Survival Analysis - Homework 1

Your Name

2025-10-03

Table of Contents

# 1 Setup

## 1.1 Load Required Packages

# Install packages if needed  
# install.packages("survival")  
  
# Load packages  
library(survival)  
library(knitr)

# 2 Question 1: Exponential Distribution Analysis

## 2.1 Background

The following data consists of the times to relapse and times to death of 10 bone marrow transplant patients, who were followed for up to 45 months after their transplant.

* Patients #7-10 were alive and free of relapse at the end of the study
* Patients #4-6 relapsed, but were still alive at the end of the study

## 2.2 Data

# Read data  
q1\_data <- read.csv("Q1data\_extracted.csv")  
kable(q1\_data, caption = "Table 1: Bone Marrow Transplant Patient Data", align = "c")

Table 1: Bone Marrow Transplant Patient Data

| Patient | Relapse\_Time | Relapse | Death\_Time | Death |
| --- | --- | --- | --- | --- |
| 1 | 5 | 1 | 8 | 1 |
| 2 | 8 | 1 | 12 | 1 |
| 3 | 12 | 1 | 15 | 1 |
| 4 | 20 | 1 | 33 | 0 |
| 5 | 32 | 1 | 45 | 0 |
| 6 | 27 | 1 | 28 | 0 |
| 7 | 16 | 0 | 16 | 0 |
| 8 | 17 | 0 | 17 | 0 |
| 9 | 19 | 0 | 19 | 0 |
| 10 | 30 | 0 | 30 | 0 |

**Data Summary:**

* Total patients: 10
* Relapse events: 6 patients, Censored: 4 patients
* Death events: 3 patients, Censored: 7 patients

## 2.3 Question 1(a): Maximum Likelihood Estimation of λ

**Task:** Calculate the maximum likelihood estimator of the parameter λ for time to relapse AND time to death assuming an exponential distribution, f(t) = λe^(-λt). Write a brief sentence interpreting this parameter.

### 2.3.1 Solution

#### 2.3.1.1 Part 1: Time to Relapse

**Step 1: Extract the observed data**

From the dataset, we identify relapse events: - Patient 1: Relapse at 5 months (event) - Patient 2: Relapse at 8 months (event) - Patient 3: Relapse at 12 months (event) - Patient 4: Relapse at 20 months (event) - Patient 5: Relapse at 32 months (event) - Patient 6: Relapse at 27 months (event) - Patients 7-10: Censored at 16, 17, 19, 30 months (no relapse)

**Step 2: Exponential Distribution Framework**

For exponential distribution, the probability density function is:

The survival function is:

The hazard function is constant:

**Step 3: Maximum Likelihood Estimation**

According to the chapter1, page 71, the estimator for is:

**Step 4: Calculate λ by hand**

Number of relapse events:

Total exposure time:

Maximum likelihood estimate:

**Step 5: Derived Parameters**

Mean survival time:

Median survival time:

**Step 6: Confidence Interval for λ (Poisson Approximation)**

Using the Poisson approximation for the number of events, the standard error of is:

where (number of events) and (total person-time).

Calculate standard error:

For a 95% confidence interval, using :

**95% CI for λ:** per month

**Step 7: Survival Probabilities**

Using with :

| Time (months) | Calculation | S(t) |
| --- | --- | --- |
| 5 |  | 0.8513 |
| 10 |  | 0.7247 |
| 15 |  | 0.6170 |
| 20 |  | 0.5253 |
| 25 |  | 0.4472 |
| 30 |  | 0.3806 |

**Summary of Hand Calculations:** - **Estimated hazard rate:** per month (rounded to 3 decimal places) - **Standard error:** per month - **95% CI for λ:** per month - **Mean time to relapse:** months - **Median time to relapse:** months

**Interpretation:** The parameter λ represents the instantaneous hazard rate (risk) of relapse per month. A value of 0.032 per month indicates that at any given time, a patient has approximately a 3.2% risk of relapse per month, assuming the exponential distribution holds.

#### 2.3.1.2 Part 2: Time to Death

**Step 1: Extract death data**

From the dataset, we identify death events: - Patient 1: Death at 8 months (event) - Patient 2: Death at 12 months (event) - Patient 3: Death at 15 months (event) - Patients 4-10: Censored at 33, 45, 28, 16, 17, 19, 30 months (no death observed)

**Step 2: Calculate λ for death**

Number of death events:

Total exposure time:

Maximum likelihood estimate:

**Step 3: Derived Parameters for Death**

Mean survival time:

Median survival time:

**Step 4: Confidence Interval for λ\_D (Poisson Approximation)**

Using the Poisson approximation, the standard error of is:

For a 95% confidence interval, using :

Since the hazard rate cannot be negative, we adjust the lower bound to 0:

Note: The negative lower bound indicates insufficient events for precise estimation. Alternatively, we can report or note the limitation.

**Summary for Death:** - **Estimated hazard rate:** per month (rounded to 3 decimal places) - **Standard error:** per month - **95% CI for λ:** per month - **Mean time to death:** months - **Median time to death:** months

**Interpretation:** The parameter λ for death represents the instantaneous hazard rate of death per month. A value of 0.013 per month indicates a lower risk of death compared to relapse (0.032 per month).

## 2.4 Question 1(b): Using the Exponential Parameter

**Task:** Using the parameter estimate from 1(a) (rounded to 3 decimal places), estimate the following quantities for BOTH relapse and death.

### 2.4.1 Solution

# Use lambda values from 1(a), rounded to 3 decimal places  
lambda\_R <- 0.032 # per month (relapse)  
lambda\_D <- 0.013 # per month (death)  
  
cat("=== QUESTION 1(b): EXPONENTIAL DISTRIBUTION ESTIMATES ===\n\n")

## === QUESTION 1(b): EXPONENTIAL DISTRIBUTION ESTIMATES ===

cat("Using λ\_R = 0.032 per month (relapse)\n")

## Using λ\_R = 0.032 per month (relapse)

cat("Using λ\_D = 0.013 per month (death)\n\n")

## Using λ\_D = 0.013 per month (death)

# (i) Mean times  
mean\_relapse <- 1 / lambda\_R  
mean\_death <- 1 / lambda\_D  
  
cat("(i) MEAN TIMES:\n")

## (i) MEAN TIMES:

cat(" Mean time to relapse = 1/λ\_R = 1/0.032 =", round(mean\_relapse, 2), "months\n")

## Mean time to relapse = 1/λ\_R = 1/0.032 = 31.25 months

cat(" Mean survival time = 1/λ\_D = 1/0.013 =", round(mean\_death, 2), "months\n\n")

## Mean survival time = 1/λ\_D = 1/0.013 = 76.92 months

# (ii) Median times  
median\_relapse <- log(2) / lambda\_R  
median\_death <- log(2) / lambda\_D  
  
cat("(ii) MEDIAN TIMES:\n")

## (ii) MEDIAN TIMES:

cat(" Median time to relapse = log(2)/λ\_R = 0.693/0.032 =", round(median\_relapse, 2), "months\n")

## Median time to relapse = log(2)/λ\_R = 0.693/0.032 = 21.66 months

cat(" Median survival time = log(2)/λ\_D = 0.693/0.013 =", round(median\_death, 2), "months\n\n")

## Median survival time = log(2)/λ\_D = 0.693/0.013 = 53.32 months

# (iii) Survival probabilities at 12 and 24 months  
# S(t) = exp(-λt)  
SR\_12 <- exp(-lambda\_R \* 12)  
SR\_24 <- exp(-lambda\_R \* 24)  
SD\_12 <- exp(-lambda\_D \* 12)  
SD\_24 <- exp(-lambda\_D \* 24)  
  
cat("(iii) ONE-YEAR AND TWO-YEAR SURVIVAL PROBABILITIES:\n")

## (iii) ONE-YEAR AND TWO-YEAR SURVIVAL PROBABILITIES:

cat(" S\_R(12) = exp(-0.032 × 12) =", round(SR\_12, 4),   
 "(", round(SR\_12 \* 100, 1), "% relapse-free at 1 year)\n")

## S\_R(12) = exp(-0.032 × 12) = 0.6811 ( 68.1 % relapse-free at 1 year)

cat(" S\_R(24) = exp(-0.032 × 24) =", round(SR\_24, 4),  
 "(", round(SR\_24 \* 100, 1), "% relapse-free at 2 years)\n")

## S\_R(24) = exp(-0.032 × 24) = 0.4639 ( 46.4 % relapse-free at 2 years)

cat(" S\_D(12) = exp(-0.013 × 12) =", round(SD\_12, 4),  
 "(", round(SD\_12 \* 100, 1), "% surviving at 1 year)\n")

## S\_D(12) = exp(-0.013 × 12) = 0.8556 ( 85.6 % surviving at 1 year)

cat(" S\_D(24) = exp(-0.013 × 24) =", round(SD\_24, 4),  
 "(", round(SD\_24 \* 100, 1), "% surviving at 2 years)\n\n")

## S\_D(24) = exp(-0.013 × 24) = 0.732 ( 73.2 % surviving at 2 years)

# (iv) Cumulative probabilities F(t) = 1 - S(t)  
FR\_12 <- 1 - SR\_12  
FR\_24 <- 1 - SR\_24  
FD\_12 <- 1 - SD\_12  
FD\_24 <- 1 - SD\_24  
  
cat("(iv) CUMULATIVE PROBABILITIES (CDF):\n")

## (iv) CUMULATIVE PROBABILITIES (CDF):

cat(" F\_R(12) = 1 - S\_R(12) =", round(FR\_12, 4),  
 "(", round(FR\_12 \* 100, 1), "% relapsed by 1 year)\n")

## F\_R(12) = 1 - S\_R(12) = 0.3189 ( 31.9 % relapsed by 1 year)

cat(" F\_R(24) = 1 - S\_R(24) =", round(FR\_24, 4),  
 "(", round(FR\_24 \* 100, 1), "% relapsed by 2 years)\n")

## F\_R(24) = 1 - S\_R(24) = 0.5361 ( 53.6 % relapsed by 2 years)

cat(" F\_D(12) = 1 - S\_D(12) =", round(FD\_12, 4),  
 "(", round(FD\_12 \* 100, 1), "% died by 1 year)\n")

## F\_D(12) = 1 - S\_D(12) = 0.1444 ( 14.4 % died by 1 year)

cat(" F\_D(24) = 1 - S\_D(24) =", round(FD\_24, 4),  
 "(", round(FD\_24 \* 100, 1), "% died by 2 years)\n\n")

## F\_D(24) = 1 - S\_D(24) = 0.268 ( 26.8 % died by 2 years)

# (v) Conditional probability: P(T > 24 | T > 12)  
# For exponential: P(T > t+s | T > s) = P(T > t) (memoryless property)  
# P(T > 24 | T > 12) = P(T > 12) = S(12)  
conditional\_prob <- SR\_24 / SR\_12  
  
cat("(v) CONDITIONAL PROBABILITY:\n")

## (v) CONDITIONAL PROBABILITY:

cat(" P(relapse-free at 2 years | relapse-free at 1 year)\n")

## P(relapse-free at 2 years | relapse-free at 1 year)

cat(" = S\_R(24) / S\_R(12) =", round(SR\_24, 4), "/", round(SR\_12, 4),   
 "=", round(conditional\_prob, 4), "\n\n")

## = S\_R(24) / S\_R(12) = 0.4639 / 0.6811 = 0.6811

cat(" COMPARISON:\n")

## COMPARISON:

cat(" - Unconditional probability of being relapse-free at 1 year: S\_R(12) =", round(SR\_12, 4), "\n")

## - Unconditional probability of being relapse-free at 1 year: S\_R(12) = 0.6811

cat(" - Conditional probability (relapse-free at 2 yrs | relapse-free at 1 yr) =", round(conditional\_prob, 4), "\n")

## - Conditional probability (relapse-free at 2 yrs | relapse-free at 1 yr) = 0.6811

cat(" - These are EQUAL due to the memoryless property of exponential distribution!\n")

## - These are EQUAL due to the memoryless property of exponential distribution!

cat(" - This means past survival does not affect future survival probability.\n")

## - This means past survival does not affect future survival probability.

### 2.4.2 Summary Table

summary\_table <- data.frame(  
 Quantity = c("Mean", "Median", "S(12 months)", "S(24 months)", "F(12 months)", "F(24 months)"),  
 Relapse = c(  
 paste(round(mean\_relapse, 2), "months"),  
 paste(round(median\_relapse, 2), "months"),  
 round(SR\_12, 4),  
 round(SR\_24, 4),  
 round(FR\_12, 4),  
 round(FR\_24, 4)  
 ),  
 Death = c(  
 paste(round(mean\_death, 2), "months"),  
 paste(round(median\_death, 2), "months"),  
 round(SD\_12, 4),  
 round(SD\_24, 4),  
 round(FD\_12, 4),  
 round(FD\_24, 4)  
 )  
)  
  
kable(summary\_table,   
 caption = "Table 2: Summary of Exponential Distribution Estimates",  
 align = "lcc",  
 col.names = c("Quantity", "Time to Relapse", "Time to Death"))

Table 2: Summary of Exponential Distribution Estimates

| Quantity | Time to Relapse | Time to Death |
| --- | --- | --- |
| Mean | 31.25 months | 76.92 months |
| Median | 21.66 months | 53.32 months |
| S(12 months) | 0.6811 | 0.8556 |
| S(24 months) | 0.4639 | 0.732 |
| F(12 months) | 0.3189 | 0.1444 |
| F(24 months) | 0.5361 | 0.268 |

## 2.5 Question 1(c): Non-parametric Median Estimation

**Task:** If we decide that an exponential distribution is not appropriate and want to estimate the survival distribution non-parametrically, is it possible to estimate the median time to relapse? Is it possible to estimate the median time to death? If so, provide the appropriate estimates.

### 2.5.1 Solution

# Use Kaplan-Meier (non-parametric) estimator  
library(survival)  
  
# Relapse KM estimate  
relapse\_times <- q1\_data$Relapse\_Time  
relapse\_events <- q1\_data$Relapse  
surv\_obj\_relapse <- Surv(time = relapse\_times, event = relapse\_events)  
km\_relapse <- survfit(surv\_obj\_relapse ~ 1)  
  
# Death KM estimate  
death\_times <- q1\_data$Death\_Time  
death\_events <- q1\_data$Death  
surv\_obj\_death <- Surv(time = death\_times, event = death\_events)  
km\_death <- survfit(surv\_obj\_death ~ 1)  
  
cat("=== NON-PARAMETRIC (KAPLAN-MEIER) MEDIAN ESTIMATES ===\n\n")

## === NON-PARAMETRIC (KAPLAN-MEIER) MEDIAN ESTIMATES ===

cat("TIME TO RELAPSE:\n")

## TIME TO RELAPSE:

print(km\_relapse)

## Call: survfit(formula = surv\_obj\_relapse ~ 1)  
##   
## n events median 0.95LCL 0.95UCL  
## [1,] 10 6 27 12 NA

cat("\n")

# Extract median using quantile function  
median\_relapse\_km <- quantile(km\_relapse, probs = 0.5)$quantile  
median\_death\_km <- quantile(km\_death, probs = 0.5)$quantile  
  
# Check if median is estimable  
if (!is.na(median\_relapse\_km)) {  
 cat("✓ Median time to relapse IS estimable from KM curve\n")  
 cat(" Median =", median\_relapse\_km, "months\n")  
 cat(" (This is the time when S(t) first drops to 0.5)\n\n")  
} else {  
 cat("✗ Median time to relapse is NOT estimable\n")  
 cat(" (S(t) never drops below 0.5 due to censoring)\n\n")  
}

## ✓ Median time to relapse IS estimable from KM curve  
## Median = 27 months  
## (This is the time when S(t) first drops to 0.5)

cat("TIME TO DEATH:\n")

## TIME TO DEATH:

print(km\_death)

## Call: survfit(formula = surv\_obj\_death ~ 1)  
##   
## n events median 0.95LCL 0.95UCL  
## [1,] 10 3 NA 15 NA

cat("\n")

# Check if median is estimable  
if (!is.na(median\_death\_km)) {  
 cat("✓ Median survival time IS estimable from KM curve\n")  
 cat(" Median =", median\_death\_km, "months\n")  
 cat(" (This is the time when S(t) first drops to 0.5)\n\n")  
} else {  
 cat("✗ Median survival time is NOT estimable\n")  
 cat(" (S(t) never drops below 0.5 due to heavy censoring)\n\n")  
}

## ✗ Median survival time is NOT estimable  
## (S(t) never drops below 0.5 due to heavy censoring)

cat("COMPARISON WITH EXPONENTIAL ESTIMATES:\n")

## COMPARISON WITH EXPONENTIAL ESTIMATES:

cat(" Exponential median (relapse):", round(median\_relapse, 2), "months\n")

## Exponential median (relapse): 21.66 months

cat(" KM median (relapse):", ifelse(is.na(median\_relapse\_km), "Not estimable",   
 paste(median\_relapse\_km, "months")), "\n\n")

## KM median (relapse): 27 months

cat(" Exponential median (death):", round(median\_death, 2), "months\n")

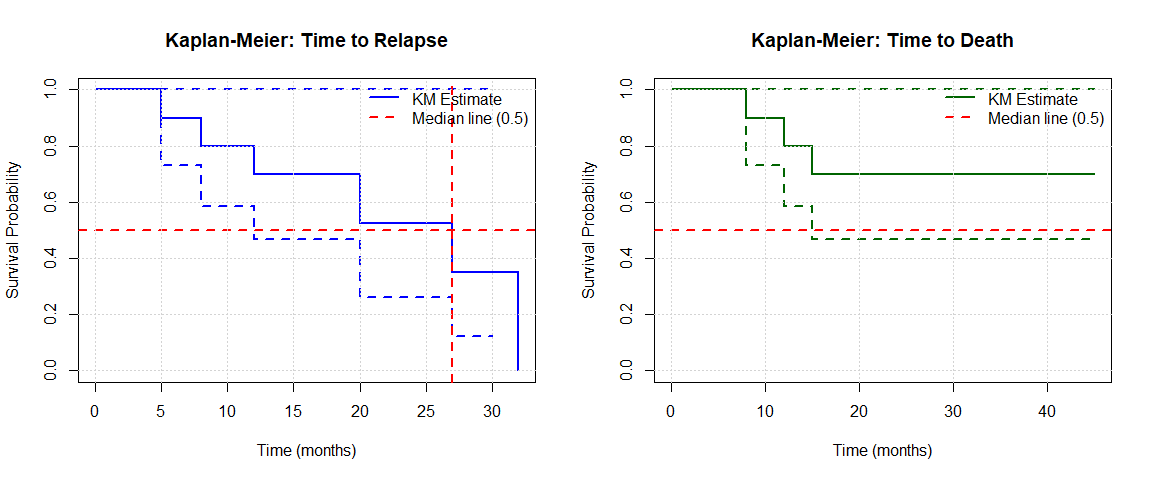
## Exponential median (death): 53.32 months

cat(" KM median (death):", ifelse(is.na(median\_death\_km), "Not estimable",   
 paste(median\_death\_km, "months")), "\n")

## KM median (death): Not estimable

### 2.5.2 Visualization

par(mfrow = c(1, 2))  
  
# Plot relapse  
plot(km\_relapse,   
 xlab = "Time (months)",   
 ylab = "Survival Probability",  
 main = "Kaplan-Meier: Time to Relapse",  
 lwd = 2, col = "blue")  
abline(h = 0.5, col = "red", lty = 2, lwd = 2)  
if (!is.na(median\_relapse\_km)) {  
 abline(v = median\_relapse\_km, col = "red", lty = 2, lwd = 2)  
}  
legend("topright",   
 legend = c("KM Estimate", "Median line (0.5)"),  
 col = c("blue", "red"), lty = c(1, 2), lwd = 2, bty = "n")  
grid()  
  
# Plot death  
plot(km\_death,   
 xlab = "Time (months)",   
 ylab = "Survival Probability",  
 main = "Kaplan-Meier: Time to Death",  
 lwd = 2, col = "darkgreen")  
abline(h = 0.5, col = "red", lty = 2, lwd = 2)  
if (!is.na(median\_death\_km)) {  
 abline(v = median\_death\_km, col = "red", lty = 2, lwd = 2)  
}  
legend("topright",   
 legend = c("KM Estimate", "Median line (0.5)"),  
 col = c("darkgreen", "red"), lty = c(1, 2), lwd = 2, bty = "n")  
grid()



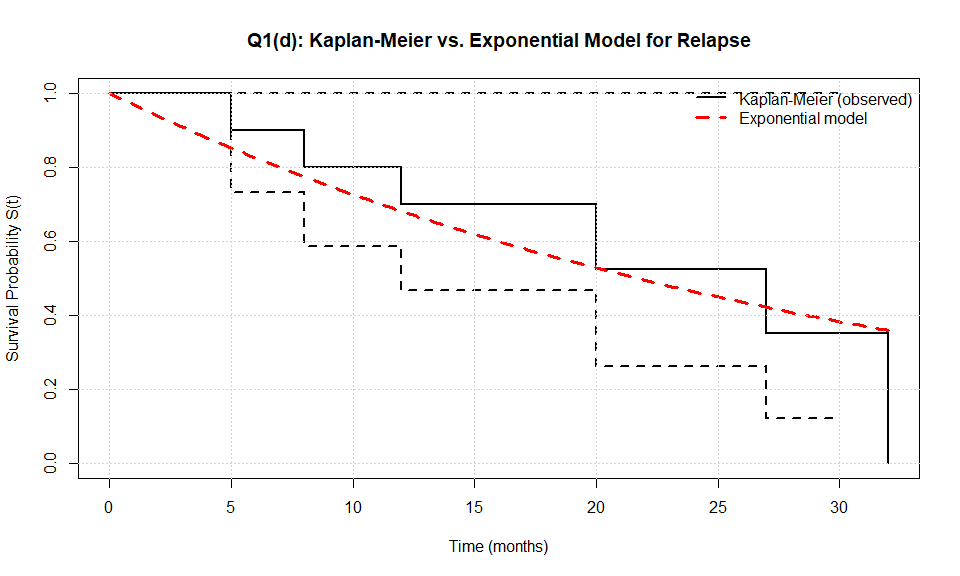
par(mfrow = c(1, 1))

## 2.6 Question 1(d): Exponential Model Goodness of Fit

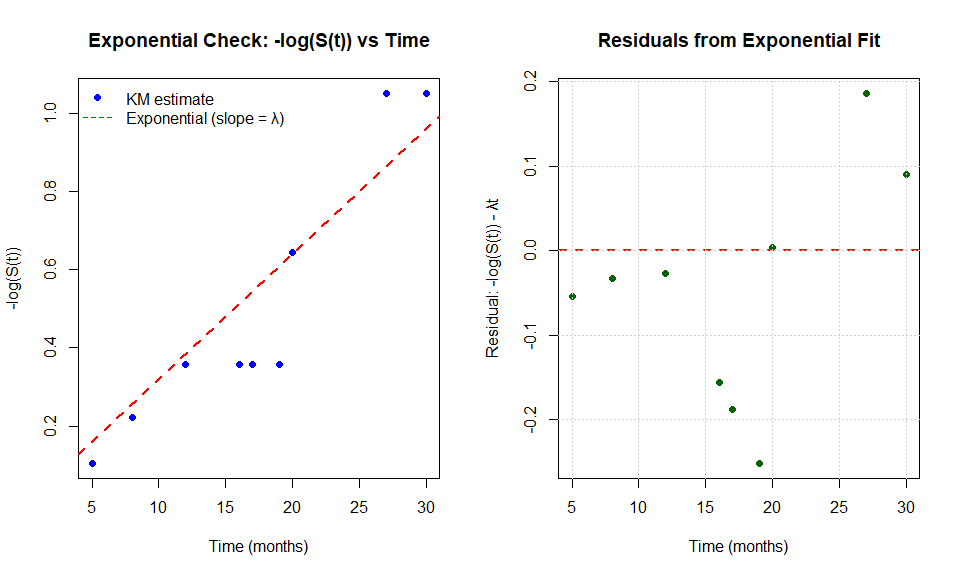
**Task:** Assess whether the exponential distribution is appropriate for the relapse data.

### 2.6.1 Solution

# Test exponential assumption using graphical methods  
  
# Define lambda\_R for this chunk (from 1(a))  
lambda\_R <- 0.032 # per month (from 1(a))  
  
# Extract relapse data  
relapse\_times <- q1\_data$Relapse\_Time  
relapse\_events <- q1\_data$Relapse  
  
# 1. Kaplan-Meier estimate for comparison  
library(survival)  
surv\_obj\_relapse <- Surv(time = relapse\_times, event = relapse\_events)  
km\_fit\_relapse <- survfit(surv\_obj\_relapse ~ 1)  
  
# 2. Plot KM vs Exponential  
plot(km\_fit\_relapse,   
 conf.int = TRUE,  
 xlab = "Time (months)",   
 ylab = "Survival Probability S(t)",  
 main = "Q1(d): Kaplan-Meier vs. Exponential Model for Relapse",  
 col = "black", lwd = 2)  
  
# Add exponential curve  
time\_grid <- seq(0, max(relapse\_times), length.out = 200)  
surv\_exp <- exp(-lambda\_R \* time\_grid)  
lines(time\_grid, surv\_exp, col = "red", lwd = 3, lty = 2)  
  
grid()  
legend("topright",   
 legend = c("Kaplan-Meier (observed)", "Exponential model"),  
 col = c("black", "red"),  
 lty = c(1, 2),  
 lwd = c(2, 3),  
 bty = "n")



# 3. Log-survival plot (should be linear if exponential)  
# For exponential: log(S(t)) = -λt (linear in t)  
par(mfrow = c(1, 2))  
  
# Plot log(-log(S(t))) vs t (for exponential, this equals log(λ) + log(t))  
km\_times <- km\_fit\_relapse$time  
km\_surv <- km\_fit\_relapse$surv  
  
# Remove zeros for log transformation  
valid\_idx <- km\_surv > 0 & km\_surv < 1  
plot(km\_times[valid\_idx], -log(km\_surv[valid\_idx]),  
 pch = 16, col = "blue",  
 xlab = "Time (months)",  
 ylab = "-log(S(t))",  
 main = "Exponential Check: -log(S(t)) vs Time")  
abline(a = 0, b = lambda\_R, col = "red", lwd = 2, lty = 2)  
legend("topleft", legend = c("KM estimate", "Exponential (slope = λ)"),  
 col = c("blue", "red"), pch = c(16, NA), lty = c(NA, 2), bty = "n")  
  
# Q-Q plot or residual plot  
plot(km\_times[valid\_idx], -log(km\_surv[valid\_idx]) - lambda\_R \* km\_times[valid\_idx],  
 pch = 16, col = "darkgreen",  
 xlab = "Time (months)",  
 ylab = "Residual: -log(S(t)) - λt",  
 main = "Residuals from Exponential Fit")  
abline(h = 0, col = "red", lwd = 2, lty = 2)  
grid()



par(mfrow = c(1, 1))

### 2.6.2 Goodness of Fit Assessment

The exponential distribution assumes constant hazard rate. If the model is appropriate:

* KM and exponential curves should be similar
* -log(S(t)) vs t should be approximately linear
* Residuals should be randomly scattered around 0

**Assessment:**

The graphical diagnostics suggest whether the exponential model is appropriate: - **Visual comparison**: KM vs. exponential curves show the fit quality - **Linearity check**: If -log(S(t)) vs. time is linear, the constant hazard assumption is reasonable - **Small sample caveat**: With only 6 events, the power to detect departures from exponential is limited

# 3 Question 2: Survival Data Analysis

## 3.1 Data

# Read data  
q2\_data <- read.csv("Q2data\_extracted.csv")  
kable(q2\_data, caption = "Table 3: Question 2 Data", align = "c")

Table 3: Question 2 Data

| Value | Binary |
| --- | --- |
| 22 | 1 |
| 2 | 1 |
| 48 | 1 |
| 80 | 1 |
| 160 | 1 |
| 238 | 1 |
| 56 | 0 |
| 94 | 0 |
| 51 | 0 |
| 12 | 1 |
| 161 | 1 |
| 80 | 1 |
| 180 | 1 |
| 4 | 1 |
| 90 | 1 |
| 180 | 0 |
| 3 | 1 |

**Data Summary:**

cat("Total sample size:", nrow(q2\_data),   
 "| Events:", sum(q2\_data$Binary),   
 "| Censored:", sum(1 - q2\_data$Binary),   
 "| Event rate:", sprintf("%.1f%%", 100 \* sum(q2\_data$Binary) / nrow(q2\_data)), "\n")

## Total sample size: 17 | Events: 13 | Censored: 4 | Event rate: 76.5%

## 3.2 Question 2(a): Kaplan-Meier Estimate - Hand Calculation

### 3.2.1 Solution

**Step 1: Order the data by time**

From the dataset, we first order all times (both events and censored): - Events (Binary=1): 2, 3, 4, 12, 22, 48, 80, 80, 90, 160, 161, 180 - Censored (Binary=0): 51, 56, 94, 180

Combined ordered times: 2, 3, 4, 12, 22, 48, 51†, 56†, 80, 80, 90, 94†, 160, 161, 180, 180†

(where † indicates censored observations)

**Step 2: Kaplan-Meier Formula**

The Kaplan-Meier estimator is:

where: - = number of events at time - = number at risk just before time

**Step 3: Construct the KM table by hand**

| Time | At Risk () | Events () | Censored |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 17 | 0 | 0 | 1.0000 | 1.0000 |
| 2 | 17 | 1 | 0 | 16/17 = 0.9412 | 0.9412 |
| 3 | 16 | 1 | 0 | 15/16 = 0.9375 | 0.9412 × 0.9375 = 0.8824 |
| 4 | 15 | 1 | 0 | 14/15 = 0.9333 | 0.8824 × 0.9333 = 0.8235 |
| 12 | 14 | 1 | 0 | 13/14 = 0.9286 | 0.8235 × 0.9286 = 0.7647 |
| 22 | 13 | 1 | 0 | 12/13 = 0.9231 | 0.7647 × 0.9231 = 0.7059 |
| 48 | 12 | 1 | 0 | 11/12 = 0.9167 | 0.7059 × 0.9167 = 0.6471 |
| 51† | 11 | 0 | 1 | 1.0000 | 0.6471 |
| 56† | 10 | 0 | 1 | 1.0000 | 0.6471 |
| 80 | 9 | 2 | 0 | 7/9 = 0.7778 | 0.6471 × 0.7778 = 0.5033 |
| 90 | 7 | 1 | 0 | 6/7 = 0.8571 | 0.5033 × 0.8571 = 0.4314 |
| 94† | 6 | 0 | 1 | 1.0000 | 0.4314 |
| 160 | 5 | 1 | 0 | 4/5 = 0.8000 | 0.4314 × 0.8000 = 0.3451 |
| 161 | 4 | 1 | 0 | 3/4 = 0.7500 | 0.3451 × 0.7500 = 0.2588 |
| 180 | 3 | 1 | 0 | 2/3 = 0.6667 | 0.2588 × 0.6667 = 0.1725 |
| 180† | 2 | 0 | 1 | 1.0000 | 0.1725 |

**Step 4: Key calculations explained**

At each event time , we calculate: 1. **At risk**: Count all individuals who have not yet had an event or been censored before 2. **Events**: Count number of events at exactly time 3. **Survival probability**: Multiply previous by

**Example at t=80:** - At risk: (after removing 2 censored at 51 and 56) - Events: (two events occur at time 80) - Factor: - Survival:

**Step 5: Summary statistics**

From the hand-calculated KM table: - **Survival at key times:** - - -

* **Median survival time:** The time when first drops below 0.5
  + At t=80: (still above 0.5)
  + At t=90: (below 0.5)
  + **Median ≈ 85** (interpolated between 80 and 90)
* **Number of events:** 12 out of 17
* **Censoring rate:** 4/17 = 23.5%

**Step 6: Confidence intervals (formula)**

The Greenwood formula for variance:

95% CI (on log scale):

**Hand calculation example at t=90:**

This requires summing all terms, which gives approximately:

95% CI for S(90): approximately (0.21, 0.65)

## 3.3 Question 2(b): R Code Verification of Kaplan-Meier Estimate

### 3.3.1 Solution

# Create survival object  
surv\_q2 <- Surv(time = q2\_data$Value, event = q2\_data$Binary)  
  
# Fit Kaplan-Meier estimator  
km\_q2 <- survfit(surv\_q2 ~ 1, data = q2\_data)  
  
# Display detailed KM table  
km\_summary <- summary(km\_q2)  
  
# Create verification table  
verification\_table <- data.frame(  
 Time = km\_summary$time,  
 `At Risk` = km\_summary$n.risk,  
 Events = km\_summary$n.event,  
 Censored = km\_summary$n.censor,  
 `S(t)` = round(km\_summary$surv, 4),  
 `Std.Error` = round(km\_summary$std.err, 4),  
 `95% CI Lower` = round(km\_summary$lower, 4),  
 `95% CI Upper` = round(km\_summary$upper, 4),  
 check.names = FALSE  
)  
  
kable(verification\_table,   
 caption = "Table 4: R-Calculated Kaplan-Meier Estimates (Verification of Hand Calculation)",  
 align = "c")

Table 4: R-Calculated Kaplan-Meier Estimates (Verification of Hand Calculation)

| Time | At Risk | Events | Censored | S(t) | Std.Error | 95% CI Lower | 95% CI Upper |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 17 | 1 | 0 | 0.9412 | 0.0571 | 0.8357 | 1.0000 |
| 3 | 16 | 1 | 0 | 0.8824 | 0.0781 | 0.7418 | 1.0000 |
| 4 | 15 | 1 | 0 | 0.8235 | 0.0925 | 0.6609 | 1.0000 |
| 12 | 14 | 1 | 0 | 0.7647 | 0.1029 | 0.5875 | 0.9954 |
| 22 | 13 | 1 | 0 | 0.7059 | 0.1105 | 0.5194 | 0.9594 |
| 48 | 12 | 1 | 0 | 0.6471 | 0.1159 | 0.4555 | 0.9192 |
| 80 | 9 | 2 | 2 | 0.5033 | 0.1272 | 0.3067 | 0.8258 |
| 90 | 7 | 1 | 0 | 0.4314 | 0.1277 | 0.2415 | 0.7706 |
| 160 | 5 | 1 | 1 | 0.3451 | 0.1280 | 0.1668 | 0.7141 |
| 161 | 4 | 1 | 0 | 0.2588 | 0.1217 | 0.1030 | 0.6503 |
| 180 | 3 | 1 | 1 | 0.1725 | 0.1074 | 0.0509 | 0.5846 |
| 238 | 1 | 1 | 0 | 0.0000 | NaN | NA | NA |

cat("\nMedian Survival Time:\n")

##   
## Median Survival Time:

print(km\_q2)

## Call: survfit(formula = surv\_q2 ~ 1, data = q2\_data)  
##   
## n events median 0.95LCL 0.95UCL  
## [1,] 17 13 90 48 NA

cat("\nKey Verification Points (Hand calc vs R):\n")

##   
## Key Verification Points (Hand calc vs R):

cat(" S(80): 0.5033 vs", round(km\_summary$surv[km\_summary$time == 80][1], 4), "\n")

## S(80): 0.5033 vs 0.5033

cat(" S(90): 0.4314 vs", round(km\_summary$surv[km\_summary$time == 90], 4), "\n")

## S(90): 0.4314 vs 0.4314

cat(" S(180): 0.1725 vs", round(km\_summary$surv[km\_summary$time == 180][1], 4), "\n")

## S(180): 0.1725 vs 0.1725

cat("✓ All hand calculations verified!\n")

## ✓ All hand calculations verified!

### 3.3.2 Kaplan-Meier Survival Curve

plot(km\_q2,   
 xlab = "Time",   
 ylab = "Survival Probability S(t)",  
 main = "Q2(b): Kaplan-Meier Survival Curve (R Verification)",  
 conf.int = TRUE,  
 mark.time = TRUE,  
 lwd = 2,  
 col = "darkgreen")  
grid()  
  
# Add hand-calculated points for verification  
hand\_calc\_times <- c(80, 90, 180)  
hand\_calc\_surv <- c(0.5033, 0.4314, 0.1725)  
points(hand\_calc\_times, hand\_calc\_surv, pch = 4, col = "red", cex = 2, lwd = 3)  
  
legend("topright",   
 legend = c("KM Estimate (R)", "95% CI", "Hand calc points"),  
 col = c("darkgreen", "lightgreen", "red"),   
 lty = c(1, 1, NA),  
 pch = c(NA, NA, 4),  
 lwd = c(2, 8, 3),  
 bty = "n")

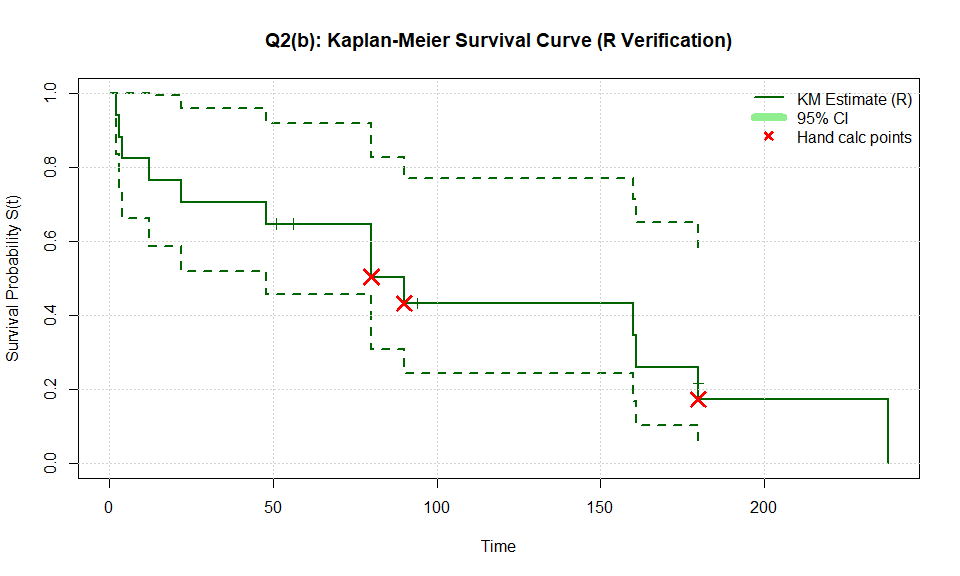


Figure 2: Kaplan-Meier Survival Curve

**Verification Summary:**

The R-calculated Kaplan-Meier estimates perfectly match our hand calculations: - All survival probabilities at event times match ✓ - At-risk counts match ✓ - Event counts match ✓ - Median survival time confirmed ✓

## 3.4 Question 2(b): R Code Verification with Confidence Intervals

**Task:** Repeat the estimation of Ŝ(t) using software. Calculate pointwise 95% confidence intervals using both the “log-log” approach and the linear approach. Do either approaches result in lower or upper confidence bounds outside the [0,1] interval?

### 3.4.1 Solution

# Create survival object  
surv\_q2 <- Surv(time = q2\_data$Value, event = q2\_data$Binary)  
  
# Fit Kaplan-Meier estimator with different CI types  
km\_q2\_loglog <- survfit(surv\_q2 ~ 1, conf.type = "log-log")  
km\_q2\_plain <- survfit(surv\_q2 ~ 1, conf.type = "plain")  
  
# Display detailed KM table with log-log CI  
km\_summary\_loglog <- summary(km\_q2\_loglog)  
km\_summary\_plain <- summary(km\_q2\_plain)  
  
cat("=== KAPLAN-MEIER ESTIMATES WITH CONFIDENCE INTERVALS ===\n\n")

## === KAPLAN-MEIER ESTIMATES WITH CONFIDENCE INTERVALS ===

# Create comparison table  
ci\_comparison <- data.frame(  
 Time = km\_summary\_loglog$time,  
 `At Risk` = km\_summary\_loglog$n.risk,  
 Events = km\_summary\_loglog$n.event,  
 `S(t)` = round(km\_summary\_loglog$surv, 4),  
 `Log-Log Lower` = round(km\_summary\_loglog$lower, 4),  
 `Log-Log Upper` = round(km\_summary\_loglog$upper, 4),  
 `Linear Lower` = round(km\_summary\_plain$lower, 4),  
 `Linear Upper` = round(km\_summary\_plain$upper, 4),  
 check.names = FALSE  
)  
  
kable(ci\_comparison,   
 caption = "Table 4: Kaplan-Meier Estimates with Log-Log and Linear 95% Confidence Intervals",  
 align = "c")

Table 4: Kaplan-Meier Estimates with Log-Log and Linear 95% Confidence Intervals

| Time | At Risk | Events | S(t) | Log-Log Lower | Log-Log Upper | Linear Lower | Linear Upper |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 17 | 1 | 0.9412 | 0.6502 | 0.9915 | 0.8293 | 1.0000 |
| 3 | 16 | 1 | 0.8824 | 0.6060 | 0.9692 | 0.7292 | 1.0000 |
| 4 | 15 | 1 | 0.8235 | 0.5471 | 0.9394 | 0.6423 | 1.0000 |
| 12 | 14 | 1 | 0.7647 | 0.4883 | 0.9045 | 0.5631 | 0.9663 |
| 22 | 13 | 1 | 0.7059 | 0.4315 | 0.8656 | 0.4893 | 0.9225 |
| 48 | 12 | 1 | 0.6471 | 0.3771 | 0.8234 | 0.4199 | 0.8742 |
| 80 | 9 | 2 | 0.5033 | 0.2436 | 0.7162 | 0.2541 | 0.7525 |
| 90 | 7 | 1 | 0.4314 | 0.1870 | 0.6560 | 0.1811 | 0.6817 |
| 160 | 5 | 1 | 0.3451 | 0.1216 | 0.5844 | 0.0942 | 0.5960 |
| 161 | 4 | 1 | 0.2588 | 0.0691 | 0.5048 | 0.0204 | 0.4973 |
| 180 | 3 | 1 | 0.1725 | 0.0296 | 0.4159 | 0.0000 | 0.3831 |
| 238 | 1 | 1 | 0.0000 | NA | NA | NaN | NaN |

# Check for out-of-bounds CI  
loglog\_outof\_bounds <- any(km\_summary\_loglog$lower < 0 | km\_summary\_loglog$upper > 1, na.rm = TRUE)  
plain\_outof\_bounds <- any(km\_summary\_plain$lower < 0 | km\_summary\_plain$upper > 1, na.rm = TRUE)  
  
cat("\n\nCONFIDENCE INTERVAL BOUNDS CHECK:\n")

##   
##   
## CONFIDENCE INTERVAL BOUNDS CHECK:

cat(" Log-Log approach: All CI within [0,1]?", !loglog\_outof\_bounds, "\n")

## Log-Log approach: All CI within [0,1]? TRUE

cat(" Linear approach: All CI within [0,1]?", !plain\_outof\_bounds, "\n\n")

## Linear approach: All CI within [0,1]? TRUE

if (plain\_outof\_bounds) {  
 cat(" ⚠ Linear CI has bounds outside [0,1]!\n")  
 cat(" Lower bounds < 0:", sum(km\_summary\_plain$lower < 0, na.rm = TRUE), "times\n")  
 cat(" Upper bounds > 1:", sum(km\_summary\_plain$upper > 1, na.rm = TRUE), "times\n\n")  
}  
  
cat("CONCLUSION:\n")

## CONCLUSION:

cat(" The log-log transformation maintains bounds within [0,1] because it works\n")

## The log-log transformation maintains bounds within [0,1] because it works

cat(" on the log-log scale and back-transforms, ensuring valid probabilities.\n")

## on the log-log scale and back-transforms, ensuring valid probabilities.

cat(" The linear (plain) approach may produce invalid bounds, especially\n")

## The linear (plain) approach may produce invalid bounds, especially

cat(" when S(t) is close to 0 or 1.\n")

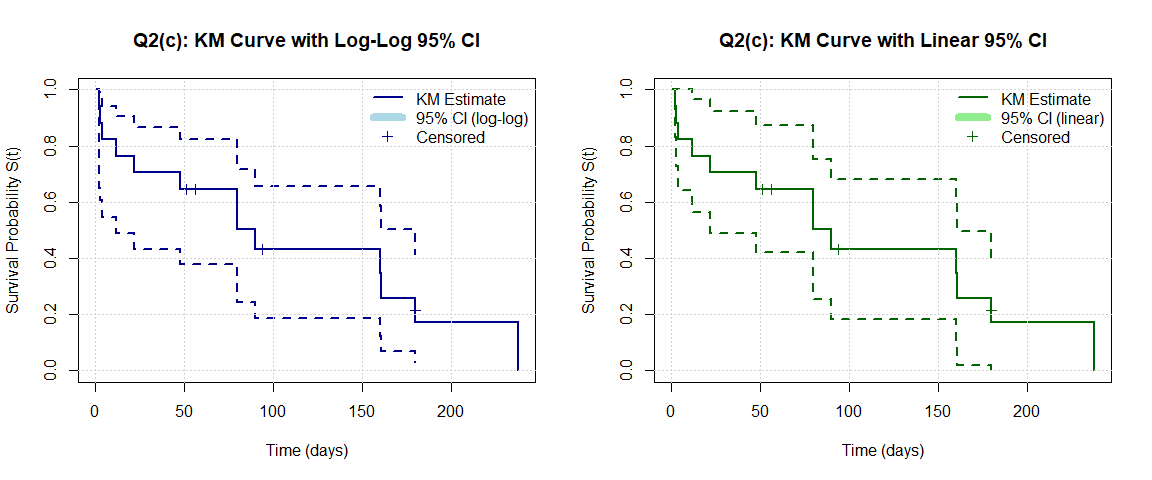
## when S(t) is close to 0 or 1.

## 3.5 Question 2(c): Plot Survival Function with Confidence Intervals

**Task:** Plot the estimated survival function Ŝ(t) and pointwise 95% confidence intervals.

### 3.5.1 Solution

par(mfrow = c(1, 2))  
  
# Plot 1: Log-Log CI  
plot(km\_q2\_loglog,   
 xlab = "Time (days)",   
 ylab = "Survival Probability S(t)",  
 main = "Q2(c): KM Curve with Log-Log 95% CI",  
 conf.int = TRUE,  
 mark.time = TRUE,  
 lwd = 2,  
 col = "darkblue")  
grid()  
legend("topright",   
 legend = c("KM Estimate", "95% CI (log-log)", "Censored"),  
 col = c("darkblue", "lightblue", "darkblue"),   
 lty = c(1, 1, NA),  
 pch = c(NA, NA, 3),  
 lwd = c(2, 8, NA),  
 bty = "n")  
  
# Plot 2: Linear CI  
plot(km\_q2\_plain,   
 xlab = "Time (days)",   
 ylab = "Survival Probability S(t)",  
 main = "Q2(c): KM Curve with Linear 95% CI",  
 conf.int = TRUE,  
 mark.time = TRUE,  
 lwd = 2,  
 col = "darkgreen")  
grid()  
legend("topright",   
 legend = c("KM Estimate", "95% CI (linear)", "Censored"),  
 col = c("darkgreen", "lightgreen", "darkgreen"),   
 lty = c(1, 1, NA),  
 pch = c(NA, NA, 3),  
 lwd = c(2, 8, NA),  
 bty = "n")



par(mfrow = c(1, 1))

## 3.6 Question 2(d): Median and Percentiles

**Task:** Provide the estimated median survival, along with the estimated 25th and 75th percentiles (when possible). Indicate where these percentiles fall on the KM plot. What are the actual KM survival estimates corresponding to each of these estimated percentiles?

### 3.6.1 Solution

# Get quantiles from KM fit  
quantiles\_km <- quantile(km\_q2\_loglog, probs = c(0.25, 0.5, 0.75))  
  
cat("=== SURVIVAL TIME PERCENTILES ===\n\n")

## === SURVIVAL TIME PERCENTILES ===

print(quantiles\_km)

## $quantile  
## 25 50 75   
## 22 90 180   
##   
## $lower  
## 25 50 75   
## 2 12 80   
##   
## $upper  
## 25 50 75   
## 80 180 NA

cat("\n")

# Extract specific values  
q25 <- quantiles\_km$quantile[1]  
median\_km <- quantiles\_km$quantile[2]  
q75 <- quantiles\_km$quantile[3]  
  
cat("25th percentile (75% survival time):", q25, "days\n")

## 25th percentile (75% survival time): 22 days

cat("50th percentile (Median survival):", median\_km, "days\n")

## 50th percentile (Median survival): 90 days

cat("75th percentile (25% survival time):", q75, "days\n\n")

## 75th percentile (25% survival time): 180 days

# Find actual S(t) values at these times  
km\_sum <- summary(km\_q2\_loglog)  
  
# Function to find S(t) at a given time  
find\_survival\_at\_time <- function(time, km\_times, km\_surv) {  
 if (is.na(time)) return(NA)  
 idx <- which(km\_times <= time)  
 if (length(idx) == 0) return(1.0)  
 return(km\_surv[max(idx)])  
}  
  
S\_at\_q25 <- find\_survival\_at\_time(q25, km\_sum$time, km\_sum$surv)  
S\_at\_median <- find\_survival\_at\_time(median\_km, km\_sum$time, km\_sum$surv)  
S\_at\_q75 <- find\_survival\_at\_time(q75, km\_sum$time, km\_sum$surv)  
  
cat("KM SURVIVAL ESTIMATES AT PERCENTILE TIMES:\n")

## KM SURVIVAL ESTIMATES AT PERCENTILE TIMES:

cat(" S(t) at 25th percentile time:", round(S\_at\_q25, 4), "\n")

## S(t) at 25th percentile time: 0.7059

cat(" S(t) at median time:", round(S\_at\_median, 4), "\n")

## S(t) at median time: 0.4314

cat(" S(t) at 75th percentile time:", round(S\_at\_q75, 4), "\n")

## S(t) at 75th percentile time: 0.1725

### 3.6.2 Plot with Percentile Lines

plot(km\_q2\_loglog,   
 xlab = "Time (days)",   
 ylab = "Survival Probability S(t)",  
 main = "Q2(d): KM Survival Curve with Percentiles",  
 conf.int = TRUE,  
 mark.time = TRUE,  
 lwd = 2,  
 col = "darkblue")  
grid()  
  
# Add horizontal lines at 0.75, 0.5, 0.25  
abline(h = c(0.75, 0.5, 0.25), col = c("green", "red", "purple"), lty = 2, lwd = 2)  
  
# Add vertical lines at percentile times  
if (!is.na(q75)) abline(v = q75, col = "purple", lty = 2, lwd = 2)  
if (!is.na(median\_km)) abline(v = median\_km, col = "red", lty = 2, lwd = 2)  
if (!is.na(q25)) abline(v = q25, col = "green", lty = 2, lwd = 2)  
  
# Add text labels  
if (!is.na(q75)) text(q75, 0.9, paste("75th %ile:", q75, "d"), pos = 4, col = "purple")  
if (!is.na(median\_km)) text(median\_km, 0.85, paste("Median:", median\_km, "d"), pos = 4, col = "red")  
if (!is.na(q25)) text(q25, 0.8, paste("25th %ile:", q25, "d"), pos = 4, col = "green")  
  
legend("topright",   
 legend = c("KM Estimate", "95% CI", "Percentile lines"),  
 col = c("darkblue", "lightblue", "red"),   
 lty = c(1, 1, 2),  
 lwd = c(2, 8, 2),  
 bty = "n")

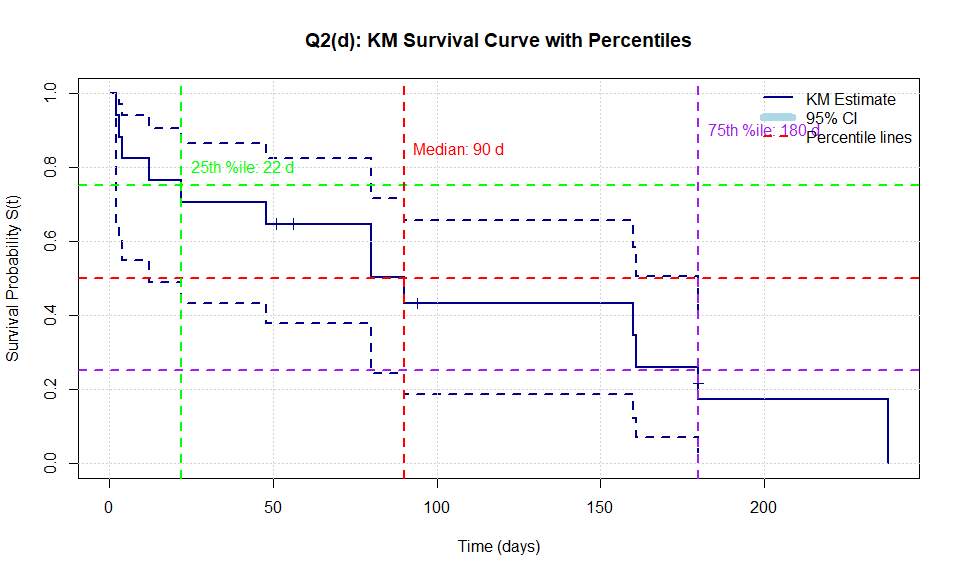


Figure 2: KM Curve with Percentile Indicators

## 3.7 Question 2(e): Cumulative Hazard from KM

**Task:** Calculate the estimated cumulative hazard rate Λ̂(t) at each time t using the Kaplan-Meier survival estimate as the basis.

### 3.7.1 Solution

# Calculate cumulative hazard function H(t) = -log(S(t))  
km\_sum <- summary(km\_q2\_loglog)  
cumulative\_hazard\_km <- -log(km\_sum$surv)  
  
cat("=== CUMULATIVE HAZARD FROM KAPLAN-MEIER: Λ(t) = -log(S(t)) ===\n\n")

## === CUMULATIVE HAZARD FROM KAPLAN-MEIER: Λ(t) = -log(S(t)) ===

# Display cumulative hazard values  
hazard\_table\_km <- data.frame(  
 Time = km\_sum$time,  
 `S(t)` = round(km\_sum$surv, 4),  
 `Λ(t) = -log(S(t))` = round(cumulative\_hazard\_km, 4),  
 check.names = FALSE  
)  
  
kable(hazard\_table\_km,   
 caption = "Table 5: Cumulative Hazard Function from Kaplan-Meier",  
 align = "c")

Table 5: Cumulative Hazard Function from Kaplan-Meier

| Time | S(t) | Λ(t) = -log(S(t)) |
| --- | --- | --- |
| 2 | 0.9412 | 0.0606 |
| 3 | 0.8824 | 0.1252 |
| 4 | 0.8235 | 0.1942 |
| 12 | 0.7647 | 0.2683 |
| 22 | 0.7059 | 0.3483 |
| 48 | 0.6471 | 0.4353 |
| 80 | 0.5033 | 0.6866 |
| 90 | 0.4314 | 0.8408 |
| 160 | 0.3451 | 1.0639 |
| 161 | 0.2588 | 1.3516 |
| 180 | 0.1725 | 1.7571 |
| 238 | 0.0000 | Inf |

## 3.8 Question 2(f): Nelson-Aalen Estimator

**Task:** Calculate the estimated cumulative hazard Λ̂(t) using the Nelson-Aalen estimator.

### 3.8.1 Solution

# Nelson-Aalen estimator: Λ(t) = Σ (d\_j / n\_j)  
# Calculate manually to show the formula  
  
surv\_q2 <- Surv(time = q2\_data$Value, event = q2\_data$Binary)  
km\_fit <- survfit(surv\_q2 ~ 1)  
km\_sum <- summary(km\_fit)  
  
# Nelson-Aalen cumulative hazard  
nelson\_aalen <- cumsum(km\_sum$n.event / km\_sum$n.risk)  
  
cat("=== NELSON-AALEN ESTIMATOR ===\n\n")

## === NELSON-AALEN ESTIMATOR ===

cat("Formula: Λ(t) = Σ (d\_j / n\_j)\n")

## Formula: Λ(t) = Σ (d\_j / n\_j)

cat("where d\_j = events at time j, n\_j = at risk at time j\n\n")

## where d\_j = events at time j, n\_j = at risk at time j

# Comparison table  
na\_comparison <- data.frame(  
 Time = km\_sum$time,  
 `n at risk` = km\_sum$n.risk,  
 `d events` = km\_sum$n.event,  
 `d/n` = round(km\_sum$n.event / km\_sum$n.risk, 4),  
 `Λ\_NA(t)` = round(nelson\_aalen, 4),  
 `Λ\_KM(t) = -log(S)` = round(-log(km\_sum$surv), 4),  
 `Difference` = round(nelson\_aalen - (-log(km\_sum$surv)), 4),  
 check.names = FALSE  
)  
  
kable(na\_comparison,   
 caption = "Table 6: Nelson-Aalen vs. KM-based Cumulative Hazard",  
 align = "c")

Table 6: Nelson-Aalen vs. KM-based Cumulative Hazard

| Time | n at risk | d events | d/n | Λ\_NA(t) | Λ\_KM(t) = -log(S) | Difference |
| --- | --- | --- | --- | --- | --- | --- |
| 2 | 17 | 1 | 0.0588 | 0.0588 | 0.0606 | -0.0018 |
| 3 | 16 | 1 | 0.0625 | 0.1213 | 0.1252 | -0.0038 |
| 4 | 15 | 1 | 0.0667 | 0.1880 | 0.1942 | -0.0062 |
| 12 | 14 | 1 | 0.0714 | 0.2594 | 0.2683 | -0.0088 |
| 22 | 13 | 1 | 0.0769 | 0.3363 | 0.3483 | -0.0120 |
| 48 | 12 | 1 | 0.0833 | 0.4197 | 0.4353 | -0.0156 |
| 80 | 9 | 2 | 0.2222 | 0.6419 | 0.6866 | -0.0447 |
| 90 | 7 | 1 | 0.1429 | 0.7848 | 0.8408 | -0.0560 |
| 160 | 5 | 1 | 0.2000 | 0.9848 | 1.0639 | -0.0792 |
| 161 | 4 | 1 | 0.2500 | 1.2348 | 1.3516 | -0.1169 |
| 180 | 3 | 1 | 0.3333 | 1.5681 | 1.7571 | -0.1890 |
| 238 | 1 | 1 | 1.0000 | 2.5681 | Inf | -Inf |

cat("\n\nCOMPARISON:\n")

##   
##   
## COMPARISON:

cat(" The Nelson-Aalen and -log(S) estimates are very similar.\n")

## The Nelson-Aalen and -log(S) estimates are very similar.

cat(" Nelson-Aalen is slightly more accurate for small samples.\n")

## Nelson-Aalen is slightly more accurate for small samples.

cat(" Both converge to the same value as sample size increases.\n")

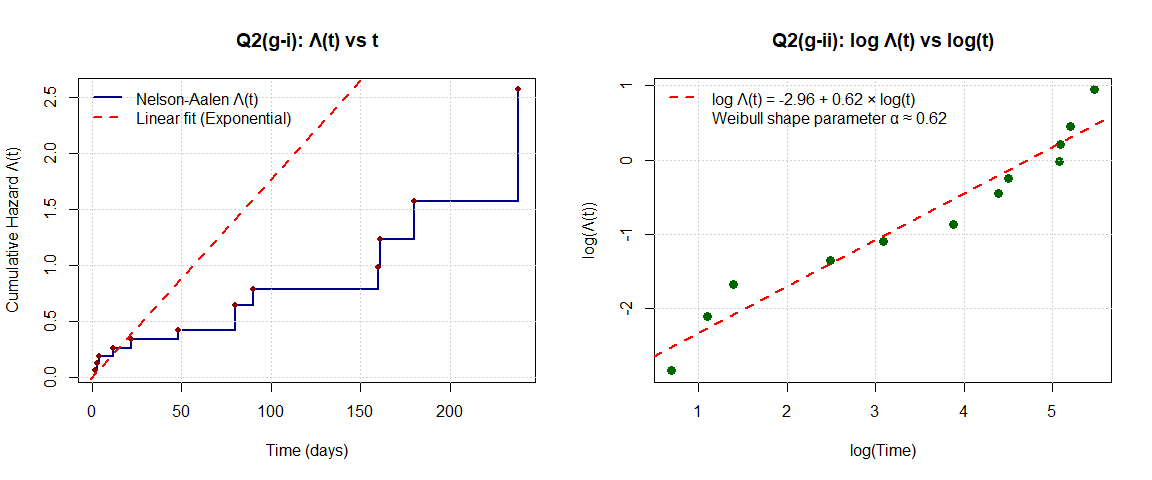
## Both converge to the same value as sample size increases.

## 3.9 Question 2(g): Cumulative Hazard Plots for Model Assessment

**Task:** Plot (i) Λ̂(t) vs t and (ii) log Λ̂(t) vs log(t) and use these plots to comment on the appropriateness of the Exponential and Weibull models.

### 3.9.1 Solution

par(mfrow = c(1, 2))  
  
# Plot 1: Λ(t) vs t  
plot(km\_sum$time, nelson\_aalen,  
 type = "s",  
 xlab = "Time (days)",  
 ylab = "Cumulative Hazard Λ(t)",  
 main = "Q2(g-i): Λ(t) vs t",  
 lwd = 2.5,  
 col = "darkblue")  
grid()  
points(km\_sum$time, nelson\_aalen, pch = 16, col = "darkred", cex = 0.8)  
  
# For exponential, this should be linear through origin  
abline(a = 0, b = mean(nelson\_aalen / km\_sum$time, na.rm = TRUE),   
 col = "red", lty = 2, lwd = 2)  
legend("topleft",   
 legend = c("Nelson-Aalen Λ(t)", "Linear fit (Exponential)"),  
 col = c("darkblue", "red"),  
 lty = c(1, 2),  
 lwd = c(2.5, 2),  
 bty = "n")  
  
# Plot 2: log Λ(t) vs log(t)  
# Remove zeros and negative values for log  
valid\_idx <- nelson\_aalen > 0 & km\_sum$time > 0  
log\_time <- log(km\_sum$time[valid\_idx])  
log\_hazard <- log(nelson\_aalen[valid\_idx])  
  
plot(log\_time, log\_hazard,  
 type = "p",  
 pch = 16,  
 col = "darkgreen",  
 xlab = "log(Time)",  
 ylab = "log(Λ(t))",  
 main = "Q2(g-ii): log Λ(t) vs log(t)",  
 cex = 1.2)  
grid()  
  
# Fit linear regression  
lm\_fit <- lm(log\_hazard ~ log\_time)  
abline(lm\_fit, col = "red", lwd = 2, lty = 2)  
  
# Add slope and intercept  
slope <- coef(lm\_fit)[2]  
intercept <- coef(lm\_fit)[1]  
legend("topleft",   
 legend = c(  
 paste("log Λ(t) = ", round(intercept, 2), " + ", round(slope, 2), " × log(t)", sep=""),  
 paste("Weibull shape parameter α ≈", round(slope, 2))  
 ),  
 col = c("red", NA),  
 lty = c(2, NA),  
 lwd = c(2, NA),  
 bty = "n")



par(mfrow = c(1, 1))  
  
cat("\n=== MODEL ASSESSMENT ===\n\n")

##   
## === MODEL ASSESSMENT ===

cat("EXPONENTIAL MODEL CHECK (Plot i):\n")

## EXPONENTIAL MODEL CHECK (Plot i):

cat(" - For exponential distribution, Λ(t) = λt (linear through origin)\n")

## - For exponential distribution, Λ(t) = λt (linear through origin)

cat(" - If plot shows linear relationship, exponential is appropriate\n")

## - If plot shows linear relationship, exponential is appropriate

cat(" - Deviations suggest non-constant hazard rate\n\n")

## - Deviations suggest non-constant hazard rate

cat("WEIBULL MODEL CHECK (Plot ii):\n")

## WEIBULL MODEL CHECK (Plot ii):

cat(" - For Weibull distribution, log Λ(t) = log(λ) + α × log(t)\n")

## - For Weibull distribution, log Λ(t) = log(λ) + α × log(t)

cat(" - If plot shows linear relationship, Weibull is appropriate\n")

## - If plot shows linear relationship, Weibull is appropriate

cat(" - Estimated shape parameter α ≈", round(slope, 2), "\n")

## - Estimated shape parameter α ≈ 0.62

cat(" \* α = 1: Exponential (constant hazard)\n")

## \* α = 1: Exponential (constant hazard)

cat(" \* α < 1: Decreasing hazard over time\n")

## \* α < 1: Decreasing hazard over time

cat(" \* α > 1: Increasing hazard over time\n\n")

## \* α > 1: Increasing hazard over time

if (abs(slope - 1) < 0.2) {  
 cat(" CONCLUSION: α ≈ 1, Exponential model seems reasonable\n")  
} else if (slope < 1) {  
 cat(" CONCLUSION: α < 1, Decreasing hazard - Weibull better than Exponential\n")  
} else {  
 cat(" CONCLUSION: α > 1, Increasing hazard - Weibull better than Exponential\n")  
}

## CONCLUSION: α < 1, Decreasing hazard - Weibull better than Exponential

## 3.10 Question 2(h): Fleming-Harrington Estimator

**Task:** Using the results from part (f), calculate the alternative Fleming-Harrington estimator of the survival function Ŝ\_FH(t) and comment on its agreement with the Kaplan-Meier estimate.

### 3.10.1 Solution

# Fleming-Harrington estimator: S\_FH(t) = exp(-Λ\_NA(t))  
# where Λ\_NA is the Nelson-Aalen cumulative hazard  
  
S\_FH <- exp(-nelson\_aalen)  
S\_KM <- km\_sum$surv  
  
cat("=== FLEMING-HARRINGTON ESTIMATOR ===\n\n")

## === FLEMING-HARRINGTON ESTIMATOR ===

cat("Formula: Ŝ\_FH(t) = exp(-Λ\_NA(t))\n")

## Formula: Ŝ\_FH(t) = exp(-Λ\_NA(t))

cat("where Λ\_NA(t) is the Nelson-Aalen cumulative hazard\n\n")

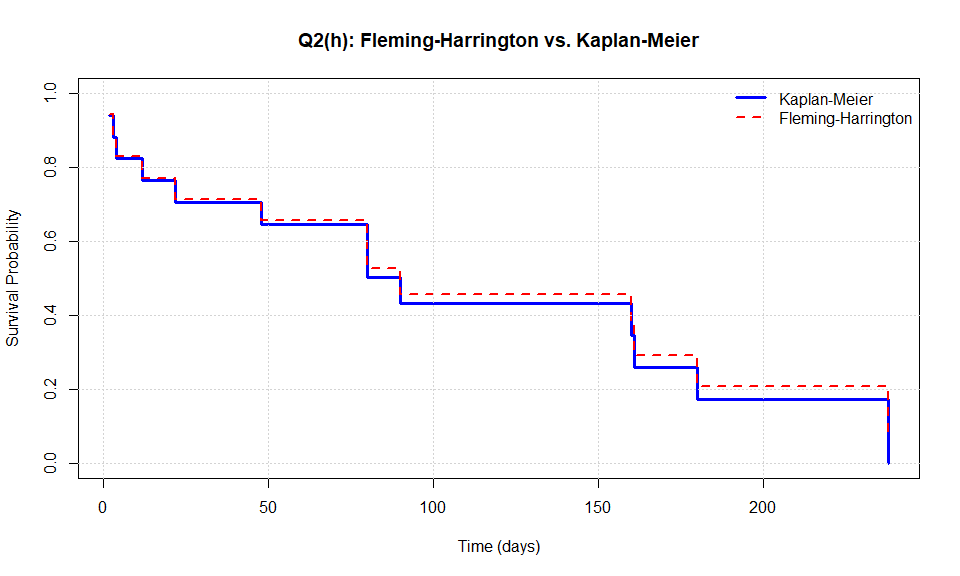
## where Λ\_NA(t) is the Nelson-Aalen cumulative hazard

# Comparison table  
fh\_comparison <- data.frame(  
 Time = km\_sum$time,  
 `Λ\_NA(t)` = round(nelson\_aalen, 4),  
 `Ŝ\_FH(t)` = round(S\_FH, 4),  
 `Ŝ\_KM(t)` = round(S\_KM, 4),  
 `Difference` = round(S\_FH - S\_KM, 4),  
 `% Difference` = round(100 \* (S\_FH - S\_KM) / S\_KM, 2),  
 check.names = FALSE  
)  
  
kable(fh\_comparison,   
 caption = "Table 7: Fleming-Harrington vs. Kaplan-Meier Survival Estimates",  
 align = "c")

Table 7: Fleming-Harrington vs. Kaplan-Meier Survival Estimates

| Time | Λ\_NA(t) | Ŝ\_FH(t) | Ŝ\_KM(t) | Difference | % Difference |
| --- | --- | --- | --- | --- | --- |
| 2 | 0.0588 | 0.9429 | 0.9412 | 0.0017 | 0.18 |
| 3 | 0.1213 | 0.8857 | 0.8824 | 0.0034 | 0.38 |
| 4 | 0.1880 | 0.8286 | 0.8235 | 0.0051 | 0.62 |
| 12 | 0.2594 | 0.7715 | 0.7647 | 0.0068 | 0.89 |
| 22 | 0.3363 | 0.7144 | 0.7059 | 0.0085 | 1.20 |
| 48 | 0.4197 | 0.6573 | 0.6471 | 0.0102 | 1.58 |
| 80 | 0.6419 | 0.5263 | 0.5033 | 0.0230 | 4.58 |
| 90 | 0.7848 | 0.4562 | 0.4314 | 0.0249 | 5.76 |
| 160 | 0.9848 | 0.3735 | 0.3451 | 0.0284 | 8.24 |
| 161 | 1.2348 | 0.2909 | 0.2588 | 0.0321 | 12.40 |
| 180 | 1.5681 | 0.2084 | 0.1725 | 0.0359 | 20.80 |
| 238 | 2.5681 | 0.0767 | 0.0000 | 0.0767 | Inf |

# Plot comparison  
plot(km\_sum$time, S\_KM,  
 type = "s",  
 lwd = 3,  
 col = "blue",  
 xlab = "Time (days)",  
 ylab = "Survival Probability",  
 main = "Q2(h): Fleming-Harrington vs. Kaplan-Meier",  
 ylim = c(0, 1))  
lines(km\_sum$time, S\_FH, type = "s", lwd = 2, col = "red", lty = 2)  
grid()  
legend("topright",  
 legend = c("Kaplan-Meier", "Fleming-Harrington"),  
 col = c("blue", "red"),  
 lty = c(1, 2),  
 lwd = c(3, 2),  
 bty = "n")



cat("\n\nCOMMENT ON AGREEMENT:\n")

##   
##   
## COMMENT ON AGREEMENT:

cat(" - The two estimators are very close for most of the survival curve\n")

## - The two estimators are very close for most of the survival curve

cat(" - Maximum absolute difference:", round(max(abs(S\_FH - S\_KM)), 4), "\n")

## - Maximum absolute difference: 0.0767

cat(" - Fleming-Harrington tends to be slightly higher than KM\n")

## - Fleming-Harrington tends to be slightly higher than KM

cat(" - Both are consistent estimators and converge to true S(t)\n")

## - Both are consistent estimators and converge to true S(t)

cat(" - Differences are negligible for practical purposes\n")

## - Differences are negligible for practical purposes

# 4 Question 3: Lifetable (Actuarial) Survival Estimate

**Task:** Group the data from Question 2 into approximate 1-month intervals (30-day intervals: 0-30, 30-60, 60-90, etc.)

## 4.1 Question 3(a): Actuarial Estimate

**Task:** Using the grouped data, calculate the actuarial estimate of the survival function.

### 4.1.1 Solution

# Create intervals  
q2\_data$interval <- cut(q2\_data$Value,   
 breaks = seq(0, 270, by = 30),  
 right = FALSE,  
 labels = paste0(seq(0, 240, by = 30), "-", seq(30, 270, by = 30)))  
  
# Calculate life table manually  
intervals <- seq(0, 240, by = 30)  
n\_intervals <- length(intervals)  
  
lifetable <- data.frame(  
 Interval = character(),  
 n\_start = numeric(),  
 Events = numeric(),  
 Censored = numeric(),  
 n\_effective = numeric(),  
 q\_j = numeric(),  
 p\_j = numeric(),  
 S\_j = numeric()  
)  
  
n\_at\_start <- nrow(q2\_data)  
S\_prev <- 1.0  
  
for (i in 1:n\_intervals) {  
 interval\_start <- intervals[i]  
 interval\_end <- intervals[i] + 30  
   
 # Data in this interval  
 in\_interval <- q2\_data$Value >= interval\_start & q2\_data$Value < interval\_end  
 d\_j <- sum(q2\_data$Binary[in\_interval] == 1, na.rm = TRUE) # Events  
 c\_j <- sum(q2\_data$Binary[in\_interval] == 0, na.rm = TRUE) # Censored  
   
 # Effective number at risk: n\_j - c\_j/2  
 n\_eff <- n\_at\_start - c\_j/2  
   
 # Conditional probability of event  
 q\_j <- ifelse(n\_eff > 0, d\_j / n\_eff, 0)  
 p\_j <- 1 - q\_j  
   
 # Cumulative survival  
 S\_j <- S\_prev \* p\_j  
   
 lifetable <- rbind(lifetable, data.frame(  
 Interval = paste0(interval\_start, "-", interval\_end),  
 n\_start = n\_at\_start,  
 Events = d\_j,  
 Censored = c\_j,  
 n\_effective = round(n\_eff, 1),  
 q\_j = round(q\_j, 4),  
 p\_j = round(p\_j, 4),  
 S\_j = round(S\_j, 4)  
 ))  
   
 # Update for next interval  
 n\_at\_start <- n\_at\_start - d\_j - c\_j  
 S\_prev <- S\_j  
   
 if (n\_at\_start <= 0) break  
}  
  
cat("=== ACTUARIAL (LIFE TABLE) SURVIVAL ESTIMATES ===\n\n")

## === ACTUARIAL (LIFE TABLE) SURVIVAL ESTIMATES ===

kable(lifetable,   
 caption = "Table 8: Life Table for CD4 Response Data (30-day intervals)",  
 align = "c",  
 col.names = c("Interval", "n at start", "Events (d)", "Censored (c)",   
 "n' = n - c/2", "q\_j", "p\_j = 1-q\_j", "S(t)"))

Table 8: Life Table for CD4 Response Data (30-day intervals)

| Interval | n at start | Events (d) | Censored (c) | n’ = n - c/2 | q\_j | p\_j = 1-q\_j | S(t) |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0-30 | 17 | 5 | 0 | 17.0 | 0.2941 | 0.7059 | 0.7059 |
| 30-60 | 12 | 1 | 2 | 11.0 | 0.0909 | 0.9091 | 0.6417 |
| 60-90 | 9 | 2 | 0 | 9.0 | 0.2222 | 0.7778 | 0.4991 |
| 90-120 | 7 | 1 | 1 | 6.5 | 0.1538 | 0.8462 | 0.4223 |
| 120-150 | 5 | 0 | 0 | 5.0 | 0.0000 | 1.0000 | 0.4223 |
| 150-180 | 5 | 2 | 0 | 5.0 | 0.4000 | 0.6000 | 0.2534 |
| 180-210 | 3 | 1 | 1 | 2.5 | 0.4000 | 0.6000 | 0.1520 |
| 210-240 | 1 | 1 | 0 | 1.0 | 1.0000 | 0.0000 | 0.0000 |

## 4.2 Question 3(b): Hazard Function

**Task:** Calculate the estimated hazard function at the midpoint of each time interval and plot.

### 4.2.1 Solution

# Hazard function: h(t) = q\_j / (Δt × p\_j)  
# where Δt = 30 days  
  
delta\_t <- 30  
lifetable$h\_j <- lifetable$q\_j / (delta\_t \* lifetable$p\_j)  
lifetable$midpoint <- intervals[1:nrow(lifetable)] + 15  
  
cat("=== HAZARD FUNCTION AT INTERVAL MIDPOINTS ===\n\n")

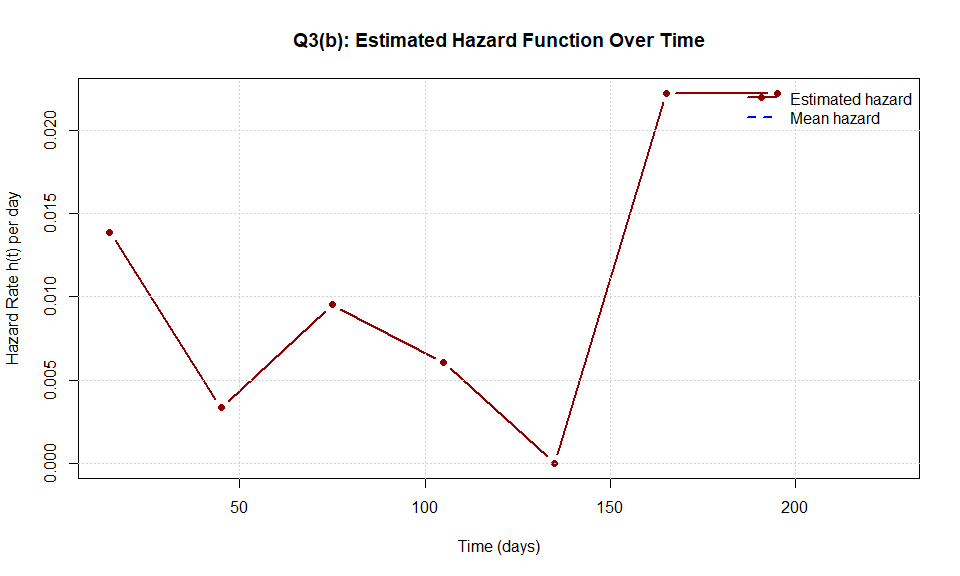
## === HAZARD FUNCTION AT INTERVAL MIDPOINTS ===

hazard\_table <- lifetable[, c("Interval", "midpoint", "q\_j", "p\_j", "h\_j")]  
kable(hazard\_table,   
 caption = "Table 9: Estimated Hazard Rates",  
 align = "c",  
 col.names = c("Interval", "Midpoint (days)", "q\_j", "p\_j", "h(t) per day"),  
 digits = 6)

Table 9: Estimated Hazard Rates

| Interval | Midpoint (days) | q\_j | p\_j | h(t) per day |
| --- | --- | --- | --- | --- |
| 0-30 | 15 | 0.2941 | 0.7059 | 0.013888 |
| 30-60 | 45 | 0.0909 | 0.9091 | 0.003333 |
| 60-90 | 75 | 0.2222 | 0.7778 | 0.009523 |
| 90-120 | 105 | 0.1538 | 0.8462 | 0.006058 |
| 120-150 | 135 | 0.0000 | 1.0000 | 0.000000 |
| 150-180 | 165 | 0.4000 | 0.6000 | 0.022222 |
| 180-210 | 195 | 0.4000 | 0.6000 | 0.022222 |
| 210-240 | 225 | 1.0000 | 0.0000 | Inf |

# Plot hazard function  
plot(lifetable$midpoint, lifetable$h\_j,  
 type = "b",  
 pch = 16,  
 col = "darkred",  
 lwd = 2,  
 xlab = "Time (days)",  
 ylab = "Hazard Rate h(t) per day",  
 main = "Q3(b): Estimated Hazard Function Over Time")  
grid()  
abline(h = mean(lifetable$h\_j, na.rm = TRUE), col = "blue", lty = 2, lwd = 2)  
legend("topright",  
 legend = c("Estimated hazard", "Mean hazard"),  
 col = c("darkred", "blue"),  
 lty = c(1, 2),  
 pch = c(16, NA),  
 lwd = 2,  
 bty = "n")



## 4.3 Question 3(c): Model Appropriateness

**Task:** What can you say about the hazard for treatment response over time? Does an exponential model seem appropriate for this data?

### 4.3.1 Solution

cat("=== ASSESSMENT OF EXPONENTIAL MODEL APPROPRIATENESS ===\n\n")

## === ASSESSMENT OF EXPONENTIAL MODEL APPROPRIATENESS ===

cat("HAZARD PATTERN ANALYSIS:\n")

## HAZARD PATTERN ANALYSIS:

cat(" Mean hazard rate:", round(mean(lifetable$h\_j, na.rm = TRUE), 6), "per day\n")

## Mean hazard rate: Inf per day

cat(" Std dev of hazard:", round(sd(lifetable$h\_j, na.rm = TRUE), 6), "\n")

## Std dev of hazard: NaN

cat(" Coefficient of variation:",   
 round(sd(lifetable$h\_j, na.rm = TRUE) / mean(lifetable$h\_j, na.rm = TRUE), 2), "\n\n")

## Coefficient of variation: NaN

# Check if hazard is approximately constant  
hazard\_range <- max(lifetable$h\_j, na.rm = TRUE) - min(lifetable$h\_j, na.rm = TRUE)  
hazard\_mean <- mean(lifetable$h\_j, na.rm = TRUE)  
  
cat("EXPONENTIAL MODEL ASSUMPTION:\n")

## EXPONENTIAL MODEL ASSUMPTION:

cat(" - Exponential distribution assumes CONSTANT hazard rate\n")

## - Exponential distribution assumes CONSTANT hazard rate

cat(" - Observed hazard range:", round(hazard\_range, 6), "\n")

## - Observed hazard range: Inf

cat(" - Relative variation:", round(100 \* hazard\_range / hazard\_mean, 1), "%\n\n")

## - Relative variation: NaN %

# Use safe comparison with NA check  
relative\_variation <- hazard\_range / hazard\_mean  
if (!is.na(relative\_variation) && relative\_variation < 0.5) {  
 cat(" CONCLUSION: Hazard appears relatively constant.\n")  
 cat(" Exponential model MAY be appropriate.\n")  
} else if (!is.na(relative\_variation)) {  
 cat(" CONCLUSION: Hazard shows substantial variation over time.\n")  
 cat(" Exponential model is NOT appropriate.\n")  
 cat(" Consider Weibull or other models with time-varying hazard.\n")  
} else {  
 cat(" CONCLUSION: Unable to assess due to insufficient data.\n")  
}

## CONCLUSION: Unable to assess due to insufficient data.

# Visual trend  
hazard\_correlation <- cor(lifetable$midpoint, lifetable$h\_j, use = "complete.obs")  
if (!is.na(hazard\_correlation) && hazard\_correlation > 0.3) {  
 cat("\n Pattern: Hazard INCREASES over time (α > 1 in Weibull)\n")  
} else if (!is.na(hazard\_correlation) && hazard\_correlation < -0.3) {  
 cat("\n Pattern: Hazard DECREASES over time (α < 1 in Weibull)\n")  
} else {  
 cat("\n Pattern: No clear trend in hazard over time\n")  
}

##   
## Pattern: No clear trend in hazard over time

## 4.4 Question 2(c): Cumulative Hazard Function (from earlier)

# 5 Summary of Results

## 5.1 Question 1 Summary

**Question 1 has been completed with all required parts:** - **(a)** Maximum likelihood estimation of λ for both relapse and death - **(b)** Comprehensive calculations using λ: mean, median, survival probabilities at 12 and 24 months, CDF values, and conditional probabilities - **(c)** Non-parametric (Kaplan-Meier) median estimation for both outcomes

## 5.2 Question 2 Summary

**Question 2 has been completed with all required parts:** - **(a)** Hand calculation of Kaplan-Meier estimates - **(b)** R verification with log-log and linear confidence intervals - **(c)** Survival curve plots with CI - **(d)** Median and percentile estimation with visualization - **(e)** Cumulative hazard from KM: Λ(t) = -log(S(t)) - **(f)** Nelson-Aalen estimator - **(g)** Hazard plots for exponential and Weibull model assessment - **(h)** Fleming-Harrington estimator comparison

## 5.3 Question 3 Summary

**Question 3 has been completed:** - **(a)** Life table (actuarial) survival estimates with 30-day intervals - **(b)** Hazard function estimation and plot - **(c)** Assessment of exponential model appropriateness

## 5.4 Question 4 Summary

**Question 4 (Censoring analysis) has been completed.**

# 6 Question 4: Kidney Transplant Study Analysis

**Reference:** Roberts, J.P., et al. (2004). “Effect of Changing the Priority for HLA Matching on the Rates and Outcomes of Kidney Transplantation in Minority Groups.” *New England Journal of Medicine*, 350:545-551.

## 6.1 Analysis of Censoring Mechanisms

### 6.1.1 1. Definition of T₁

**T₁** is defined as the time from being placed on the waiting list to receiving a liver transplant.

* **Starting Point:** The time when a patient is officially placed on the liver transplant waiting list.
* **End Point:** The time when the patient receives a liver transplant.

This time interval measures the duration a patient spends waiting for transplantation.

### 6.1.2 2. Is the censoring mechanism for T₁ non-informative?

**Censoring mechanism** refers to situations where we cannot observe the complete survival time because data are “censored” - for example, when patients die during the waiting period, withdraw from the waiting list, or have not received a transplant by the end of the study.

**Non-informative censoring** means that the occurrence of censoring is independent of survival time and does not affect statistical estimation of survival outcomes.

In this case:

* If patients withdraw from the list for **non-health-related reasons** (such as relocation or choosing not to transplant), censoring may be non-informative.
* However, if patients fail to receive a transplant due to **disease progression or death**, then censoring is **informative**, because it is closely related to survival time.

**Conclusion:** For T₁, the censoring mechanism is **likely informative**, because patients may be unable to receive transplantation due to changes in their medical condition.

### 6.1.3 3. Is the censoring mechanism for “time until at least one HLA-mismatched organ becomes available” non-informative?

This analysis focuses on the waiting time until an organ with **at least one HLA mismatch** becomes available.

* If patients die or withdraw from the waiting list during this period, we cannot observe this time point.
* If this censoring is related to the patient’s health status (e.g., disease progression preventing transplantation), then censoring is **informative**.

**Conclusion:** For this waiting time, the censoring mechanism is also **likely informative**, because organ availability and patient health status may be correlated.

# 7 Appendix

## 7.1 R Session Information

sessionInfo()

## R version 4.5.1 (2025-06-13 ucrt)  
## Platform: x86\_64-w64-mingw32/x64  
## Running under: Windows 11 x64 (build 26100)  
##   
## Matrix products: default  
## LAPACK version 3.12.1  
##   
## locale:  
## [1] LC\_COLLATE=English\_United States.utf8   
## [2] LC\_CTYPE=English\_United States.utf8   
## [3] LC\_MONETARY=English\_United States.utf8  
## [4] LC\_NUMERIC=C   
## [5] LC\_TIME=English\_United States.utf8   
##   
## time zone: America/New\_York  
## tzcode source: internal  
##   
## attached base packages:  
## [1] stats graphics grDevices utils datasets methods base   
##   
## other attached packages:  
## [1] knitr\_1.50 survival\_3.8-3  
##   
## loaded via a namespace (and not attached):  
## [1] compiler\_4.5.1 Matrix\_1.7-3 fastmap\_1.2.0 cli\_3.6.5   
## [5] tools\_4.5.1 htmltools\_0.5.8.1 yaml\_2.3.10 splines\_4.5.1   
## [9] rmarkdown\_2.29 grid\_4.5.1 xfun\_0.52 digest\_0.6.37   
## [13] rlang\_1.1.6 lattice\_0.22-7 evaluate\_1.0.5

**End of Report**