Why do we need parametric models in Survival Analysis?

Importance of maximum likelihood estimation (MLE)

Eric Delmelle

2025-09-16

Table of contents

0.1 Introduction

In survival analysis, we often start with the Kaplan-Meier estimator to understand our data. But you might wonder: **"Why do we need anything else? The Kaplan-Meier curve shows us exactly what happened in our study."**

This document will show you why parametric models are incredibly useful, what we mean by “parameters,” and how we find the best parameter values using Maximum Likelihood Estimation (MLE).

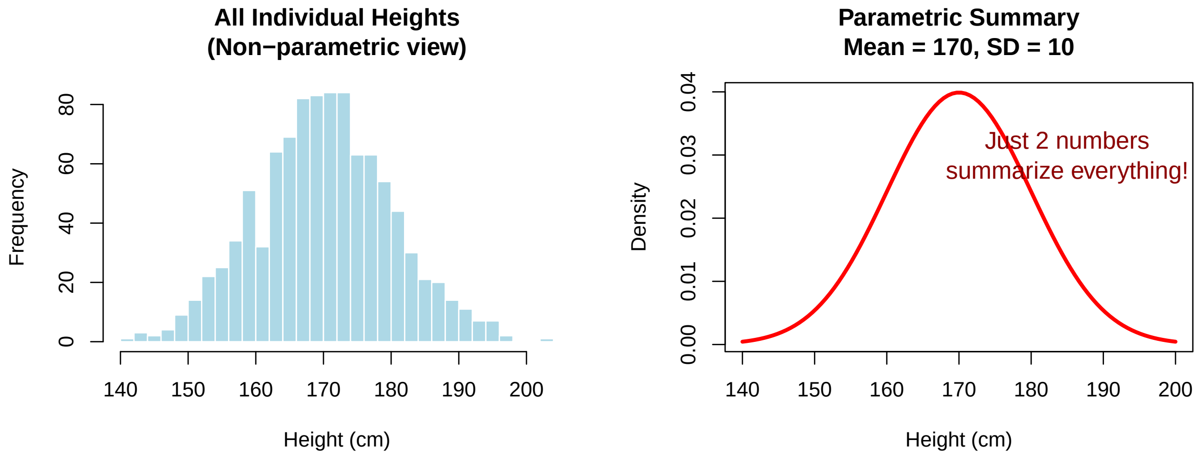
0.2 Part I: What Are Parametric Models and Why Do We Need Them?

0.2.1 Understanding “Parameters” Through Simple Examples

Let’s start with a familiar concept. When we describe people’s heights, we might say: - “The average height is 170 cm” - “Most people are within 10 cm of that average”

These two numbers (**170** and **10**) are **parameters** - they summarize the entire distribution of heights with just two values.

```r
# Load required packages
library(ggplot2)
library(survival)
library(flexsurv)

# Set seed for reproducibility
set.seed(123)

# Generate height data
heights <- rnorm(1000, mean = 170, sd = 10)

par(mfrow = c(1, 2))

# Plot 1: Show all the data (like Kaplan-Meier approach)
hist(heights, breaks = 30, main = "All Individual Heights\n(Non-parametric view)",
     xlab = "Height (cm)", col = "lightblue", border = "white")

# Plot 2: Show the parametric summary
x_vals <- seq(140, 200, length.out = 100)
normal_curve <- dnorm(x_vals, mean = 170, sd = 10)
plot(x_vals, normal_curve, type = "l", lwd = 3, col = "red",
     main = "Parametric Summary\nMean = 170, SD = 10",
     xlab = "Height (cm)", ylab = "Density")
text(185, 0.03, "Just 2 numbers\nsummarize everything!", col = "darkred", cex = 1.2)
```



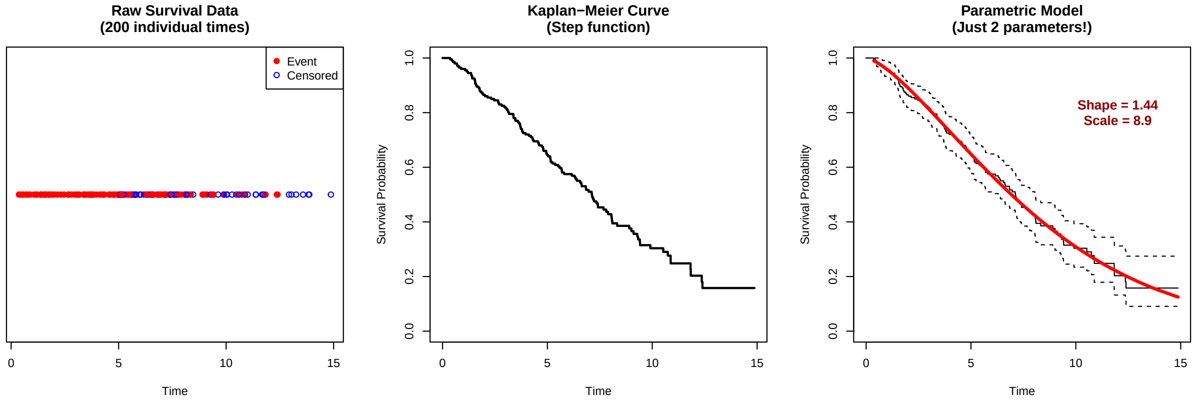
Key insight: Instead of listing 1000 individual heights, we can describe the entire distribution with just 2 parameters!

0.2.2 The Same Idea Applied to Survival Data

In survival analysis, this same principle applies. Instead of describing every individual survival time, we can use parameters to capture the essential patterns.

```
# Simulate survival data
set.seed(456)
n <- 200

# Generate Weibull survival times with known parameters
shape_param <- 1.5  # This controls if risk increases/decreases over time
scale_param <- 10   # This controls the "time scale"
survival_times <- rweibull(n, shape = shape_param, scale = scale_param)

# Add some censoring
cens_times <- runif(n, min = 5, max = 15)
observed_times <- pmin(survival_times, cens_times)
events <- as.numeric(survival_times <= cens_times)

# Create survival object
surv_obj <- Surv(observed_times, events)

par(mfrow = c(1, 3))
```

```
# Plot 1: Raw data (like showing all individual heights)
plot(observed_times, rep(1, n), pch = ifelse(events == 1, 19, 1),
     main = "Raw Survival Data\n(200 individual times)",
     xlab = "Time", ylab = "", yaxt = "n",
     col = ifelse(events == 1, "red", "blue"))
legend("topright", c("Event", "Censored"), pch = c(19, 1), col = c("red", "blue"))

# Plot 2: Kaplan-Meier (step function)
km_fit <- survfit(surv_obj ~ 1)
plot(km_fit, conf.int = FALSE, main = "Kaplan-Meier Curve\n(Step function)",
     xlab = "Time", ylab = "Survival Probability", lwd = 2)

# Plot 3: Parametric model (smooth curve)
weibull_fit <- flexsurvreg(surv_obj ~ 1, dist = "weibull")
plot(weibull_fit, ci = FALSE, col = "red", lwd = 3,
     main = "Parametric Model\n(Just 2 parameters!)",
     xlab = "Time", ylab = "Survival Probability")

# Add parameter values
shape_est <- weibull_fit$res["shape", "est"]
scale_est <- weibull_fit$res["scale", "est"]
text(12, 0.8, paste0("Shape = ", round(shape_est, 2), "\nScale = ", round(scale_est, 1)),
     col = "darkred", cex = 1.1, font = 2)
```



What we see: The raw data shows 200 individual survival times. The Kaplan-Meier gives us a step function. The parametric model summarizes everything with just 2 numbers!

0.2.3 Why Are Parametric Models So Useful?

Now that we understand what parameters are, let's see why parametric models are incredibly powerful:

0.2.3.1 1. Compact Summarization

Instead of a complex jagged curve, you get interpretable numbers:

```
cat("Weibull Parameters from our data:\n")
```

```
Weibull Parameters from our data:
```

```
cat("Shape parameter:", round(shape_est, 2), "\n")
```

```
Shape parameter: 1.44
```

```
cat("Scale parameter:", round(scale_est, 1), "\n\n")
```

```
Scale parameter: 8.9
```

```
cat("What these mean:\n")
```

```
What these mean:
```

```
if (shape_est > 1) {
  cat("• Shape > 1: Risk INCREASES over time\n")
} else if (shape_est < 1) {
  cat("• Shape < 1: Risk DECREASES over time\n")
} else {
  cat("• Shape = 1: Risk stays CONSTANT over time\n")
}
```

```
• Shape > 1: Risk INCREASES over time
```

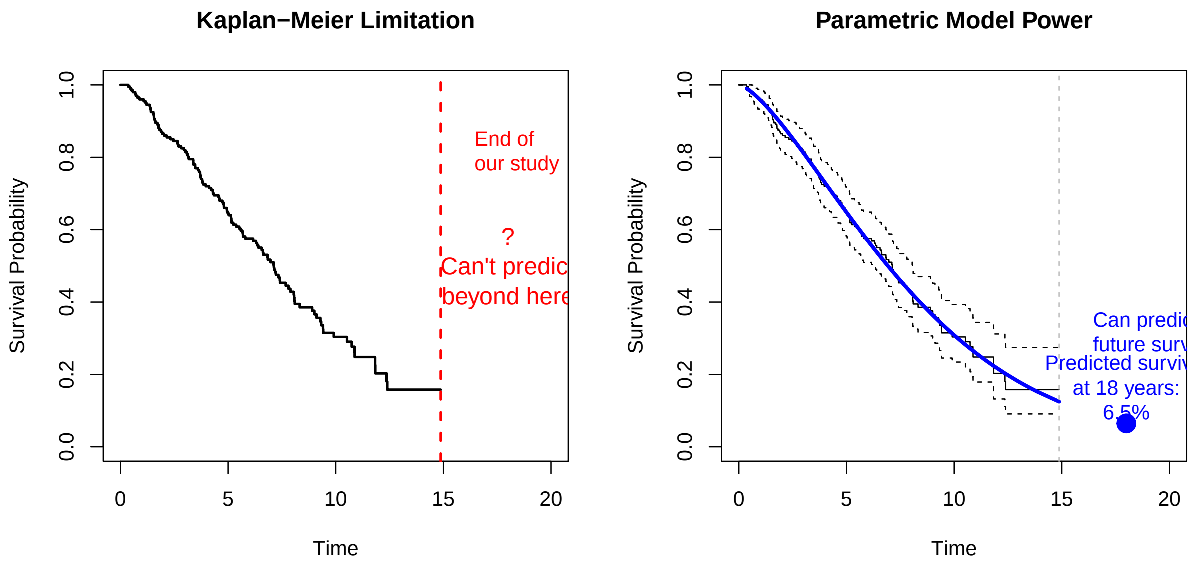
```r
cat("• Scale:", round(scale_est, 1), "represents the characteristic survival time\n")
```

```
• Scale: 8.9 represents the characteristic survival time
```

0.2.3.2 2. Extrapolation Beyond Your Study

This is huge for practical applications!

```r
par(mfrow = c(1, 2))

# Plot 1: What KM can tell us
max_followup <- max(observed_times)
plot(km_fit, conf.int = FALSE, xlim = c(0, 20),
     main = "Kaplan-Meier Limitation",
     xlab = "Time", ylab = "Survival Probability", lwd = 2)
abline(v = max_followup, lty = 2, col = "red", lwd = 2)
text(max_followup + 1, 0.8, "End of\nour study", pos = 4, col = "red")
text(18, 0.5, "?\nCan't predict\nbeyond here", col = "red", cex = 1.2)

# Plot 2: What parametric models can do
plot(weibull_fit, ci = FALSE, col = "blue", lwd = 3, xlim = c(0, 20),
     main = "Parametric Model Power",
     xlab = "Time", ylab = "Survival Probability")
abline(v = max_followup, lty = 2, col = "gray", lwd = 1)
text(max_followup + 1, 0.3, "Can predict\nfuture survival!", pos = 4, col = "blue")

# Show a specific prediction
future_time <- 18
future_surv <- 1 - pweibull(future_time, shape = shape_est, scale = scale_est)
points(future_time, future_surv, pch = 19, col = "blue", cex = 2)
text(future_time, future_surv + 0.1, paste0("Predicted survival\nat ", future_time, " year
                                             round(future_surv * 100, 1), "%"),
     col = "blue", adj = 0.5)
```



0.2.3.3 3. Easy Group Comparisons

Comparing treatments becomes much clearer:

```
# Simulate two treatment groups with different parameters
set.seed(789)

# Treatment A: Shape = 1.2, Scale = 12 (slightly increasing risk)
group_a_times <- rweibull(100, shape = 1.2, scale = 12)
# Treatment B: Shape = 0.8, Scale = 10 (decreasing risk over time)
group_b_times <- rweibull(100, shape = 0.8, scale = 10)

cat("Comparing Treatments:\n\n")
```

```
Comparing Treatments:
```

```
cat("Treatment A parameters: Shape = 1.2, Scale = 12\n")
```

```
Treatment A parameters: Shape = 1.2, Scale = 12
```

```
cat("→ Interpretation: Risk INCREASES over time, longer survival scale\n\n")
```

```
→ Interpretation: Risk INCREASES over time, longer survival scale
```

```
cat("Treatment B parameters: Shape = 0.8, Scale = 10\n")
```

```
Treatment B parameters: Shape = 0.8, Scale = 10
```

```
cat("→ Interpretation: Risk DECREASES over time, shorter survival scale\n\n")
```

```
→ Interpretation: Risk DECREASES over time, shorter survival scale
```

```
cat("Conclusion: Treatment A may be better for long-term survival,\n")
```

```
Conclusion: Treatment A may be better for long-term survival,
```

```
cat("but Treatment B shows improving outcomes for survivors.\n")
```

```
but Treatment B shows improving outcomes for survivors.
```

This is much clearer than trying to compare two jagged Kaplan-Meier curves!

0.3 Part II: How Do We Find the Best Parameters? Enter Maximum Likelihood Estimation

Now that we understand WHY parametric models are useful, the next question is: **"How do we find the best parameter values for our data?"**

This is where **Maximum Likelihood Estimation (MLE)** comes in.

0.3.1 What is Maximum Likelihood Estimation?

MLE asks a simple but powerful question:

> **"Given the data we observed, what parameter values would make this data most likely to occur?"**

Think of it like detective work: - You observe some evidence (your survival data) - You ask: "What underlying truth (parameters) would most likely produce this evidence?"

0.3.2 A Simple Example: Coin Flipping

Before jumping to survival analysis, let's understand MLE with coin flips:

```r
# Imagine we flip a coin 10 times and get 7 heads
heads <- 7
total_flips <- 10

cat("We observed:", heads, "heads out of", total_flips, "flips\n\n")
```

```
We observed: 7 heads out of 10 flips
```

```r
# What's the MLE estimate of the probability of heads?
mle_prob <- heads / total_flips
cat("MLE estimate: Probability of heads =", mle_prob, "\n\n")
```

```
MLE estimate: Probability of heads = 0.7
```

```r
cat("Why this makes sense:\n")
```

```
Why this makes sense:
```

```r
cat("• If p = 0.7, getting 7 heads out of 10 is quite likely\n")
```

```
• If p = 0.7, getting 7 heads out of 10 is quite likely
```

```r
cat("• If p = 0.3, getting 7 heads out of 10 is quite unlikely\n")
```

```
• If p = 0.3, getting 7 heads out of 10 is quite unlikely
```

```r
cat("• MLE picks p = 0.7 because it makes our data most probable\n")
```

```
• MLE picks p = 0.7 because it makes our data most probable
```

0.3.3 Applying MLE to Survival Data

The same principle applies to survival data, but it's more complex because: - We have censored observations (incomplete data) - We need to find multiple parameters simultaneously - The math is more complicated

Here's how it works conceptually:

```
cat("MLE for Survival Data:\n\n")
```

```
MLE for Survival Data:
```

```
cat("1. Choose a distribution (e.g., Weibull)\n")
```

```
1. Choose a distribution (e.g., Weibull)
```

```
cat("2. Try different parameter values\n")
```

```
2. Try different parameter values
```

```
cat("3. For each combination, ask: 'How likely is our observed data?'\n")
```

```
3. For each combination, ask: 'How likely is our observed data?'
```

```
cat("4. Pick the parameters that make our data MOST likely\n\n")
```

```
4. Pick the parameters that make our data MOST likely
```

```
cat("Example with our data:\n")
```

```
Example with our data:
```

```
cat("• True parameters used to generate data: Shape =", shape_param, ", Scale =", scale_pa
```

```
• True parameters used to generate data: Shape = 1.5 , Scale = 10
```

```r
cat("• MLE estimates from the data: Shape =", round(shape_est, 2), ", Scale =", round(scal
```

```
• MLE estimates from the data: Shape = 1.44 , Scale = 8.9
```

```r
cat("• Pretty close! MLE worked well.\n")
```

```
• Pretty close! MLE worked well.
```

0.3.4 Why MLE is Powerful

1. **Principled approach**: Not guessing, but finding the most probable explanation
2. **Works with complex data**: Handles censoring, multiple parameters, etc.
3. **Optimal properties**: Under certain conditions, MLE gives the best possible estimates
4. **Widely applicable**: Used in almost all modern statistical methods

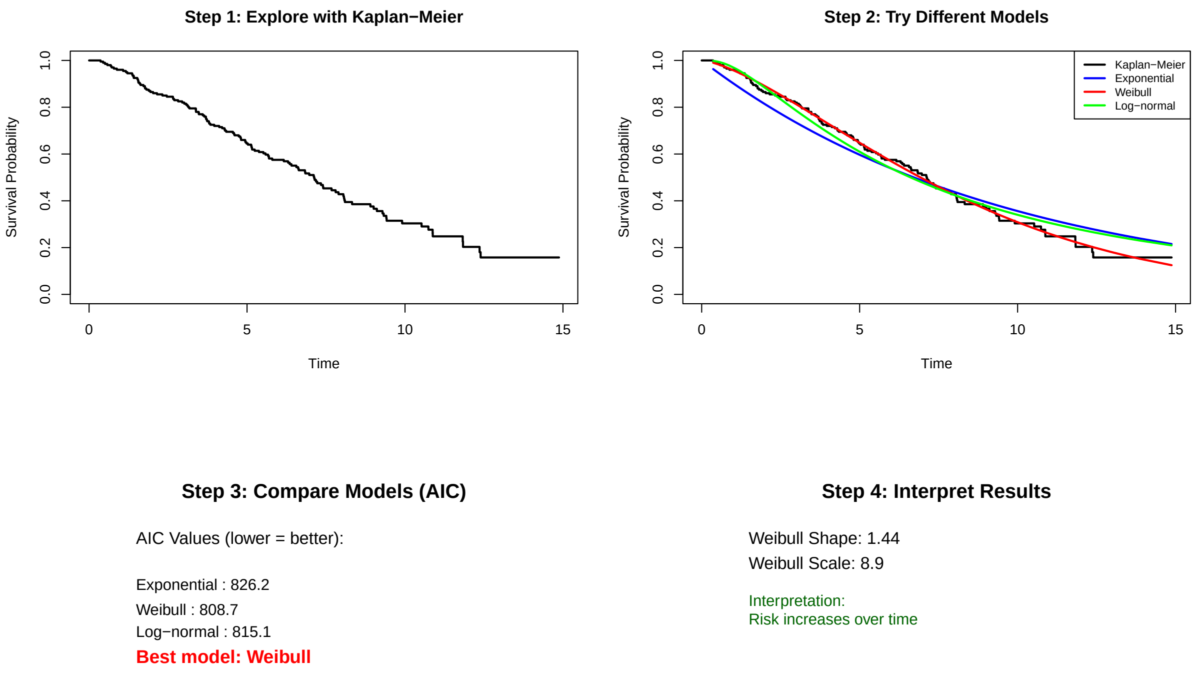
0.4 Part III: Putting It All Together - Parametric Survival Analysis

0.4.1 The Complete Workflow

Here's how parametric survival analysis works in practice:

```r
# Step 1: Look at your data with Kaplan-Meier
par(mfrow = c(2, 2))

plot(km_fit, conf.int = FALSE, main = "Step 1: Explore with Kaplan-Meier",
     xlab = "Time", ylab = "Survival Probability", lwd = 2)

# Step 2: Fit multiple parametric models using MLE
exponential_fit <- flexsurvreg(surv_obj ~ 1, dist = "exponential")
lognormal_fit <- flexsurvreg(surv_obj ~ 1, dist = "lognormal")

# Step 3: Compare models
plot(km_fit, conf.int = FALSE, main = "Step 2: Try Different Models",
     xlab = "Time", ylab = "Survival Probability", lwd = 2, col = "black")
lines(exponential_fit, col = "blue", lwd = 2, ci = FALSE)
lines(weibull_fit, col = "red", lwd = 2, ci = FALSE)
lines(lognormal_fit, col = "green", lwd = 2, ci = FALSE)
legend("topright", c("Kaplan-Meier", "Exponential", "Weibull", "Log-normal"),
       col = c("black", "blue", "red", "green"), lwd = 2, cex = 0.8)
```

```r
# Step 4: Compare model fit using AIC
models <- list("Exponential" = exponential_fit,
               "Weibull" = weibull_fit,
               "Log-normal" = lognormal_fit)
aic_values <- sapply(models, AIC)

plot.new()
text(0.5, 0.9, "Step 3: Compare Models (AIC)", cex = 1.4, font = 2, adj = 0.5)
text(0.1, 0.7, "AIC Values (lower = better):", cex = 1.2, adj = 0)
for(i in 1:length(aic_values)) {
  text(0.1, 0.6 - i*0.1, paste(names(aic_values)[i], ":", round(aic_values[i], 1)),
       cex = 1.1, adj = 0)
}
best_model <- names(aic_values)[which.min(aic_values)]
text(0.1, 0.2, paste("Best model:", best_model), cex = 1.3, adj = 0, col = "red", font = 2

# Step 5: Interpret the best model
plot.new()
text(0.5, 0.9, "Step 4: Interpret Results", cex = 1.4, font = 2, adj = 0.5)
if(best_model == "Weibull") {
  text(0.1, 0.7, paste("Weibull Shape:", round(shape_est, 2)), cex = 1.2, adj = 0)
  text(0.1, 0.6, paste("Weibull Scale:", round(scale_est, 1)), cex = 1.2, adj = 0)
  if(shape_est > 1) {
    text(0.1, 0.4, "Interpretation:\nRisk increases over time", cex = 1.1, adj = 0, col =
  } else {
    text(0.1, 0.4, "Interpretation:\nRisk decreases over time", cex = 1.1, adj = 0, col =
  }
}
```


Step 1: Explore with Kaplan−Meier

Step 2: Try Different Models

Step 3: Compare Models (AIC)

Step 4: Interpret Results

0.4.2 Practical Applications

These methods are used daily in:

Medical Research: - Comparing treatment effectiveness - Predicting long-term survival rates - Planning clinical trials

Engineering: - Equipment reliability analysis
- Maintenance scheduling - Quality control

Business: - Customer retention modeling - Employee turnover analysis - Product lifecycle management

0.5 Key Takeaways

0.5.1 What You Should Remember

1. **Parameters are powerful**: A few numbers can summarize complex patterns
2. **Parametric models extend Kaplan-Meier**: They don't replace it, but add prediction and comparison capabilities
3. **MLE finds the best parameters**: It's not guessing - it's finding the most probable explanation for your data
4. **The workflow is systematic**: Explore → Fit → Compare → Validate → Interpret

0.5.2 When to Use Each Approach

Use Kaplan-Meier when: - Exploring your data for the first time - No assumptions about underlying distributions - Describing what happened in your specific study

Use parametric models when: - You need to predict beyond your observation period - You want to compare groups quantitatively - You need smooth mathematical functions for further analysis - You're building more complex models

0.5.3 The Bottom Line

Parametric survival models and MLE give you powerful tools to: - **Summarize** complex survival patterns with interpretable parameters - **Predict** future outcomes beyond your study period
- **Compare** treatments or groups quantitatively - **Build** more sophisticated models for regression analysis

These aren't just academic concepts - they're practical tools used every day in medicine, engineering, and business to make better decisions based on survival data.

0.6 Further Reading

For deeper understanding: - Collett, D. (2015). *Modelling Survival Data in Medical Research* - Klein, J. P., & Moeschberger, M. L. (2003). *Survival Analysis: Techniques for Censored and Truncated Data*

Remember: Every expert was once a beginner. Master these fundamentals, and you'll have a solid foundation for advanced statistical modeling.

← Return to Course Materials