Comparing Raw Survival Data to Exponential Models

Eric Delmelle

2025-09-16

Table of contents

0.1 Load Data and Create Kaplan-Meier Curve

- note that we create a new variable ‘time_years’ where we divide the time into years.

```r
library(survival)
library(ggplot2)
library(dplyr)

# Load lung cancer data
data(lung)
lung_clean <- lung %>%
  filter(!is.na(time)) %>%
  mutate(time_years = time / 365.25)

# Create Kaplan-Meier estimate
surv_obj <- Surv(lung_clean$time_years, lung_clean$status - 1)
km_fit <- survfit(surv_obj ~ 1)

print(paste("Sample size:", length(lung_clean$time_years)))
```

```
[1] "Sample size: 228"
```

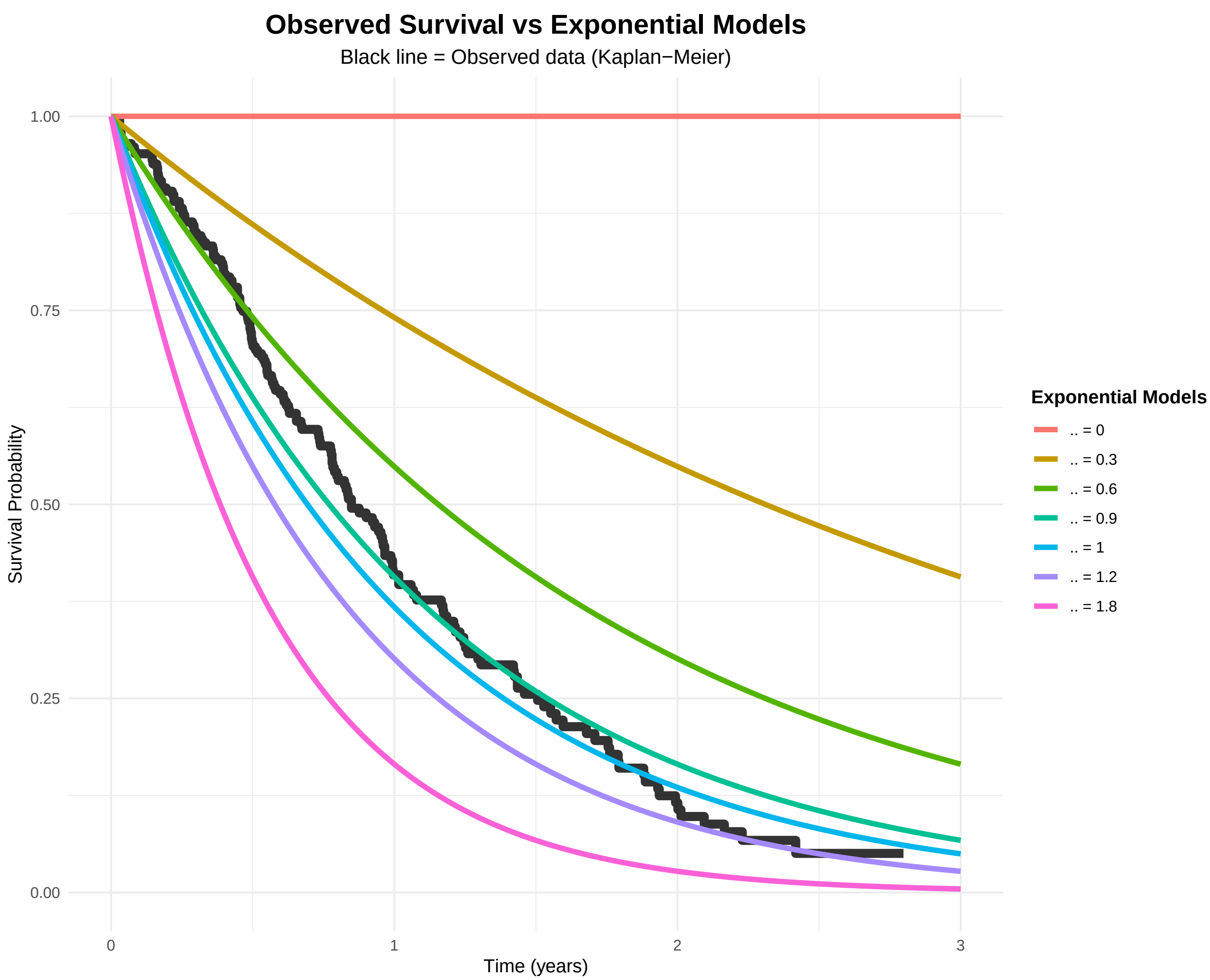
```r
print(paste("Number of deaths:", sum(lung_clean$status == 2)))
```

```
[1] "Number of deaths: 165"
```

0.2 Compare Different Exponential Models

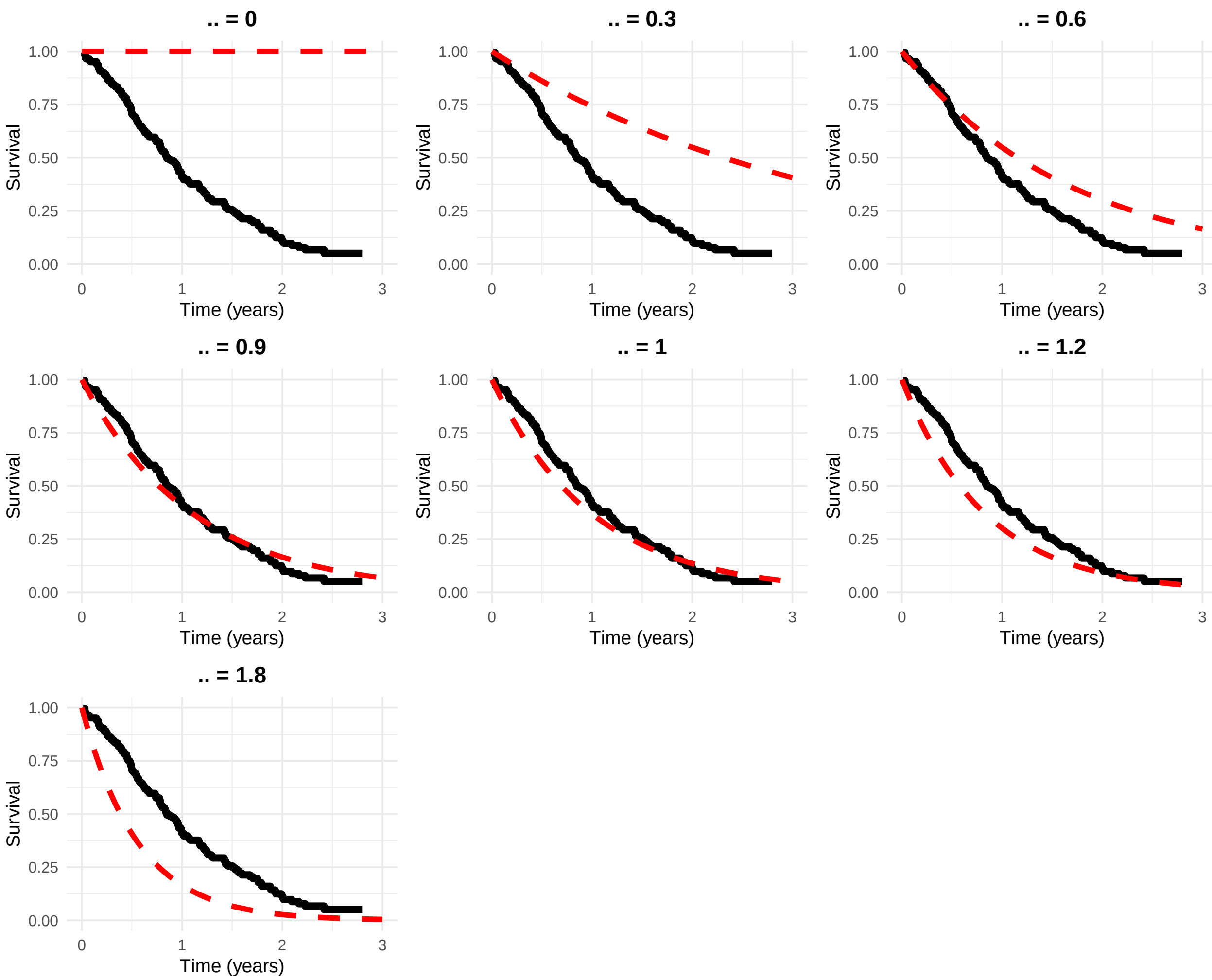
```r
# Time points for plotting exponential curves
time_grid <- seq(0, 3, length.out = 200)

# Try different lambda values
lambda_values <- c(0, 0.3, 0.6, 0.9, 1, 1.2, 1.8)

# Create plot data
plot_data <- data.frame()
for(lambda in lambda_values) {
  temp_data <- data.frame(
    time = time_grid,
    survival = exp(-lambda * time_grid),
    lambda = paste("  =", lambda)
  )
  plot_data <- rbind(plot_data, temp_data)
}

# Extract KM data
km_data <- data.frame(
  time = km_fit$time,
  survival = km_fit$surv
)

# Create the comparison plot
ggplot() +
  # Kaplan-Meier curve (observed data)
  geom_step(data = km_data,
            aes(x = time, y = survival),
            color = "black", linewidth = 2.5, alpha = 0.8) +

  # Different exponential models
  geom_line(data = plot_data,
            aes(x = time, y = survival, color = lambda),
            linewidth = 1.5) +

  xlim(0, 3) + ylim(0, 1) +
  labs(
    title = "Observed Survival vs Exponential Models",
    subtitle = "Black line = Observed data (Kaplan-Meier)",
```

```
    x = "Time (years)",
    y = "Survival Probability",
    color = "Exponential Models"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
    plot.subtitle = element_text(hjust = 0.5, size = 12),
    legend.position = "right",
    legend.title = element_text(face = "bold")
  )
```



0.3 Side-by-Side Comparison

```r
# Create individual plots for each lambda
plots <- list()

for(i in 1:length(lambda_values)) {
  lambda <- lambda_values[i]
  exp_data <- data.frame(
    time = time_grid,
    survival = exp(-lambda * time_grid)
  )

  p <- ggplot() +
    geom_step(data = km_data,
              aes(x = time, y = survival),
              color = "black", linewidth = 2) +
    geom_line(data = exp_data,
              aes(x = time, y = survival),
              color = "red", linewidth = 1.5, linetype = "dashed") +
    xlim(0, 3) + ylim(0, 1) +
    labs(title = paste("  =", lambda),
         x = "Time (years)", y = "Survival") +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5, face = "bold"))

  plots[[i]] <- p
}

# Arrange plots
library(gridExtra)
do.call(grid.arrange, c(plots, ncol = 3))
```



0.4 Summary

From these plots we can see:

- = **0.3**: Too small - curve drops too slowly, overestimates long-term survival
- = **0.6**: Decent - follows the general trend but a bit optimistic

- = **0.9**: Good fit - closely matches the observed curve
- = **1.2**: Decent - slightly pessimistic but reasonable
- = **1.8**: Too large - drops too quickly, underestimates survival

The exponential model S(t) = exp(- t) provides a simple way to model survival, but finding the right value is crucial for a good fit to the data!

0.5 Finding the Best Lambda Using Maximum Likelihood

Now let's use mathematics to find the optimal value and see how it compares to our visual assessment:

```r
# Prepare data for MLE calculation
lung_clean <- lung_clean %>%
  mutate(event = status - 1)  # Convert to 0/1 coding

times <- lung_clean$time_years
events <- lung_clean$event

# Calculate key statistics for MLE
n <- length(times)
d <- sum(events)  # number of deaths
total_time <- sum(times)  # sum of all observed times

cat("=== Data Summary for MLE ===\n")
```

```
=== Data Summary for MLE ===
```

```r
cat("Sample size (n):", n, "\n")
```

```
Sample size (n): 228
```

```r
cat("Number of deaths (d):", d, "\n")
```

```
Number of deaths (d): 165
```

```r
cat("Number censored:", n - d, "\n")
```

```
Number censored: 63
```

```r
cat("Total observed time:", round(total_time, 2), "person-years\n\n")
```

```
Total observed time: 190.54 person-years
```

```r
# Test many lambda values to find the best one
test_lambdas <- seq(0.1, 2.0, by = 0.05)  # More fine-grained search

results <- data.frame()
for(lam in test_lambdas) {
  # Log-likelihood formula: d * log( ) -   * Σt_i
  ll <- d * log(lam) - lam * total_time
  results <- rbind(results, data.frame(
```

```
    lambda = lam,
    log_likelihood = ll
  ))
}

# Find the best lambda
best_result <- results[which.max(results$log_likelihood), ]
best_lambda <- best_result$lambda
best_ll <- best_result$log_likelihood

cat("=== Search Results ===\n")
```

```
=== Search Results ===
```

```
cat("Best   from search:", best_lambda, "\n")
```

```
Best   from search: 0.85
```

```
cat("Log-likelihood at best  :", round(best_ll, 2), "\n")
```

```
Log-likelihood at best  : -188.77
```

```
# Compare with our visual guesses
visual_lambdas <- c(0.3, 0.6, 0.9, 1.2, 1.8)
cat("\n=== How Our Visual Guesses Compare ===\n")
```

```
=== How Our Visual Guesses Compare ===
```

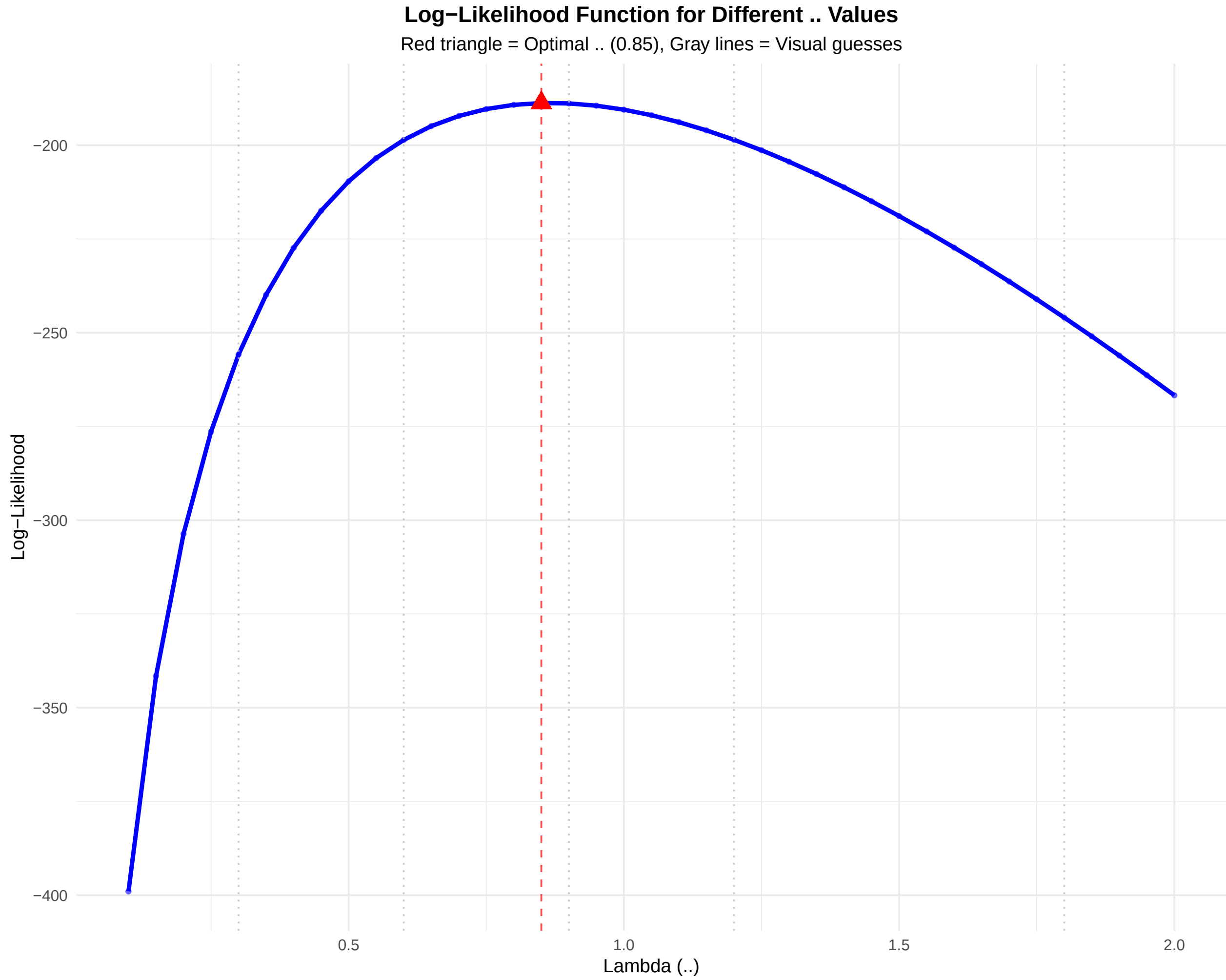
```
for(lam in visual_lambdas) {
  ll <- d * log(lam) - lam * total_time
  diff <- best_ll - ll
  cat("  =", lam, ": Log-likelihood =", round(ll, 2),
      ", Difference from best:", round(diff, 2), "\n")
}
```

```
  = 0.3 : Log-likelihood = -255.82 , Difference from best: 67.05
  = 0.6 : Log-likelihood = -198.61 , Difference from best: 9.84
  = 0.9 : Log-likelihood = -188.87 , Difference from best: 0.1
  = 1.2 : Log-likelihood = -198.56 , Difference from best: 9.79
  = 1.8 : Log-likelihood = -245.98 , Difference from best: 57.21
```

0.6 Plot: Likelihood Function

Let's visualize how the likelihood changes across different   values:

```
# Create the likelihood plot
ggplot(results, aes(x = lambda, y = log_likelihood)) +
  geom_line(color = "blue", linewidth = 1.2) +
  geom_point(color = "blue", size = 1, alpha = 0.6) +

  # Mark the optimal lambda
  geom_point(aes(x = best_lambda, y = best_ll),
             color = "red", size = 4, shape = 17) +
  geom_vline(xintercept = best_lambda, color = "red",
             linetype = "dashed", alpha = 0.7) +

  # Mark our visual lambda guesses
  geom_vline(data = data.frame(lam = visual_lambdas),
             aes(xintercept = lam),
             color = "gray", linetype = "dotted", alpha = 0.8) +

  labs(
    title = "Log-Likelihood Function for Different   Values",
    subtitle = paste("Red triangle = Optimal   (", best_lambda, "), Gray lines = Visual guess
    x = "Lambda ( )",
    y = "Log-Likelihood"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    plot.subtitle = element_text(hjust = 0.5)
  )
```




```
# Add some annotations for the visual guesses
cat("\n=== Visual Assessment vs Mathematical Optimum ===\n")
```

```
=== Visual Assessment vs Mathematical Optimum ===
```

```
cat("Our visual 'good fit' was   = 0.9\n")
```

```
Our visual 'good fit' was   = 0.9
```

```
cat("Mathematical optimum is   =", best_lambda, "\n")
```

```
Mathematical optimum is   = 0.85
```

```
cat("Difference:", round(abs(0.9 - best_lambda), 3), "\n")
```

```
Difference: 0.05
```

0.7 Analytical Solution

The exponential distribution has a simple analytical solution for the MLE:

```
# The MLE formula:  _hat = d / Σt_i
lambda_mle_analytical <- d / total_time
ll_analytical <- d * log(lambda_mle_analytical) - lambda_mle_analytical * total_time

cat("=== Analytical MLE Solution ===\n")
```

```
=== Analytical MLE Solution ===
```

```
cat(" _MLE = d / Σt_i = ", d, " / ", round(total_time, 2), " = ", round(lambda_mle_analytica
```

```
 _MLE = d / Σt_i = 165 / 190.54 = 0.866
```

```
cat("Log-likelihood:", round(ll_analytical, 2), "\n")
```

```
Log-likelihood: -188.74
```

```
cat("\n=== Comparison of Methods ===\n")
```

```
=== Comparison of Methods ===
```

```
cat("Grid search best  :", best_lambda, "\n")
```

```
Grid search best  : 0.85
```

```
cat("Analytical MLE  :", round(lambda_mle_analytical, 4), "\n")
```

```
Analytical MLE  : 0.866
```

```
cat("Difference:", round(abs(best_lambda - lambda_mle_analytical), 4), "\n")
```

```
Difference: 0.016
```

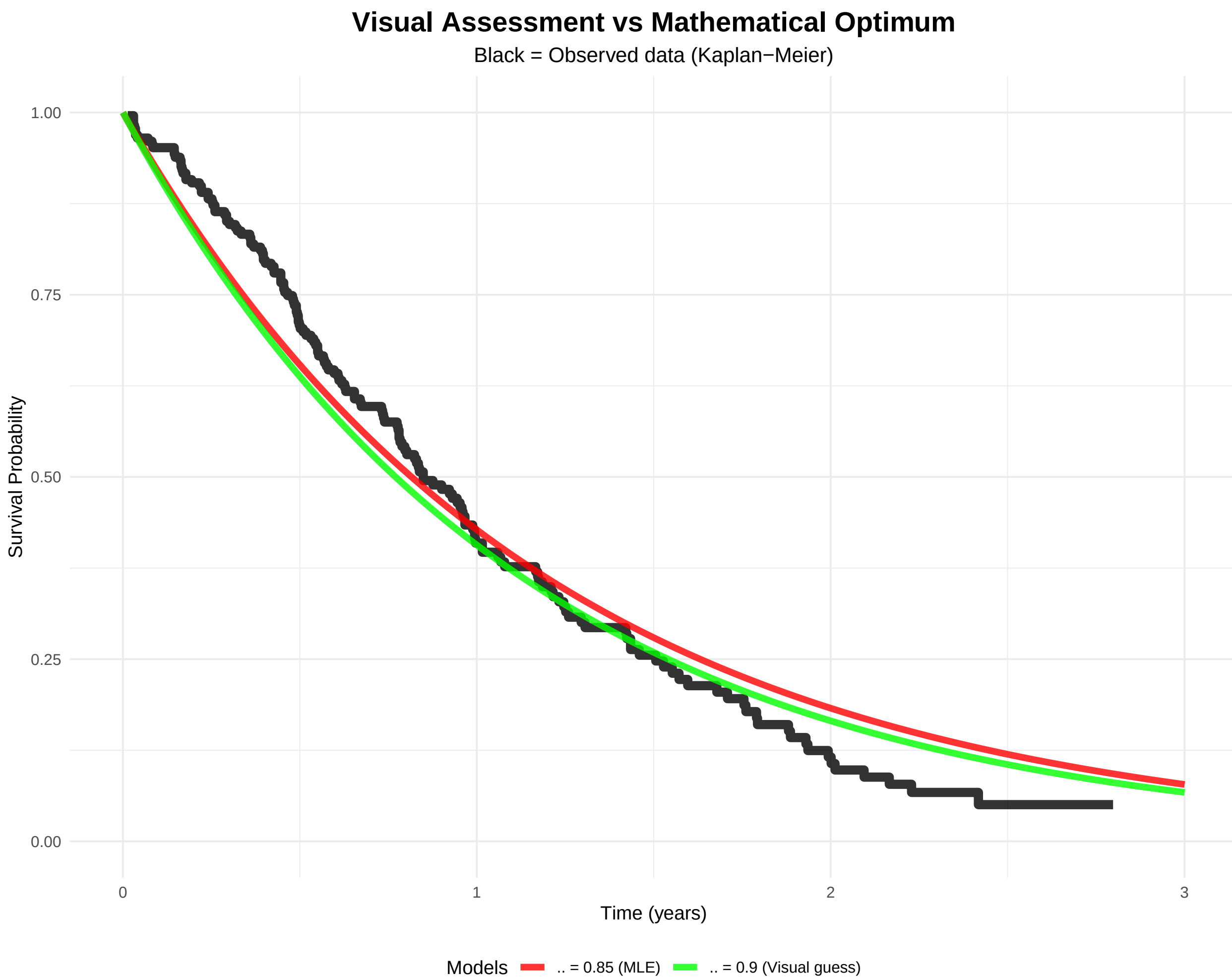
```
cat("\nThe analytical solution is exact - any tiny difference is due to our grid spacing.\n"
```

```
The analytical solution is exact - any tiny difference is due to our grid spacing.
```

0.8 Final Comparison: Visual vs Mathematical

```r
# Create a final comparison plot showing survival curves
final_lambdas <- c(0.9, best_lambda)
final_labels <- c("  = 0.9 (Visual guess)", paste("  =", best_lambda, "(MLE)"))

final_plot_data <- data.frame()
for(i in 1:length(final_lambdas)) {
  temp_data <- data.frame(
    time = time_grid,
    survival = exp(-final_lambdas[i] * time_grid),
    model = final_labels[i]
  )
  final_plot_data <- rbind(final_plot_data, temp_data)
}

# Create color mapping
mle_label <- paste("  =", best_lambda, "(MLE)")
color_mapping <- c("  = 0.9 (Visual guess)" = "green")
color_mapping[mle_label] <- "red"

ggplot() +
  # Kaplan-Meier curve
  geom_step(data = km_data,
            aes(x = time, y = survival),
            color = "black", linewidth = 2.5, alpha = 0.8) +

  # Comparison models
  geom_line(data = final_plot_data,
            aes(x = time, y = survival, color = model),
            linewidth = 1.8, alpha = 0.8) +

  scale_color_manual(values = color_mapping) +

  xlim(0, 3) + ylim(0, 1) +
  labs(
    title = "Visual Assessment vs Mathematical Optimum",
    subtitle = "Black = Observed data (Kaplan-Meier)",
    x = "Time (years)",
    y = "Survival Probability",
    color = "Models"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
    plot.subtitle = element_text(hjust = 0.5, size = 12),
```

```
    legend.position = "bottom"
  )
```




```
cat("\n=== Conclusion ===\n")
```

```
=== Conclusion ===
```

```
cat("• Visual assessment (  = 0.9) was very close to optimal!\n")
```

```
• Visual assessment (  = 0.9) was very close to optimal!
```

```
cat("• Mathematical MLE gives   =", round(lambda_mle_analytical, 3), "\n")
```

```
• Mathematical MLE gives   = 0.866
```

```
cat("• Both models fit the data quite well\n")
```

```
• Both models fit the data quite well
```

```
cat("• The likelihood plot shows a clear single peak at the MLE\n")
```

```
• The likelihood plot shows a clear single peak at the MLE
```

← Return to Course Materials

Link to qmd (quarto markdown)