Day9 exercise solutions

Ali Movasati, Isabelle Caroline Rose Cretton, Tristan Koning

Nov. 15th, 2024

Problem 1: Log-logistic distribution

(a) Plot density, distribution, survival, and hazard functions

```r
# Define the functions
loglogistic_density <- function(x, alpha = 1, beta) {
  (beta/alpha) * (x/alpha)^(beta-1) / (1 + (x/alpha)^beta)^2
}

loglogistic_distribution <- function(x, alpha = 1, beta) {
  (x^beta) / (alpha^beta + x^beta)
}

loglogistic_survival <- function(x, alpha = 1, beta) {
  1 - loglogistic_distribution(x, alpha, beta)
}

loglogistic_hazard <- function(x, alpha = 1, beta) {
  loglogistic_density(x, alpha, beta) / loglogistic_survival(x, alpha, beta)
}

# Create x values
x <- seq(0.01, 5, length.out = 1000)
beta_values <- c(0.25, 0.5, 1, 4, 8)

# Create plots
par(mfrow = c(2, 2))

# Density plot
plot(x, loglogistic_density(x, beta = beta_values[1]), type = "l",
     ylim = c(0, 2), main = "Density Function",
     xlab = "x", ylab = "f(x)")
for(beta in beta_values[-1]) {
  lines(x, loglogistic_density(x, beta = beta), col = which(beta_values == beta))
}
legend("topright", legend = paste("beta =", beta_values),
       col = 1:length(beta_values), lty = 1)

# Distribution plot
plot(x, loglogistic_distribution(x, beta = beta_values[1]), type = "l",
```

```r
      ylim = c(0, 1), main = "Distribution Function",
      xlab = "x", ylab = "F(x)")
for(beta in beta_values[-1]) {
  lines(x, loglogistic_distribution(x, beta = beta), col = which(beta_values == beta))
}
legend("bottomright", legend = paste("beta =", beta_values),
       col = 1:length(beta_values), lty = 1)

# Survival plot
plot(x, loglogistic_survival(x, beta = beta_values[1]), type = "l",
     ylim = c(0, 1), main = "Survival Function",
     xlab = "x", ylab = "S(x)")
for(beta in beta_values[-1]) {
  lines(x, loglogistic_survival(x, beta = beta), col = which(beta_values == beta))
}
legend("topright", legend = paste("beta =", beta_values),
       col = 1:length(beta_values), lty = 1)

# Hazard plot
plot(x, loglogistic_hazard(x, beta = beta_values[1]), type = "l",
     ylim = c(0, 5), main = "Hazard Function",
     xlab = "x", ylab = "h(x)")
for(beta in beta_values[-1]) {
  lines(x, loglogistic_hazard(x, beta = beta), col = which(beta_values == beta))
}
legend("topright", legend = paste("beta =", beta_values),
       col = 1:length(beta_values), lty = 1)
```
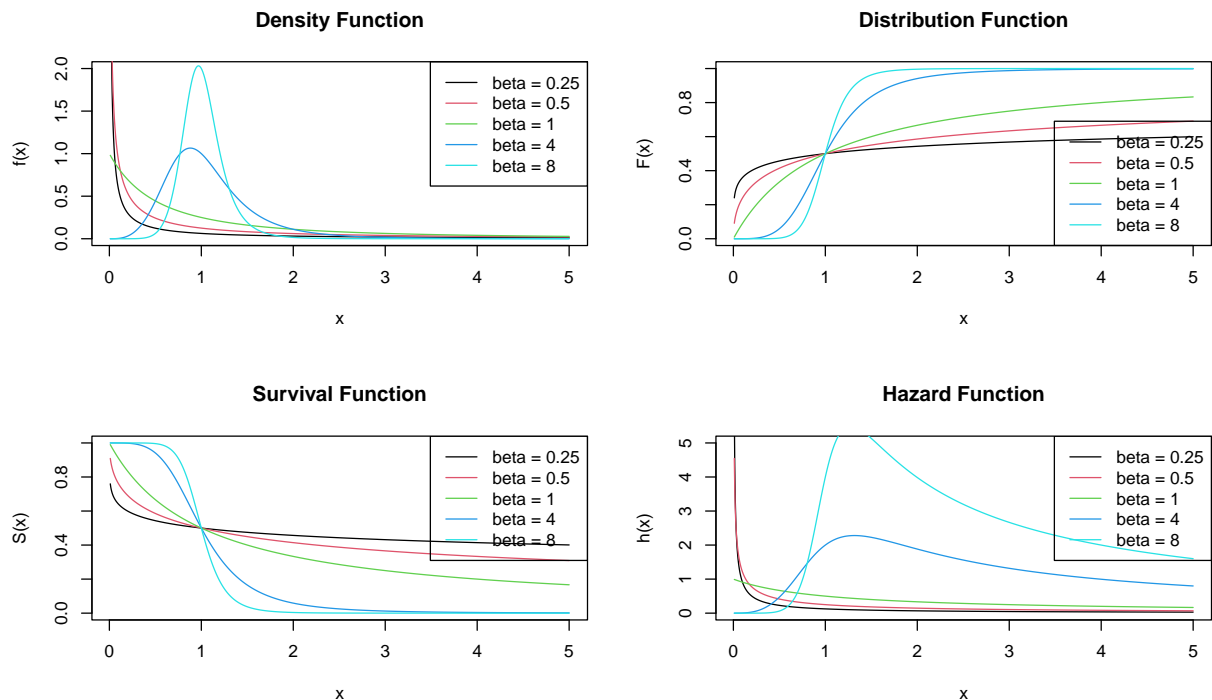
```r
# Create summary table for log-logistic distribution characteristics
log_logistic_chars <- data.frame(
  "Parameter" = c("beta < 1", "beta = 1", "beta > 1"),
  "Density Shape" = c("Right-skewed with high peak near zero",
                      "Balanced shape",
                      "More symmetric, bell-shaped"),
  "Hazard Behavior" = c("Decreasing hazard",
                        "Constant hazard",
                        "Initially increasing then decreasing")
)

kable(log_logistic_chars,
      caption = "Log-logistic Distribution Characteristics",
      align = c('l', 'l', 'l'),
      booktabs = TRUE) %>%
  kable_styling(latex_options = c("striped", "hold_position"),
                full_width = FALSE)
```

Table 1: Log-logistic Distribution Characteristics

| Parameter	Density.Shape	Hazard.Behavior
beta $<$ 1	Right-skewed with high peak near zero	Decreasing hazard
beta $=$ 1	Balanced shape	Constant hazard
beta $>$ 1	More symmetric, bell-shaped	Initially increasing then decreasing

(b) Weibull Distribution Analysis

```r
# Define Weibull density function
weibull_density <- function(x, sigma = 1, kappa) {
  (kappa/sigma) * (x/sigma)^(kappa-1) * exp(-(x/sigma)^kappa)
}


# Create plots for different kappa values
x <- seq(0.01, 5, length.out = 1000)
kappa_values <- c(0.5, 1, 2)  # One from each case

par(mfrow = c(1,1))
plot(x, weibull_density(x, kappa = kappa_values[1]), type = "l",
     ylim = c(0, 1.5), main = "Weibull Density for Different Kappa Values",
     xlab = "x", ylab = "f(x)")
for(kappa in kappa_values[-1]) {
  lines(x, weibull_density(x, kappa = kappa), col = which(kappa_values == kappa))
}
legend("topright", legend = paste("kappa =", kappa_values),
       col = 1:length(kappa_values), lty = 1)
```

Weibull Density for Different Kappa Values
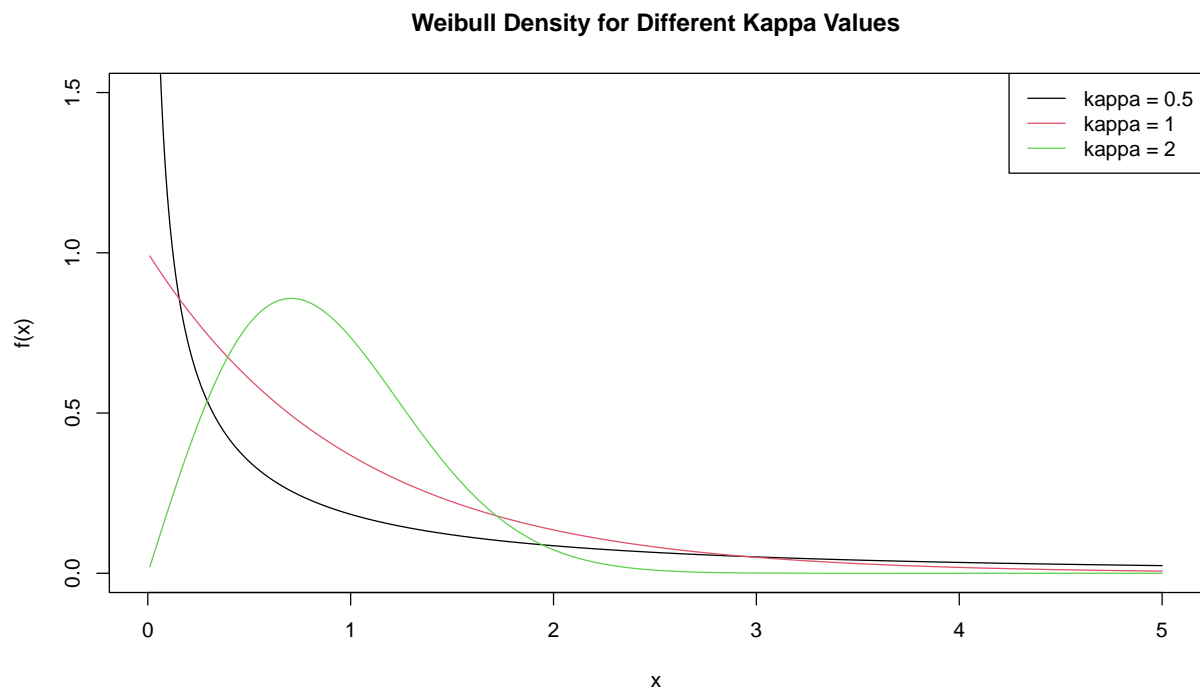

```r
# Create summary table for Weibull distribution characteristics
weibull_chars <- data.frame(
  "Parameter Range" = c("0 < kappa < 1", "kappa = 1", "kappa > 1"),
  "Characteristics" = c("Monotonically decreasing, infinite peak at x = 0",
                        "Reduces to exponential distribution",
                        "Peak away from zero, right-skewed")
)

kable(weibull_chars,
      caption = "Weibull Distribution Characteristics (sigma = 1)",
      align = c('l', 'l'),
      booktabs = TRUE) %>%
  kable_styling(latex_options = c("striped", "hold_position"),
                full_width = FALSE)
```

Table 2: Weibull Distribution Characteristics (sigma = 1)

Parameter.Range	Characteristics
0 < kappa < 1	Monotonically decreasing, infinite peak at x = 0
kappa = 1	Reduces to exponential distribution
kappa > 1	Peak away from zero, right-skewed

Problem 2: Survival Analysis of PBC Data

```r
# Load and prepare data
pbc_data <- read.table("data/pbc.txt", header = TRUE)

# Convert treat to factor
pbc_data$treat <- factor(pbc_data$treat,
                         levels = c(1, 2),
                         labels = c("Placebo", "Treatment"))

# Create survival object
surv_obj <- Surv(time = pbc_data$time, event = pbc_data$d)

# Fit survival curves
km_fit <- survfit(surv_obj ~ treat, data = pbc_data)
```
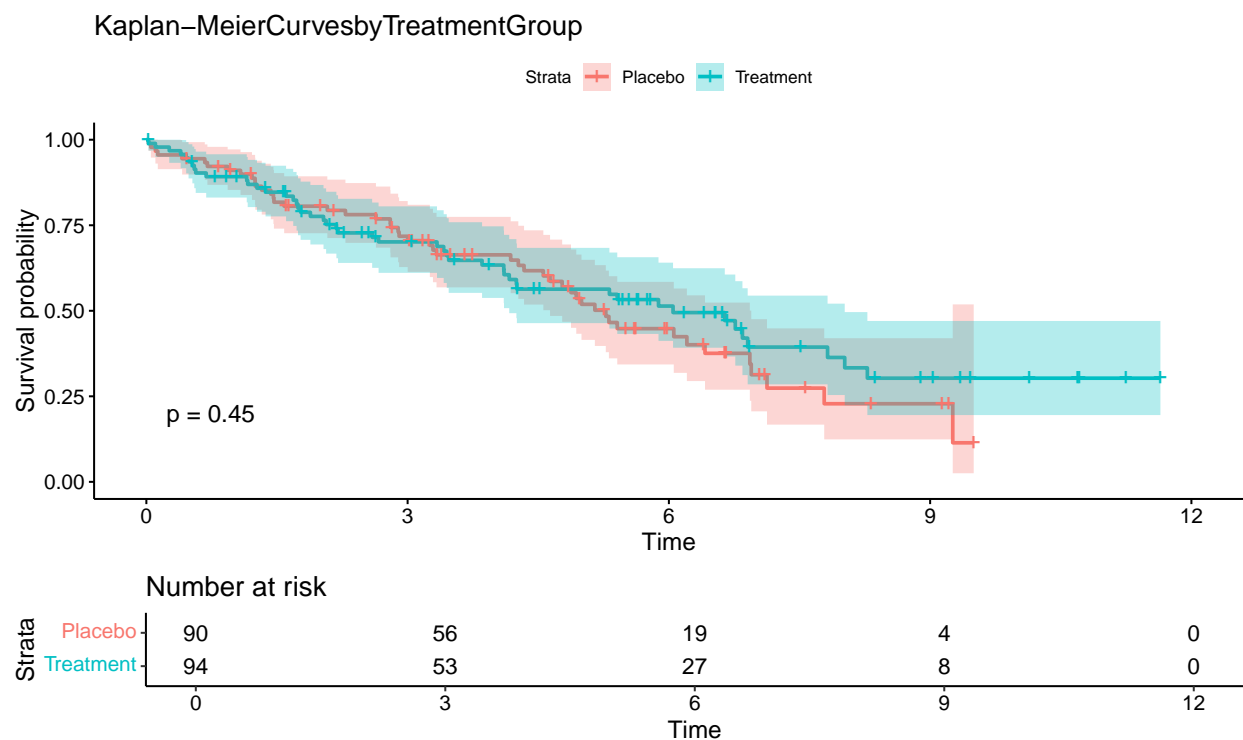
Kaplan-Meier Analysis

```r
# Plot Kaplan-Meier curves
par(mfrow = c(1,1))
 ggsurvplot(km_fit,
 data= pbc_data,
 pval= TRUE,
 conf.int=TRUE,
 risk.table= TRUE,
 title= "Kaplan-MeierCurvesbyTreatmentGroup",
 legend.labs= c("Placebo", "Treatment"))
```
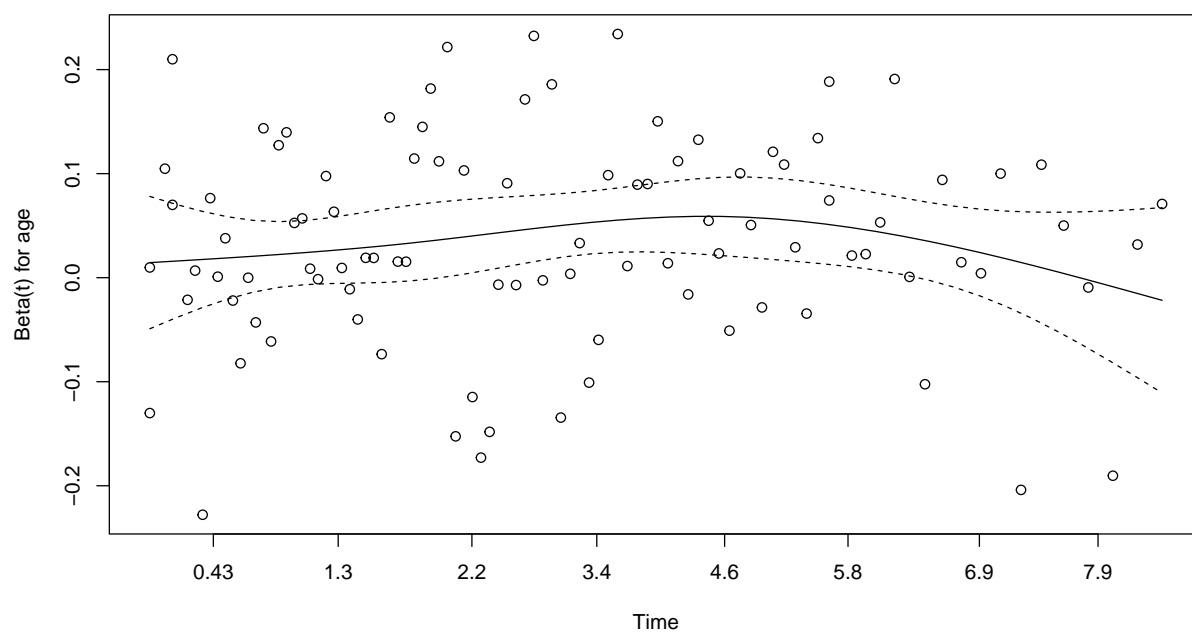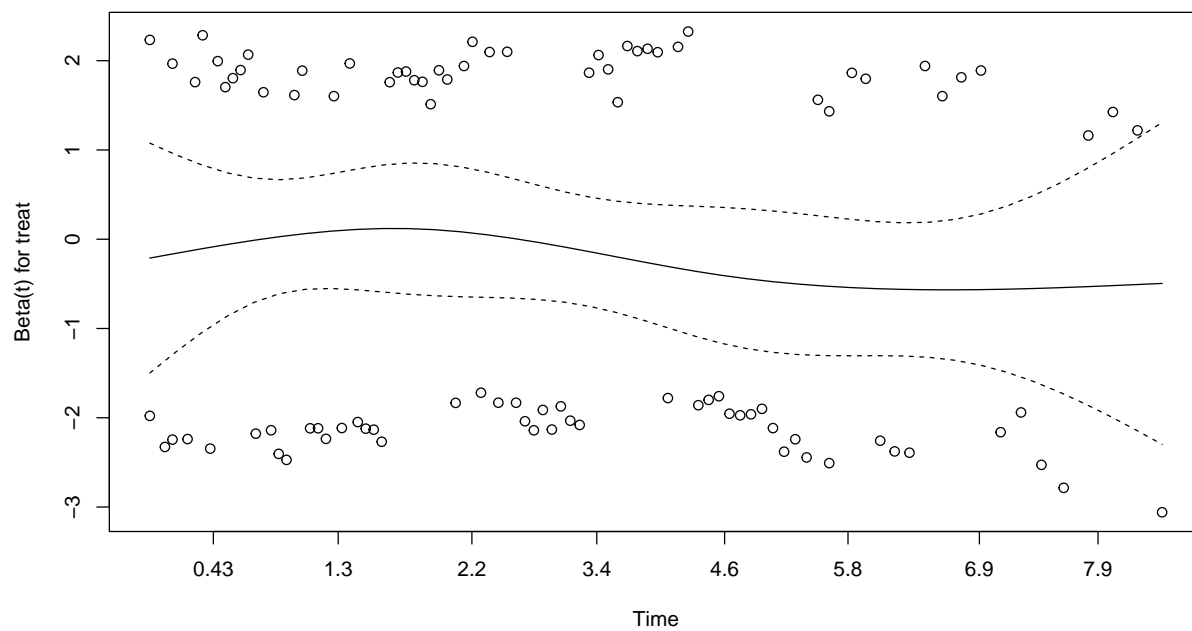
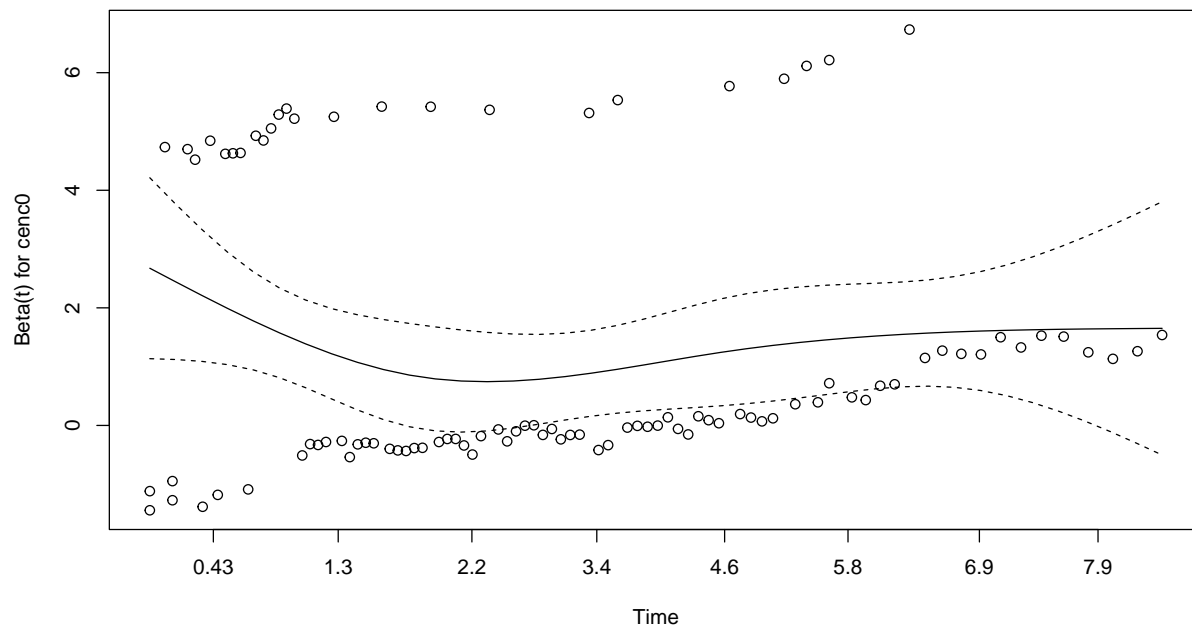### Kaplan−MeierCurvesbyTreatmentGroup

Statistical Analysis

```r
# Fit models
cox_model1 <- coxph(surv_obj ~ treat, data = pbc_data)
cox_model2 <- coxph(surv_obj ~ treat + age + cenc0, data = pbc_data)
log_rank <- survdiff(surv_obj ~ treat, data = pbc_data)
test.ph1 <- cox.zph(cox_model1)
test.ph2 <- cox.zph(cox_model2)
plot(test.ph1)
```



```r
plot(test.ph2)
```
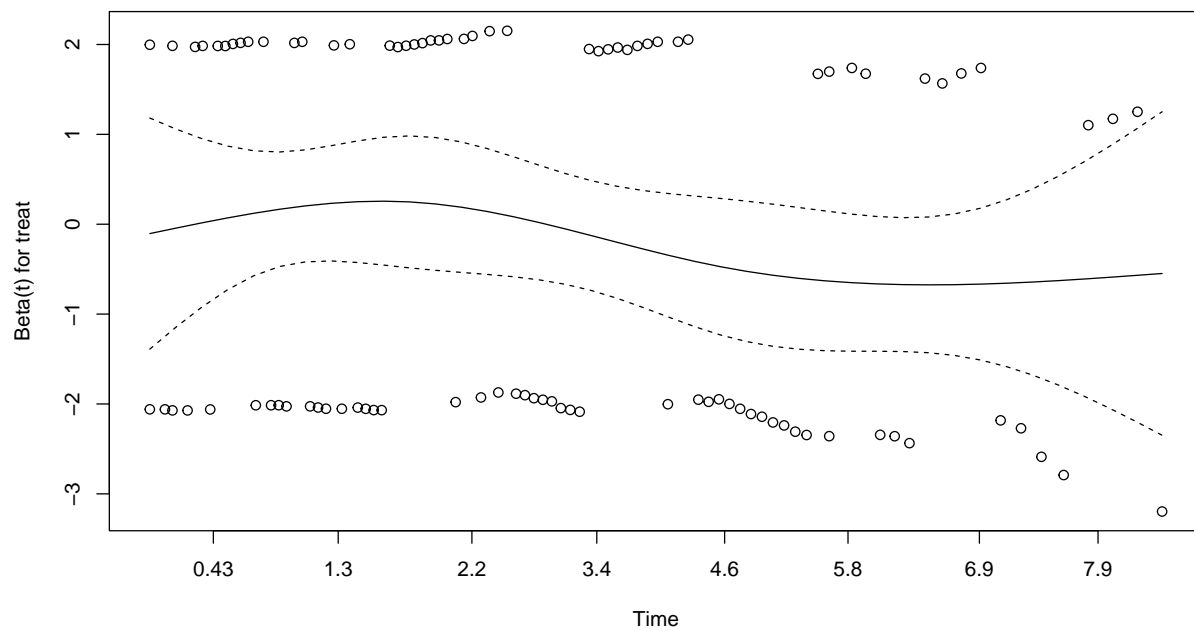
```r
# Function to create formatted Cox model results
create_cox_table <- function(model) {
  coef_table <- summary(model)$coefficients
  conf_table <- summary(model)$conf.int

  results_df <- data.frame(
    Variable = rownames(coef_table),
    "Hazard Ratio" = sprintf("%.2f", conf_table[, "exp(coef)"]),
    "95% CI" = sprintf("(%.2f, %.2f)", conf_table[, "lower .95"], conf_table[, "upper .95"]),
    "P-value" = sprintf("%.3f", coef_table[, "Pr(>|z|)"]),
    check.names = FALSE
  )

  concordance <- sprintf("%.3f", summary(model)$concordance[1])

  kable(results_df,
        caption = paste("Cox Model Results (Concordance =", concordance, ")"),
        align = c('l', 'c', 'c', 'c'),
        booktabs = TRUE) %>%
    kable_styling(latex_options = c("striped", "hold_position"),
                  full_width = FALSE)
}

# Basic Cox Model Results
cat("\n### Basic Cox Model\n")
```

Basic Cox Model

```
create_cox_table(cox_model1)
```

Table 3: Cox Model Results (Concordance = 0.503)

Variable	Hazard Ratio	95% CI	P-value
treatTreatment	0.86	(0.57, 1.28)	0.451

```
# Extended Cox Model Results
cat("\n### Extended Cox Model\n")
```

Extended Cox Model

```
create_cox_table(cox_model2)
```

Table 4: Cox Model Results (Concordance = 0.669)

Variable	Hazard Ratio	95% CI	P-value
treatTreatment	0.83	(0.55, 1.24)	0.362
age	1.04	(1.02, 1.06)	0.001
cenc0	3.85	(2.38, 6.23)	0.000

```
# Log-rank test results
cat("\n### Log-rank Test Results\n")
```

Log-rank Test Results

```
logrank_df <- data.frame(
  Group = c("Placebo", "Treatment"),
  Observed = c(log_rank$obs[1], log_rank$obs[2]),
  Expected = c(log_rank$exp[1], log_rank$exp[2]),
  "(O-E)^2/E" = c(
    (log_rank$obs[1] - log_rank$exp[1])^2/log_rank$exp[1],
    (log_rank$obs[2] - log_rank$exp[2])^2/log_rank$exp[2]
  )
)

kable(logrank_df,
      caption = sprintf("Log-rank Test Results (p-value = %.3f)",
                        1 - pchisq(log_rank$chisq, df = 1)),
      align = c('l', 'c', 'c', 'c'),
      digits = 2,
      booktabs = TRUE) %>%
  kable_styling(latex_options = c("striped", "hold_position"),
                full_width = FALSE)
```

Table 5: Log-rank Test Results (p-value = 0.450)

Group	Observed	Expected	X.O.E..2.E
Placebo	49	45.33	0.30
Treatment	47	50.67	0.27

```r
# Proportional hazards test results
cat("\n### Proportional Hazards Test Results\n")
```

Proportional Hazards Test Results

```r
ph_df <- data.frame(
  Variable = rownames(test.ph2$table),
  "Chi-square" = sprintf("%.2f", test.ph2$table[, "chisq"]),
  "P-value" = sprintf("%.3f", test.ph2$table[, "p"]),
  check.names = FALSE
)

kable(ph_df,
      caption = "Proportional Hazards Assumption Test",
      align = c('l', 'c', 'c'),
      booktabs = TRUE) %>%
  kable_styling(latex_options = c("striped", "hold_position"),
                full_width = FALSE)
```

Table 6: Proportional Hazards Assumption Test

Variable	Chi-square	P-value
treat	1.04	0.308
age	0.04	0.839
cenc0	0.26	0.607
GLOBAL	1.33	0.722

```r
# Key findings summary
cat("\n### Key Findings\n")
```

Key Findings

```r
findings_df <- data.frame(
  Category = c("Treatment Effect", "Risk Factors", "Model Performance", "Assumptions"),
  Finding = c(
    "17% reduction in hazard (non-significant, p > 0.05)",
    "Age: 4% increased risk per year (p < 0.001)\nCentral cholestasis: 284% increased risk (p < 0.001)"
    "Extended model concordance: 0.669\nSignificant improvement over basic model",
    "Proportional hazards satisfied for all variables (p > 0.05)"
  )
```

```
)

kable(findings_df,
      caption = "Summary of Key Findings",
      col.names = c("Category", "Details"),
      align = c('l', 'l'),
      booktabs = TRUE) %>%
  kable_styling(latex_options = c("striped", "hold_position"),
                full_width = FALSE)
```

Table 7: Summary of Key Findings

Category	Details
Treatment Effect	17% reduction in hazard (non-significant, $p > 0.05$)
Risk Factors	Age: 4% increased risk per year ($p < 0.001$) Central cholestasis: 284% increased risk ($p < 0.001$)
Model Performance	Extended model concordance: 0.669 Significant improvement over basic model
Assumptions	Proportional hazards satisfied for all variables ($p > 0.05$)