存活分析_HW2(無母數估計存活函數)

B082040005 高念慈

2023-04-16

HW 73-150行 & 200-217行

K-M estimate

```
data("drug6mp")
# head(drug6mp) # packages 'KMsurv'
```

```
drug6mp_event = drug6mp[drug6mp$relapse==1,]
sort(drug6mp_event$t2)          # 6  6  6  7 10 13 16 22 23 (6-MP 的事件發生時間:t1...tD)
```

```
## [1]  6  6  6  7 10 13 16 22 23
```

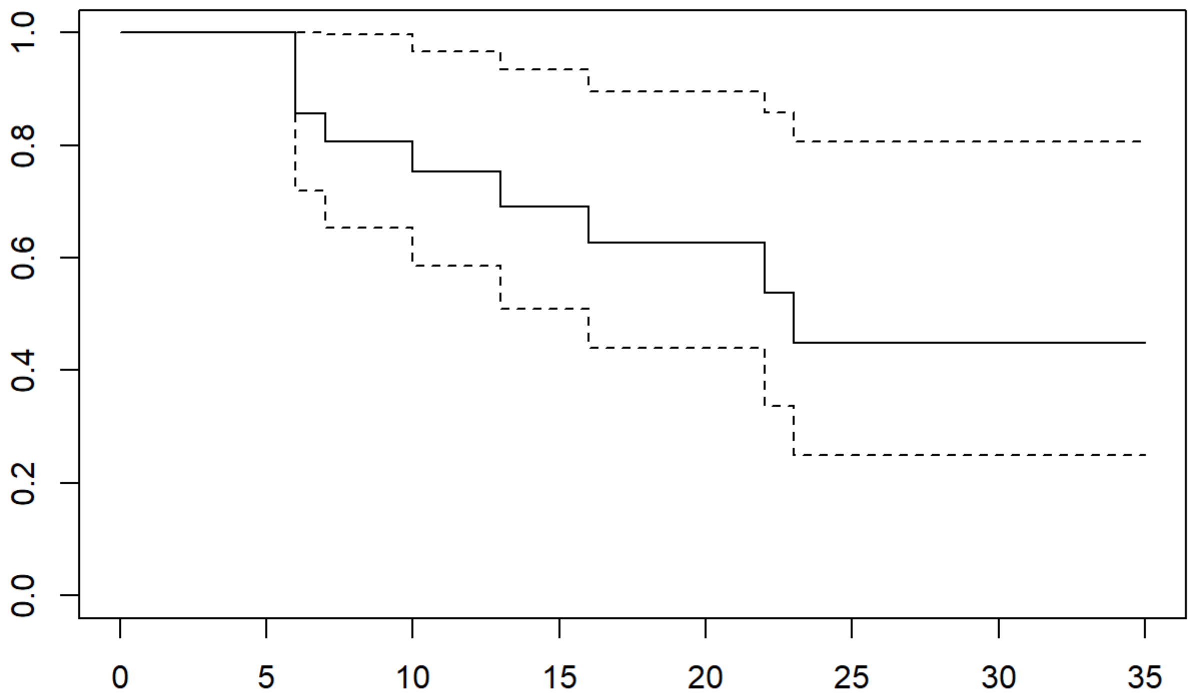
```
Surv(drug6mp$t2,drug6mp$relapse) # use Surv function to create survival object，Ti=min(xi,ci)
```

```
##  [1] 10   7  32+ 23  22   6  16  34+ 32+ 25+ 11+ 20+ 19+  6  17+ 35+  6  13   9+
## [20]  6+ 10+
```

```
fit.km.6mp = survfit(Surv(t2,relapse)~1, data=drug6mp)
temp = summary(fit.km.6mp)
temp
```

```
## Call: survfit(formula = Surv(t2, relapse) ~ 1, data = drug6mp)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     6     21       3    0.857  0.0764        0.720        1.000
##     7     17       1    0.807  0.0869        0.653        0.996
##    10     15       1    0.753  0.0963        0.586        0.968
##    13     12       1    0.690  0.1068        0.510        0.935
##    16     11       1    0.627  0.1141        0.439        0.896
##    22      7       1    0.538  0.1282        0.337        0.858
##    23      6       1    0.448  0.1346        0.249        0.807
```

```
plot(fit.km.6mp, conf.int=T)  # K-M estimate plot
```




```
fit.km.6mp$n.risk            # Yi...YD
```

```
##  [1] 21 17 16 15 13 12 11 10  9  8  7  6  5  4  2  1
```

- Compute a Survival Curve for Censored Data(截尾數據) (http://stat.ethz.ch/R-manual/R-patched/library/survival/html/survfit.formula.html)
- 在 R 中為生存選擇 conf.type (https://stats.stackexchange.com/questions/361354/choosing-conf-type-for-survfit-in-r)

```
fit.km.6mp = survfit(Surv(t2,relapse)~1, data=drug6mp, conf.type="log-log")
summary(fit.km.6mp)  # conf.type=c("log","log-log","plain","none")
```

```
## Call: survfit(formula = Surv(t2, relapse) ~ 1, data = drug6mp, conf.type = "log-log")
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     6     21       3    0.857  0.0764        0.620        0.952
##     7     17       1    0.807  0.0869        0.563        0.923
##    10     15       1    0.753  0.0963        0.503        0.889
##    13     12       1    0.690  0.1068        0.432        0.849
##    16     11       1    0.627  0.1141        0.368        0.805
##    22      7       1    0.538  0.1282        0.268        0.747
##    23      6       1    0.448  0.1346        0.188        0.680
```

type 方法
+ “kaplan-meier”，相當於 stype=1、ctype=1 + “fleming-harrington”，相當於 stype=2、ctype=1 和 “fh2”

```
fit.km.6mp = survfit(Surv(t2,relapse)~1, data=drug6mp, type="fh2")
summary(fit.km.6mp)
```

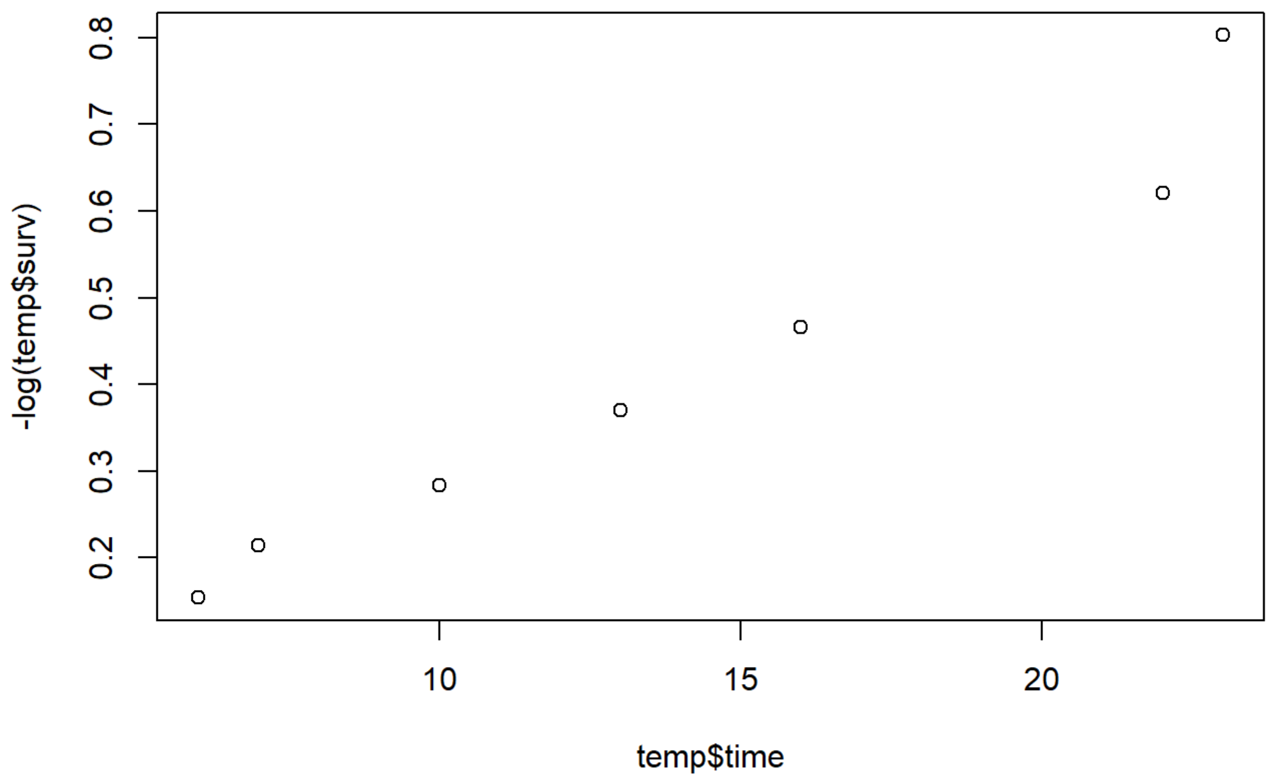
```
## Call: survfit(formula = Surv(t2, relapse) ~ 1, data = drug6mp, type = "fh2")
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     6     21       3    0.860  0.0747        0.726        1.000
##     7     17       1    0.811  0.0851        0.661        0.996
##    10     15       1    0.759  0.0943        0.595        0.968
##    13     12       1    0.698  0.1045        0.521        0.936
##    16     11       1    0.638  0.1116        0.452        0.899
##    22      7       1    0.553  0.1249        0.355        0.861
##    23      6       1    0.468  0.1314        0.270        0.811
```

- 累積風險函數
- $S(t) = e^{-H(t)}$

```
list(temp$time,-log(temp$surv))
```

```
## [[1]]
## [1]  6  7 10 13 16 22 23
##
## [[2]]
## [1] 0.1541507 0.2147753 0.2837682 0.3707796 0.4660897 0.6202404 0.8025620
```

```
plot(temp$time,-log(temp$surv))
```



Homework: (6mp, t1(placebo), relapse)

(1)

Calculate K-M estimator for S(t) and variance estimate and 99% confidence interval for S(t)

```
fit.km.6mp = survfit(Surv(t1,relapse)~1, data=drug6mp, ctype=1, conf.int = .99, conf.type="lo
g-log")
summary(fit.km.6mp)
```

```
## Call: survfit(formula = Surv(t1, relapse) ~ 1, data = drug6mp, ctype = 1,
##     conf.int = 0.99, conf.type = "log-log")
##
##  time n.risk n.event survival std.err lower 99% CI upper 99% CI
##     1     21       1    0.952  0.0465       0.5266        0.996
##     2     19       1    0.902  0.0657       0.5287        0.984
##     5     14       1    0.838  0.0871       0.4479        0.962
##     8     12       1    0.768  0.1041       0.3713        0.932
##    11      8       1    0.672  0.1279       0.2555        0.891
##    12      6       1    0.560  0.1477       0.1539        0.836
##    15      4       1    0.420  0.1642       0.0627        0.762
##    17      3       1    0.280  0.1583       0.0184        0.667
##    22      2       1    0.140  0.1267       0.0016        0.549
```

- Be careful with standard errors in `survival::survfit` (https://dominicmagirr.github.io/post/2022-01-18-be-careful-with-standard-errors-in-survival-survfit/)

```
round(cbind("累積危害的標準誤差" = fit.km.6mp$std.err,
      '個別標準誤差' = summary((fit.km.6mp))$std.err,
      'variance estimate' = (summary((fit.km.6mp))$std.err)^2),4)
```

```
##       累積危害的標準誤差 個別標準誤差 variance estimate
##  [1,]            0.0488       0.0465            0.0022
##  [2,]            0.0728       0.0657            0.0043
##  [3,]            0.0728       0.0871            0.0076
##  [4,]            0.0728       0.1041            0.0108
##  [5,]            0.1039       0.1279            0.0164
##  [6,]            0.1356       0.1477            0.0218
##  [7,]            0.1903       0.1642            0.0270
##  [8,]            0.2638       0.1583            0.0251
##  [9,]            0.3910       0.1267            0.0161
## [10,]            0.5653       0.0465            0.0022
## [11,]            0.9053       0.0657            0.0043
## [12,]            0.9053       0.0871            0.0076
```

- K-M estimator
- variance estimate
- 99% confidence interval

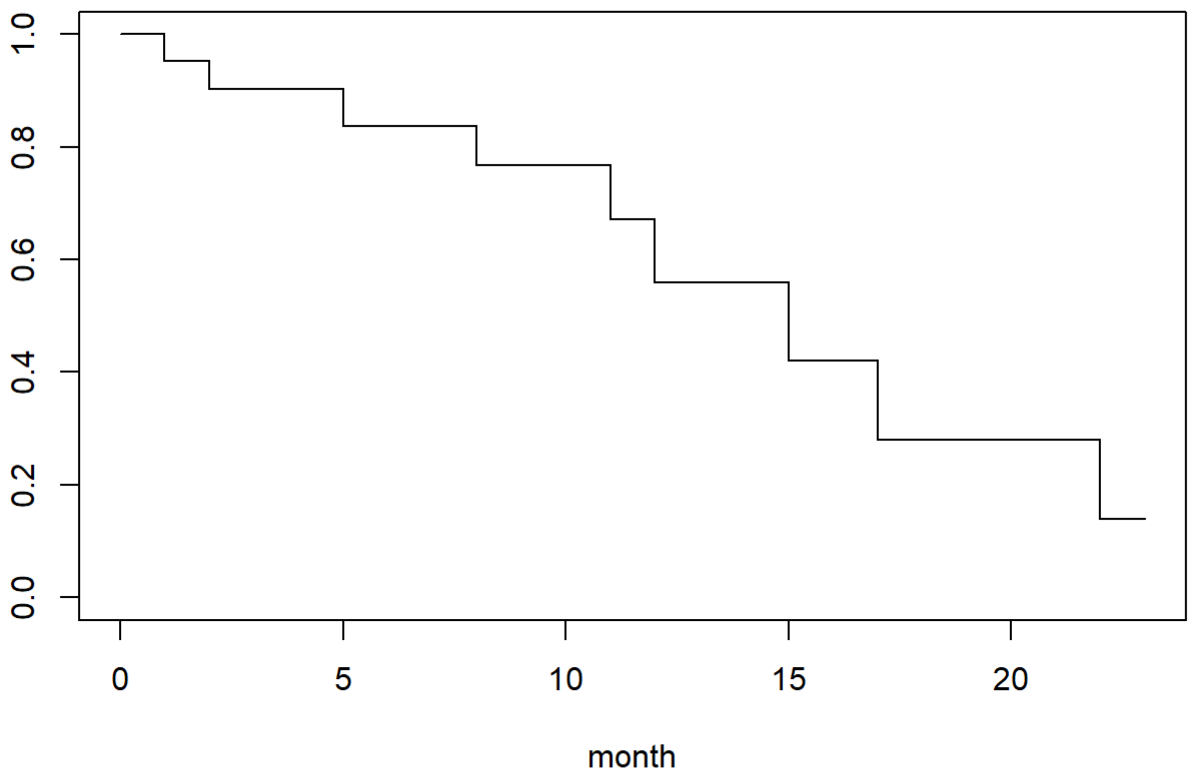
```r
round(cbind('K-M estimator'=fit.km.6mp$surv,
            'variance estimate'=(summary((fit.km.6mp))$std.err)^2,
            '99% confidence lower'=fit.km.6mp$lower,
            '99% confidence upper'=fit.km.6mp$upper),4)
```

```
##       K-M estimator variance estimate 99% confidence lower 99% confidence upper
##  [1,]        0.9524            0.0022               0.5266               0.9963
##  [2,]        0.9023            0.0043               0.5287               0.9835
##  [3,]        0.9023            0.0076               0.5287               0.9835
##  [4,]        0.9023            0.0108               0.5287               0.9835
##  [5,]        0.8378            0.0164               0.4479               0.9618
##  [6,]        0.7680            0.0218               0.3713               0.9321
##  [7,]        0.6720            0.0270               0.2555               0.8907
##  [8,]        0.5600            0.0251               0.1539               0.8356
##  [9,]        0.4200            0.0161               0.0627               0.7621
## [10,]        0.2800            0.0022               0.0184               0.6666
## [11,]        0.1400            0.0043               0.0016               0.5485
## [12,]        0.1400            0.0076               0.0016               0.5485
```

(2)

Plot K-M estimators vs. time

```r
plot(fit.km.6mp, conf.int=F, xlab='month')  # K-M estimate plot
```



(3)

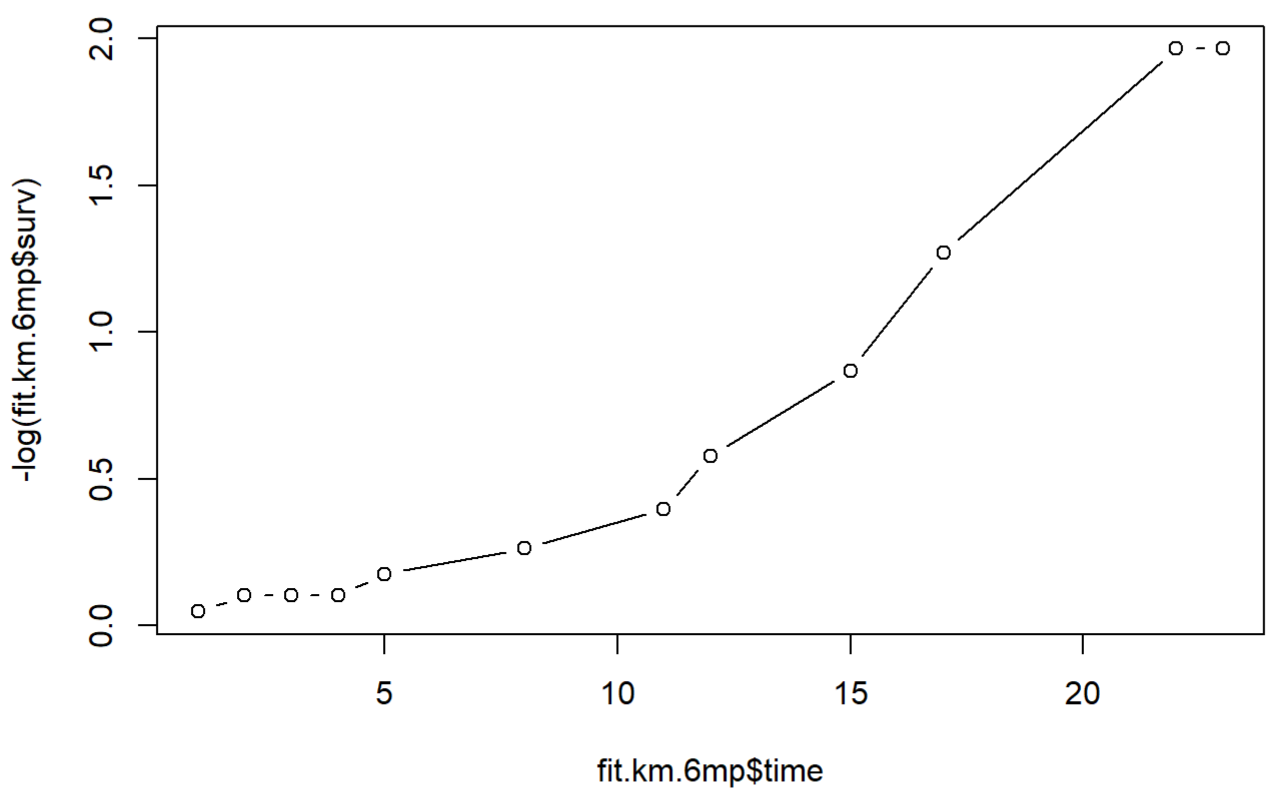
Calculate the cumulative hazard function H(t) using the K-M estimator

- 累積風險函數
- $S(t) = e^{-H(t)}$
- $-log(S(t) = H(t))$

```
cbind("time" = fit.km.6mp$time,
      "H(t)" = -log(fit.km.6mp$surv))
```

```
##       time       H(t)
##  [1,]    1 0.04879016
##  [2,]    2 0.10285739
##  [3,]    3 0.10285739
##  [4,]    4 0.10285739
##  [5,]    5 0.17696536
##  [6,]    8 0.26397673
##  [7,]   11 0.39750813
##  [8,]   12 0.57982968
##  [9,]   15 0.86751176
## [10,]   17 1.27297686
## [11,]   22 1.96612405
## [12,]   23 1.96612405
```

```
plot(fit.km.6mp$time, -log(fit.km.6mp$surv), type="b")
```



(4)

Calculate the Nelson-Aalen estimator for H(t) and S(t)

```
H_t = rep(0,length(fit.km.6mp$time))
s_t = rep(0,length(fit.km.6mp$time))

for (i in length(fit.km.6mp$time):1){
  H_t[i] = sum((fit.km.6mp$n.event/fit.km.6mp$n.risk)[1:i])
}

s_t = exp(-H_t)

cbind('NA_H(t)' = H_t,
      "NA_S(t)" = s_t)
```

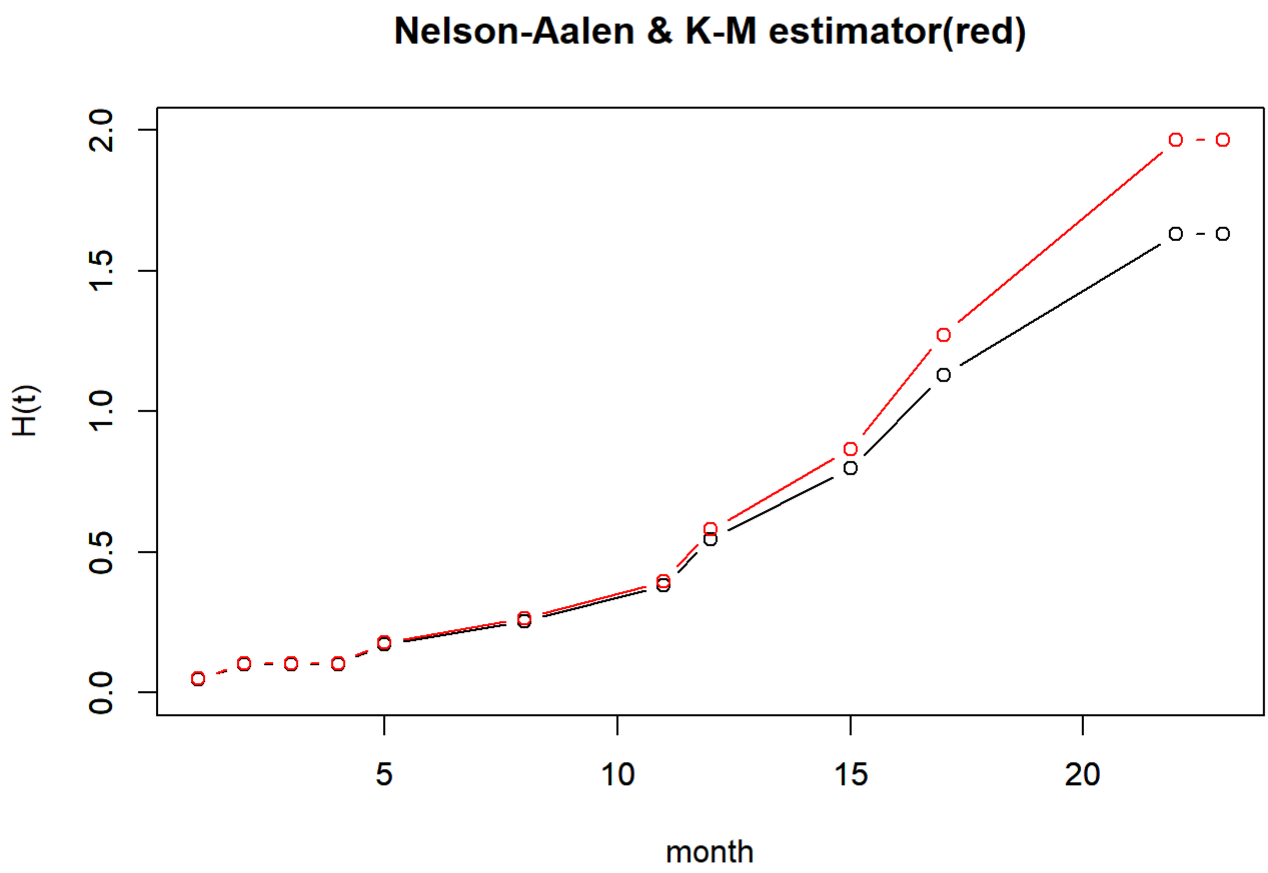
```
##          NA_H(t)   NA_S(t)
##  [1,] 0.04761905 0.9534970
##  [2,] 0.10025063 0.9046107
##  [3,] 0.10025063 0.9046107
##  [4,] 0.10025063 0.9046107
##  [5,] 0.17167920 0.8422493
##  [6,] 0.25501253 0.7749068
##  [7,] 0.38001253 0.6838528
##  [8,] 0.54667920 0.5788689
##  [9,] 0.79667920 0.4508236
## [10,] 1.13001253 0.3230292
## [11,] 1.63001253 0.1959271
## [12,] 1.63001253 0.1959271
```

(5)

Plot H(t) vs. t to check whether the distribution of the time-to-event random variable is exponential

- Yes

```
plot(fit.km.6mp$time, H_t, type="b", xlab="month", ylim=c(0,2), main="Nelson-Aalen & K-M esti
mator(red)", ylab='H(t)')
par(new=TRUE)
plot(fit.km.6mp$time, -log(fit.km.6mp$surv), type="b", col="red", xlab='', ylab='' , ylim=c
(0,2))
```



Left-truncated and right-censored data

```
data(channing)

channing_new = channing[channing[,"ageentry"]<channing[,"age"],] # 4筆進院=出院
# only keep the subjects whose entry time (Lj) are less than the corresponding study time (T
j)

channing_new_male = channing_new[channing_new[,"gender"]==1,]
# only keep the male subjects
```

- codes for calculating the risk sets for male subjects

```
sort(channing_new_male$ageentry) # 排進院時間
```

```
##  [1]  751  759  782  806  817  820  821  823  830  835  835  836  836  837  843
## [16]  846  847  847  852  853  854  856  856  856  863  865  865  866  871  871
## [31]  875  876  878  878  879  883  885  886  890  891  893  894  898  900  906
## [46]  906  909  915  919  919  921  923  925  926  936  936  938  943  943  946
## [61]  953  955  955  956  959  960  962  962  964  966  967  967  969  969  971
## [76]  978  978  981  982  984  984  988 1007 1010 1010 1016 1020 1021 1027 1036
## [91] 1039 1041 1046 1051 1063 1073
```

```
sort(channing_new_male[channing_new_male[,"death"]==1,"age"]) # 排死亡時間
```

```
## [1] 777 781 869 872 876 893 894 898 907 909 911 927 932 945 948
## [16] 957 966 969 971 983 985 989 993 993 998 1009 1012 1012 1022 1025
## [31] 1029 1031 1033 1036 1043 1044 1053 1055 1059 1060 1080 1085 1094 1094 1128
## [46] 1139
```

```
# cbind(channing_new$ageentry, channing_new$age)

fit = survfit(Surv(ageentry,age,death)~1, data=channing_new, subset=(gender==1))
summary(fit)
```

```
## Call: survfit(formula = Surv(ageentry, age, death) ~ 1, data = channing_new,
##     subset = (gender == 1))
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##   777      2       1      0.5   0.354        0.125            1
##   781      1       1      0.0     NaN           NA           NA
##   869     24       1      0.0     NaN           NA           NA
##   872     25       1      0.0     NaN           NA           NA
##   876     25       1      0.0     NaN           NA           NA
##   893     33       1      0.0     NaN           NA           NA
##   894     33       1      0.0     NaN           NA           NA
##   898     32       1      0.0     NaN           NA           NA
##   907     34       1      0.0     NaN           NA           NA
##   909     33       1      0.0     NaN           NA           NA
##   911     33       1      0.0     NaN           NA           NA
##   927     37       1      0.0     NaN           NA           NA
##   932     36       1      0.0     NaN           NA           NA
##   945     36       1      0.0     NaN           NA           NA
##   948     35       1      0.0     NaN           NA           NA
##   957     36       1      0.0     NaN           NA           NA
##   966     37       1      0.0     NaN           NA           NA
##   969     38       1      0.0     NaN           NA           NA
##   971     38       1      0.0     NaN           NA           NA
##   983     39       1      0.0     NaN           NA           NA
##   985     39       1      0.0     NaN           NA           NA
##   989     39       1      0.0     NaN           NA           NA
##   993     38       2      0.0     NaN           NA           NA
##   998     35       1      0.0     NaN           NA           NA
##  1009     31       1      0.0     NaN           NA           NA
##  1012     32       2      0.0     NaN           NA           NA
##  1022     28       1      0.0     NaN           NA           NA
##  1025     26       1      0.0     NaN           NA           NA
##  1029     25       1      0.0     NaN           NA           NA
##  1031     24       1      0.0     NaN           NA           NA
##  1033     21       1      0.0     NaN           NA           NA
##  1036     20       1      0.0     NaN           NA           NA
##  1043     22       1      0.0     NaN           NA           NA
##  1044     20       1      0.0     NaN           NA           NA
##  1053     18       1      0.0     NaN           NA           NA
##  1055     17       1      0.0     NaN           NA           NA
##  1059     15       1      0.0     NaN           NA           NA
##  1060     14       1      0.0     NaN           NA           NA
##  1080     11       1      0.0     NaN           NA           NA
##  1085     10       1      0.0     NaN           NA           NA
##  1094      8       2      0.0     NaN           NA           NA
##  1128      3       1      0.0     NaN           NA           NA
##  1139      2       1      0.0     NaN           NA           NA
```

- Now we only consider those deaths that occur after age 68 years old, that is P(X>t|X>68 years old)

```
fit_age = survfit(Surv(ageentry,age,death)~1, data=channing_new, subset=(gender==1 & age>=12*
68))
summary(fit_age)
```

```
## Call: survfit(formula = Surv(ageentry, age, death) ~ 1, data = channing_new,
##     subset = (gender == 1 & age >= 12 * 68))
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##   869     24       1   0.9583  0.0408      0.88163        1.000
##   872     25       1   0.9200  0.0543      0.81957        1.000
##   876     25       1   0.8832  0.0634      0.76737        1.000
##   893     33       1   0.8564  0.0668      0.73495        0.998
##   894     33       1   0.8305  0.0697      0.70456        0.979
##   898     32       1   0.8045  0.0722      0.67482        0.959
##   907     34       1   0.7809  0.0738      0.64879        0.940
##   909     33       1   0.7572  0.0753      0.62314        0.920
##   911     33       1   0.7343  0.0764      0.59877        0.900
##   927     37       1   0.7144  0.0769      0.57855        0.882
##   932     36       1   0.6946  0.0773      0.55850        0.864
##   945     36       1   0.6753  0.0775      0.53926        0.846
##   948     35       1   0.6560  0.0776      0.52016        0.827
##   957     36       1   0.6378  0.0776      0.50245        0.810
##   966     37       1   0.6205  0.0774      0.48596        0.792
##   969     38       1   0.6042  0.0771      0.47056        0.776
##   971     38       1   0.5883  0.0767      0.45571        0.759
##   983     39       1   0.5732  0.0762      0.44180        0.744
##   985     39       1   0.5585  0.0756      0.42835        0.728
##   989     39       1   0.5442  0.0750      0.41535        0.713
##   993     38       2   0.5156  0.0738      0.38950        0.682
##   998     35       1   0.5008  0.0731      0.37622        0.667
##  1009     31       1   0.4847  0.0725      0.36150        0.650
##  1012     32       2   0.4544  0.0711      0.33441        0.617
##  1022     28       1   0.4381  0.0704      0.31984        0.600
##  1025     26       1   0.4213  0.0696      0.30471        0.582
##  1029     25       1   0.4044  0.0689      0.28968        0.565
##  1031     24       1   0.3876  0.0680      0.27478        0.547
##  1033     21       1   0.3691  0.0672      0.25830        0.528
##  1036     20       1   0.3507  0.0664      0.24200        0.508
##  1043     22       1   0.3347  0.0652      0.22847        0.490
##  1044     20       1   0.3180  0.0641      0.21424        0.472
##  1053     18       1   0.3003  0.0629      0.19921        0.453
##  1055     17       1   0.2827  0.0616      0.18435        0.433
##  1059     15       1   0.2638  0.0603      0.16851        0.413
##  1060     14       1   0.2450  0.0589      0.15292        0.392
##  1080     11       1   0.2227  0.0576      0.13414        0.370
##  1085     10       1   0.2004  0.0560      0.11594        0.347
##  1094      8       2   0.1503  0.0520      0.07631        0.296
##  1128      3       1   0.1002  0.0536      0.03511        0.286
##  1139      2       1   0.0501  0.0444      0.00881        0.285
```

- 以下不需要 run,但是可以看到為什麼不加 age>68 的限制時會發生的問題

```
attach(channing_new_male) # 無需實際鍵入數據框的名稱，就可使數據框中的對象可訪問
aa <- cbind(ageentry, age, death)
## aa[order(-aa[,3], aa[,2]), ]
# aa[order(aa[,2]), ]
# aa[order(aa[,1]), ]
```

- 一次出現兩個Model

```
fit_age = survfit(Surv(ageentry,age,death)~gender, data=channing_new, subset=(age>=12*68))
summary(fit_age)
```

```
## Call: survfit(formula = Surv(ageentry, age, death) ~ gender, data = channing_new,
##     subset = (age >= 12 * 68))
##
##                 gender=1
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##   869     24       1   0.9583  0.0408      0.88163        1.000
##   872     25       1   0.9200  0.0543      0.81957        1.000
##   876     25       1   0.8832  0.0634      0.76737        1.000
##   893     33       1   0.8564  0.0668      0.73495        0.998
##   894     33       1   0.8305  0.0697      0.70456        0.979
##   898     32       1   0.8045  0.0722      0.67482        0.959
##   907     34       1   0.7809  0.0738      0.64879        0.940
##   909     33       1   0.7572  0.0753      0.62314        0.920
##   911     33       1   0.7343  0.0764      0.59877        0.900
##   927     37       1   0.7144  0.0769      0.57855        0.882
##   932     36       1   0.6946  0.0773      0.55850        0.864
##   945     36       1   0.6753  0.0775      0.53926        0.846
##   948     35       1   0.6560  0.0776      0.52016        0.827
##   957     36       1   0.6378  0.0776      0.50245        0.810
##   966     37       1   0.6205  0.0774      0.48596        0.792
##   969     38       1   0.6042  0.0771      0.47056        0.776
##   971     38       1   0.5883  0.0767      0.45571        0.759
##   983     39       1   0.5732  0.0762      0.44180        0.744
##   985     39       1   0.5585  0.0756      0.42835        0.728
##   989     39       1   0.5442  0.0750      0.41535        0.713
##   993     38       2   0.5156  0.0738      0.38950        0.682
##   998     35       1   0.5008  0.0731      0.37622        0.667
##  1009     31       1   0.4847  0.0725      0.36150        0.650
##  1012     32       2   0.4544  0.0711      0.33441        0.617
##  1022     28       1   0.4381  0.0704      0.31984        0.600
##  1025     26       1   0.4213  0.0696      0.30471        0.582
##  1029     25       1   0.4044  0.0689      0.28968        0.565
##  1031     24       1   0.3876  0.0680      0.27478        0.547
##  1033     21       1   0.3691  0.0672      0.25830        0.528
##  1036     20       1   0.3507  0.0664      0.24200        0.508
##  1043     22       1   0.3347  0.0652      0.22847        0.490
##  1044     20       1   0.3180  0.0641      0.21424        0.472
##  1053     18       1   0.3003  0.0629      0.19921        0.453
##  1055     17       1   0.2827  0.0616      0.18435        0.433
##  1059     15       1   0.2638  0.0603      0.16851        0.413
##  1060     14       1   0.2450  0.0589      0.15292        0.392
##  1080     11       1   0.2227  0.0576      0.13414        0.370
##  1085     10       1   0.2004  0.0560      0.11594        0.347
##  1094      8       2   0.1503  0.0520      0.07631        0.296
##  1128      3       1   0.1002  0.0536      0.03511        0.286
##  1139      2       1   0.0501  0.0444      0.00881        0.285
##
##                 gender=2
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##   822     36       1   0.9722  0.0274      0.92000        1.000
##   830     46       1   0.9511  0.0340      0.88676        1.000
##   840     58       1   0.9347  0.0371      0.86465        1.000
##   845     66       1   0.9205  0.0392      0.84684        1.000
##   861     90       1   0.9103  0.0401      0.83506        0.992
##   868    100       1   0.9012  0.0407      0.82488        0.985
```

```
##    873    106        1   0.8927  0.0412      0.81552         0.977
##    883    116        1   0.8850  0.0415      0.80722         0.970
##    885    119        1   0.8776  0.0418      0.79925         0.964
##    895    137        1   0.8712  0.0420      0.79255         0.958
##    897    140        1   0.8649  0.0422      0.78607         0.952
##    901    145        1   0.8590  0.0423      0.77991         0.946
##    905    149        2   0.8474  0.0425      0.76805         0.935
##    908    148        2   0.8360  0.0427      0.75632         0.924
##    911    151        1   0.8305  0.0428      0.75068         0.919
##    912    151        1   0.8250  0.0429      0.74508         0.913
##    915    149        1   0.8194  0.0429      0.73945         0.908
##    919    150        1   0.8140  0.0430      0.73391         0.903
##    923    158        1   0.8088  0.0430      0.72872         0.898
##    926    160        1   0.8037  0.0431      0.72364         0.893
##    928    157        1   0.7986  0.0431      0.71850         0.888
##    930    157        1   0.7935  0.0431      0.71340         0.883
##    931    157        1   0.7885  0.0431      0.70833         0.878
##    932    157        1   0.7835  0.0431      0.70331         0.873
##    934    155        1   0.7784  0.0432      0.69825         0.868
##    936    164        1   0.7737  0.0432      0.69354         0.863
##    940    162        1   0.7689  0.0432      0.68879         0.858
##    941    160        1   0.7641  0.0432      0.68402         0.854
##    944    166        2   0.7549  0.0431      0.67492         0.844
##    948    160        1   0.7502  0.0431      0.67025         0.840
##    954    159        1   0.7454  0.0431      0.66559         0.835
##    959    161        1   0.7408  0.0431      0.66102         0.830
##    963    157        1   0.7361  0.0431      0.65636         0.826
##    966    155        1   0.7313  0.0430      0.65167         0.821
##    969    162        2   0.7223  0.0430      0.64280         0.812
##    970    160        1   0.7178  0.0429      0.63837         0.807
##    975    158        1   0.7133  0.0429      0.63391         0.803
##    976    155        1   0.7087  0.0429      0.62939         0.798
##    978    153        1   0.7040  0.0429      0.62485         0.793
##    982    153        2   0.6948  0.0428      0.61582         0.784
##    983    151        1   0.6902  0.0427      0.61132         0.779
##    986    146        1   0.6855  0.0427      0.60668         0.775
##    989    144        1   0.6807  0.0427      0.60201         0.770
##    990    141        4   0.6614  0.0426      0.58306         0.750
##    991    137        1   0.6566  0.0425      0.57834         0.745
##    992    136        1   0.6518  0.0425      0.57361         0.741
##    994    134        2   0.6420  0.0424      0.56410         0.731
##    995    132        2   0.6323  0.0423      0.55460         0.721
##    996    129        3   0.6176  0.0422      0.54025         0.706
##    998    125        1   0.6127  0.0421      0.53543         0.701
##    999    123        1   0.6077  0.0421      0.53058         0.696
##   1000    122        1   0.6027  0.0420      0.52573         0.691
##   1001    120        1   0.5977  0.0420      0.52083         0.686
##   1003    118        1   0.5926  0.0419      0.51590         0.681
##   1004    117        1   0.5875  0.0419      0.51097         0.676
##   1005    117        1   0.5825  0.0418      0.50609         0.670
##   1006    115        2   0.5724  0.0417      0.49625         0.660
##   1010    110        1   0.5672  0.0416      0.49119         0.655
##   1011    107        1   0.5619  0.0416      0.48603         0.650
##   1012    104        2   0.5511  0.0415      0.47551         0.639
##   1013     99        1   0.5455  0.0414      0.47008         0.633
##   1014    100        1   0.5401  0.0414      0.46477         0.628
```

```
##  1015     96       1   0.5344  0.0413      0.45929        0.622
##  1017     95       1   0.5288  0.0413      0.45381        0.616
##  1018     94       2   0.5176  0.0411      0.44288        0.605
##  1019     92       2   0.5063  0.0410      0.43198        0.593
##  1020     86       1   0.5004  0.0410      0.42625        0.587
##  1021     83       1   0.4944  0.0409      0.42038        0.581
##  1023     81       2   0.4822  0.0408      0.40850        0.569
##  1024     77       1   0.4759  0.0407      0.40240        0.563
##  1027     77       1   0.4697  0.0407      0.39641        0.557
##  1029     78       1   0.4637  0.0406      0.39059        0.551
##  1030     76       1   0.4576  0.0405      0.38470        0.544
##  1033     71       1   0.4512  0.0405      0.37844        0.538
##  1040     67       4   0.4242  0.0402      0.35229        0.511
##  1041     63       3   0.4040  0.0400      0.33283        0.490
##  1043     60       1   0.3973  0.0399      0.32637        0.484
##  1044     58       1   0.3905  0.0398      0.31981        0.477
##  1047     56       1   0.3835  0.0397      0.31312        0.470
##  1056     48       2   0.3675  0.0396      0.29756        0.454
##  1063     42       1   0.3588  0.0396      0.28897        0.445
##  1064     40       1   0.3498  0.0396      0.28016        0.437
##  1068     38       2   0.3314  0.0396      0.26217        0.419
##  1070     36       1   0.3222  0.0396      0.25326        0.410
##  1072     35       1   0.3130  0.0395      0.24440        0.401
##  1073     33       1   0.3035  0.0394      0.23528        0.391
##  1074     32       1   0.2940  0.0393      0.22623        0.382
##  1083     30       1   0.2842  0.0392      0.21688        0.372
##  1084     29       1   0.2744  0.0391      0.20760        0.363
##  1085     28       2   0.2548  0.0386      0.18928        0.343
##  1086     25       1   0.2446  0.0384      0.17979        0.333
##  1089     23       1   0.2340  0.0382      0.16991        0.322
##  1093     21       1   0.2228  0.0380      0.15957        0.311
##  1097     21       1   0.2122  0.0376      0.14994        0.300
##  1115     15       1   0.1981  0.0377      0.13644        0.288
##  1122     13       1   0.1828  0.0377      0.12201        0.274
##  1131     12       1   0.1676  0.0375      0.10805        0.260
##  1132     11       1   0.1524  0.0371      0.09456        0.246
##  1142     10       1   0.1371  0.0364      0.08153        0.231
##  1152      8       1   0.1200  0.0356      0.06704        0.215
##  1172      7       1   0.1028  0.0344      0.05337        0.198
##  1192      4       1   0.0771  0.0341      0.03244        0.183
##  1200      3       2   0.0257  0.0239      0.00417        0.159
```

HW:女性

```
fit_age = survfit(Surv(ageentry,age,death)~1,
                  data=channing_new,
                  subset=(gender==2 & age>=12*68),
                  conf.type="log-log")
summary(fit_age)
```

```
## Call: survfit(formula = Surv(ageentry, age, death) ~ 1, data = channing_new,
##     subset = (gender == 2 & age >= 12 * 68), conf.type = "log-log")
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##   822     36       1   0.9722  0.0274      0.81873        0.996
##   830     46       1   0.9511  0.0340      0.81656        0.988
##   840     58       1   0.9347  0.0371      0.80736        0.979
##   845     66       1   0.9205  0.0392      0.79707        0.970
##   861     90       1   0.9103  0.0401      0.79031        0.963
##   868    100       1   0.9012  0.0407      0.78385        0.957
##   873    106       1   0.8927  0.0412      0.77744        0.950
##   883    116       1   0.8850  0.0415      0.77151        0.944
##   885    119       1   0.8776  0.0418      0.76555        0.938
##   895    137       1   0.8712  0.0420      0.76049        0.933
##   897    140       1   0.8649  0.0422      0.75546        0.928
##   901    145       1   0.8590  0.0423      0.75057        0.923
##   905    149       2   0.8474  0.0425      0.74091        0.913
##   908    148       2   0.8360  0.0427      0.73101        0.903
##   911    151       1   0.8305  0.0428      0.72618        0.898
##   912    151       1   0.8250  0.0429      0.72133        0.893
##   915    149       1   0.8194  0.0429      0.71640        0.888
##   919    150       1   0.8140  0.0430      0.71151        0.883
##   923    158       1   0.8088  0.0430      0.70692        0.878
##   926    160       1   0.8037  0.0431      0.70240        0.874
##   928    157       1   0.7986  0.0431      0.69778        0.869
##   930    157       1   0.7935  0.0431      0.69318        0.864
##   931    157       1   0.7885  0.0431      0.68859        0.860
##   932    157       1   0.7835  0.0431      0.68402        0.855
##   934    155       1   0.7784  0.0432      0.67939        0.850
##   936    164       1   0.7737  0.0432      0.67507        0.846
##   940    162       1   0.7689  0.0432      0.67071        0.841
##   941    160       1   0.7641  0.0432      0.66631        0.837
##   944    166       2   0.7549  0.0431      0.65789        0.828
##   948    160       1   0.7502  0.0431      0.65354        0.823
##   954    159       1   0.7454  0.0431      0.64918        0.819
##   959    161       1   0.7408  0.0431      0.64491        0.814
##   963    157       1   0.7361  0.0431      0.64054        0.810
##   966    155       1   0.7313  0.0430      0.63613        0.805
##   969    162       2   0.7223  0.0430      0.62778        0.797
##   970    160       1   0.7178  0.0429      0.62360        0.792
##   975    158       1   0.7133  0.0429      0.61938        0.788
##   976    155       1   0.7087  0.0429      0.61509        0.783
##   978    153       1   0.7040  0.0429      0.61077        0.779
##   982    153       2   0.6948  0.0428      0.60218        0.770
##   983    151       1   0.6902  0.0427      0.59788        0.765
##   986    146       1   0.6855  0.0427      0.59344        0.761
##   989    144       1   0.6807  0.0427      0.58896        0.756
##   990    141       4   0.6614  0.0426      0.57075        0.737
##   991    137       1   0.6566  0.0425      0.56619        0.733
##   992    136       1   0.6518  0.0425      0.56163        0.728
##   994    134       2   0.6420  0.0424      0.55244        0.718
##   995    132       2   0.6323  0.0423      0.54324        0.709
##   996    129       3   0.6176  0.0422      0.52933        0.694
##   998    125       1   0.6127  0.0421      0.52465        0.689
##   999    123       1   0.6077  0.0421      0.51993        0.684
```

```
##   1000    122       1   0.6027  0.0420      0.51521        0.679
##   1001    120       1   0.5977  0.0420      0.51044        0.674
##   1003    118       1   0.5926  0.0419      0.50564        0.669
##   1004    117       1   0.5875  0.0419      0.50083        0.664
##   1005    117       1   0.5825  0.0418      0.49607        0.659
##   1006    115       2   0.5724  0.0417      0.48647        0.649
##   1010    110       1   0.5672  0.0416      0.48152        0.644
##   1011    107       1   0.5619  0.0416      0.47647        0.639
##   1012    104       2   0.5511  0.0415      0.46616        0.628
##   1013     99       1   0.5455  0.0414      0.46083        0.622
##   1014    100       1   0.5401  0.0414      0.45563        0.617
##   1015     96       1   0.5344  0.0413      0.45024        0.611
##   1017     95       1   0.5288  0.0413      0.44486        0.606
##   1018     94       2   0.5176  0.0411      0.43412        0.595
##   1019     92       2   0.5063  0.0410      0.42340        0.583
##   1020     86       1   0.5004  0.0410      0.41777        0.577
##   1021     83       1   0.4944  0.0409      0.41198        0.571
##   1023     81       2   0.4822  0.0408      0.40027        0.559
##   1024     77       1   0.4759  0.0407      0.39425        0.553
##   1027     77       1   0.4697  0.0407      0.38833        0.547
##   1029     78       1   0.4637  0.0406      0.38260        0.541
##   1030     76       1   0.4576  0.0405      0.37679        0.535
##   1033     71       1   0.4512  0.0405      0.37061        0.528
##   1040     67       4   0.4242  0.0402      0.34473        0.501
##   1041     63       3   0.4040  0.0400      0.32549        0.481
##   1043     60       1   0.3973  0.0399      0.31911        0.474
##   1044     58       1   0.3905  0.0398      0.31261        0.467
##   1047     56       1   0.3835  0.0397      0.30600        0.460
##   1056     48       2   0.3675  0.0396      0.29053        0.445
##   1063     42       1   0.3588  0.0396      0.28197        0.436
##   1064     40       1   0.3498  0.0396      0.27320        0.427
##   1068     38       2   0.3314  0.0396      0.25526        0.409
##   1070     36       1   0.3222  0.0396      0.24639        0.400
##   1072     35       1   0.3130  0.0395      0.23758        0.391
##   1073     33       1   0.3035  0.0394      0.22851        0.382
##   1074     32       1   0.2940  0.0393      0.21951        0.372
##   1083     30       1   0.2842  0.0392      0.21021        0.362
##   1084     29       1   0.2744  0.0391      0.20099        0.353
##   1085     28       2   0.2548  0.0386      0.18281        0.333
##   1086     25       1   0.2446  0.0384      0.17339        0.323
##   1089     23       1   0.2340  0.0382      0.16358        0.312
##   1093     21       1   0.2228  0.0380      0.15331        0.301
##   1097     21       1   0.2122  0.0376      0.14378        0.290
##   1115     15       1   0.1981  0.0377      0.13025        0.276
##   1122     13       1   0.1828  0.0377      0.11581        0.262
##   1131     12       1   0.1676  0.0375      0.10190        0.247
##   1132     11       1   0.1524  0.0371      0.08852        0.232
##   1142     10       1   0.1371  0.0364      0.07569        0.217
##   1152      8       1   0.1200  0.0356      0.06140        0.200
##   1172      7       1   0.1028  0.0344      0.04808        0.182
##   1192      4       1   0.0771  0.0341      0.02752        0.161
##   1200      3       2   0.0257  0.0239      0.00243        0.108
```

survival curve

```
plot(fit_age$time, fit_age$surv, type='l', xlab='months', ylab='S(t)', xlim=c(800,1200), ylim
=c(0,1), col='red')
par(new=T)
plot(fit_age, xlab='months', ylab='S(t)', xlim=c(800,1200), ylim=c(0,1))
```