Survival Analysis

Rui

2026-05-26

目录

1 这章在讲什么

生存分析（survival analysis）解决的是「事件什么时候发生」类的问题。这类数据的特殊之处在于**右删失**（right censoring）：研究结束时一部分被试还没发生事件（比如复发、死亡、离职），你只知道「至少活到了这个时间」，不知道确切的事件时间。直接把这些观测当作「永远没事件」会低估事件率，扔掉他们又损失信息——生存分析的 Kaplan–Meier 估计和 Cox 回归专门处理这种数据。

心理学的典型用法：跟踪干预 / 治疗的效果在多长时间内保持、研究戒断（戒烟、戒毒）的「复发时间」、教育研究里的「辍学时间」。本章用 `survival` 包自带的 `lung` 数据集（肺癌患者）演示 K–M 估计 + log-rank 检验 + Cox 比例风险模型完整流程。

两个关键对象

生存函数 $S(t) = P(T > t)$：到时刻 t 还没发生事件的概率，KM 估计：$\hat{S}(t) = \prod_{t_i \le t}(1 - \frac{d_i}{n_i})$，其中 d_i 是 t_i 时发生事件的人数，n_i 是 t_i 之前还在「风险集」里的人数。

风险函数 $h(t) = \lim_{\Delta t \to 0} \frac{P(t \le T < t+\Delta t \mid T \ge t)}{\Delta t}$：到了 t 还活着，下一瞬间发生事件的「瞬时率」。Cox 模型假设 $h(t \mid X) = h_0(t)\exp(\beta' X)$——协变量只乘一个常数因子在基线风险上（**比例风险假设**）。系数 $\exp(\beta_j)$ 解读为风险比（HR）：X_j 增加 1 单位时风险翻几倍。

2 Packages

- Survival Analysis
  - survival
  - survminer
- Plots of Survival Analysis
  - ggfortify
  - ggplot

```
library(survival)
library(survminer)
library(ggplot2)
library(ggfortify)
```

Dataset “lung” in package “survival” is of interest, which contains the information about survival in patients with advanced lung cancer from the North Central Cancer Treatment Group. Performance scores rate how well the patient can perform usual daily activities. Variables include:

- inst: Institution code
- time: Survival time in days
- status: censoring status `1=censored`, `2=dead`
- age: Age in years
- sex: `Male=1 Female=2`
- ph.ecog: ECOG performance score as rated by the physician. `0=asymptomatic`, `1= symptomatic` but completely ambulatory, `2= in bed <50% of the day`, `3= in bed > 50% of the day but not bedbound`, `4 = bedbound`
- ph.karno: Karnofsky performance score (`bad=0-good=100`) rated by physician
- pat.karno: Karnofsky performance score as rated by patient
- meal.cal: Calories consumed at meals
- wt.loss: Weight loss in last six months (pounds)

```
attach(lung)
```

3 Survival Object

`survival::Surv()` creates a survival object, with censored data marked with "`+`" following up censoring time and death cases simply showing the death time. The result is usually used as a response variable in a model formula.

Two most important parameters:

- `time` :For right censored data, this is the follow up time. For interval data, the first argument is the starting time for the interval.
- `event`: The status indicator, normally `0=alive`, `1=dead`. Other choices are `TRUE/FALSE` (`TRUE = death`) or `1/2` (`2=death`). For interval censored data, the status indicator is 0=right censored, 1=event at time, 2=left censored, 3=interval censored. For multiple endpoint data the event variable will be a factor, whose first level is treated as censoring. Although unusual, the event indicator can be

omitted, in which case all subjects are assumed to have an event.

```
survobj = Surv(time, status==2)
```

4 Surv Model

For the overall survival analysis, just specify the null hypothesis with `Surv~1`, i.e., NO groups.

```
sfit <- survfit(Surv(time, status==2)~1, data=lung)
```

The fitted survival model itself will tell you the results, with observations, events, median and 95% CI.

```
sfit
```

```
## Call: survfit(formula = Surv(time, status == 2) ~ 1, data = lung)
##
##         n events median 0.95LCL 0.95UCL
## [1,] 228    165    310     285     363
```

If more details are needed, i.e., the instantaneous events, and estimates of survival, std.err, 95% CI, just `summary(sfit)`.

Moreover, if `time` is designated, then the `summary()` function will return the estimated results.

```
summary(sfit,time = 200)
```

```
## Call: survfit(formula = Surv(time, status == 2) ~ 1, data = lung)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##   200    144      72     0.68  0.0311        0.622        0.744
```

However, if group is of interest, specify it in the formula with `Surv~group`.

```
sfit_sex = survfit(Surv(time, status==2) ~ sex, data = lung)
sfit_sex
```

```
## Call: survfit(formula = Surv(time, status == 2) ~ sex, data = lung)
##
##         n events median 0.95LCL 0.95UCL
## sex=1 138    112    270     212     310
## sex=2  90     53    426     348     550
```

From the model itself, we can get an overall picture of the differences between groups.

```
summary(sfit_sex,time = 365)
```

```
## Call: survfit(formula = Surv(time, status == 2) ~ sex, data = lung)
##
##                 sex=1
##         time       n.risk      n.event     survival      std.err lower 95% CI
##     365.0000      35.0000      85.0000       0.3361       0.0434       0.2609
## upper 95% CI
##       0.4329
##
##                 sex=2
##         time       n.risk      n.event     survival      std.err lower 95% CI
##     365.0000      30.0000      36.0000       0.5265       0.0597       0.4215
## upper 95% CI
##       0.6576
```

It's ideal that the 95% CI of groups doesn't intersect, and a significant result can be obtained instantly. However, that's not the case, take a step further and use Cox regression to see exactly.

5 Cox Regression

```
coxph(Surv(time, status==2)~sex)
```

```
## Call:
```

```
## coxph(formula = Surv(time, status == 2) ~ sex)
##
##        coef exp(coef) se(coef)      z       p
## sex -0.5310    0.5880   0.1672 -3.176 0.00149
##
## Likelihood ratio test=10.63  on 1 df, p=0.001111
## n= 228, number of events= 165
```

where "`exp(coef)`" is the HR (hazard ratio), which measures the relative risk between two groups.

Theoretically, it's better only open to binary variables in Cox regression. However, for categorical variables with more than 2 levels, use "`as.factor(cat_var)`" with "`model = TRUE`" to generate corresponding dummy variables. For continuous variables, they'll be included into the regression model as usual, but the model will be plotted with expected value of continuou variables.

However, similar to `coxph()` which returns the differences through the view of regression, the simpler version of such comparison can be accomplished by "`survdiff()`", also in package "survival". Both of the functions have the same formula.

From "`survdiff()`", we can get the intermediate products when computing the log rank, or the chi-square test. However, this wouldn't return the coefficient.

Jointly speaking, the combined results of two functions emphasize the fact that the log rank test (named Mantel-Haenszel test), is a chi-square test in essence.

```
surv_diff = survdiff(Surv(time, status==2)~sex)
surv_diff
```

```
## Call:
## survdiff(formula = Surv(time, status == 2) ~ sex)
##
```

```
##         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
```

```
surv_diff$chisq
```

```
## [1] 10.32674
```

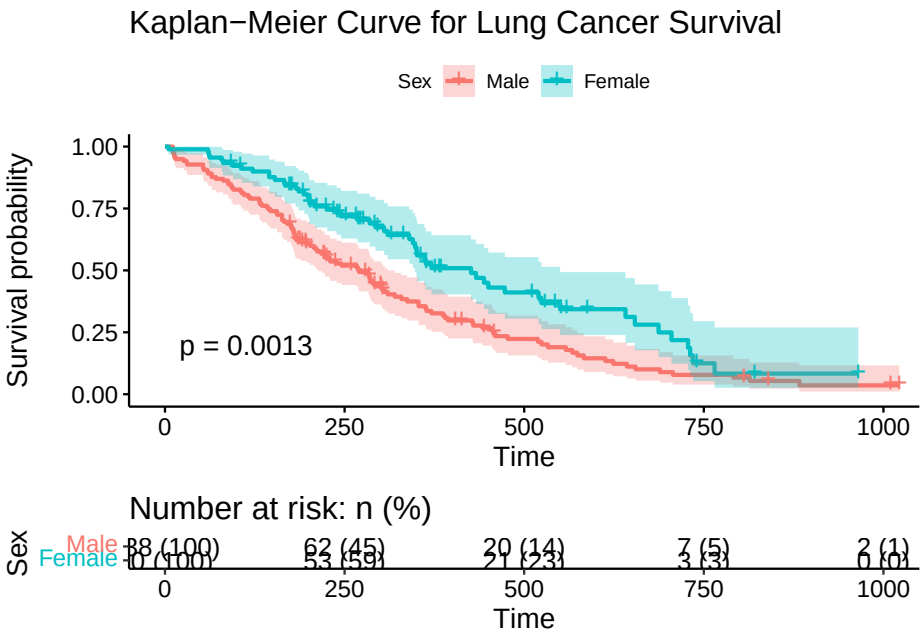
6 Surv Plots

Finally, we can visualize the survival analysis.

Note that even if analysed data has been attached, if the "`data`" isn't passed into `survfit()`, then an error will be raised.

The parameter "`risk.table`" provides you with the freedom to decide whether or not to add a risk table below the Kaplan-Meier curve(s). Meanwhile, various options are available!

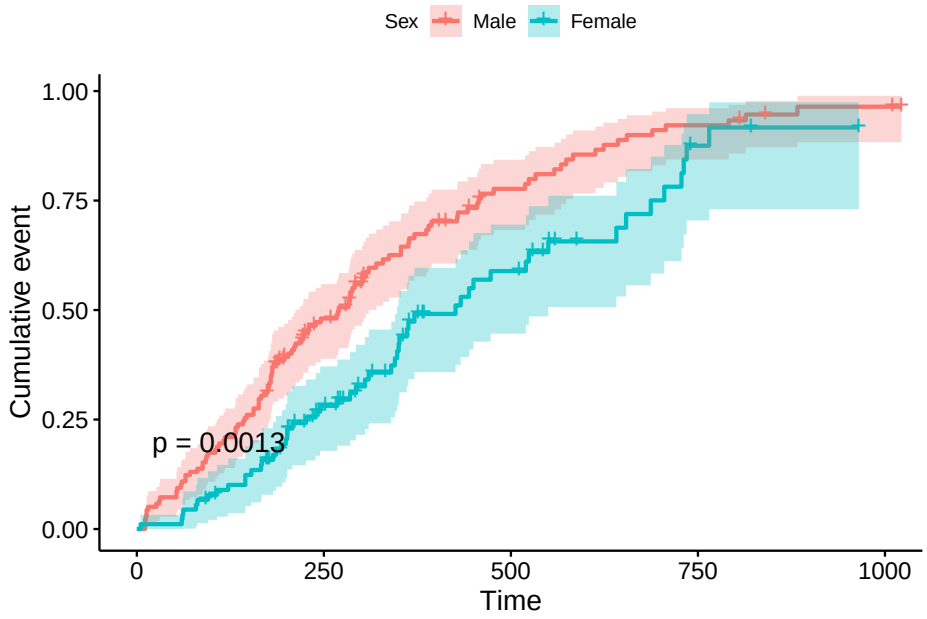
```
ggsurvplot(sfit_sex,data = lung,
           conf.int=TRUE, pval=TRUE,
           risk.table='abs_pct',
           legend.labs=c("Male", "Female"), legend.title="Sex",
           title="Kaplan-Meier Curve for Lung Cancer Survival")
```



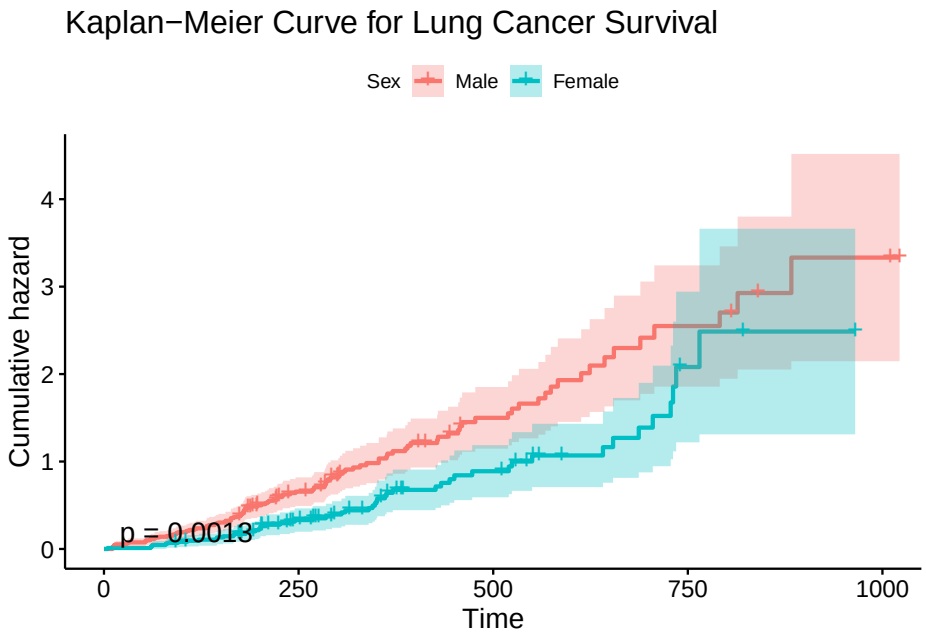
Moreover, the parameter "`fun`" is quite useful, which will define a transformation of the survival curve.

- "pct": for survival probability in percentage, **by default**.
- "event": plots cumulative events, $f(y) = 1 - y$.
- "cumhaz": plots the cumulative hazard function, $f(y) = -\log(y)$.

```
ggsurvplot(sfit_sex,data = lung,
           conf.int=TRUE, pval=TRUE,
           # risk.table='abs_pct',
           legend.labs=c("Male", "Female"), legend.title="Sex",
           # title="Kaplan-Meier Curve for Lung Cancer Survival",
           fun = 'event')
```




```
ggsurvplot(sfit_sex,data = lung,
           conf.int=TRUE, pval=TRUE,
           # risk.table='abs_pct',
           legend.labs=c("Male", "Female"), legend.title="Sex",
           title="Kaplan-Meier Curve for Lung Cancer Survival",
           fun = 'cumhaz')
```



7 容易踩的坑

生存分析最常见的三个误区

1. **忘了检验比例风险假设**：Cox 模型最核心的假设是「协变量的影响在所有时点上是常数比例」。检验方法：`cox.zph(fit)`，对每个协变量画 Schoenfeld 残差与时间的图，残差应该围绕水平线，$p > 0.05$。若违反，要么对应变量做分层（`strata()`）、要么引入时间交互项 $\beta(t) = \beta_0 + \beta_1 \cdot t$。本笔记上方代码没显式跑 `cox.zph` ——读者实际用时务必补上。
2. **把删失当作「没发生」**：原始数据里 `status` 列通常 1 = 删失 / 2 = 事件（或 0/1）。`Surv(time, status == 2)` 里必须显式说明「哪个值代表事件」。看错编码会让事件率算反，KM 曲线整个倒过来。
3. **log-rank 显著但 HR** ≈ 1：log-rank 检验对「曲线交叉」的差异最敏感，Cox 假设曲线不交叉（比例风险）。两组曲线在前期分得开、后期合并 → log-rank 可能显著但 HR 接近 1，因为 HR 是「平

均效应」。这种情况下不该用 Cox，可以试 AFT 模型（`survreg`）或分段 Cox（按时间段拆）。