Survival Analysis HW1

313657003 周佳萱

2025-03-23

Problem 1.

(a)

$$S(t) = \exp\left(-\int_0^t \lambda(u)\,du\right) = \exp\left(-\int_0^t c\,du\right) = e^{-ct}$$
$$f(t) = -\frac{dS(t)}{dt} = -\frac{de^{-ct}}{dt} = ce^{-ct}$$

Median failure time for c=2, 5, 11

Let t_η is median failure time,

$$\begin{aligned} 0.5 &= 1 - e^{-ct_\eta} \\ -\log 2 &= -ct_\eta \\ t_\eta &= \frac{\log 2}{c} \end{aligned}$$

```
c = c(2,5,11)
t = log(2) / c
t
```

```
## [1] 0.34657359 0.13862944 0.06301338
```

$c = 2$, $t_\eta = \frac{\log 2}{2} = 0.3466$
$c = 5$, $t_\eta = \frac{\log 2}{5} = 0.1386$
$c = 11$, $t_\eta = \frac{\log 2}{11} = 0.0630$

(b)

$$f(t) = -\frac{dS(t)}{dt} = -\frac{d\exp(-\theta t^\beta)}{dt} = \theta\beta t^{\beta-1}\exp(-\theta t^\beta)$$
$$\lambda(t) = \frac{f(t)}{S(t)} = \frac{\theta\beta t^{\beta-1}\exp(-\theta t^\beta)}{\exp(-\theta t^\beta)} = \theta\beta t^{\beta-1}$$

Problem 2.

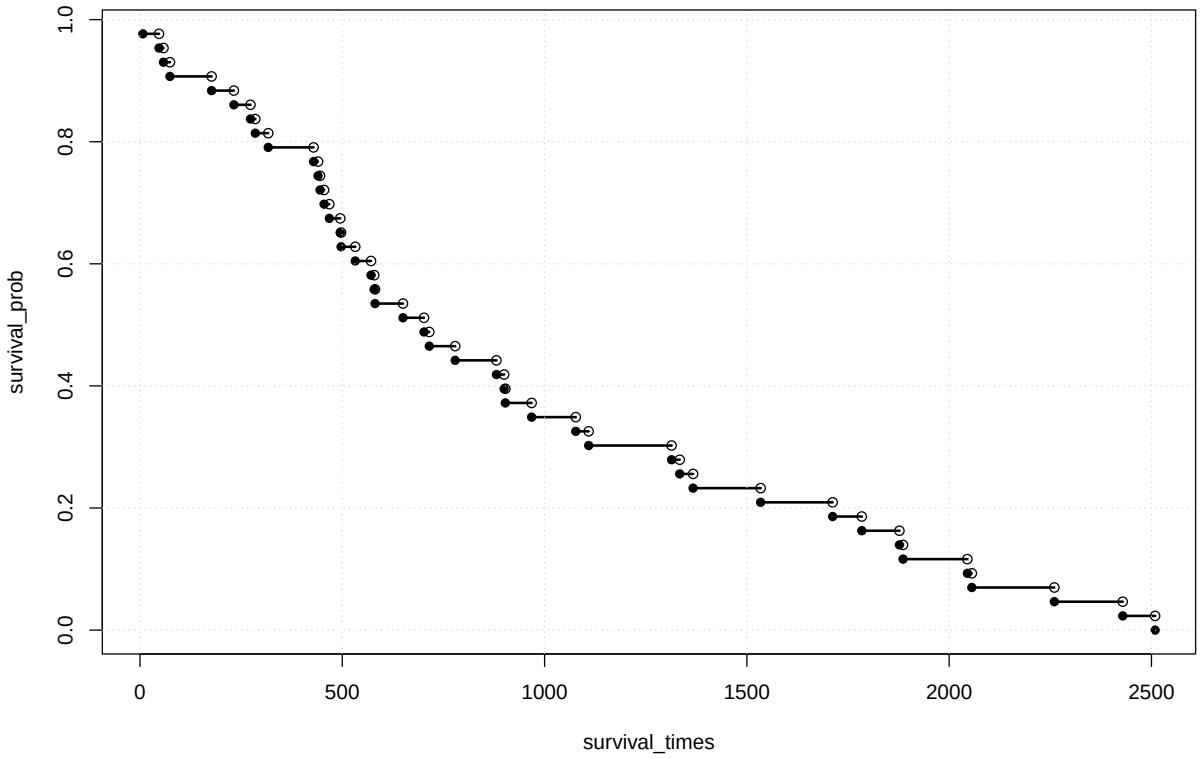
(a) Calculate the empirical survival function and graph it.

```
library(knitr)

survival_times <- c(7, 47, 58, 74, 177, 232, 273, 285, 317, 429, 440,
                    445, 455, 468, 495, 497, 532, 571, 579, 581, 650, 702,
                    715, 779, 881, 900, 903, 968, 1077, 1109, 1314, 1334,
                    1367, 1534, 1712, 1784, 1877, 1886, 2045, 2056, 2260,
                    2429, 2509)

n <- length(survival_times)

survival_prob <- ((n-1):0) / n

survival_table_individual <- data.frame(Survival_Time = survival_times, Survival_Prob = survival_prob)

kable(survival_table_individual, align = "c")
```

Survival_Time	Survival_Prob
7	0.9767442
47	0.9534884
58	0.9302326
74	0.9069767
177	0.8837209
232	0.8604651
273	0.8372093
285	0.8139535
317	0.7906977
429	0.7674419
440	0.7441860
445	0.7209302
455	0.6976744
468	0.6744186
495	0.6511628
497	0.6279070
532	0.6046512
571	0.5813953
579	0.5581395
581	0.5348837
650	0.5116279
702	0.4883721
715	0.4651163
779	0.4418605
881	0.4186047
900	0.3953488
903	0.3720930
968	0.3488372
1077	0.3255814
1109	0.3023256

Survival_Time	Survival_Prob
1314	0.2790698
1334	0.2558140
1367	0.2325581
1534	0.2093023
1712	0.1860465
1784	0.1627907
1877	0.1395349
1886	0.1162791
2045	0.0930233
2056	0.0697674
2260	0.0465116
2429	0.0232558
2509	0.0000000

```
plot(survival_times, survival_prob, pch = 16, col = "black")
points(survival_times[-1], ((n-1):1) / n, pch = 1, col = "black")
segments(survival_times[-n], survival_prob[-n], survival_times[-1], ((n-1):1) / n
         , col = "black", lwd = 2)
grid()
```



(b) Estimate the 1-year, 2-year, 3-year, 4-year and 5-year survival rates by non-parametric methods.

```
survival_table <- data.frame(Year = 1:5, Rate = rep(0, 5))

# Survival Rate
for(i in 1:5){
  survival_table$Rate[i] <- length(survival_times[survival_times > 365 * i]) / n
}


library(knitr)

kable(survival_table
      , caption = "Estimated Survival Rates"
      , align = "c")
```

Table 2: Estimated Survival Rates

Year	Rate
1	0.7906977
2	0.4651163
3	0.3255814
4	0.2325581
5	0.1627907

(c) Under nonparametric model, construct 95% con dence intervals for the 1-year, 2-year, 3-year, 4-year and 5-year survival rates.

```
survival_table$LowerCI <- rep(0, 5)
survival_table$UpperCI <- rep(0, 5)

for(i in 1:5){
  sur_exp <- survival_table$Rate[i]
  sur_var <- (sur_exp * (1 - sur_exp)) / n

  survival_table$LowerCI[i] <- sur_exp - (1.96 * sqrt(sur_var))
  survival_table$UpperCI[i] <- sur_exp + (1.96 * sqrt(sur_var))
}

survival_table <- survival_table[, c("Year", "LowerCI", "Rate", "UpperCI")]

kable(survival_table, caption = "Estimated Survival Rates with 95% Confidence Interval", align = "c")
```

Table 3: Estimated Survival Rates with 95% Confidence Interval

Year	LowerCI	Rate	UpperCI
1	0.6691031	0.7906977	0.9122922
2	0.3160318	0.4651163	0.6142007
3	0.1855207	0.3255814	0.4656421
4	0.1062852	0.2325581	0.3588311
5	0.0524454	0.1627907	0.2731360

(d) Estimate the median survival time.

According to Table 1 in (a), we can see that $S(650) = 0.5116279$ and $S(702) = 0.4883721$, so the median survival time is 650.

(e) Under the exponential model with the constant hazard λ, calculate the maximum likelihood estimate of λ and construct a 95% confidence interval for λ.

$$
\begin{aligned}
&X \sim \text{Exp}(\lambda), \quad f(x \mid \lambda) = \lambda e^{-\lambda x} \\
&\Rightarrow \mathcal{L}(\lambda \mid x) = \prod_{i=1}^{43} f(x_i \mid \lambda) = \lambda^{43} e^{-\lambda \sum_{i=1}^{43} x_i} \\
&\Rightarrow \log \mathcal{L}(\lambda \mid x) = 43 \log \lambda - \lambda \sum_{i=1}^{43} x_i \\
&\Rightarrow \frac{\partial}{\partial \lambda} \log \mathcal{L}(\lambda \mid x) = \frac{43}{\lambda} - \sum_{i=1}^{43} x_i \equiv 0 \\
&\Rightarrow \hat{\lambda} = \frac{43}{\sum_{i=1}^{43} x_i} \\
&\Rightarrow \text{The maximum likelihood of } \lambda \text{ is } \frac{43}{\sum_{i=1}^{43} x_i}
\end{aligned}
$$

$$
\begin{aligned}
&\sum_{i=1}^{43} X_i \sim \Gamma(43, \lambda) \\
&\Rightarrow 2\lambda \sum_{i=1}^{43} X_i \sim \Gamma(43, \frac{1}{2}) \equiv \chi^2(86) \\
&\Rightarrow 0.95 = P(\chi^2_{0.025}(86) \leq 2\lambda \sum_{i=1}^{43} x_i \leq \chi^2_{0.975}(86)) \\
&\Rightarrow 0.95 = P\left(\frac{\chi^2_{0.025}(86)}{2\sum_{i=1}^{43} x_i} \leq \lambda \leq \frac{\chi^2_{0.975}(86)}{2\sum_{i=1}^{43} x_i} \right)
\end{aligned}
$$

```r
chi_low <- qchisq(0.025, df = 86)
chi_high <- qchisq(0.975, df = 86)

sum_survival <- sum(survival_times)

LowerCI <- chi_low / (2 * sum_survival)
```

```r
UpperCI <- chi_high / (2 * sum_survival)

cat("95% Confidence Interval for Lambda:\n")
```

```
## 95% Confidence Interval for Lambda:
```

```r
cat("Lower Bound:", LowerCI, "\n")
```

```
## Lower Bound: 0.0007828167
```

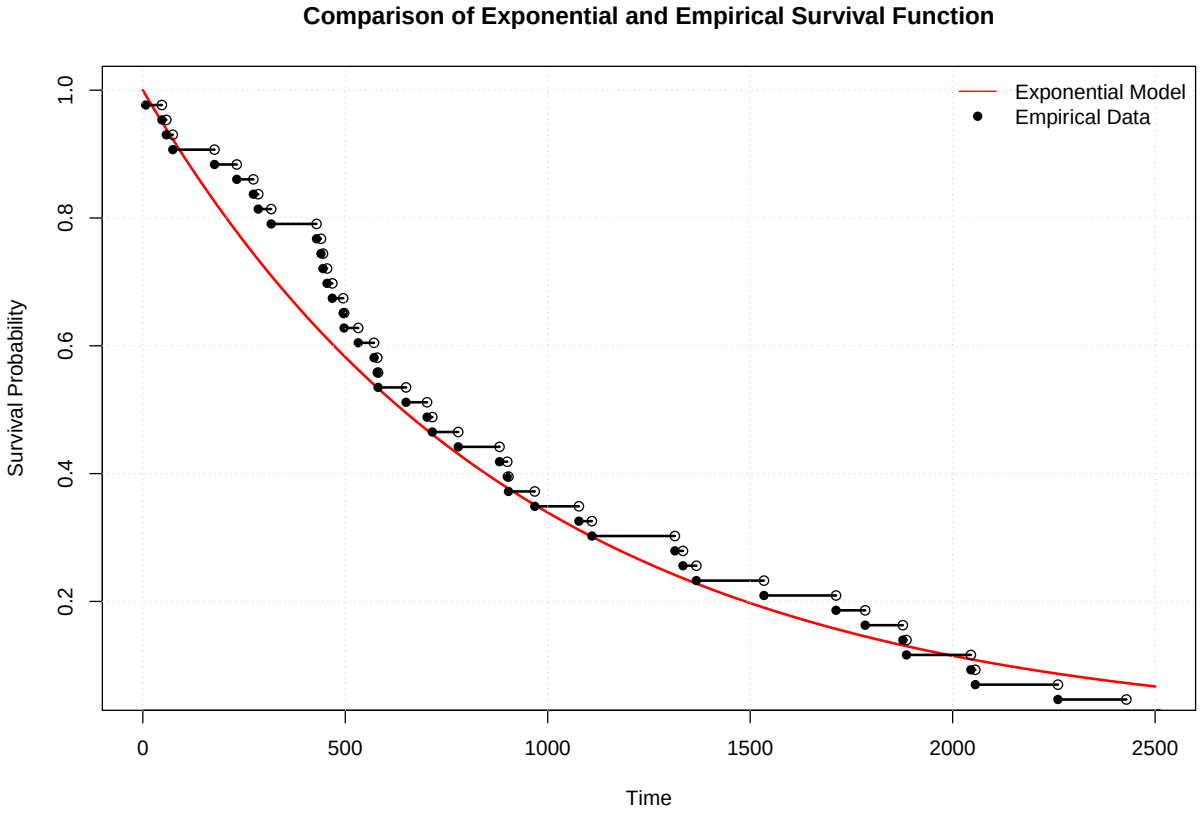
```r
cat("Upper Bound:", UpperCI, "\n")
```

```
## Upper Bound: 0.001428114
```

95% Confidence Interval for λ is $(0.000783, 0.001428)$

(f)Do you think the exponential model is appropriate for the data? Explain.

```r
# MLE
mle <- 43 / sum_survival

exp_fun <- function(x) {
  exp(-mle * x)
}

x <- 0:2500
exp_val <- exp_fun(x)

# Exponential
plot(x, exp_val, type = "l", col = "red", lwd = 2,
     xlab = "Time", ylab = "Survival Probability",
     main = "Comparison of Exponential and Empirical Survival Function")

# Original Survival
points(survival_times, survival_prob, pch = 16, col = "black")
points(survival_times[-1], ((n-1):1) / n, pch = 1, col = "black")
segments(survival_times[-n], survival_prob[-n], survival_times[-1], ((n-1):1) / n
         , col = "black", lwd = 2)

grid()

legend("topright", legend = c("Exponential Model", "Empirical Data"),
       col = c("red", "black"), lty = c(1, NA), pch = c(NA, 16), bty = "n")
```



Compared the exponential survival function with the empirical survival function, we observe that they have a similar shape. Therefore, the exponential model is appropriate for this data.

Problem 3

$$
\begin{aligned}
m(t) &= E(T - t \mid T > t) \\
&= \int_t^{\infty} (T - t) \frac{f(T)}{S(t)} dT \\
&= \int_t^{\infty} T \frac{f(T)}{S(t)} dT - \int_t^{\infty} \frac{t f(T)}{S(t)} dT \\
&= \frac{1}{S(t)} [\int_t^{\infty} T f(T) dT] - t
\end{aligned}
$$

$$
\begin{aligned}
m'(t) &= \frac{-t f(t) S(t) + f(t) \int_t^{\infty} T f(T) dT}{S(t)^2} \\
&= \frac{f(t)}{S^2(t)} \int_t^{\infty} T f(T) dT - \frac{t f(t)}{S(t)} - 1
\end{aligned}
$$

then, we can make the relationship between $m(t)$ and $m'(t)$

$$\frac{m(t)f(t)}{S(t)} = \frac{f(t)\int_t^\infty Tf(T)dT}{S(t)^2} - \frac{tf(t)}{S(t)}$$

$$\Rightarrow m'(t) = \frac{m(t)f(t)}{S(t)} - 1$$

$$\Rightarrow m'(t) + 1 = \frac{f(t)}{S(t)}m(t)$$

$$\Rightarrow -\frac{S'(t)}{S(t)} = \frac{m'(t)}{m(t)} + \frac{1}{m(t)}$$

$$\Rightarrow -\log S(t) = \log m(t) + \int_0^t \frac{1}{m(u)}du + c$$

$$\Rightarrow S(t) = \frac{1}{m(t)}\exp\left(-\int_0^t \frac{1}{m(u)}du - c\right)$$

$$\Rightarrow 1 = S(0) = \frac{1}{m(0)}e^{-c} \Rightarrow \ c = -\log m(0)$$

$$\therefore S(t) = \frac{1}{m(t)}\exp\left(-\int_0^t \frac{1}{m(u)}du + \log m(0)\right) = \frac{m(0)}{m(t)}\exp\left(-\int_0^t \frac{1}{m(u)}du\right)$$

Problem 4

(a) Calculate both the Kaplan-Meier estimate and the empirical survival distribution on the basis of the placebo data. What is the relationship between the Kaplan-Meier estimate and the empirical survival distribution from your calculation?

```
library(knitr)

placebo <- c(1, 1, 2, 2, 3, 4, 4, 5, 5, 8, 8, 8, 8, 11, 11, 12, 12, 15, 17, 22, 23)
n_placebo <- length(placebo)

time_points <- sort(unique(placebo))

placebo_survival_table <- data.frame(
  Time = time_points,
  KM_Estimate = numeric(length(time_points)),
  Empirical_Survival = numeric(length(time_points))
)

last_survival <- 1

#  KM and Empirical Survival
for (i in seq_along(time_points)) {
  t <- time_points[i]
  d <- sum(placebo == t)
  n_risk <- sum(placebo >= t)

  s_t <- last_survival * (1 - d / n_risk)
```

```r
  placebo_survival_table[i, "KM_Estimate"] <- s_t
  last_survival <- s_t

  placebo_survival_table[i, "Empirical_Survival"] <- sum(placebo > t) / n_placebo
}


kable(
  placebo_survival_table,
  caption = "Kaplan-Meier Estimate and Empirical Survival Distribution for Placebo Group",
  align = "c",
)
```

Table 4: Kaplan-Meier Estimate and Empirical Survival Distribution for Placebo Group

Time	KM_Estimate	Empirical_Survival
1	0.9047619	0.9047619
2	0.8095238	0.8095238
3	0.7619048	0.7619048
4	0.6666667	0.6666667
5	0.5714286	0.5714286
8	0.3809524	0.3809524
11	0.2857143	0.2857143
12	0.1904762	0.1904762
15	0.1428571	0.1428571
17	0.0952381	0.0952381
22	0.0476190	0.0476190
23	0.0000000	0.0000000

There is no difference between KM estimator and empirical survival function on placebo group.

(b) Do you observe a similar phenomenon which also holds for other uncensored survival data? Explain.

Under uncensored data, empirical survival function is same as Kaplan-Meier estimator.

$$
\begin{aligned}
\hat{S}_{KM}(t) &= \prod_{y_{(j)} \le t} \left(1 - \frac{d_{(j)}}{N_{(j)}}\right) \\
&= \prod_{y_{(j)} \le t} \frac{\sum_{i=1}^{N} I(y_i > y_{(j)})}{\sum_{i=1}^{N} I(y_i \ge y_{(j)})} \\
&= \prod_{y_{(j)} \le t} \frac{\sum_{i=1}^{N} I(y_i > y_{(j)})/N}{\sum_{i=1}^{N} I(y_i > y_{(j-1)})/N} \quad \because \text{uncensored} \\
&= \prod_{y_{(j)} \le t} \frac{\hat{S}_e(y_{(j)})}{\hat{S}_e(y_{(j-1)})} \\
&= \frac{\hat{S}_e(y_{(1)})}{\hat{S}_e(y_{(0)})} \times \cdots \times \frac{\hat{S}_e(y_{(i-1)})}{\hat{S}_e(y_{(i-2)})} \\
&= \frac{\hat{S}_e(y_{(i-1)})}{\hat{S}_e(y_{(0)})} \\
&= \hat{S}_e(y_{(i-1)}) \\
&= \sum_{i=1}^{N} \frac{I(y_s > y_{(i-1)})}{N} \\
&= \hat{S}_e(t)
\end{aligned}
$$

(c) Calculate the Kaplan-Meier estimate on the basis of the 6MP data.

```r
MP<-c(6, 6, 6, 7, 10, 13, 16, 22, 23,6, 9, 10, 11, 17, 19, 20, 25, 32, 32, 34, 35)
unsensor_MP <- c(6, 6, 6, 7, 10, 13, 16, 22, 23)

n_MP <- length(MP)

time_points <- sort(unique(unsensor_MP))

MP_survival_table <- data.frame(
  Time = time_points,
  KM_Estimate = numeric(length(time_points))
)

last_survival <- 1

# KM and Empirical Survival
for (i in seq_along(time_points)) {
  t <- time_points[i]
  d <- sum(unsensor_MP == t)
  n_risk <- sum(MP >= t)

  s_t <- last_survival * (1 - d / n_risk)
  MP_survival_table[i, "KM_Estimate"] <- s_t
  last_survival <- s_t
}
```

```
kable(
  MP_survival_table,
  caption = "Kaplan-Meier Estimate for MP Group",
  align = "c",
)
```

Table 5: Kaplan-Meier Estimate for MP Group

Time	KM_Estimate
6	0.8571429
7	0.8067227
10	0.7529412
13	0.6901961
16	0.6274510
22	0.5378151
23	0.4481793

(d) Use the variance estimates that you learn from class to produce a 95% confidence interval at each observed failure time for the empirical survival distribution on the basis of the placebo data. Comment on your results.

```
placebo_survival_table$LowerCI <- rep(0, nrow(placebo_survival_table))
placebo_survival_table$UpperCI <- rep(0, nrow(placebo_survival_table))

for(i in 1:nrow(placebo_survival_table)){

  sur_exp <- placebo_survival_table$Empirical_Survival[i]
  sur_var <- (sur_exp * (1 - sur_exp)) / n_placebo
  sur_low <- sur_exp - (1.96 * sqrt(sur_var))
  sur_upper <- sur_exp + (1.96 * sqrt(sur_var))

  placebo_survival_table$LowerCI[i] <- ifelse(sur_low <= 0, 0, sur_low)
  placebo_survival_table$UpperCI[i] <- ifelse(sur_upper >= 1, 1, sur_upper)
}

placebo_survival_table_CI <- placebo_survival_table[, c("Time", "LowerCI", "Empirical_Survival", "UpperC

kable(placebo_survival_table_CI
      , caption = "Empirical Survival with 95% Confidence Interval (Placebo Group)"
      , align = "c")
```

Table 6: Empirical Survival with 95% Confidence Interval (Placebo Group)

Time	LowerCI	Empirical_Survival	UpperCI
1	0.7792113	0.9047619	1.0000000
2	0.6415732	0.8095238	0.9774744
3	0.5797367	0.7619048	0.9440728

Time	LowerCI	Empirical_Survival	UpperCI
4	0.4650436	0.6666667	0.8682897
5	0.3597685	0.5714286	0.7830887
8	0.1732489	0.3809524	0.5886559
11	0.0924959	0.2857143	0.4789326
12	0.0225256	0.1904762	0.3584268
15	0.0000000	0.1428571	0.2925234
17	0.0000000	0.0952381	0.2207887
22	0.0000000	0.0476190	0.1387031
23	0.0000000	0.0000000	0.0000000

When calculating the 95% confidence interval for each empirical survival point, we observe that some lower bounds fall below 0. It occurs because the sample size is relatively small (only 21 data points), making the normal approximation less accurate in the tails.

(e) Use the variance estimates that you learn from class to produce a 95% condence interval, at each uncensored time, for the the Kaplan-Meier estimate on the basis of the 6MP data. Comment on your results.

```
time_points <- sort(unique(unsensor_MP))

MP_survival_table$LowerCI <- rep(0,length(time_points))
MP_survival_table$UpperCI <- rep(0,length(time_points))

temp <- 0

for (i in seq_along(time_points)) {
  t <- time_points[i]

  d <- sum(unsensor_MP == t)

  n_risk <- sum(MP >= t)

  temp <- temp + d/(n_risk*(n_risk-d))
  km <- MP_survival_table$KM_Estimate[i]

  sur_var <- km^2 * temp

  lower<-km-1.96*sqrt(sur_var)
  upper<-km+1.96*sqrt(sur_var)

  MP_survival_table$LowerCI[i] <- ifelse(lower < 0, 0, lower)
  MP_survival_table$UpperCI[i] <- ifelse(upper > 1, 1, upper)

}

MP_survival_table_CI <- MP_survival_table[, c("Time", "LowerCI", "KM_Estimate", "UpperCI")]


kable(
```

```
  MP_survival_table_CI,
  caption = "Kaplan-Meier Estimate with 95% Confidence Interval (6MP Group)",
  align = "c",
)
```

Table 7: Kaplan-Meier Estimate with 95% Confidence Interval (6MP Group)

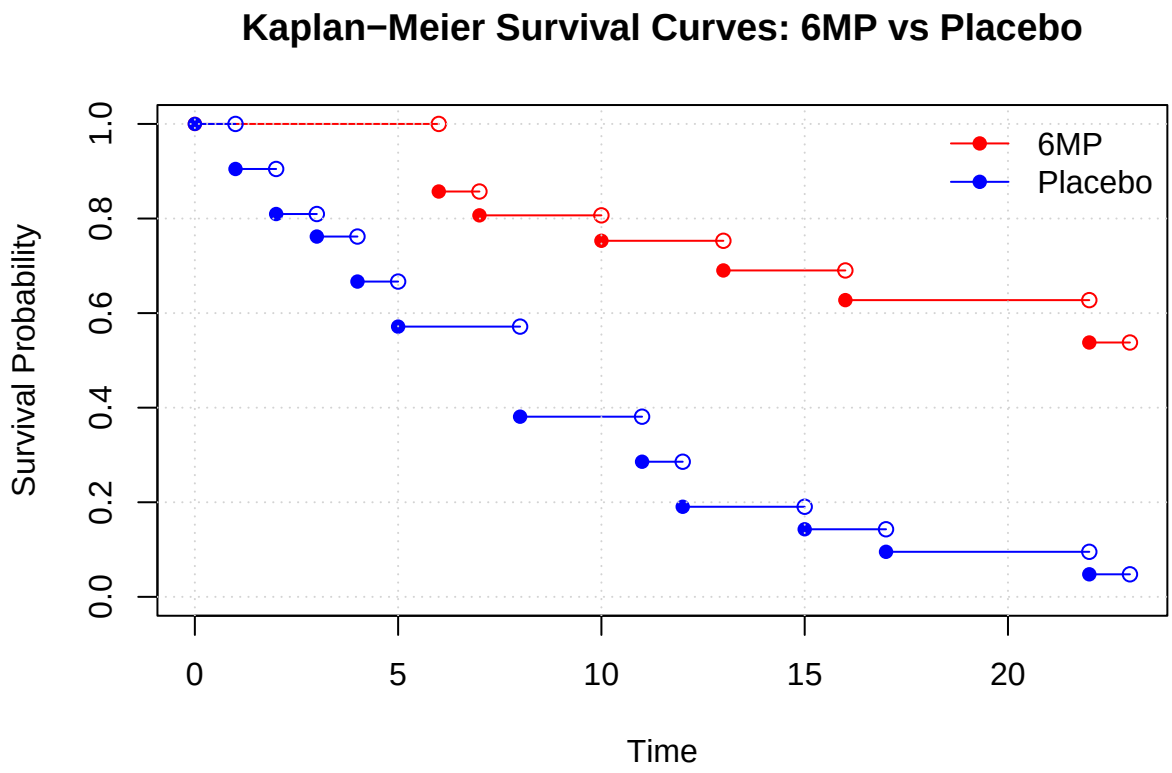
Time	LowerCI	KM_Estimate	UpperCI
6	0.7074766	0.8571429	1.0000000
7	0.6363295	0.8067227	0.9771158
10	0.5640959	0.7529412	0.9417865
13	0.4808393	0.6901961	0.8995529
16	0.4039054	0.6274510	0.8509966
22	0.2864770	0.5378151	0.7891533
23	0.1843800	0.4481793	0.7119785

Compared to part (d), the confidence intervals are less likely to go beyond [0,1]. Still, with only 21 data points, the intervals widen over time.

(f) Compare the survival function estimates from the placebo and 6MP groups. Interpret your results.

```
plot(0, 0, type = "n", xlim = c(0, max(MP_survival_table$Time, placebo_survival_table$Time)),
     ylim = c(0, 1), xlab = "Time", ylab = "Survival Probability",
     main = "Kaplan-Meier Survival Curves: 6MP vs Placebo")

# --- 6MP Group ---

points(c(0, MP_survival_table$Time[-nrow(MP_survival_table)])
       , c(1, MP_survival_table$KM_Estimate[-nrow(MP_survival_table)])
       , pch = 16, col = "red")

points(MP_survival_table$Time,
       c(1, MP_survival_table$KM_Estimate[-nrow(MP_survival_table)]),
       pch = 1, col = "red")

segments(x0 = c(0, MP_survival_table$Time[-nrow(MP_survival_table)]),
         y0 = c(1, MP_survival_table$KM_Estimate[-nrow(MP_survival_table)]),
         x1 = MP_survival_table$Time,
         y1 = c(1, MP_survival_table$KM_Estimate[-nrow(MP_survival_table)]),
         col = "red")

# --- Placebo Group ---

points(c(0, placebo_survival_table$Time[-nrow(placebo_survival_table)])
       , c(1, placebo_survival_table$KM_Estimate[-nrow(placebo_survival_table)])
       , pch = 16, col = "blue")
points(placebo_survival_table$Time,
```

```
       c(1, placebo_survival_table$KM_Estimate[-nrow(placebo_survival_table)]),
       pch = 1, col = "blue")

segments(x0 = c(0, placebo_survival_table$Time[-nrow(placebo_survival_table)]),
         y0 = c(1, placebo_survival_table$KM_Estimate[-nrow(placebo_survival_table)]),
         x1 = placebo_survival_table$Time,
         y1 = c(1, placebo_survival_table$KM_Estimate[-nrow(placebo_survival_table)]),
         col = "blue")
grid()

legend("topright", legend = c("6MP", "Placebo"),
       col = c("red", "blue"), pch = 16, lty = 1, bty = "n")
```



From the graph above, we can see that the survival function of the 6MP treatment group is higher than that of the placebo group. It indicates that leukemia patients who received 6MP treatment had a higher survival rate compared to those who were given a placebo.

Problem 5. Simulate survival data and compute the Nelson-Aalen estimator.

(a) Simulate 100 event times with constant hazard 0.9 (using rexp()). Also simulate 100 censoring times with uniform distribution on [0,5]. Next, compute the observed times and an event indicator. How many observed events/censorings do you have?

```
set.seed(123)

event_time <- rexp(100, 0.9) #T
censor_time <- runif(100, 0, 5) #C

observed_time <- pmin(event_time, censor_time)   #Y
event_indicator <- as.numeric(event_time <= censor_time) #I


n_event <- sum(event_indicator)
n_censor <- sum(event_indicator == 0)


cat("Number of observed events:", n_event, "\n")
```

```
## Number of observed events: 85
```

```
cat("Number of censored observations:", n_censor, "\n")
```

```
## Number of censored observations: 15
```

(b) compute Nelson-Aalen (cumulative hazard).

```
data <- data.frame(time = observed_time, status = event_indicator)

event_times <- sort(unique(data$time[data$status == 1]))

hazard_table <- data.frame(
  time = event_times,
  d = numeric(length(event_times)),
  n_risk = numeric(length(event_times)),
  d_over_n = numeric(length(event_times)),
  cum_hazard = numeric(length(event_times)),
  survival = numeric(length(event_times))
)


cum_hazard <- 0
for (i in seq_along(event_times)) {
  t <- event_times[i]
```

```
  d_j <- sum(data$time == t & data$status == 1)
  n_j <- sum(data$time >= t)
  d_over_n <- d_j / n_j
  cum_hazard <- cum_hazard + d_over_n

  hazard_table$d[i] <- d_j
  hazard_table$n_risk[i] <- n_j
  hazard_table$d_over_n[i] <- d_over_n
  hazard_table$cum_hazard[i] <- cum_hazard
  hazard_table$survival[i] <- exp(-cum_hazard)
}

kable(
  head(hazard_table[, c("time", "cum_hazard", "survival")], 10),
  caption = "Nelson-Aalen Cumulative Hazard Estimate (First 10 Events)",
  align = "c",
)
```

Table 8: Nelson-Aalen Cumulative Hazard Estimate (First 10 Events)

time	cum_hazard	survival
0.0051101	0.0100000	0.9900498
0.0323927	0.0202041	0.9799987
0.0350860	0.0305134	0.9699475
0.0352975	0.0409300	0.9598963
0.0467648	0.0514563	0.9498451
0.0469683	0.0620946	0.9397939
0.0624566	0.0728473	0.9297428
0.0748621	0.0837169	0.9196916
0.1006571	0.0948280	0.9095294
0.1095215	0.1060640	0.8993671

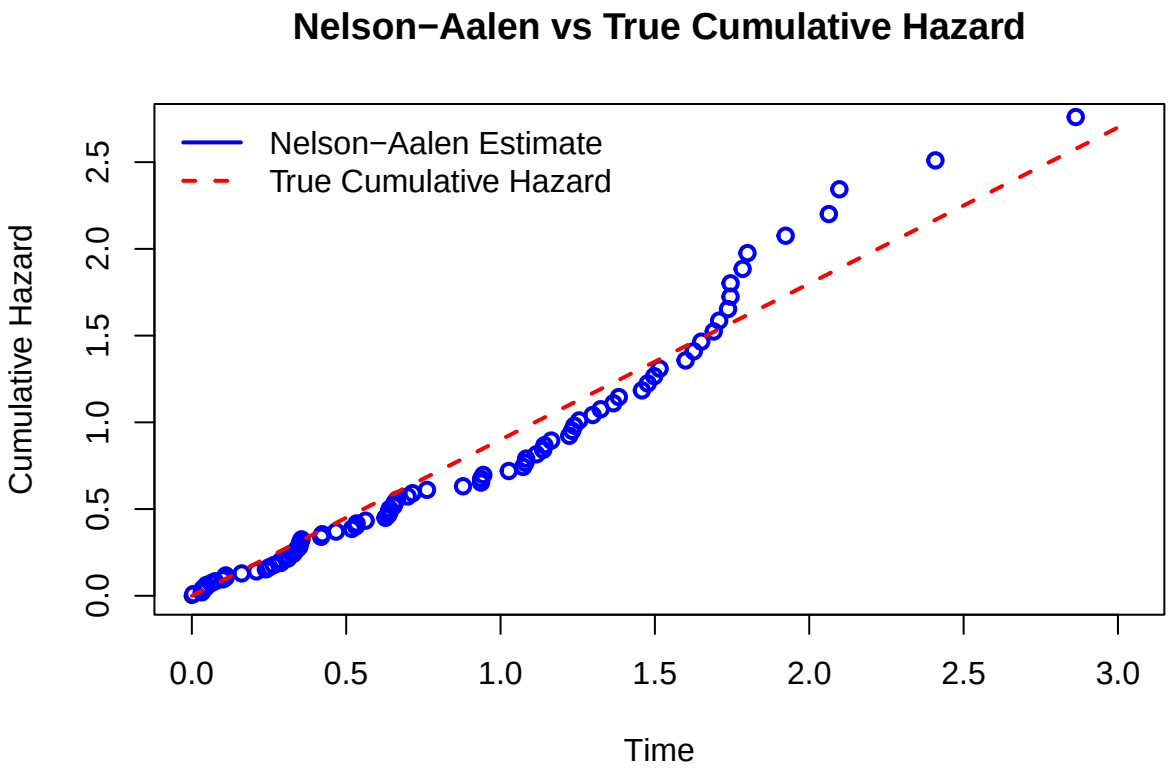
(c) Plot the Nelson-Aalen estimator. Compare it to the true cumulative hazard using curve. For comparison, you may want to restrict plotting to time regions where there is more data. E.g., restrict to the true median of the survival distribution.

$$f(t) = 0.9exp(-0.9t)$$
$$S(t) = exp(-0.9t)$$
$$\therefore \Lambda(t) = 0.9t$$

True cumulative hazard is $\Lambda(t) = 0.9t$

```
max_time <- max(hazard_table$time)

plot(0, 0, xlab = "Time", ylab = "Cumulative Hazard",
     main = "Nelson-Aalen vs True Cumulative Hazard",
     xlim = c(0, max_time), ylim = c(0, 0.9 * max_time))
```

```r
points(hazard_table$time, hazard_table$cum_hazard, col = "blue", lwd = 2,)
curve(0.9 * x, from = 0, to = max_time, col = "red", lty = 2, lwd = 2, add = TRUE)


legend("topleft", legend = c("Nelson-Aalen Estimate", "True Cumulative Hazard"),
       col = c("blue", "red"), lty = c(1, 2), lwd = 2, bty = "n")
```



When t $<$ 1.5, Nelson-Aalen estimate closely follows the true cumulative hazard line. When t $>$ 1.5, Nelson-Aalen estimate begins to deviate from the true curve.

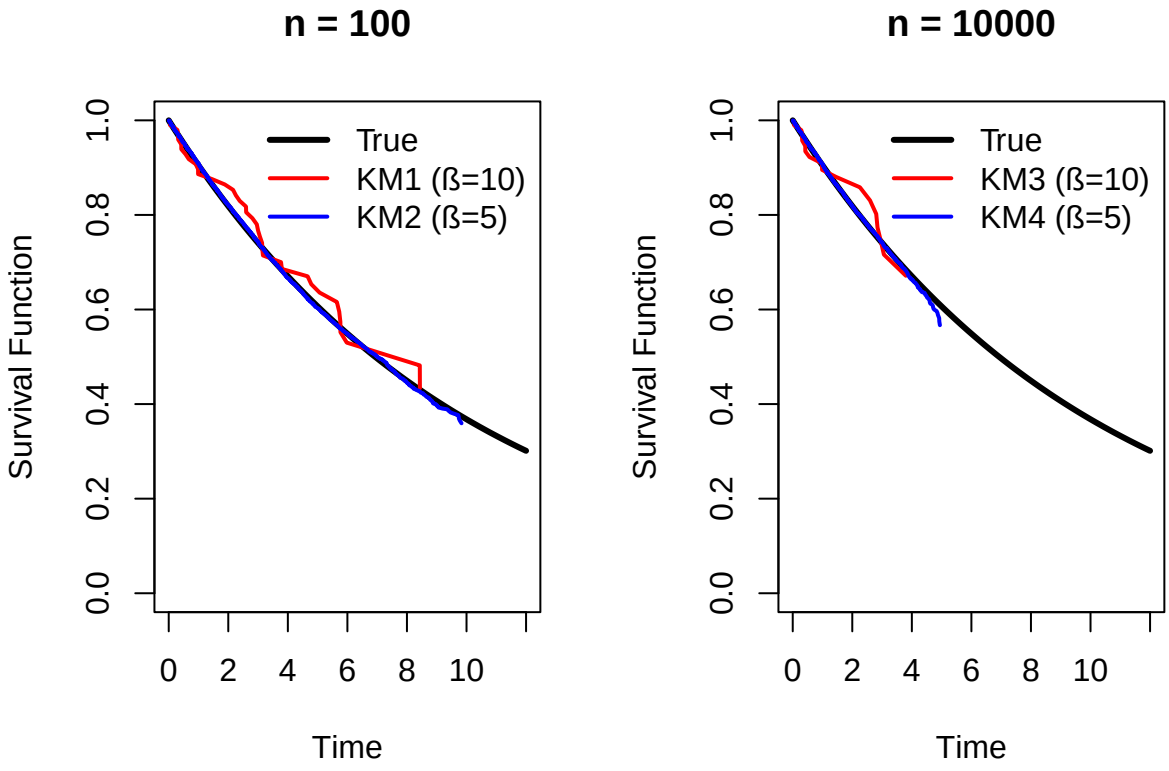
Problem 6.

(a) Reproduce simulation study in Updated Survival Notes, Second Chapter, Page30. (Attached Code)

```r
set.seed(123)

n_list <- c(100, 10000)
lmbd <- 0.1
beta_list <- c(10, 5)
label_matrix <- matrix(c("KM1", "KM2", "KM3", "KM4"), nrow = 2)

true_surv <- function(t) exp(-lmbd * t)
```

```r
KM1 <- KM2 <- KM3 <- KM4 <- NULL

for (ni in seq_along(n_list)) {
  n <- n_list[ni]
  event_time <- rexp(n, rate = lmbd)

  for (bi in seq_along(beta_list)) {
    beta <- beta_list[bi]
    label <- label_matrix[ni, bi]

    censor_time <- runif(n, 0, beta)
    observed_time <- pmin(event_time, censor_time)
    event_indicator <- as.numeric(event_time <= censor_time)
    data <- data.frame(time = observed_time, status = event_indicator)

    time_points <- sort(unique(data$time[data$status == 1]))

    KM_df <- data.frame(
      time = time_points,
      true = true_surv(time_points),
      KM = numeric(length(time_points))
    )

    km <- 1
    for (j in seq_along(time_points)) {
      t <- time_points[j]
      d <- sum(data$time == t & data$status == 1)
      n_risk <- sum(data$time >= t)
      km <- km * (1 - d / n_risk)
      KM_df$KM[j] <- km
    }

    assign(label, KM_df)
  }
}

par(mfrow = c(1, 2))  # 1 列 2 欄

# ----- Left：n = 100 -----
plot(0, 0, type = "n",
     xlim = c(0, 12), ylim = c(0, 1),
     xlab = "Time", ylab = "Survival Function",
     main = "n = 100")


curve(true_surv(x), from = 0, to = 12, col = "black", lwd = 3, add = TRUE)

lines(KM1$time, KM1$KM, col = "red", lwd = 2)
lines(KM2$time, KM2$KM, col = "blue", lwd = 2)

legend("topright",
       legend = c("True", "KM1 ( =10)", "KM2 ( =5)"),
       col = c("black", "red", "blue"),
```

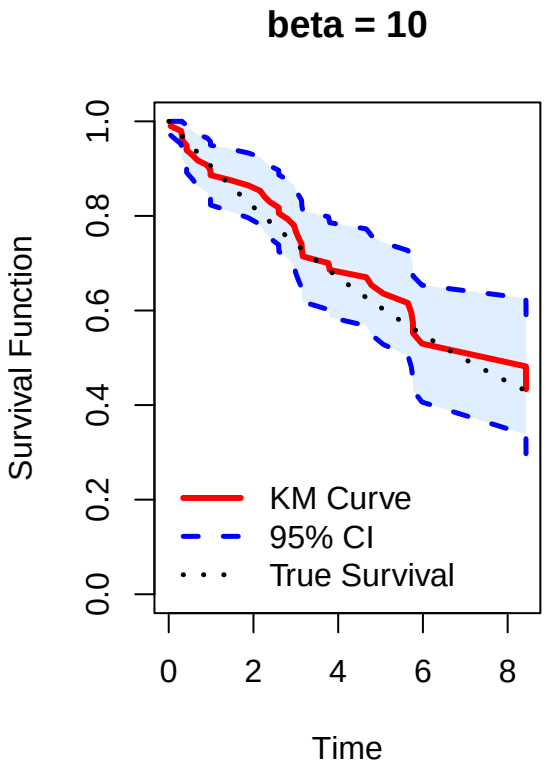
```
       lty = 1, lwd = c(3,2,2), bty = "n")

# ----- Right : n = 10000 -----
plot(0, 0, type = "n",
     xlim = c(0, 12), ylim = c(0, 1),
     xlab = "Time", ylab = "Survival Function",
     main = "n = 10000")

curve(true_surv(x), from = 0, to = 12, col = "black", lwd = 3, add = TRUE)

lines(KM3$time, KM3$KM, col = "red", lwd = 2)
lines(KM4$time, KM4$KM, col = "blue", lwd = 2)

legend("topright",
       legend = c("True", "KM3 ( =10)", "KM4 ( =5)"),
       col = c("black", "red", "blue"),
       lty = 1, lwd = c(3,2,2), bty = "n")
```

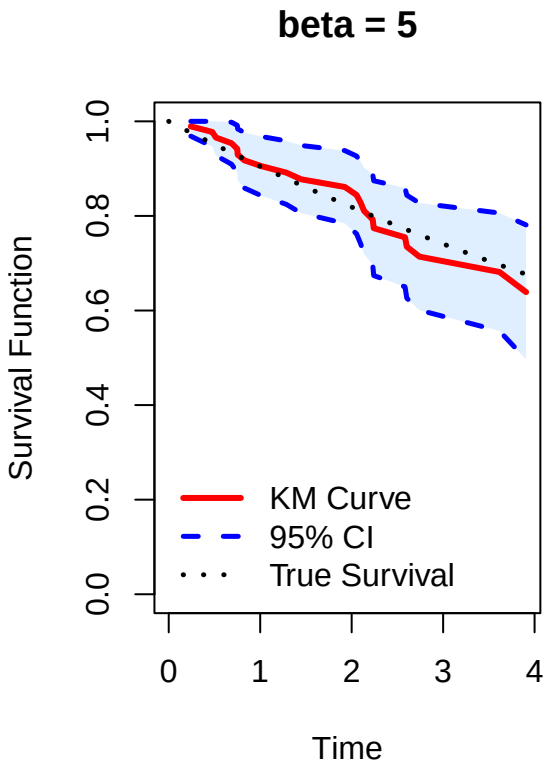


(b) Regenerate 100 simulation studies in (a). Based on these 100 simulation results, plot the average of estimated survival curve and 95% condence interval. (Attached Code)

```r
set.seed(123)

n <- 100
lmbd <- 0.1
beta_list <- c(10, 5)
label_list <- c("KM1", "KM2")

for (bi in seq_along(beta_list)) {
  beta <- beta_list[bi]
  label <- label_list[bi]

  event_time <- rexp(n, rate = lmbd)
  censor_time <- runif(n, 0, beta)
  observed_time <- pmin(event_time, censor_time)
  status <- as.numeric(event_time <= censor_time)

  time_points <- sort(unique(observed_time[status == 1]))
  KM_df <- data.frame(
    time = time_points,
    KM = numeric(length(time_points)),
    LowerCI = numeric(length(time_points)),
    UpperCI = numeric(length(time_points))
  )

  km <- 1
  temp <- 0
  for (j in seq_along(time_points)) {
    t <- time_points[j]
    d <- sum(observed_time == t & status == 1)
    n_risk <- sum(observed_time >= t)

    km <- km * (1 - d / n_risk)
    KM_df$KM[j] <- km


    temp <- temp + d / (n_risk * (n_risk - d))
    var <- km^2 * temp
    KM_df$LowerCI[j] <- max(0, km - 1.96 * sqrt(var))
    KM_df$UpperCI[j] <- min(1, km + 1.96 * sqrt(var))
  }

  assign(label, KM_df)
}

par(mfrow = c(1, 2))

# ----- Left : beta = 10 -----
plot(0, 0, type = "n",
     xlim = c(0, max(KM1$time)), ylim = c(0, 1),
     xlab = "Time", ylab = "Survival Function",
     main = "beta = 10")

polygon(c(KM1$time, rev(KM1$time)),
```

```r
        c(KM1$UpperCI, rev(KM1$LowerCI)),
        col = rgb(0.6, 0.8, 1, 0.3), border = NA)

lines(KM1$time, KM1$KM, col = "red", lwd = 3)
lines(KM1$time, KM1$LowerCI, col = "blue", lwd = 2.5, lty = 2)
lines(KM1$time, KM1$UpperCI, col = "blue", lwd = 2.5, lty = 2)


curve(exp(-0.1 * x), from = 0, to = max(KM1$time), add = TRUE,
      col = "black", lwd = 2.5, lty = 3)

legend("bottomleft",
       legend = c("KM Curve", "95% CI", "True Survival"),
       col = c("red", "blue", "black"),
       lty = c(1, 2, 3),
       lwd = c(3, 2.5, 2.5),
       bty = "n")


# ----- Right：beta = 5 -----
plot(0, 0, type = "n",
     xlim = c(0, max(KM2$time)), ylim = c(0, 1),
     xlab = "Time", ylab = "Survival Function",
     main = "beta = 5")

polygon(c(KM2$time, rev(KM2$time)),
        c(KM2$UpperCI, rev(KM2$LowerCI)),
        col = rgb(0.6, 0.8, 1, 0.3), border = NA)

lines(KM2$time, KM2$KM, col = "red", lwd = 3)
lines(KM2$time, KM2$LowerCI, col = "blue", lwd = 2.5, lty = 2)
lines(KM2$time, KM2$UpperCI, col = "blue", lwd = 2.5, lty = 2)

curve(exp(-0.1 * x), from = 0, to = max(KM2$time), add = TRUE,
      col = "black", lwd = 2.5, lty = 3)

legend("bottomleft",
       legend = c("KM Curve", "95% CI", "True Survival"),
       col = c("red", "blue", "black"),
       lty = c(1, 2, 3),
       lwd = c(3, 2.5, 2.5),
       bty = "n")
```


beta = 10
Survival Function
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
Time
KM Curve
95% CI
True Survival


beta = 5
Survival Function
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
Time
KM Curve
95% CI
True Survival