Survival Analyis HW3

313657003 周佳萱

2025-04-26

Problem 1

1.

$$
\begin{aligned}
P(U \le u) &= P(-\log S(T) \le u) \\
&= P(S(T) \ge e^{-u}) \quad \text{(since } -\log \text{ is decreasing)} \\
&= P(T \le S^{-1}(e^{-u})) \quad \text{(by definition of } S(t) = P(T > t)) \\
&= 1 - S(S^{-1}(e^{-u})) \\
&= 1 - e^{-u}
\end{aligned}
$$

Therefore, $U \sim \text{Exponential}(1)$

Problem 2

(a)

Let $U = S(T \mid x) \sim \text{Uniform}(0, 1)$. Given the conditional survival function:

$$
S(t \mid x) = \exp\left\{-\Lambda_0(t) e^{x\beta}\right\}
$$

Then we have

$$
\begin{aligned}
U &= \exp\left\{-\Lambda_0(t) e^{x\beta}\right\} \\
\Rightarrow \log U &= -\Lambda_0(t) e^{x\beta} \\
\Rightarrow -\log U &= \Lambda_0(t) e^{x\beta} \\
\Rightarrow \Lambda_0(t) &= -\log U \cdot e^{-x\beta} \\
\Rightarrow T &= \Lambda_0^{-1}\left(-\log U \cdot e^{-x\beta}\right)
\end{aligned}
$$

where $U \sim \text{Uniform}(0, 1)$.

(b)

$$
S_0(t) = \int_t^{\infty} \alpha\nu s^{\nu-1} e^{-\alpha s^{\nu}} \mathrm{d}s = -e^{-\alpha x^{\nu}}|_{x=t}^{\infty} = e^{-\alpha t^{\nu}} (\text{Let} u = \alpha s^{\nu},\, du = \alpha\nu s^{\nu-1} ds)
$$

$$
\Rightarrow S(T|x) = S_0(T)^{e^{x\beta}} = e^{-\alpha T^{\nu} e^{x\beta}}
$$

Let $U = S(T|x)$

$$\begin{aligned} P(U \leq u) = P(S(T|x) \leq u) &= P(-\alpha T^{\nu} e^{x\beta} \leq \log u) \\ &= P(T \geq (-\frac{\log u}{\alpha e^{x\beta}})^{\frac{1}{\nu}}) = S_T((-\frac{\log u}{\alpha e^{x\beta}})^{\frac{1}{\nu}}|x) \\ \Rightarrow T &= (-\frac{\log u}{\alpha e^{x\beta}})^{\frac{1}{\nu}} \end{aligned}$$

$$S(t|x) = e^{-\alpha t^{\nu} e^{x\beta}} \Rightarrow \Lambda(t|x) = \alpha t^{\nu} e^{x\beta}$$

$$\lambda(t|x) = \frac{\partial}{\partial t}\Lambda(t|x) = \frac{\partial}{\partial t}\alpha t^{\nu} e^{x\beta} = \alpha e^{x\beta} \nu t^{\nu-1}$$

(c)

$$f_T(t|x) = -\frac{\mathrm{d}S(t|x)}{\mathrm{d}t} = \alpha e^{x\beta} \nu t^{\nu-1} e^{-\alpha e^{x\beta} t^{\nu}}$$

T follows Weibull distribution with shape parameter ν and scale parameter $\alpha e^{x\beta}$

(d)

```r
set.seed(2025)
simulate_data <- function() {
    n <- 1000
    alpha <- 0.1
    beta <- 1
    nu <- 2

    x <- rep(c(1, 0), length.out = n)
    scale_param <- (alpha * exp(x * beta))^(-1 / nu)
    T <- rweibull(n, shape = nu, scale = scale_param)
    C <- runif(n, 0, 10)

    Y <- pmin(T, C)
    delta <- as.numeric(T <= C)

    return(data.frame(Y = Y, delta = delta, x = x))
}

df <- simulate_data()
```

(d1)

$$\begin{aligned}
L(\alpha,\beta,\nu|x,y,\delta) &= \prod_{i=1}^{n}[f(x_i,y_i|\alpha,\beta,\nu)]^{\delta_i}[S(x_i,y_i|\alpha,\beta,\nu)]^{1-\delta_i} \\
&= \prod_{i=1}^{n}[\alpha e^{x_i\beta}\nu {y_i}^{\nu-1}e^{-\alpha e^{x_i\beta}{y_i}^{\nu}}]^{\delta_i}[e^{-\alpha e^{x_i\beta}{y_i}^{\nu}}]^{1-\delta_i} \\
\Rightarrow \log L(\alpha,\beta,\nu|x,y,\delta) &= \sum_{i=1}^{n}\{\delta_i[\log(\alpha\nu)+x_i\beta+(\nu-1)\log(y_i)-\alpha e^{x_i\beta}y_i^{\nu}]+(1-\delta_i)[-\alpha e^{x_i\beta}y_i^{\nu}]\} \\
\Rightarrow \frac{\partial}{\partial\alpha}\log L &= \sum_{i=1}^{n}[\delta_i(\frac{1}{\alpha}-e^{x_i\beta}{y_i}^{\nu})+(1-\delta_i)(-e^{x_i\beta}{y_i}^{\nu})] \equiv 0 \\
\Rightarrow \hat{\alpha}_{MLE} &= \frac{\sum_{i=1}^{n}\delta_i}{\sum_{i=1}^{n}e^{x_i\beta}{y_i}^{\nu}}
\end{aligned}$$

```
set.seed(2025)
loglik_beta_nu <- function(para, input = df){
  alpha_mle <- sum(input$delta) / sum((input$Y^para[2]) * exp(input$x * para[1]))

  ll <- sum(
    input$delta * (log(alpha_mle * para[2]) + input$x * para[1] + (para[2] - 1) * log(input$Y)
                  - alpha_mle * (input$Y^para[2]) * exp(input$x * para[1]))  +
      (1 - input$delta) * (-alpha_mle * (input$Y^para[2]) * exp(input$x * para[1]))
  )
  return(-ll)
}

init <- c(1, 2)

res <- optim(init, loglik_beta_nu, method = "BFGS")
names(res$par) <- c("beta_hat", "nu_hat")
print(res$par)
```

```
##  beta_hat    nu_hat
## 0.9055847 1.9909114
```

We set the initial value at $(1,2)$, and the MLE of (β,ν) is $(0.9055847, 1.9909114)$, so $\hat{\beta}_M \approx 0.9055847$

```
library(survival)

fit_cox <- coxph(Surv(Y, delta) ~ x, data = df)
beta_p <-fit_cox$coefficients
print(beta_p)
```

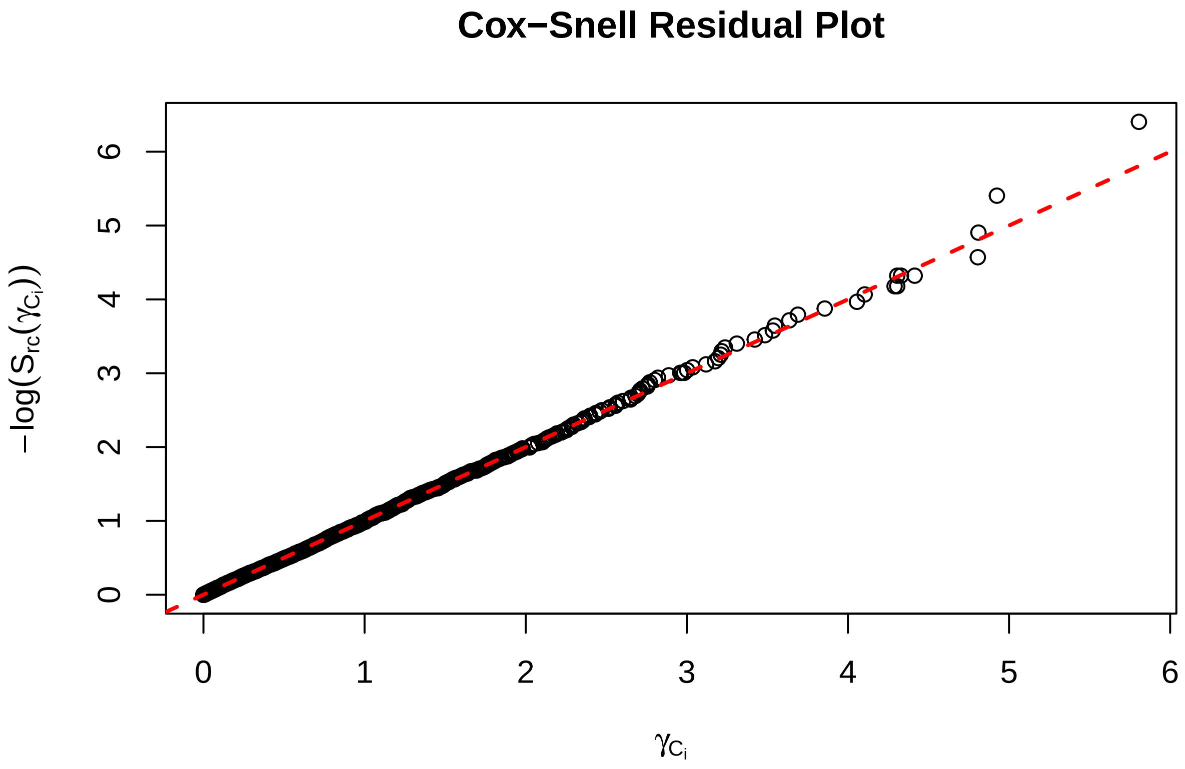
(d2)

```
##         x
## 0.9005799
```

The estimator computed by the function coxph $\hat{\beta}_P \approx 0.9005799$

(e)

```r
# 1. Cum Hazard
Y_sorted <- sort(unique(df$Y))  # unique time points

Lambda_0 <- numeric(length(Y_sorted))
names(Lambda_0) <- Y_sorted

for (i in seq_along(Y_sorted)) {
  t <- Y_sorted[i]
  d_i <- sum(df$Y == t & df$delta == 1)

  at_risk <- which(df$Y >= t)
  event_at_t <- which(df$Y == t & df$delta == 1)

  risk_sum <- sum(exp(df$x[at_risk] * beta_p))

  Lambda_0[i] <- ifelse(risk_sum > 0, d_i / risk_sum, 0)
}

Lambda_0_cum <- cumsum(Lambda_0)
names(Lambda_0_cum) <- Y_sorted

# Step 2: Cox-Snell
coxsnell <- numeric(nrow(df))
for (i in 1:nrow(df)) {
  ti <- df$Y[i]
  idx <- max(which(Y_sorted <= ti))
  lambda <- Lambda_0_cum[idx]
  coxsnell[i] <- exp(df$x[i] * beta_p) * lambda
}

ord <- order(coxsnell)
cox_sorted <- coxsnell[ord]
delta_sorted <- df$delta[ord]

n <- length(cox_sorted)
H_hat <- numeric(n)
for (i in 1:n) {
  risk <- n - i + 1
  H_hat[i] <- ifelse(risk > 0, delta_sorted[i] / risk, 0)
}
H_cum <- cumsum(H_hat)

# Step 3: Plot Cox-Snell residual
plot(cox_sorted, H_cum, type = "p",
     main = "Cox-Snell Residual Plot",
     xlab = expression(gamma[C[i]]),
     ylab = expression(-log(hat(S)[rc](gamma[C[i]]))),
     col = "black")
abline(0, 1, col = "red", lty = 2, lwd = 2)
```



The Cox-Snell residual plot shows that most points lie close to the 45-degree line, indicating that the Cox proportional hazards model fits the data well.

The deviation from the line for residuals greater than 4 suggests a poorer fit in the upper tail, likely due to fewer observations in that region.

(f)

```r
library(knitr)
set.seed(2025)

# --- Simulation function ---
n_sim <- 100
beta_m <- numeric(n_sim)
beta_p <- numeric(n_sim)

for (i in 1:n_sim) {
  df_sim <- simulate_data()

  res <- optim(par = c(1, 2), fn = loglik_beta_nu, input = df_sim, method = "BFGS")
  beta_m[i] <- res$par[1]

  fit_cox <- coxph(Surv(Y, delta) ~ x, data = df_sim)
  beta_p[i] <- coef(fit_cox)
}
```

```r
compute_ci <- function(beta_vec, method_name) {
  mean_val <- mean(beta_vec)
  sd_val <- sd(beta_vec)
  error <- qnorm(0.975) * sd_val
  c(Method = method_name, LowerCI = mean_val - error, Mean = mean_val, UpperCI = mean_val + error)
}

ci_table <- as.data.frame(rbind(
  compute_ci(beta_m, "MLE (beta_M)"),
  compute_ci(beta_p, "Partial Likelihood (beta_P)")
), stringsAsFactors = FALSE)

ci_table[, 2:4] <- lapply(ci_table[, 2:4], as.numeric)

kable(ci_table, digits = 4, align = "c", caption = "95% Confidence Intervals for Estimates")
```

Table 1: 95% Confidence Intervals for Estimates

Method	LowerCI	Mean	UpperCI
MLE (beta_M)	0.8391	0.9963	1.1535
Partial Likelihood (beta_P)	0.8281	0.9938	1.1595

(g)

After conducting 100 simulations, we observe that the estimates from the maximum likelihood method, $\hat{\beta}_M$ is very close to that of the partial likelihood method, $\hat{\beta}_P$. However, the confidence interval for $\hat{\beta}_P$ is slightly wider than that of $\hat{\beta}_M$. This is likely because the partial likelihood method may discard some information about β, that is retained in the full likelihood approach. Nevertheless, $\hat{\beta}_P$ still performs well and provides approximately same estimation of $\hat{\beta}_M$.

(h)

```r
set.seed(2025)
reject_count <- 0

for (i in 1:n_sim) {
  df_sim <- simulate_data()

  fit <- coxph(Surv(Y, delta) ~ x, data = df_sim)
  beta_hat <- coef(fit)
  se_beta <- sqrt(diag(fit$var))

  lower <- beta_hat - qnorm(0.975) * se_beta
  upper <- beta_hat + qnorm(0.975) * se_beta

  if (lower > 1 || upper < 1) {
    reject_count <- reject_count + 1
  }
}

cat("Number of rejections of H0 (beta = 1):", reject_count, "out of", n_sim, "simulations\n")
```

```
## Number of rejections of H0 (beta = 1): 5 out of 100 simulations
```

Using Wald's test:

$$H_0 : \beta = 1$$
$$H_1 : \beta \neq 1$$

We reject H_0, if the 95% confidence interval$(\hat{\beta}_P - 1.96 s.e.(\hat{\beta}_P), \hat{\beta}_P - 1.96 s.e.(\hat{\beta}_P))$ does not cover 1.

Based on the simulation results, only 5 out of 100 data led to the rejection of H_0, which is consistent with the 5% significance level.