

survival package

(AST405) Lifetime data analysis

Md Rasel Biswas

Lecture Outline

Subsection 1

Introduction to R package survival

Time-to-event data

- Time-to-event data are defined as $\{(t_i, \delta_i), i = 1, \dots, n\}$
 - ▶ $t_i \rightarrow$ observed time
 - ▶ $\delta_i \rightarrow$ censoring indicator (1=failure, 0=censored)

Time-to-event data

- Defining time and censoring indicator in R for the following sample of five observations

20, 13⁺, 10, 25, 18⁺

```
dat0 <- tibble(  
  time = c(20, 13, 10, 25, 18),  
  status = c(1, 0, 1, 1, 0)  
)
```

Time-to-event data

```
dat0
```

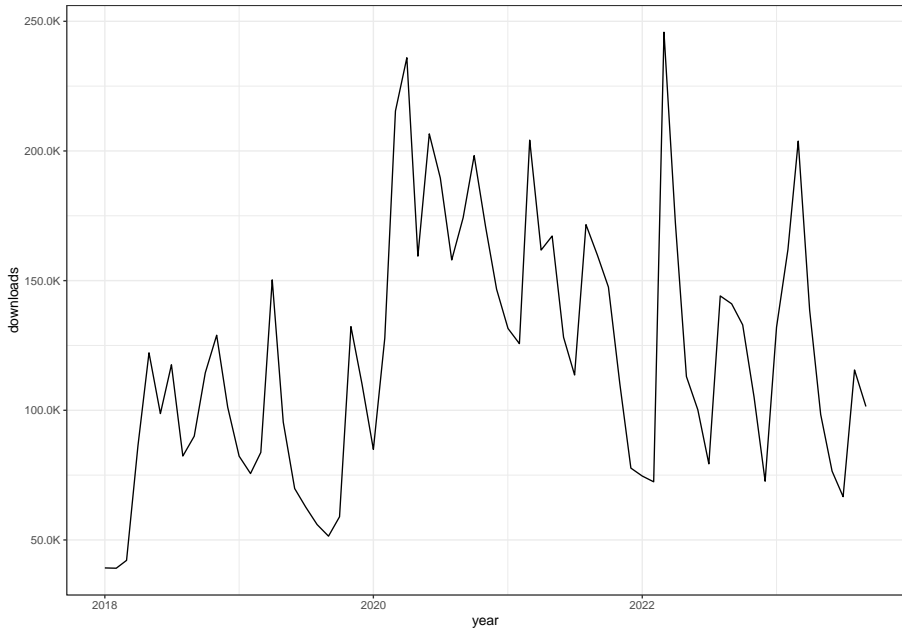
```
# A tibble: 5 x 2
```

```
  time status  
  <dbl> <dbl>  
1    20     1  
2    13     0  
3    10     1  
4    25     1  
5    18     0
```

survival package

- There are a number of R packages available for analyzing time-to-event data
- The `survival` package will be used for the course, the current (January 2025) version of survival package is 3.8-3
- Author, contributors, and maintainer of survival package
 - ▶ Terry M Therneau [aut, cre]
 - ▶ Thomas Lumley
 - ▶ Atkinson Elizabeth
 - ▶ Crowson Cynthia
- So far there are about 8.41 millions downloads of the survival package (between the years 2015 and 2023)

survival package



PL estimate

- `survival::survfit()` function is used to obtain non-parametric estimate survivor function

```
survfit(Surv(time, time2, event) ~ x, data)
```

- `Surv` → creates a survival object, which has arguments such as, `time`, `time2`, `event`, and `type` (censoring type “right”, “left”, “interval”, etc.)
- `time` → observed time (numeric object)
- `event` → censoring status (1=failure, 0=censored)
- `x` → strata (categorical variable), `x=1` when stratified analysis is not required (single curve)

PL estimate

- Some important arguments of `survfit()` function
 - ▶ `stype` → 1 (direct estimation of survivor function), 2 (survivor function estimated from cumulative hazard function)
 - ▶ `ctype` → method used to estimate cumulative hazard function (1=Nelson Aalen, 2=Fleming-Harrington)
 - ▶ `conf.type` → available options include (“none”, “plain”, “log” (default), “log-log”, and “logit”)

Example I

```
dim(gehan65)
```

```
[1] 42  3
```

```
print(gehan65, n = 6)
```

```
# A tibble: 42 x 3  
  time status drug  
  <dbl> <dbl> <chr>  
1     6     1 6-MP  
2     6     1 6-MP  
3     6     1 6-MP  
4     6     0 6-MP  
5     7     1 6-MP  
6     9     0 6-MP  
# i 36 more rows
```

Example 1

```
gehan65 %>%  
  count(drug, status)
```

```
# A tibble: 3 x 3
```

	drug	status	n
	<chr>	<dbl>	<int>
1	6-MP	0	12
2	6-MP	1	9
3	placebo	1	21

Example 1

```
dat6MP <- gehan65 %>%  
  filter(drug == "6-MP")
```

```
mod1 <- survfit(Surv(time = time, event = status) ~ 1,  
               conf.type = "plain",  
               data = dat6MP)
```

- Default `conf.type` is `log`

Example 1

```
names(mod1)
```

```
[1] "n"           "time"        "n.risk"      "n.event"     "n.censor"
[7] "std.err"    "cumhaz"     "std.chaz"   "type"        "logse"
[13] "conf.type"  "lower"      "upper"      "t0"          "call"
```

Example I

```
head(mod1$time)
```

```
[1] 6 7 9 10 11 13
```

```
head(mod1$n.risk)
```

```
[1] 21 17 16 15 13 12
```

```
head(mod1$n.event)
```

```
[1] 3 1 0 1 0 1
```

```
head(mod1$surv)
```

```
[1] 0.8571429 0.8067227 0.8067227 0.7529412 0.7529412 0.6901961
```

Example 1

```
names(summary(mod1))
```

```
[1] "n"                "time"            "n.risk"          "n.event"  
[5] "n.censor"        "surv"            "std.err"         "cumhaz"  
[9] "std.chaz"        "type"            "logse"           "conf.int"  
[13] "conf.type"       "lower"           "upper"           "t0"  
[17] "call"            "table"           "rmean.endtime"
```


Example I

```
summary(mod1)$table
```

```
  records      n.max    n.start    events    rmean se(rmean) 23
21.000000 21.000000 21.000000  9.000000 23.287395  2.827468 23
0.95UCL
      NA
```

Example 1

```
# broom::tidy(mod1)
print(broom::tidy(mod1), n = 6)
```

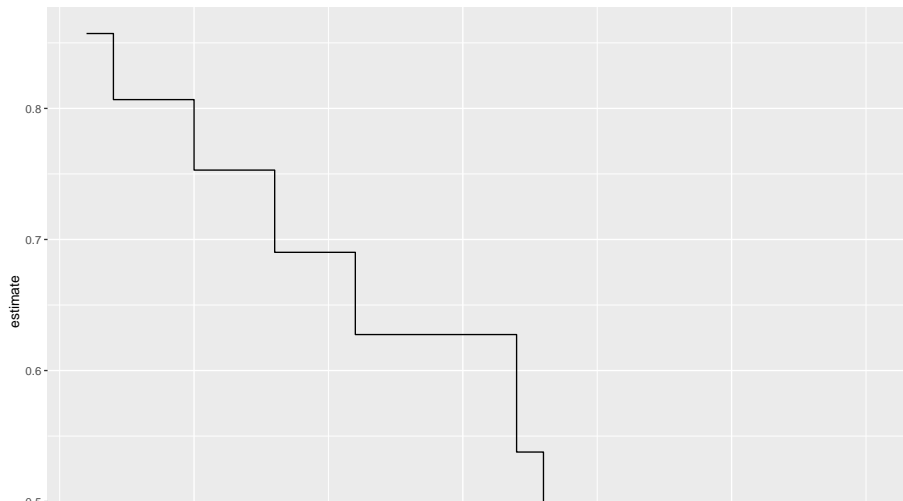
```
# A tibble: 16 x 8
```

	time	n.risk	n.event	n.censor	estimate	std.error	conf.high	conf.low
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	6	21	3	1	0.857	0.0891	1	0.977
2	7	17	1	0	0.807	0.108	0.977	0.942
3	9	16	0	1	0.807	0.108	0.977	0.942
4	10	15	1	1	0.753	0.128	0.942	0.900
5	11	13	0	1	0.753	0.128	0.942	0.900
6	13	12	1	0	0.690	0.155	0.900	0.857

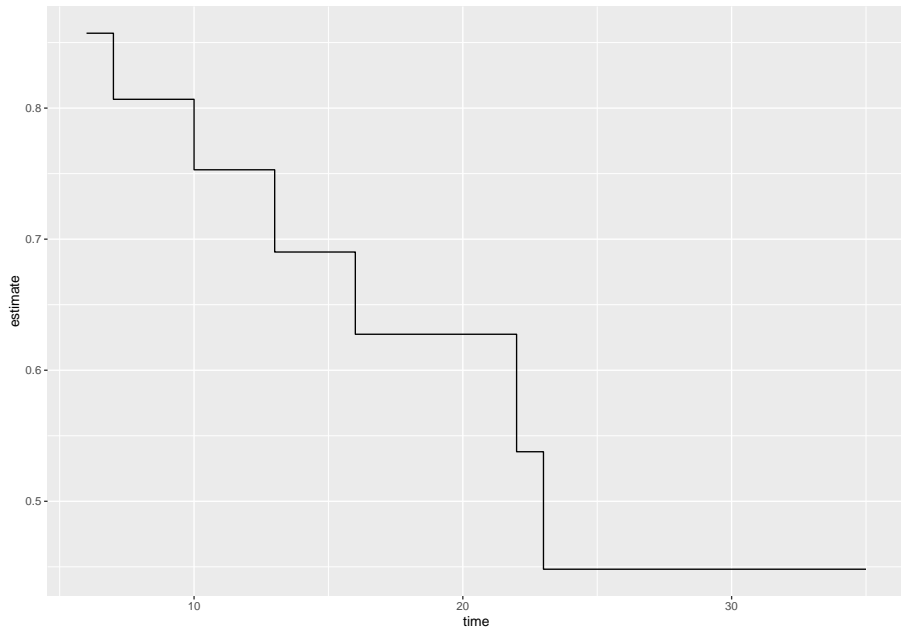
```
# i 10 more rows
```

Plot of survivor function

```
broom::tidy(mod1) %>%  
  ggplot() +  
  geom_step(aes(time, estimate))
```

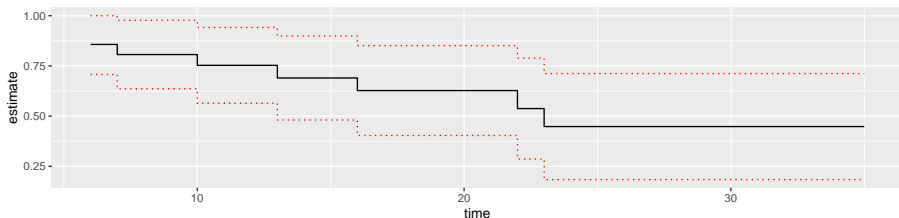


Plot of survivor function

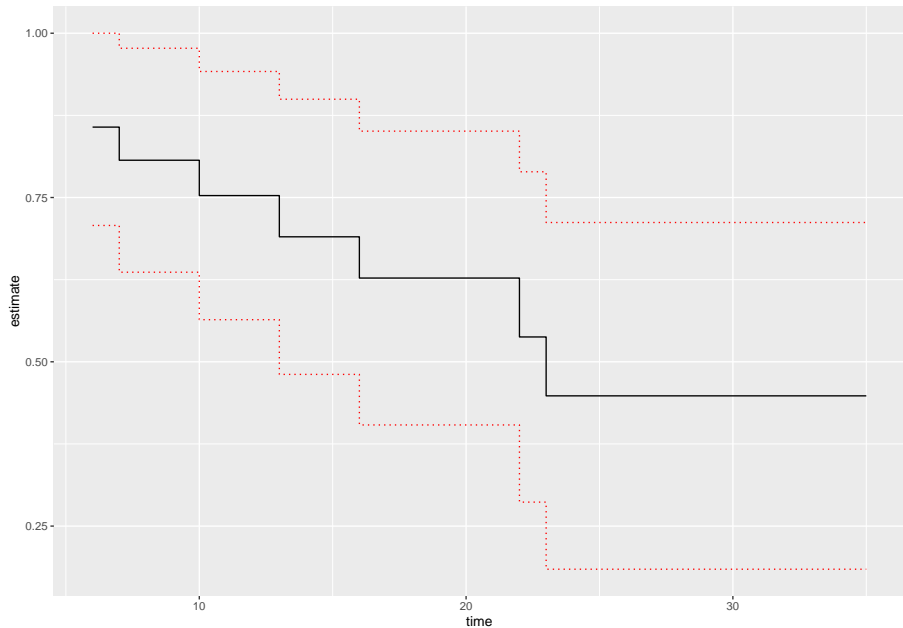


Plot of survivor function

```
tidy(mod1) %>%  
  ggplot() +  
  geom_step(aes(time, estimate)) +  
  geom_step(aes(time, conf.high), linetype = "dotted", col = "red") +  
  geom_step(aes(time, conf.low), linetype = "dotted", col = "red")
```



Plot of survivor function



Quantiles

```
qsurv <- function(mod, p) {  
  tmod <- broom::tidy(mod)  
  if (min(tmod$estimate) > (1 - p)) return(NA_real_)  
  else {  
    tmod %>%  
      filter(estimate <= (1 - p)) %>%  
      pull(time) %>%  
      min()  
  }  
}
```

Quantiles

- First quartile

```
qsurv(mod1, .25)
```

```
[1] 13
```

- Median

```
qsurv(mod1, .5)
```

```
[1] 23
```


Example 1

```
mod2 <- survfit(  
  Surv(time = time, event = status) ~ drug,  
  data = gehan65,  
  conf.type = "plain"  
)
```

Example 1

```
broom::tidy(mod2)
```

```
# A tibble: 28 x 9
```

```
   time n.risk n.event n.censor estimate std.error conf.high conf.low strata
<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>
1     6     21      3       1  0.857  0.0891     1     0.707 drug=6-MP
2     7     17      1       0  0.807  0.108     0.977  0.636 drug=6-MP
3     9     16      0       1  0.807  0.108     0.977  0.636 drug=6-MP
4    10     15      1       1  0.753  0.128     0.942  0.564 drug=6-MP
5    11     13      0       1  0.753  0.128     0.942  0.564 drug=6-MP
6    13     12      1       0  0.690  0.155     0.900  0.481 drug=6-MP
7    16     11      1       0  0.627  0.182     0.851  0.404 drug=6-MP
8    17     10      0       1  0.627  0.182     0.851  0.404 drug=6-MP
9    19      9      0       1  0.627  0.182     0.851  0.404 drug=6-MP
10   20      8      0       1  0.627  0.182     0.851  0.404 drug=6-MP
# i 18 more rows
```

Example 1

```
broom::tidy(mod2) %>%  
  select(time, estimate, strata)
```

```
# A tibble: 28 x 3  
  time estimate strata  
  <dbl>   <dbl> <chr>  
1     6     0.857 drug=6-MP  
2     7     0.807 drug=6-MP  
3     9     0.807 drug=6-MP  
4    10     0.753 drug=6-MP  
5    11     0.753 drug=6-MP  
6    13     0.690 drug=6-MP  
7    16     0.627 drug=6-MP  
8    17     0.627 drug=6-MP  
9    19     0.627 drug=6-MP  
10   20     0.627 drug=6-MP  
# i 18 more rows
```

Example 1

```
broom::tidy(mod2) %>%  
  ggplot() +  
  geom_step(aes(time, estimate, col = strata))
```

