MSMS 308 : Practical 04

Ananda Biswas

Exam Roll No. : 24419STC053

August 6, 2025

Question

Consider a lifetime variable that follows a **Weibull distribution** with shape parameter α and scale parameter λ. The probability density function of this distribution is given by:

$$f(t;\alpha,\lambda) = \frac{\alpha}{\lambda}\left(\frac{t}{\lambda}\right)^{\alpha-1} e^{-(t/\lambda)^{\alpha}}, \quad t > 0.$$

Estimate the parameters by MLE and get respective Bias and Mean Squared Errors. Then plot the PDF, Survival Function, Hazard Function and Cumulative Hazard Function using the obtained estimates.

Build-up for obtaining MLE

For a sample of size n, the likelihood function is given by

$$\begin{aligned} L(\alpha,\lambda) &= \prod_{i=1}^{n} \frac{\alpha}{\lambda}\left(\frac{t_i}{\lambda}\right)^{\alpha-1} e^{-(t_i/\lambda)^{\alpha}} \\ &= \left(\frac{\alpha}{\lambda}\right)^{n} \left(\prod_{i=1}^{n} \frac{t_i}{\lambda}\right)^{\alpha-1} \exp\left\{-\sum_{i=1}^{n}\left(\frac{t_i}{\lambda}\right)^{\alpha}\right\}. \end{aligned}$$

The log-likelihood function is given by

$$l(\alpha,\lambda) = n\log\alpha - n\log\lambda + (\alpha-1)\sum_{i=1}^{n}\log\left(\frac{t_i}{\lambda}\right) - \sum_{i=1}^{n}\left(\frac{t_i}{\lambda}\right)^{\alpha}.$$

Now,

$$\frac{\partial}{\partial\alpha} l(\alpha,\lambda) = \frac{n}{\alpha} + \sum_{i=1}^{n}\log\left(\frac{t_i}{\lambda}\right) - \sum_{i=1}^{n}\left(\frac{t_i}{\lambda}\right)^{\alpha}\log\left(\frac{t_i}{\lambda}\right) = u(\alpha,\lambda), \text{ say} \tag{1}$$

and

$$
\begin{aligned}
\frac{\partial}{\partial\lambda}l(\alpha,\lambda) &= -\frac{n}{\lambda}+(\alpha-1)\sum_{i=1}^{n}\frac{\lambda}{t_i}\cdot\left(-\frac{t_i}{\lambda^2}\right)+\sum_{i=1}^{n}\frac{\alpha\cdot t_i^{\alpha}}{\lambda^{\alpha+1}} \\
&= -\frac{n}{\lambda}-(\alpha-1)\frac{n}{\lambda}+\sum_{i=1}^{n}\frac{\alpha\cdot t_i^{\alpha}}{\lambda^{\alpha+1}} \\
&= -\frac{n\alpha}{\lambda}+\sum_{i=1}^{n}\frac{\alpha\cdot t_i^{\alpha}}{\lambda^{\alpha+1}} = v(\alpha,\lambda), \text{ say.}
\end{aligned}
$$

Setting $v(\alpha,\lambda)=0$ we get,

$$
\begin{aligned}
-\frac{n\alpha}{\lambda}+\sum_{i=1}^{n}\frac{\alpha\cdot t_i^{\alpha}}{\lambda^{\alpha+1}} &= 0 \\
\Rightarrow \frac{n}{\lambda} &= \sum_{i=1}^{n}\frac{t_i^{\alpha}}{\lambda^{\alpha+1}} \\
\Rightarrow \frac{n}{\lambda} &= \frac{1}{\lambda^{\alpha+1}}\sum_{i=1}^{n}t_i^{\alpha} \\
\Rightarrow \lambda^{\alpha} &= \frac{1}{n}\sum_{i=1}^{n}t_i^{\alpha} \\
\therefore \lambda &= \left(\frac{1}{n}\sum_{i=1}^{n}t_i^{\alpha}\right)^{\frac{1}{\alpha}} \qquad (2)
\end{aligned}
$$

Setting $u(\alpha,\lambda)=0$ does not yield any closed form solution. So for getting the ML estimate of α, we resort to numerical methods (here Newton-Raphson method).

Now,

$$
u_\alpha(\alpha,\lambda)=\frac{\partial}{\partial\alpha}u(\alpha,\lambda)=-\frac{n}{\alpha^2}-\sum_{i=1}^{n}\left(\frac{t_i}{\lambda}\right)^{\alpha}\left[\log\left(\frac{t_i}{\lambda}\right)\right]^2; \qquad (3)
$$

At each iteration, with the present value of α we calculate λ by using (2); then we use the obtained value of λ in (1) and (3) to improve the estimate of α by Newton-Raphson method.

R Program

```
estimate_lambda <- function(s, alpha) mean(s^alpha)^(1/alpha)
```

```
u <- function(alpha, lambda, s){

  a <- length(s) / alpha
```

```r
  b <- sum(log(s / lambda))

  c <- sum((s / lambda)^alpha * log(s / lambda))

  return(a + b - c)
}
```

```r
u_alpha <- function(alpha, lambda, s){

  a <- - length(s) / alpha^2

  b <- sum((s / lambda)^alpha * log(s / lambda)^2)

  return(a - b)
}
```

```r
estimate_alpha <- function(s, initial, epsilon = 0.0001, iterations = 100){

  alphas <- c(initial)

  for (i in 2:iterations) {
    l <- estimate_lambda(s, alphas[i-1])

    alphas[i] <- alphas[i-1] - u(alphas[i-1], l, s) / u_alpha(alphas[i-1], l, s)

    if(abs((alphas[i] - alphas[i-1])) < epsilon) break
  }

  return(alphas[length(alphas)])
}
```

```r
true_alpha <- 3; true_lambda <- 2
```

```r
sample_size <- 100
```

```r
alpha_hat = lambda_hat = c()
```

```r
x <- rweibull(sample_size, shape = true_alpha, scale = true_lambda)

alpha_hat <- estimate_alpha(x, 1)

lambda_hat <- estimate_lambda(x, alpha_hat)
```

✎ Estimates of the parameters for sample size = 100 are as follows :

```
alpha_hat; lambda_hat

## [1] 3.232285
## [1] 2.025045
```

Now we shall evaluate the bias and MSE of the estimates.

```
alpha_bias = lambda_bias = alpha_MSE = lambda_MSE = c()
```

```
alpha_estimates = lambda_estimates = c()

for (i in 1:100){

  x <- rweibull(sample_size, shape = true_alpha, scale = true_lambda)

  alpha_estimates[i] <- estimate_alpha(x, 1)

  lambda_estimates[i] <- estimate_lambda(x, alpha_estimates[i])
}

alpha_bias <- mean(alpha_estimates) - true_alpha

lambda_bias <- mean(lambda_estimates) - true_lambda

alpha_MSE <- mean( (alpha_estimates - true_alpha)^2 )

lambda_MSE <- mean( (lambda_estimates - true_lambda)^2 )
```

```
alpha_bias; lambda_bias; alpha_MSE; lambda_MSE

## [1] 0.02121329
## [1] 0.002625044
## [1] 0.06575218
## [1] 0.004191812
```

We take a mean of the different estimates to be our final estimate.

```
alpha <- mean(alpha_estimates); lambda <- mean(lambda_estimates)
```

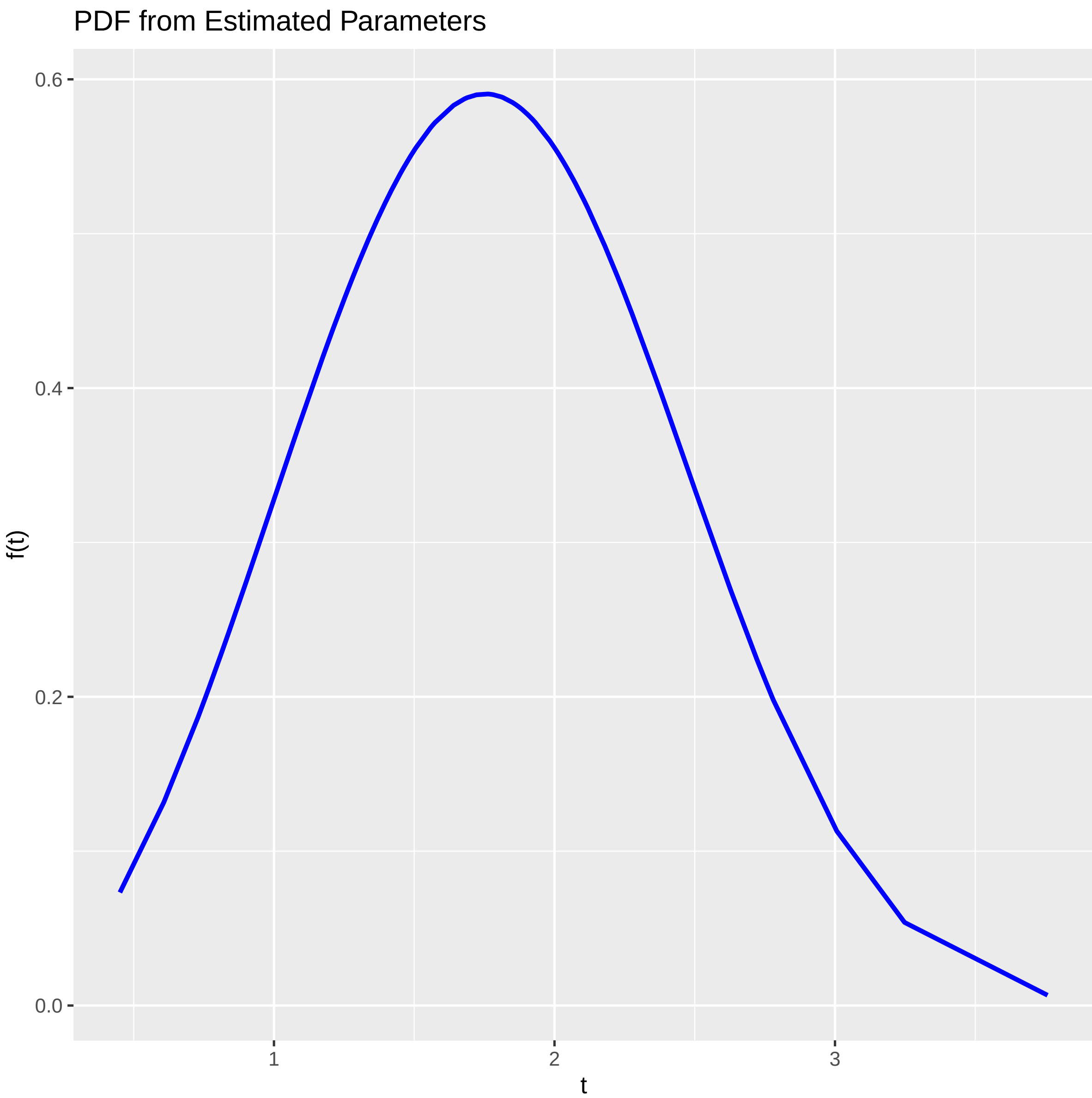
```
alpha; lambda

## [1] 3.021213
## [1] 2.002625
```

```r
t_values <- rweibull(sample_size, scale = lambda, shape = alpha)
```

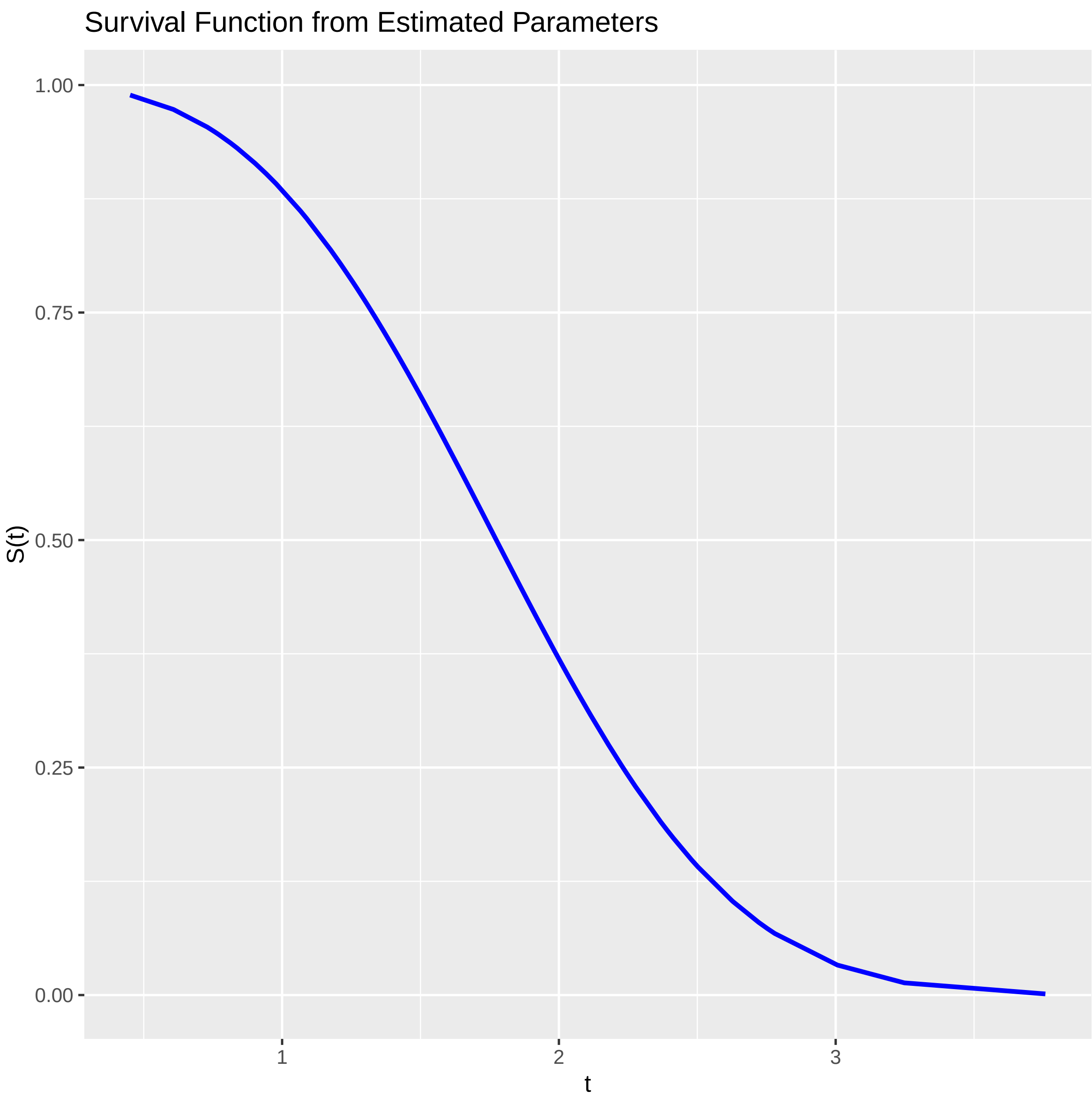
```r
df1 <- data.frame(t = t_values,
                  ft = dweibull(t_values, shape = alpha,
                                scale = lambda))

df1 %>%
  ggplot(aes(x = t, y = ft)) +
  geom_line(col = 'blue', linewidth = 1) +
  labs(x = "t", y = "f(t)",
       title = "PDF from Estimated Parameters")
```



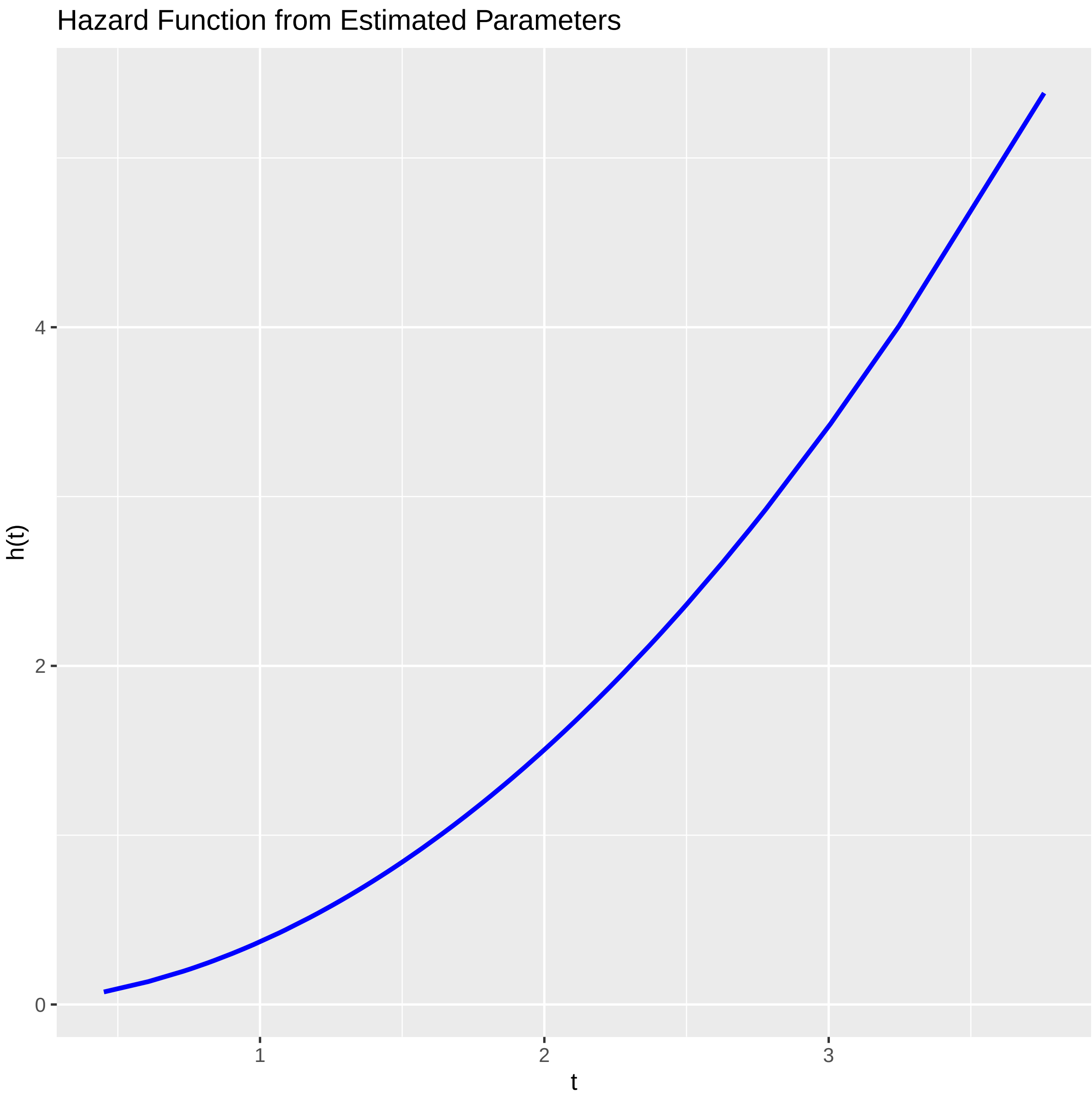
```r
df2 <- data.frame(t = t_values,
                  St = 1 - pweibull(t_values, shape = alpha,
                                    scale = lambda))

df2 %>%
  ggplot(aes(x = t, y = St)) +
  geom_line(col = 'blue', linewidth = 1) +
  labs(x = "t", y = "S(t)",
       title = "Survival Function from Estimated Parameters")
```



```
df3 <- data.frame(t = t_values,
                  ht = df1$ft / df2$St)

df3 %>%
  ggplot(aes(x = t, y = ht)) +
  geom_line(col = 'blue', linewidth = 1) +
  labs(x = "t", y = "h(t)",
       title = "Hazard Function from Estimated Parameters")
```



```
df4 <- data.frame(t = t_values,
                  Ht = -log(df2$St))

df4 %>%
  ggplot(aes(x = t, y = Ht)) +
  geom_line(col = 'blue', linewidth = 1) +
  labs(x = "t", y = "H(t)",
       title = "Cumulative Hazard Function from Estimated Parameters")
```