074     28       1    0.603  0.0678
##  1156     27       1    0.581  0.0688
##  2204      6       1    0.484  0.1054
```

```
# AML High Risk
fit.high <- survfit(Surv(bmt$t1[bmt$group == 3], bmt$d1[bmt$group == 3]) ~ 1,
conf.type = 'none')
summary(fit.high)
```

```
## Call: survfit(formula = Surv(bmt$t1[bmt$group == 3], bmt$d1[bmt$group ==
##     3]) ~ 1, conf.type = "none")
##
##  time n.risk n.event survival std.err
##     2     45       1    0.978  0.0220
##    16     44       1    0.956  0.0307
##    62     43       1    0.933  0.0372
##    63     42       1    0.911  0.0424
##    73     41       1    0.889  0.0468
##    74     40       1    0.867  0.0507
##    80     39       1    0.844  0.0540
##    93     38       1    0.822  0.0570
##    97     37       1    0.800  0.0596
##   105     36       2    0.756  0.0641
##   121     34       1    0.733  0.0659
##   122     33       1    0.711  0.0676
##   128     32       1    0.689  0.0690
##   129     31       1    0.667  0.0703
##   153     30       1    0.644  0.0714
##   162     29       1    0.622  0.0723
##   164     28       1    0.600  0.0730
##   168     27       1    0.578  0.0736
##   183     26       1    0.556  0.0741
##   195     25       1    0.533  0.0744
##   248     24       1    0.511  0.0745
##   265     23       1    0.489  0.0745
##   318     22       1    0.467  0.0744
##   341     21       1    0.444  0.0741
##   363     20       1    0.422  0.0736
##   392     19       1    0.400  0.0730
##   469     18       1    0.378  0.0723
##   491     17       1    0.356  0.0714
##   515     16       1    0.333  0.0703
##   547     15       1    0.311  0.0690
##   677     14       1    0.289  0.0676
##   732     13       1    0.267  0.0659
##  1298      9       1    0.237  0.0649
```

b) Estimate the cumulative hazard rates and their standard errors for the AML low risk and AML high risk groups.

```
# Nelson-Aalen Estimate of H(t)

# AML Low Risk
h_low <- fit.low$n.event/fit.low$n.risk
H.na_low <- cumsum(h_low) #Nelson-Aalen estimates
```

```r
s_low <- fit.low$n.event/fit.low$n.risk^2
V.na_low <- cumsum(s_low) #Variance estimate for the N-A estimator
NAest_low <- cbind(fit.low$time,H.na_low, sqrt(V.na_low))
colnames(NAest_low) <- c("time","NAEst.","Std. Err")
NAest_low <- as.data.frame(NAest_low)
NAest_low
```

```
##    time     NAEst.   Std. Err
## 1    10 0.01851852 0.01851852
## 2    35 0.03738644 0.02643736
## 3    48 0.05661721 0.03269184
## 4    53 0.07622506 0.03812118
## 5    79 0.09622506 0.04304909
## 6    80 0.11663322 0.04764155
## 7   105 0.13746655 0.05199755
## 8   222 0.15874315 0.05618219
## 9   288 0.18048228 0.06024142
## 10  390 0.20270450 0.06420947
## 11  393 0.22543177 0.06811303
## 12  414 0.24868759 0.07197373
## 13  431 0.27249711 0.07580970
## 14  481 0.29688735 0.07963664
## 15  522 0.32188735 0.08346853
## 16  583 0.34752838 0.08731814
## 17  641 0.37384417 0.09119747
## 18  653 0.40087120 0.09511802
## 19  704 0.42864897 0.09909108
## 20  847 0.42864897 0.09909108
## 21  848 0.42864897 0.09909108
## 22  860 0.42864897 0.09909108
## 23  932 0.42864897 0.09909108
## 24  957 0.42864897 0.09909108
## 25 1030 0.42864897 0.09909108
## 26 1063 0.46313173 0.10491951
## 27 1074 0.49884602 0.11083147
## 28 1156 0.53588306 0.11685613
## 29 1258 0.53588306 0.11685613
## 30 1324 0.53588306 0.11685613
## 31 1356 0.53588306 0.11685613
## 32 1363 0.53588306 0.11685613
## 33 1384 0.53588306 0.11685613
## 34 1447 0.53588306 0.11685613
## 35 1470 0.53588306 0.11685613
## 36 1499 0.53588306 0.11685613
## 37 1527 0.53588306 0.11685613
## 38 1535 0.53588306 0.11685613
## 39 1562 0.53588306 0.11685613
## 40 1568 0.53588306 0.11685613
## 41 1674 0.53588306 0.11685613
## 42 1709 0.53588306 0.11685613
## 43 1799 0.53588306 0.11685613
## 44 1829 0.53588306 0.11685613
## 45 1843 0.53588306 0.11685613
## 46 1850 0.53588306 0.11685613
## 47 1857 0.53588306 0.11685613
## 48 1870 0.53588306 0.11685613
## 49 2204 0.70254972 0.20355130
## 50 2218 0.70254972 0.20355130
## 51 2246 0.70254972 0.20355130
## 52 2409 0.70254972 0.20355130
## 53 2506 0.70254972 0.20355130
## 54 2569 0.70254972 0.20355130
```

```
# AML High Risk
h_high <- fit.high$n.event/fit.high$n.risk
H.na_high <- cumsum(h_high) #Nelson-Aalen estimates
s_high <- fit.high$n.event/fit.high$n.risk^2
V.na_high <- cumsum(s_high) #Variance estimate for the N-A estimator
NAest_high <- cbind(H.na_high, sqrt(V.na_high))
colnames(NAest_high) <- c("NA-Est.","Std. Err")
NAest_high
```

```
##          NA-Est.   Std. Err
##  [1,] 0.02222222 0.02222222
##  [2,] 0.04494949 0.03178610
##  [3,] 0.06820531 0.03938514
##  [4,] 0.09201483 0.04602263
##  [5,] 0.11640508 0.05208614
##  [6,] 0.14140508 0.05777514
##  [7,] 0.16704610 0.06320940
##  [8,] 0.19336189 0.06846860
##  [9,] 0.22038892 0.07360985
## [10,] 0.27594447 0.08343632
## [11,] 0.30535624 0.08846848
## [12,] 0.33565927 0.09351441
## [13,] 0.36690927 0.09859771
## [14,] 0.39916733 0.10374049
## [15,] 0.43250067 0.10896422
## [16,] 0.46698343 0.11429025
## [17,] 0.50269771 0.11974044
## [18,] 0.53973475 0.12533760
## [19,] 0.57819629 0.13110608
## [20,] 0.61819629 0.13707226
## [21,] 0.65986295 0.14326519
## [22,] 0.70334121 0.14971732
## [23,] 0.74879576 0.15646530
## [24,] 0.79641481 0.16355110
## [25,] 0.84641481 0.17102328
## [26,] 0.89904639 0.17893867
## [27,] 0.95460194 0.18736453
## [28,] 1.01342547 0.19638145
## [29,] 1.07592547 0.20608718
## [30,] 1.14259214 0.21660187
## [31,] 1.21402071 0.22807545
## [32,] 1.29094379 0.24069809
## [33,] 1.29094379 0.24069809
## [34,] 1.29094379 0.24069809
## [35,] 1.29094379 0.24069809
## [36,] 1.40205490 0.26510611
## [37,] 1.40205490 0.26510611
## [38,] 1.40205490 0.26510611
## [39,] 1.40205490 0.26510611
## [40,] 1.40205490 0.26510611
## [41,] 1.40205490 0.26510611
## [42,] 1.40205490 0.26510611
## [43,] 1.40205490 0.26510611
## [44,] 1.40205490 0.26510611
```

(d) Estimate the mean time to death and find 95% confidence intervals for the mean survival time for both the AML low risk and AML high risk groups.

```
t_l <- 2569 #Looking at restricted mean from [0, 2569]
print(fit.low, print.rmean = TRUE, rmean = t_l) #Low
```

```
## Call: survfit(formula = Surv(bmt$t1[bmt$group == 2], bmt$d1[bmt$group ==
##     2]) ~ 1, conf.type = "none")
##
##          n     events     *rmean *se(rmean)     median
##         54         23       1645        147       2204
##     * restricted mean with upper limit =  2569
```

- The estimated mean time to death is 1645 days. With the SE = 147, we get the 95% CI for mean survival time for AML low risk group is (1645-1.96 * 147, 1645+1.96*147) = (1357, 1933).

```
t_h <- 2640
print(fit.high, print.rmean = TRUE, rmean = t_h) #high
```

```
## Call: survfit(formula = Surv(bmt$t1[bmt$group == 3], bmt$d1[bmt$group ==
##     3]) ~ 1, conf.type = "none")
##
##          n     events     *rmean *se(rmean)     median
##         45         34        834        155        265
##     * restricted mean with upper limit =  2640
```

- The estimated mean time to deat is 834 days. With the SE = 155, we can get the 95% CI for mean survival time for AML high risk group is (834-1.96*155, 834+1.96*155) = (530, 1138).

(e) Work out estimates of the median time to death and find 95% confidence intervals for the median survival time for both the AML low risk and AML high risk groups using the linear, log-transformed, and arcsine formulas.

- Low Risk
- Linear median time: 2204
- Log median time:2204
- Arcsine median time: 2204

```
#Low risk
data_directory <- "/Users/huiyuhu/Desktop/Study/UCLA_Biostat/M215"
source('/Users/huiyuhu/Desktop/Study/UCLA_Biostat/M215/bc_median_ci.R')
#linear
bc.median.ci(fit.low, table = F)

## $median
## [1] 2204
##
## $lower
## [1] 1063
##
## $upper
## [1] NA
##
## $type
## [1] "linear"
##
## $alpha
## [1] 0.05

#log-transformed
bc.median.ci(fit.low, type = "log", table = F)

## $median
## [1] 2204
##
## $lower
## [1] 1063
##
## $upper
## [1] NA
##
## $type
## [1] "log"
##
## $alpha
## [1] 0.05

#arcsine
bc.median.ci(fit.low, type = "asin", table = F)

## $median
## [1] 2204
##
## $lower
## [1] 1063
##
## $upper
## [1] NA
##
## $type
## [1] "asin"
##
## $alpha
## [1] 0.05
```

- High Risk
- Linear median time: 265
- Log median time:265
- Arcsine median time: 265

```
#High risk
data_directory <- "/Users/huiyuhu/Desktop/Study/UCLA_Biostat/M215"
source('/Users/huiyuhu/Desktop/Study/UCLA_Biostat/M215/bc_median_ci.R')
#linear
bc.median.ci(fit.high, table = F)
```

```
## $median
## [1] 265
##
## $lower
## [1] 162
##
## $upper
## [1] 469
##
## $type
## [1] "linear"
##
## $alpha
## [1] 0.05
```

```
#log-transformed
bc.median.ci(fit.high, type = "log", table = F)
```

```
## $median
## [1] 265
##
## $lower
## [1] 153
##
## $upper
## [1] 469
##
## $type
## [1] "log"
```

```
##
## $alpha
## [1] 0.05
```

```
#arcsine
bc.median.ci(fit.high, type = "asin", table = F)
```

```
## $median
## [1] 265
##
## $lower
## [1] 162
##
## $upper
## [1] 469
##
## $type
## [1] "asin"
##
## $alpha
## [1] 0.05
```

(f) Find 95% confidence intervals for the survival functions at 300 days post-transplant for both the AML low risk and AML high risk groups using the log- and arcsine-transformed formulas.

- Log-log
- Low Risk: S(300) = 0.833 [using time = 288 to estimate time = 300], C.I. = [0.704, 0.910]
- High Risk: S(300) = 0.489 [using time = 268 to estimate time = 300], C.I. = [0.337, 0.624]

```
# log-log - Low
fit.low_log <- survfit(Surv(bmt$t1[bmt$group == 2], bmt$d1[bmt$group == 2]) ~
1, conf.type = 'log-log')
summary(fit.low_log)
```

```
## Call: survfit(formula = Surv(bmt$t1[bmt$group == 2], bmt$d1[bmt$group ==
##     2]) ~ 1, conf.type = "log-log")
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##    10     54       1    0.981  0.0183        0.876        0.997
##    35     53       1    0.963  0.0257        0.860        0.991
##    48     52       1    0.944  0.0312        0.838        0.982
##    53     51       1    0.926  0.0356        0.815        0.972
##    79     50       1    0.907  0.0394        0.792        0.960
##    80     49       1    0.889  0.0428        0.769        0.948
##   105     48       1    0.870  0.0457        0.747        0.936
##   222     47       1    0.852  0.0483        0.726        0.923
##   288     46       1    0.833  0.0507        0.704        0.910
##   390     45       1    0.815  0.0529        0.683        0.896
##   393     44       1    0.796  0.0548        0.662        0.882
##   414     43       1    0.778  0.0566        0.642        0.867
##   431     42       1    0.759  0.0582        0.622        0.852
##   481     41       1    0.741  0.0596        0.602        0.837
##   522     40       1    0.722  0.0610        0.582        0.822
##   583     39       1    0.704  0.0621        0.563        0.807
##   641     38       1    0.685  0.0632        0.543        0.791
##   653     37       1    0.667  0.0642        0.524        0.775
##   704     36       1    0.648  0.0650        0.505        0.759
##  1063     29       1    0.626  0.0665        0.481        0.740
##  1074     28       1    0.603  0.0678        0.458        0.721
##  1156     27       1    0.581  0.0688        0.435        0.702
##  2204      6       1    0.484  0.1054        0.271        0.669
```

```
# log-log - High
fit.high_log <- survfit(Surv(bmt$t1[bmt$group == 3], bmt$d1[bmt$group == 3])
~ 1, conf.type = 'log-log')
summary(fit.high_log)
```

```
## Call: survfit(formula = Surv(bmt$t1[bmt$group == 3], bmt$d1[bmt$group ==
##     3]) ~ 1, conf.type = "log-log")
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     2     45       1    0.978  0.0220        0.853        0.997
##    16     44       1    0.956  0.0307        0.834        0.989
##    62     43       1    0.933  0.0372        0.807        0.978
##    63     42       1    0.911  0.0424        0.780        0.966
##    73     41       1    0.889  0.0468        0.753        0.952
##    74     40       1    0.867  0.0507        0.727        0.938
##    80     39       1    0.844  0.0540        0.701        0.923
##    93     38       1    0.822  0.0570        0.676        0.907
##    97     37       1    0.800  0.0596        0.651        0.891
##   105     36       2    0.756  0.0641        0.602        0.856
##   121     34       1    0.733  0.0659        0.578        0.839
##   122     33       1    0.711  0.0676        0.555        0.821
##   128     32       1    0.689  0.0690        0.532        0.802
##   129     31       1    0.667  0.0703        0.509        0.784
##   153     30       1    0.644  0.0714        0.487        0.765
##   162     29       1    0.622  0.0723        0.465        0.746
##   164     28       1    0.600  0.0730        0.443        0.726
##   168     27       1    0.578  0.0736        0.421        0.706
##   183     26       1    0.556  0.0741        0.400        0.686
##   195     25       1    0.533  0.0744        0.379        0.666
##   248     24       1    0.511  0.0745        0.358        0.645
##   265     23       1    0.489  0.0745        0.337        0.624
##   318     22       1    0.467  0.0744        0.317        0.603
##   341     21       1    0.444  0.0741        0.297        0.582
##   363     20       1    0.422  0.0736        0.278        0.560
##   392     19       1    0.400  0.0730        0.258        0.538
##   469     18       1    0.378  0.0723        0.239        0.516
##    491      17        1    0.356  0.0714         0.220         0.493
##    515      16        1    0.333  0.0703         0.202         0.470
##    547      15        1    0.311  0.0690         0.184         0.447
##    677      14        1    0.289  0.0676         0.166         0.424
##    732      13        1    0.267  0.0659         0.149         0.400
##   1298       9        1    0.237  0.0649         0.124         0.371
```

- Arcsine Low Risk: S(300) = 0.833 [using time = 288 to estimate time = 300], C.I. = [0.723, 0.920] High Risk: S(300) = 0.489 [using time = 268 to estimate time = 300], C.I. = [0.345, 0.633]

```
# Arcsine - Low
source('/Users/huiyuhu/Desktop/Study/UCLA_Biostat/M215/arcsin_ci.R')
fit.low <- survfit(Surv(bmt$t1[bmt$group == 2], bmt$d1[bmt$group == 2]) ~ 1,
conf.type = 'none')
res.low <- arcsin.ci(fit.low, alpha = 0.05)
res.low
```

```
##    time  surv sigma lower upper
## 1    10 0.981 0.019 0.929 1.000
## 2    35 0.963 0.027 0.897 0.996
## 3    48 0.944 0.033 0.868 0.989
## 4    53 0.926 0.038 0.842 0.980
## 5    79 0.907 0.043 0.817 0.969
## 6    80 0.889 0.048 0.792 0.958
## 7   105 0.870 0.053 0.769 0.946
## 8   222 0.852 0.057 0.746 0.933
## 9   288 0.833 0.061 0.723 0.920
## 10  390 0.815 0.065 0.701 0.906
## 11  393 0.796 0.069 0.680 0.892
## 12  414 0.778 0.073 0.658 0.878
## 13  431 0.759 0.077 0.637 0.863
## 14  481 0.741 0.081 0.617 0.848
## 15  522 0.722 0.084 0.596 0.832
## 16  583 0.704 0.088 0.576 0.817
## 17  641 0.685 0.092 0.556 0.801
## 18  653 0.667 0.096 0.537 0.785
## 19  704 0.648 0.100 0.517 0.769
## 20  847 0.648 0.100 0.517 0.769
## 21  848 0.648 0.100 0.517 0.769
## 22  860 0.648 0.100 0.517 0.769
## 23  932 0.648 0.100 0.517 0.769
## 24  957 0.648 0.100 0.517 0.769
## 25 1030 0.648 0.100 0.517 0.769
## 26 1063 0.626 0.106 0.493 0.750
## 27 1074 0.603 0.112 0.468 0.731
## 28 1156 0.581 0.118 0.445 0.711
## 29 1258 0.581 0.118 0.445 0.711
## 30 1324 0.581 0.118 0.445 0.711
## 31 1356 0.581 0.118 0.445 0.711
## 32 1363 0.581 0.118 0.445 0.711
## 33 1384 0.581 0.118 0.445 0.711
## 34 1447 0.581 0.118 0.445 0.711
## 35 1470 0.581 0.118 0.445 0.711
## 36 1499 0.581 0.118 0.445 0.711
## 37 1527 0.581 0.118 0.445 0.711
## 38 1535 0.581 0.118 0.445 0.711
## 39 1562 0.581 0.118 0.445 0.711
## 40 1568 0.581 0.118 0.445 0.711
## 41 1674 0.581 0.118 0.445 0.711
## 42 1709 0.581 0.118 0.445 0.711
## 43 1799 0.581 0.118 0.445 0.711
## 44 1829 0.581 0.118 0.445 0.711
## 45 1843 0.581 0.118 0.445 0.711
## 46 1850 0.581 0.118 0.445 0.711
## 47 1857 0.581 0.118 0.445 0.711
## 48 1870 0.581 0.118 0.445 0.711
## 49 2204 0.484 0.218 0.285 0.686
## 50 2218 0.484 0.218 0.285 0.686
## 51 2246 0.484 0.218 0.285 0.686
## 52 2409 0.484 0.218 0.285 0.686
## 53 2506 0.484 0.218 0.285 0.686
## 54 2569 0.484 0.218 0.285 0.686
```

```
# Arcsine - High
fit.high <- survfit(Surv(bmt$t1[bmt$group == 3], bmt$d1[bmt$group == 3]) ~ 1,
conf.type = 'none')
res.high <- arcsin.ci(fit.high, alpha = 0.05)
res.high
```

```
##    time  surv sigma lower upper
## 1     2 0.978 0.022 0.915 1.000
## 2    16 0.956 0.032 0.877 0.996
## 3    62 0.933 0.040 0.843 0.987
## 4    63 0.911 0.047 0.812 0.976
## 5    73 0.889 0.053 0.782 0.963
## 6    74 0.867 0.058 0.753 0.949
## 7    80 0.844 0.064 0.725 0.934
## 8    93 0.822 0.069 0.698 0.919
## 9    97 0.800 0.075 0.672 0.902
## 10  105 0.756 0.085 0.621 0.869
## 11  121 0.733 0.090 0.596 0.851
## 12  122 0.711 0.095 0.572 0.833
## 13  128 0.689 0.100 0.548 0.814
## 14  129 0.667 0.105 0.524 0.795
## 15  153 0.644 0.111 0.500 0.776
## 16  162 0.622 0.116 0.477 0.757
## 17  164 0.600 0.122 0.455 0.737
## 18  168 0.578 0.127 0.432 0.717
## 19  183 0.556 0.133 0.410 0.696
## 20  195 0.533 0.139 0.388 0.676
## 21  248 0.511 0.146 0.367 0.655
## 22  265 0.489 0.152 0.345 0.633
## 23  318 0.467 0.159 0.324 0.612
## 24  341 0.444 0.167 0.304 0.590
## 25  363 0.422 0.174 0.283 0.568
## 26  392 0.400 0.183 0.263 0.545
## 27  469 0.378 0.191 0.243 0.523
## 28  491 0.356 0.201 0.224 0.500
## 29  515 0.333 0.211 0.205 0.476
## 30  547 0.311 0.222 0.186 0.452
## 31  677 0.289 0.234 0.167 0.428
## 32  732 0.267 0.247 0.149 0.404
## 33  845 0.267 0.247 0.149 0.404
## 34 1136 0.267 0.247 0.149 0.404
## 35 1238 0.267 0.247 0.149 0.404
## 36 1298 0.237 0.274 0.123 0.374
## 37 1345 0.237 0.274 0.123 0.374
## 38 1631 0.237 0.274 0.123 0.374
## 39 2024 0.237 0.274 0.123 0.374
## 40 2133 0.237 0.274 0.123 0.374
## 41 2140 0.237 0.274 0.123 0.374
## 42 2252 0.237 0.274 0.123 0.374
## 43 2430 0.237 0.274 0.123 0.374
## 44 2640 0.237 0.274 0.123 0.374
```

- Likelihood ratio method. Low Risk: S(300) C.I. = [0.72, 0.916] High Risk: S(300) C.I. = [0.347, 0.632]

```
source('/Users/huiyuhu/Desktop/Study/UCLA_Biostat/M215/LRci.R')
# LR - Low
LRci.surv(fit.low, t = 300)[3]

## $conf.int
##        LL    UL
## [1,] 0.72 0.916

# LR - High
LRci.surv(fit.high, t = 300)[3]

## $conf.int
##         LL    UL
## [1,] 0.347 0.632
```

- Bootstrap method
- (use the bootstrap function by modifying the quatile_boots from GitHub)
- Low risk: [0.741, 0.926]
- High risk: [0.356, 0.644]

```
source('/Users/huiyuhu/Desktop/Study/UCLA_Biostat/M215/quantile_boots_1.R')
t1 <- bmt$t1[bmt$group == 2]
d1 <- bmt$d1[bmt$group == 2]
t2 <- bmt$t1[bmt$group == 3]
```

```
d2 <- bmt$d1[bmt$group == 3]

# Bootstrap - low
quantile_boots(t1, d1)

## $lower
##      2.5%
## 0.7407407
##
## $upper
##     97.5%
## 0.9259259

# Bootstrp - high
quantile_boots(t2, d2)

## $lower
##      2.5%
## 0.3555556
##
## $upper
##     97.5%
## 0.6222222
```

(g) Find 95% EP confidence bands for the survival functions over the range 100–400 days post-transplant for both the AML low risk and AML high risk groups using the linear, log-transformed, and arcsinetransformed formulas.

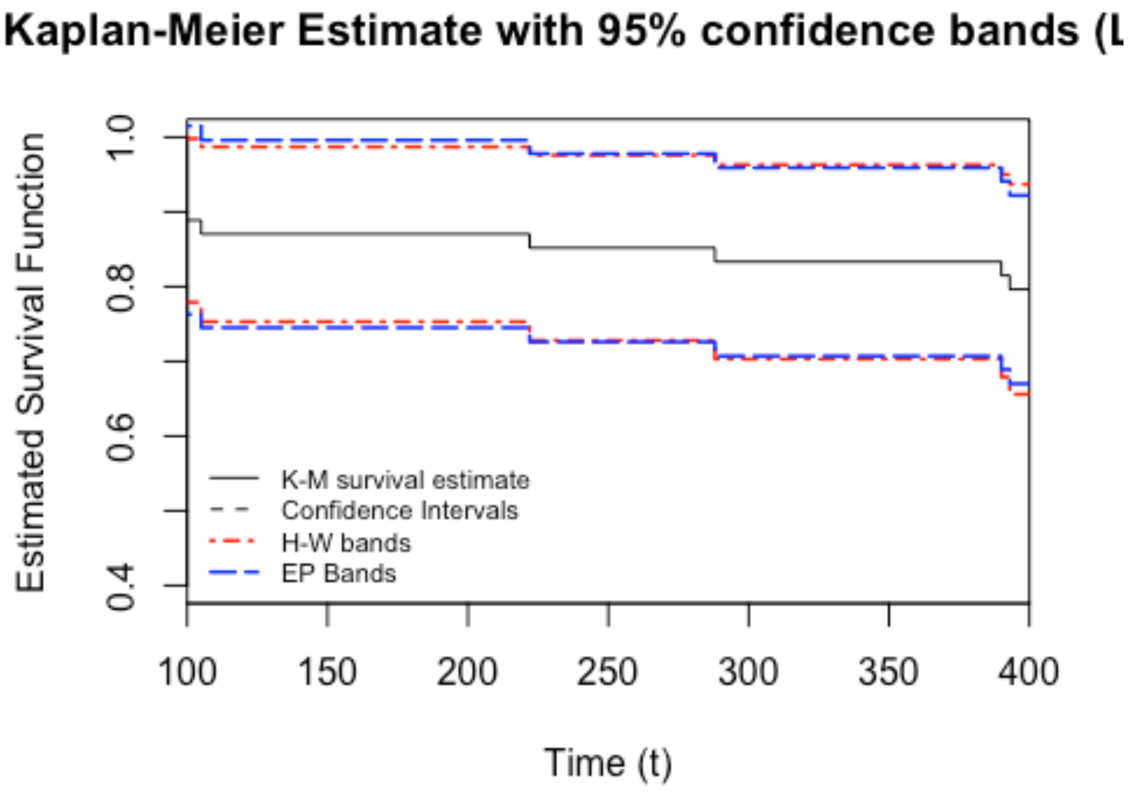
```
source('/Users/huiyuhu/Desktop/Study/UCLA_Biostat/M215/conf_band.R')
# Low
cbands.region(fit.low, tL = 80, tU = 414)

## Find critical regions in Klein and Moeschberger 2nd ed. (Appendix C.3a -
C.4c)

## $aL
## [1] 0.1
##
## $aU
## [1] 0.2

# aL = 0.1; aU = 0.2
c_l <- 2.5602
ep.band1 <- cbands.interval(fit.low, tL = 80, tU = 414, crit.value = c_l,
type = "linear", method = "ep")

## Returning linear-type confidence bands using ep method.

ep.band2 <- cbands.interval(fit.low, tL = 80, tU = 414, crit.value = c_l,
type = "log", method = "ep")

## Returning log-type confidence bands using ep method.
```

```
ep.band3 <- cbands.interval(fit.low, tL = 80, tU = 414, crit.value = c_l,
type = "asin", method = "ep")

## Returning asin-type confidence bands using ep method.

# High
cbands.region(fit.high, tL = 97, tU = 467)

## Find critical regions in Klein and Moeschberger 2nd ed. (Appendix C.3a -
C.4c)

## $aL
## [1] 0.2
##
## $aU
## [1] 0.6

# aL = 0.2; aU = 0.6
c_h <- 2.7666
ep.band10 <- cbands.interval(fit.high, tL = 97, tU = 467, crit.value = c_h,
type = "linear", method = "ep")

## Returning linear-type confidence bands using ep method.

ep.band20 <- cbands.interval(fit.high, tL = 97, tU = 467, crit.value = c_h,
type = "log", method = "ep")

## Returning log-type confidence bands using ep method.

ep.band30 <- cbands.interval(fit.high, tL = 97, tU = 467, crit.value = c_h,
type = "asin", method = "ep")

## Returning asin-type confidence bands using ep method.
```

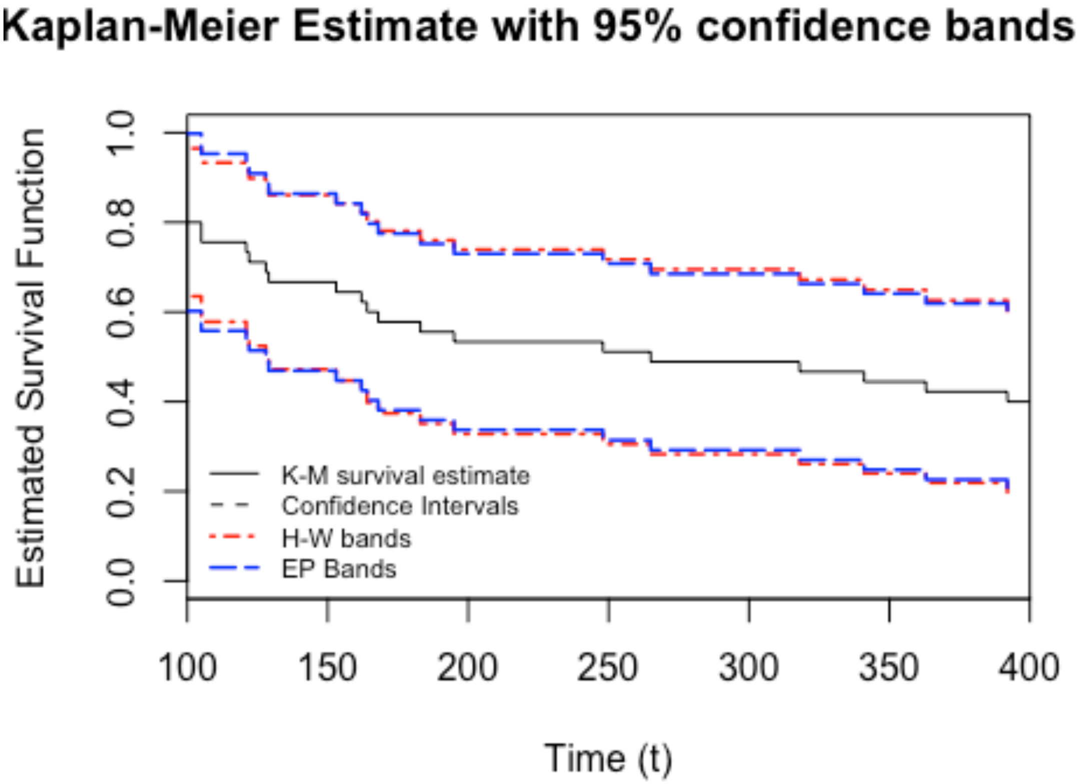
(h) Find 95% HW confidence bands for the survival functions over the range 100–400 days post-transplant for both the AML low risk and AML high risk groups using the linear, log-transformed, and arcsinetransformed formulas.

```
source('/Users/huiyuhu/Desktop/Study/UCLA_Biostat/M215/conf_band.R')
# Low
c0_l <- 0.9247
hw.band1 <- cbands.interval(fit.low, tL = 80, tU = 414, crit.value = c0_l,
type = "linear", method = "hw")

## Returning linear-type confidence bands using hw method.

hw.band2 <- cbands.interval(fit.low, tL = 80, tU = 414, crit.value = c0_l,
type = "log", method = "hw")

## Returning log-type confidence bands using hw method.

hw.band3 <- cbands.interval(fit.low, tL = 80, tU = 414, crit.value = c0_l,
type = "asin", method = "hw")
```

```r
## Returning asin-type confidence bands using hw method.

# High
c0_h <- 1.3191
hw.band10 <- cbands.interval(fit.high, tL = 97, tU = 467, crit.value = c0_h,
type = "linear", method = "hw")

## Returning linear-type confidence bands using hw method.

hw.band20 <- cbands.interval(fit.high, tL = 97, tU = 467, crit.value = c0_l,
type = "log", method = "hw")

## Returning log-type confidence bands using hw method.

hw.band30 <- cbands.interval(fit.high, tL = 97, tU = 467, crit.value = c0_l,
type = "asin", method = "hw")

## Returning asin-type confidence bands using hw method.

#Plotting them - low
plot(fit.low, xlim = c(100, 400), ylim = c(0.4, 1), main="Kaplan-Meier
Estimate with 95% confidence bands (Low)",
xlab="Time (t)", ylab="Estimated Survival Function")
lines(ep.band1$LL ~ ep.band1$t, lty = 4, col = "red", lwd = 1.5, type = 's')
lines(ep.band1$UL ~ ep.band1$t, lty = 4, col = "red", lwd = 1.5, type = 's')
lines(hw.band1$LL ~ hw.band1$t, lty = 5, col = "blue", lwd = 1.5, type = 's')
lines(hw.band1$UL ~ hw.band1$t, lty = 5, col = "blue", lwd = 1.5, type = 's')
legend("bottomleft", legend=
c("K-M survival estimate",
"Confidence Intervals",
"H-W bands", "EP Bands"), lty=c(1, 2, 4,5),
bty = "n", lwd = c(1, 1, 1.5, 1.5), cex = .7,
col = c("black", "black", "red", "blue"))
```



```r
#Plotting them - high
plot(fit.high, xlim = c(100, 400), main="Kaplan-Meier Estimate with 95%
confidence bands (High)",
xlab="Time (t)", ylab="Estimated Survival Function")
lines(ep.band10$LL ~ ep.band10$t, lty = 4, col = "red", lwd = 1.5, type =
's')
lines(ep.band10$UL ~ ep.band10$t, lty = 4, col = "red", lwd = 1.5, type =
's')
lines(hw.band10$LL ~ hw.band10$t, lty = 5, col = "blue", lwd = 1.5, type =
's')
lines(hw.band10$UL ~ hw.band10$t, lty = 5, col = "blue", lwd = 1.5, type =
's')
legend("bottomleft", legend=
c("K-M survival estimate",
"Confidence Intervals",
"H-W bands", "EP Bands"), lty=c(1, 2, 4,5),
bty = "n", lwd = c(1, 1, 1.5, 1.5), cex = .7,
col = c("black", "black", "red", "blue"))
```



(i) Based on the results above and those discussed in Example 4.2 and its continuations, how do the survival experiences of the ALL, AML low risk, and AML high risk groups compare?

- Based on the result above, low risk group has higher survival time.

Klein and Moeschberger 4.7

Consider a hypothetical study of the mortality experience of diabetics. Thirty diabetic subjects are recruited at a clinic and followed until death or the end of the study. The subject's age at entry into the study and their age at the end of study or death are given in the table below. Of interest is estimating the survival curve for a 60- or for a 70-year-old diabetic.

```
library("tidyverse")

## ── Attaching packages ────────────────────────────────── tidyverse 1.2.1
──

## ✔ ggplot2 2.2.1     ✔ purrr   0.2.4
## ✔ tibble  1.4.2     ✔ dplyr   0.7.4
## ✔ tidyr   0.8.0     ✔ stringr 1.3.0
## ✔ readr   1.1.1     ✔ forcats 0.3.0
## ── Conflicts ──────────────────────────────────── tidyverse_conflicts()
──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
data <-
read.csv("/Users/huiyuhu/Desktop/Study/UCLA_Biostat/M215/Ex47Data.csv",
header = T)
data1 <- as_data_frame(data)
```

(a) Since the diabetics needed to survive long enough from birth until the study began, the data is left truncated. Construct a table showing the number of subjects at risk, Y, as a function of age.

```
summary(survfit(Surv(entry.age, exit.age, death.indicator, type = 'counting')
~ 1, data = data1))

## Call: survfit(formula = Surv(entry.age, exit.age, death.indicator,
##     type = "counting") ~ 1, data = data1)
##
##  time n.risk n.event censored survival std.err lower 95% CI upper 95% CI
##    60      3       1        0   0.6667  0.2722       0.2995        1.000
##    62      6       1        0   0.5556  0.2485       0.2312        1.000
##    63      8       1        0   0.4861  0.2269       0.1947        1.000
##    65     10       2        0   0.3889  0.1916       0.1480        1.000
##    66      8       1        0   0.3403  0.1737       0.1251        0.926
##    68     12       2        0   0.2836  0.1493       0.1010        0.796
##    69     11       2        2   0.2320  0.1266       0.0796        0.676
##    70     10       2        0   0.1856  0.1054       0.0610        0.565
##    71     11       2        0   0.1519  0.0889       0.0482        0.478
##    72     10       2        1   0.1215  0.0737       0.0370        0.399
##    73      9       1        1   0.1080  0.0667       0.0322        0.362
##    74      9       1        1   0.0960  0.0604       0.0280        0.329
##    76      7       1        1   0.0823  0.0533       0.0231        0.293
##    77      5       1        0   0.0658  0.0451       0.0172        0.252
```

(b) Estimate the conditional survival function for the age of death of a diabetic patient who has survived to age 60.

```
fit_60 <- survfit(Surv(entry.age, exit.age, death.indicator,
type='counting') ~ 1,
data = data, subset = (exit.age >= 60))
summary(fit_60)

## Call: survfit(formula = Surv(entry.age, exit.age, death.indicator,
##     type = "counting") ~ 1, data = data, subset = (exit.age >=
##     60))
##
##  time n.risk n.event censored survival std.err lower 95% CI upper 95% CI
##    60      3       1        0   0.6667  0.2722       0.2995        1.000
##    62      6       1        0   0.5556  0.2485       0.2312        1.000
##    63      8       1        0   0.4861  0.2269       0.1947        1.000
##    65     10       2        0   0.3889  0.1916       0.1480        1.000
##    66      8       1        0   0.3403  0.1737       0.1251        0.926
##    68     12       2        0   0.2836  0.1493       0.1010        0.796
##    69     11       2        2   0.2320  0.1266       0.0796        0.676
##    70     10       2        0   0.1856  0.1054       0.0610        0.565
##    71     11       2        0   0.1519  0.0889       0.0482        0.478
##    72     10       2        1   0.1215  0.0737       0.0370        0.399
##    73      9       1        1   0.1080  0.0667       0.0322        0.362
##    74      9       1        1   0.0960  0.0604       0.0280        0.329
##    76      7       1        1   0.0823  0.0533       0.0231        0.293
##    77      5       1        0   0.0658  0.0451       0.0172        0.252
```

(c) Estimate the conditional survival function for the age of death of a diabetic patient who has survived to age 70.

```
fit_70 <- survfit(Surv(entry.age, exit.age, death.indicator,
type='counting') ~ 1,
data = data, subset = (exit.age > 70))
summary(fit_70)
```

```
## Call: survfit(formula = Surv(entry.age, exit.age, death.indicator,
##     type = "counting") ~ 1, data = data, subset = (exit.age >
##     70))
##
##  time n.risk n.event censored survival std.err lower 95% CI upper 95% CI
##    71     11       2        0    0.818   0.116        0.619        1.000
##    72     10       2        1    0.655   0.139        0.431        0.993
##    73      9       1        1    0.582   0.141        0.361        0.937
##    74      9       1        1    0.517   0.140        0.305        0.878
##    76      7       1        1    0.443   0.138        0.241        0.816
##    77      5       1        0    0.355   0.136        0.167        0.751
```

(d) Suppose an investigator incorrectly ignored the left truncation and simply treated the data as right censored. Repeat parts a–c.

```
summary(survfit(Surv(exit.age, death.indicator, type = 'right') ~ 1, data =
data1))
```

```
## Call: survfit(formula = Surv(exit.age, death.indicator, type = "right") ~
##     1, data = data1)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##    60     30       1    0.967  0.0328        0.905        1.000
##    62     29       1    0.933  0.0455        0.848        1.000
##    63     28       1    0.900  0.0548        0.799        1.000
##    65     27       2    0.833  0.0680        0.710        0.978
##    66     25       1    0.800  0.0730        0.669        0.957
##    68     24       2    0.733  0.0807        0.591        0.910
##    69     22       2    0.667  0.0861        0.518        0.859
##    70     18       2    0.593  0.0911        0.438        0.801
##    71     16       2    0.519  0.0935        0.364        0.738
##    72     14       2    0.444  0.0937        0.294        0.672
##    73     11        1    0.404  0.0935        0.257        0.636
##    74      9        1    0.359  0.0933        0.216        0.597
##    76      7        1    0.308  0.0930        0.170        0.556
##    77      5        1    0.246  0.0926        0.118        0.514
```

```
fit_60 <- survfit(Surv(exit.age, death.indicator,  type='right') ~ 1,
data = data, subset = (exit.age > 60))
summary(fit_60)
```

```
## Call: survfit(formula = Surv(exit.age, death.indicator, type = "right") ~
##     1, data = data, subset = (exit.age > 60))
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##    62     29        1    0.966  0.0339        0.901        1.000
##    63     28        1    0.931  0.0471        0.843        1.000
##    65     27        2    0.862  0.0640        0.745        0.997
##    66     25        1    0.828  0.0701        0.701        0.977
##    68     24        2    0.759  0.0795        0.618        0.932
##    69     22        2    0.690  0.0859        0.540        0.880
##    70     18        2    0.613  0.0919        0.457        0.822
##    71     16        2    0.536  0.0950        0.379        0.759
##    72     14        2    0.460  0.0957        0.306        0.691
##    73     11        1    0.418  0.0957        0.267        0.655
##    74      9        1    0.372  0.0956        0.224        0.615
##    76      7        1    0.318  0.0956        0.177        0.573
##    77      5        1    0.255  0.0954        0.122        0.531
```

```
fit_70 <- survfit(Surv(exit.age, death.indicator,  type='right') ~ 1,
data = data, subset = (exit.age > 70))
summary(fit_70)
```

```
## Call: survfit(formula = Surv(exit.age, death.indicator, type = "right") ~
##     1, data = data, subset = (exit.age > 70))
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##    71     16        2    0.875  0.0827        0.727        1.000
##    72     14        2    0.750  0.1083        0.565        0.995
##    73     11        1    0.682  0.1179        0.486        0.957
##    74      9        1    0.606  0.1269        0.402        0.913
##    76      7        1    0.519  0.1351        0.312        0.865
##    77      5        1    0.416  0.1425        0.212        0.814
```

- Based on the calculation, I get the result showed as below (different with result above). Left Truncated:

Time	Y	S(t	Alive at 60)
58	2	1.0000	1.0000
59	3	1.0000	1.0000
60	5	0.8000	1.0000
61	6	0.8000	1.0000

62	9	0.7111	1.0000
63	10	0.6400	1.0000
64	10	0.6400	1.0000
65	10	0.5120	0.8000
66	10	0.4608	0.7200
67	12	0.4608	0.7200
68	13	0.3899	0.6092
69	14	0.3342	0.5222
70	13	0.2828	0.4419
71	12	0.2357	0.3682
72	12	0.1964	0.3068
73	11	0.1785	0.2790
74	9	0.1587	0.2480
76	7	0.1360	0.2125
77	5	0.1088	0.1700
78	4	0.1088	0.1700
79	3	0.1088	0.1700
80	1	0.1088	0.1700

No Truncation:

Time	Y	S(t	Alive at 60)
58	30	1.0000	1.0000
59	30	1.0000	1.0000
60	30	0.9667	1.0000
61	30	0.9667	1.0000
62	29	0.9333	1.0000
63	28	0.9000	1.0000
64	28	0.9000	1.0000
65	27	0.8333	0.9259
66	25	0.8000	0.8889
67	25	0.8000	0.8889
68	24	0.7333	0.8148
69	22	0.6667	0.7407
70	18	0.5926	0.6584
71	16	0.5185	0.5761
72	14	0.4444	0.4938

73	11	0.4040	0.4489
74	9	0.3591	0.3991
76	7	0.3078	0.3420
77	5	0.2463	0.2736
78	4	0.2463	0.2736
79	3	0.2463	0.2736
80	1	0.2463	0.2736