Class 3 Cox Model

Yubo Rasmussen

# Introduction

This notebook demonstrates how to fit and analyze a Cox proportional hazards model in R using the survival package. Additionally, it explores numerical derivatives, partial log-likelihood, and hypothesis tests such as the Wald, Likelihood Ratio Test (LRT), and Score tests. Along the way, questions are posed to deepen understanding of the methods and their applications.

## Example: Cox Model Estimate with coxph()

### Load and Prepare the Data

library(survival)  
  
# Time to event (in arbitrary units)  
Time <- c(2,5,9,11,14, 4,7,10,12,15)  
  
# Censoring indicator (1 = event occurred, 0 = censored)  
Censor <- c(1,0,1,0,1, 1,0,0,0,1)  
  
# Grouping variable (1 = Male, 2 = Female)  
Sex <- c(rep(1,5), rep(2,5))  
  
# Display the dataset  
cbind(Time, Censor, Sex)

Time Censor Sex  
 [1,] 2 1 1  
 [2,] 5 0 1  
 [3,] 9 1 1  
 [4,] 11 0 1  
 [5,] 14 1 1  
 [6,] 4 1 2  
 [7,] 7 0 2  
 [8,] 10 0 2  
 [9,] 12 0 2  
[10,] 15 1 2

# Create a survival object  
Surv(Time, Censor)

[1] 2 5+ 9 11+ 14 4 7+ 10+ 12+ 15

**Questions and Answers**:

1. **What does the Surv() function do, and why is it essential in survival analysis?** - The Surv() function creates a survival object, which encodes the survival times and censoring information. It is essential because it provides the format needed for survival analysis models, like Kaplan-Meier or Cox regression.

2. **Why do we include a censoring indicator, and how does it affect the analysis?** - The censoring indicator specifies whether an event was observed (1) or censored (0). This ensures that censored data is appropriately accounted for, preventing biased results.

### Fit the Cox Model

# Fit the Cox model  
Cox.fit <- coxph(Surv(Time, Censor) ~ Sex)  
  
# Display the model summary  
summary(Cox.fit)

Call:  
coxph(formula = Surv(Time, Censor) ~ Sex)  
  
 n= 10, number of events= 5   
  
 coef exp(coef) se(coef) z Pr(>|z|)  
Sex -1.1567 0.3145 1.1558 -1.001 0.317  
  
 exp(coef) exp(-coef) lower .95 upper .95  
Sex 0.3145 3.179 0.03265 3.03  
  
Concordance= 0.609 (se = 0.151 )  
Likelihood ratio test= 1.16 on 1 df, p=0.3  
Wald test = 1 on 1 df, p=0.3  
Score (logrank) test = 1.12 on 1 df, p=0.3

**Questions and Answers**:

1. **What does the coefficient for Sex represent in the Cox model?** - The coefficient represents the log of the hazard ratio for the comparison between the two groups (e.g., Male vs. Female).

2. **How would you interpret the hazard ratio derived from the coefficient?** - The hazard ratio indicates how much the hazard of an event changes for one group compared to another. A hazard ratio >1 suggests higher risk, while <1 indicates lower risk.

3. **What does the p-value in the output tell us about the relationship between Sex and survival time?** - The p-value tests the null hypothesis that the coefficient for Sex is zero. A small p-value (e.g., <0.05) indicates a statistically significant relationship.

## Partial Log-Likelihood and Score Function

### Define the Partial Log-Likelihood

PartialPL <- function(b){  
 p <- exp(b)  
 b - 3\*log(1 + p) - log(4 + 5\*p)  
}

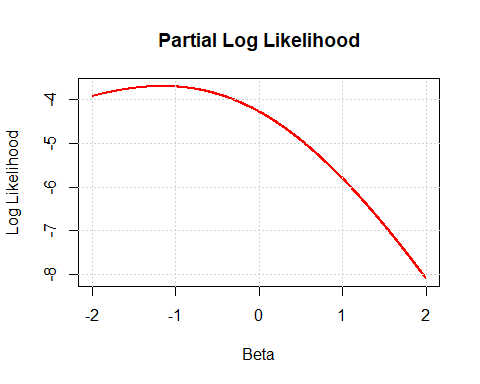
**Questions and Answers**:

1. **What does the partial log-likelihood represent in the context of the Cox model?** - It quantifies the fit of the model to the data for a given parameter value, focusing on the relative risks without considering baseline hazards.

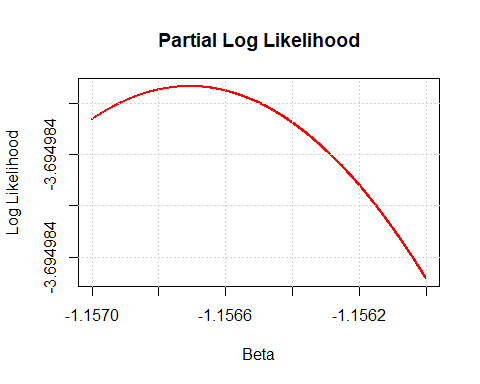
2. **How does the log-likelihood depend on the parameter b (beta)?** - The log-likelihood increases as b improves the model’s fit to the data and peaks at the MLE of b.

### Plot the Partial Log-Likelihood

PlotPL <- function(Left, Right){  
 Beta <- seq(Left, Right, length = 1000)  
 plot(Beta, PartialPL(Beta), type = "l", main = "Partial Log Likelihood",  
 xlab = "Beta", ylab = "Log Likelihood", col = "red", lwd = 2)  
 grid()  
}  
  
# Example plots  
PlotPL(-2, 2)



PlotPL(-1.157, -1.156)



# Estimated beta  
Beta.hat <- -1.16

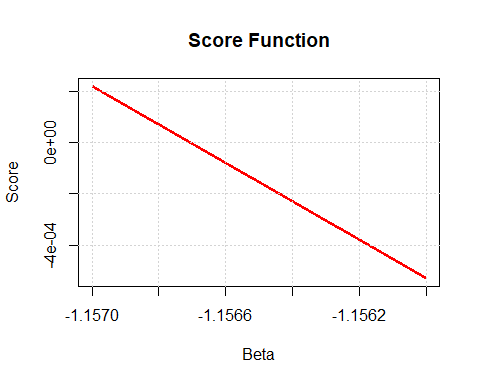
**Questions and Answers**:

1. **What does the maximum of the log-likelihood plot indicate?** - The maximum indicates the value of b that maximizes the likelihood, i.e., the MLE.

2. **How would narrowing the range of Beta values improve the accuracy of locating the maximum?** - It increases resolution around the peak, allowing for a more precise estimate of the MLE.

### Define and Plot the Score Function

Score <- function(b){  
 p <- exp(b)  
 1 - 3\*p/(1 + p) - 5\*p/(4 + 5\*p)  
}  
  
PlotScore <- function(Left, Right){  
 Beta <- seq(Left, Right, length = 1000)  
 plot(Beta, Score(Beta), type = "l", main = "Score Function",  
 xlab = "Beta", ylab = "Score", col = "red", lwd = 2)  
 grid()  
}  
  
# Plot the score function  
PlotScore(-1.157, -1.156)



**Questions and Answers**:

1. **How is the score function related to the log-likelihood function?** - The score function is the derivative of the log-likelihood with respect to b.

### Numerical Derivatives and Standard Error

# Numerical derivative at Beta.hat  
Derive <- (Score(Beta.hat + 0.01) - Score(Beta.hat - 0.01)) / 0.02  
  
# Standard error  
SE <- sqrt(-1 / Derive)  
SE

[1] 1.156717

## Numerical Derivatives and Tests

### Numerical Derivatives

PartialL <- function(x){  
 y <- exp(x)  
 x - 3\*log(1 + y) - log(4 + 5\*y)  
}  
  
Deriv <- function(x, h) (PartialL(x + h/2) - PartialL(x - h/2)) / h  
Deriv2 <- function(x, h) (PartialL(x + h) - 2\*PartialL(x) + PartialL(x - h)) / h^2  
  
# Calculate derivative and second derivative  
Info <- -Deriv2(Beta.hat, 10^(-4))  
Var <- 1 / Info  
Info

[1] 0.747389

**Questions and Answers**:

1. **How do numerical derivatives approximate the gradient and curvature of the log-likelihood function?** - By using small changes in x, they estimate the slope (gradient) and the rate of change of the slope (curvature).

2. **What are the implications of the information matrix for parameter estimation?** - The information matrix quantifies the precision of the estimates; its inverse gives the variance.

### Wald, LRT, and Score Tests

# Wald test  
W <- (Beta.hat - 0)^2 / Var  
  
# Likelihood Ratio Test (LRT)  
Lambda <- -2 \* (PartialL(0) - PartialL(Beta.hat))  
  
# Score test  
U0 <- Deriv(0, 10^(-4))  
I0 <- -Deriv2(0, 10^(-4))  
Score <- U0^2 / I0  
  
# Combine results  
Tests <- c(W, Lambda, Score)  
names(Tests) <- c("Wald", "LRT", "Score")  
Tests

Wald LRT Score   
1.005687 1.163356 1.117647

**Questions and Answers**:

1. **What is the null hypothesis tested by each of the Wald, LRT, and Score tests?** - The null hypothesis is that Beta.hat = 0, meaning no effect of the predictor.

2. **How are these three tests related, and why might they yield similar or different results?** - They all test the same hypothesis but use different approximations of the likelihood. Differences may arise due to sample size or data distribution.

3. **What does the test statistic from each test imply about the significance of Beta.hat?** - A large test statistic suggests strong evidence against the null hypothesis, indicating that Beta.hat is significantly different from zero.