MATH 284/FMPH 291 Survival Analysis Supplementary Learing Materials

Yuyao Wang

2024-04-28

Tutorial

Here is the link of a tutorial for survival analysis in R by Emily C. Zabor: https://www.emilyzabor.com/tutorials/survival_analysis_in_r_tutorial.html#Part_1:_Introduction_to_Survival_Analysis

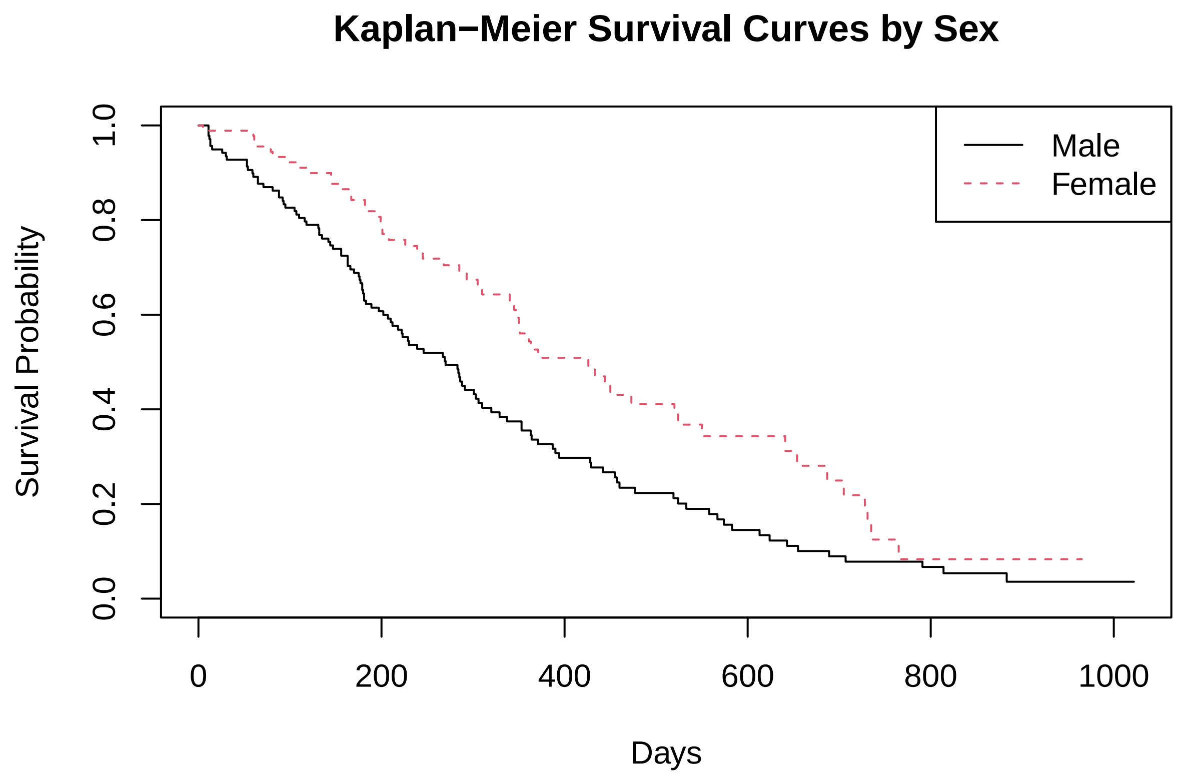
Part 1 in the above tutorial includes creating survival objects and curve, Kaplan-Meier plots, Estimating x-year survival, estimating median survival time, comparing survival times between groups (using log-rank test), and Cox regression model.

Below are some additional material for different topics.

Kaplan-Meier Curve

The tutorial uses the '`ggsurvfit`' R package for doing Kaplan-Meier plots. Alternatively, survival plots can be created using base R or the '`survminer`' package. For example, below is an example of fiting KM curves for the two sex groups using the 'lung' data set from the '`survival`' R package. You may find more details of the usage of the functions (for example how to add confidence intervals) by checking the R document for `survfit()` and `plot.survfit()`.

```r
# Load the survival package
library(survival)
# Fit the Kaplan-Meier model stratified by sex
# The lung dataset is automatically available after loading the survival package
km_fit <- survfit(Surv(time, status) ~ sex, data = lung)
# Plotting the Kaplan-Meier curves for two sex groups
plot(km_fit, main = "Kaplan-Meier Survival Curves by Sex",
        xlab = "Days", ylab = "Survival Probability", col = c(1, 2), lty = 1:2)
# Adding a legend to the plot
legend("topright", legend = c("Male", "Female"), col = c(1, 2), lty = 1:2)
```



Log-rank test

The function `survdiff()` used in the above tutorial for comparing survival times between groups is a function that conduct G-rho family tests. The default `rho=0`, which corresponds to log-rank test. More details can be found in the R documentation of this package. An alternative way of doing logrank test is using the score test for coxph score test. Try the code below, and you will find the two approaches gives exactly the same results!

BTW, `survdiff()` does not handle left truncated data, but `coxph()` can properly handle it by specifying the survival object to be `Surv(Q,X,Delta)`, where Q is the left truncation time, X is the censored event time, and Delta is the event indicator.

```
# logrank test using G^\rho test with rho = 0
logrank.1 = survdiff(Surv(time, status) ~ sex, data = lung)
print(logrank.1)
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ sex, data = lung)
##
##         N Observed Expected (O-E)^2/E (O-E)^2/V
## sex=1 138      112     91.6      4.55      10.3
## sex=2  90       53     73.4      5.68      10.3
##
##  Chisq= 10.3  on 1 degrees of freedom, p= 0.001
```

```
# logrank test using coxph score test
coxphfit = coxph(Surv(time, status) ~ sex, data = lung)
summary(coxphfit)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ sex, data = lung)
##
##   n= 228, number of events= 165
##
##        coef exp(coef) se(coef)      z Pr(>|z|)
## sex -0.5310    0.5880   0.1672 -3.176  0.00149 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
##     exp(coef) exp(-coef) lower .95 upper .95
## sex     0.588      1.701    0.4237     0.816
##
## Concordance= 0.579  (se = 0.021 )
## Likelihood ratio test= 10.63  on 1 df,   p=0.001
## Wald test            = 10.09  on 1 df,   p=0.001
## Score (logrank) test = 10.33  on 1 df,   p=0.001
```

The dual between estimating T distribution and estimating C distribution

The observed censored event time $X = \min(T, C)$ not only contains information of the event time T, but also information of the censoring time C. The even indicator $\Delta = I(T < C)$. In other words, when $\Delta = 1$, $X = T$; when $\Delta = 0$, $X = C$.

When we focus on estimating T distribution, T is censored by C, we construct survival object using `Surv(X, delta)`. When we focus on estimating C distribution, C can be viewed as censored by T. With this point of view think about how can we construct the survival object.

Notes on the exponenetial model and weibull model fitted using `survreg()`

The location-scale parameterization of a Weibull distribution found in `survreg` is not the same as the parameterization of rweibull. See the Example for `survreg()` and the chunk for `survreg.distributions` in ‘`survival`’ R package documentation for more details on how to interpret the outputs.

When fitting an exponential model using `fit = survreg(..., dist = "exponential")`, `1/exp(fit$icoef)` is the `rexp` rate parameter. Try the exmaple below.

```r
n=10000
lambda = 3
TT = rexp(n, rate = lambda)
C = runif(n, 1, 2)
X = pmin(TT,C)
delta = as.numeric(TT<C)

dat = data.frame(X = X, delta = delta)

fit = survreg(Surv(X, delta)~1, data = dat, dist = "exponential")
1/exp(fit$icoef)
```

```
## (Intercept)
##    2.994122
```