1.4 Estimation procedure

Introduction

This section is about obtaining

- Point estimates
- Interval estimates

In many settings, analytical solutions are not available, so we will need **numerical methods** (e.g., Newton-Raphson algorithm).

Recommended reading:

- A. J. Dobson & A. G. Barnett (2018) An introduction to generalised linear models, Chapman and Hall. Chapters 3,4,5.

Estimation procedure

Let's consider a dataset including lifetimes (times to failure in hours) of Kevlar epoxy strand pressure vessels at 70% stress level. (see Andrews and Herzberg, 1985.)

```
pressure <- c(1051, 1337, 1389, 1921, 1942, 2322, 3629, 4006, 4012,
              4063, 4921, 5445, 5620, 5817, 5905, 5956, 6068, 6121,
              6473, 7501, 7886, 8108, 8546, 8666, 8831, 9106, 9711,
              9806, 10205, 10396, 10861, 11026, 11214, 11362, 11604,
              11608, 11745, 11762, 11895, 12044, 13520, 13670, 14110,
              14496, 15395, 16179, 17092, 17568, 17568)
```

The Weibull distribution

A commonly used model for times to failure (or survival times) is the **Weibull distribution**:

$$f(y|\lambda,\theta) = \frac{\lambda y^{\lambda-1}}{\theta^{\lambda}} \exp\left[-\left(\frac{y}{\theta}\right)^{\lambda}\right] \tag{1.4.1}$$

where

- $y > 0$ is the time to failure
- λ is the shape parameter
- θ is the scale parameter

The pdf of the Weibull distribution $(1.4.1)$ can be rewritten as

$$f(y|\lambda,\theta) = \exp\left[\log\lambda + (\lambda-1)\log y - \lambda\log\theta - \left(\frac{y}{\theta}\right)^{\lambda}\right] \tag{1.4.2}$$

i Do you know what this means?

Likelihood function for the Weibull distribution

Let $\mathrm{Y}_1, \ldots, \mathrm{Y}_N$ be the data and $N = 49$. Let's assume that the pressure vessels are independent and identically distributed; then the **likelihood** is

$$f(y_1, \ldots, y_N | \lambda, \theta) = \prod_{i=1}^{N} \frac{\lambda y_i^{\lambda-1}}{\theta^\lambda} \exp\left[-\left(\frac{y_i}{\theta}\right)^\lambda\right]$$

and the **log-likelihood**

$$\ell(\theta, \lambda; y_1, \ldots, y_N) = \sum_{i=1}^{N} \left[(\lambda - 1)\log y_i + \log\lambda - \lambda\log\theta - \left(\frac{y_i}{\theta}\right)^\lambda\right] \tag{1.4.3}$$

Estimation of the Weibull parameters

Let's focus on θ and consider for the moment λ as known.

To obtain the maximum likelihood estimator we need to derive the *derivative of the log-likelihood with respect to* θ. This is also known as the **score function**:

$$\frac{d\ell}{d\theta} = \mathrm{U} = \sum_{i=1}^{N} \left[-\frac{\lambda}{\theta} + \frac{\lambda y_i^{\lambda}}{\theta^{\lambda+1}} \right] \tag{1.4.4}$$

and the **maximum likelihood estimator** is obtained by setting $\mathrm{U} = 0$

Let's suppose $\lambda = 2$. In this simple case the MLE is

$$\begin{aligned} & \textstyle\sum_{i=1}^{N} \left[-\frac{\lambda}{\theta} + \frac{\lambda y_i^{\lambda}}{\theta^{\lambda+1}} \right] && = 0 \\ \iff & -\frac{N\lambda}{\theta} + \frac{\lambda}{\theta^{\lambda+1}} \textstyle\sum_{i=1}^{N} y_i^{\lambda} && = 0 \\ \iff & \frac{-N\lambda\theta^{\lambda} + \lambda \sum_{i=1}^{N} y_i^{\lambda}}{\theta^{\lambda+1}} && = 0 \end{aligned}$$

Consequently we have $\lambda \sum_{i=1}^{N} y_i^{\lambda} = N\lambda\theta^{\lambda}$ which implies

$$\hat{\theta} = \left(\frac{\sum_{i=1}^{N} y_i^{\lambda}}{N} \right)^{1/\lambda} = 9,892.177$$

```
pressure <- c(1051, 1337, 1389, 1921, 1942, 2322, 3629, 4006, 4012,
              4063, 4921, 5445, 5620, 5817, 5905, 5956, 6068, 6121,
              6473, 7501, 7886, 8108, 8546, 8666, 8831, 9106, 9711,
              9806, 10205, 10396, 10861, 11026, 11214, 11362, 11604,
              11608, 11745, 11762, 11895, 12044, 13520, 13670, 14110,
              14496, 15395, 16179, 17092, 17568, 17568)

hist(pressure, nclass=17, xlim=c(0,20000), xlab="Time to failure (hours)")

## Maximum likelihood estimator of the Weibull scale parameter (theta)
## The shape parameter (lambda) is fixed.

theta.hat <- function(y, lambda){
  ( sum(y^lambda) / length(y) )^(1/lambda)
}

theta.hat(y=pressure, lambda=2)
```

Newton-Raphson algorithm in 1D

Suppose $f : \mathbb{R} \to \mathbb{R}$ is differentiable and attains the value 0 at x_0. The **Newton-Raphson algorithm** finds x_0 iteratively. The slope of f at a value $x^{(m-1)}$ is given by

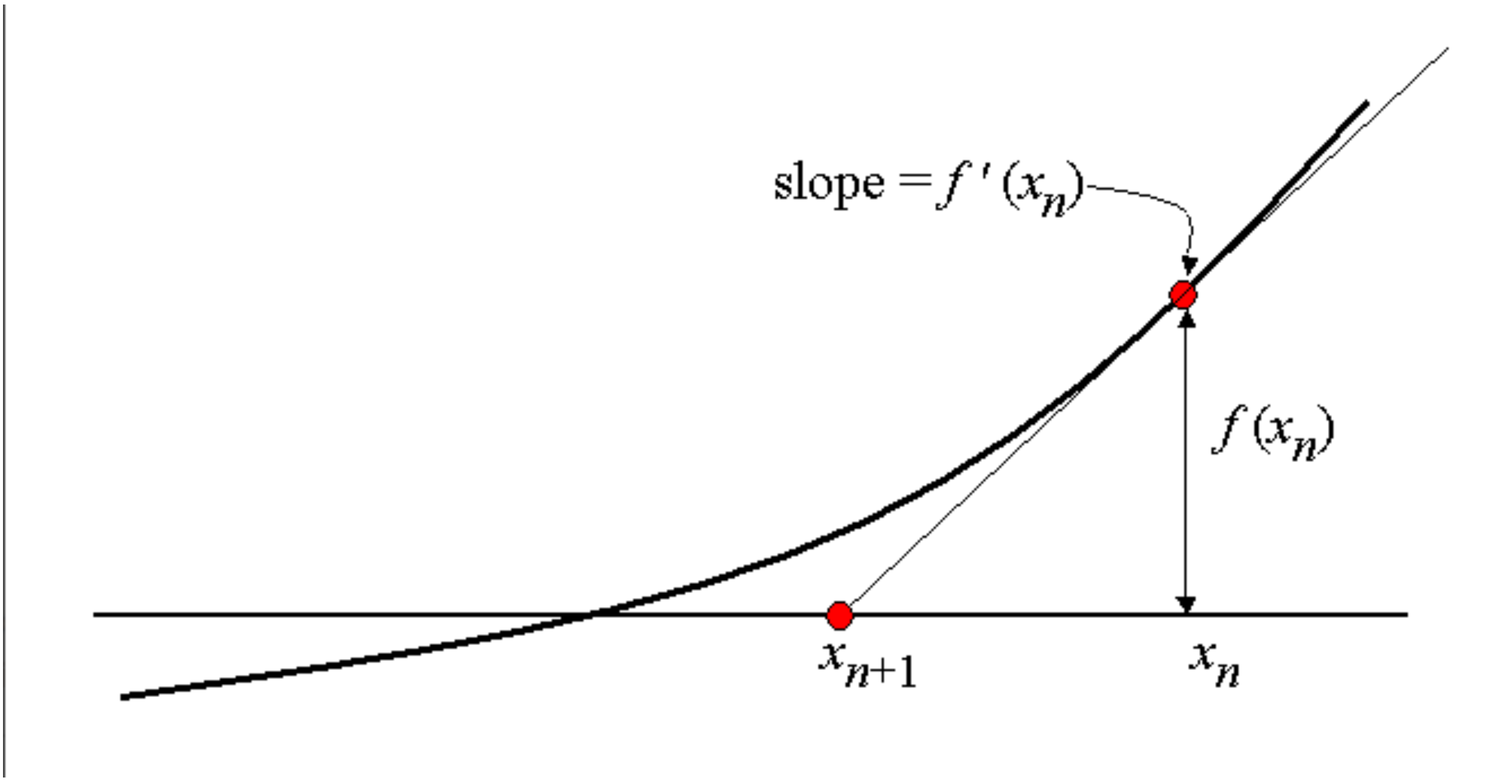
$$\left[\frac{df}{dx}\right]_{x=x^{(m-1)}} = f'(x^{(m-1)}) \approx \frac{f(x^{(m)}) - f(x^{(m-1)})}{x^{(m)} - x^{(m-1)}}$$

if the distance $x^{(m)} - x^{(m-1)}$ is small (mean value theorem). If $x^{(m)} = x_0$ is the required solution so that $f(x^m) = 0$ holds, then the above can be re-arranged to

$$x^{(m)} = x^{(m-1)} - \frac{f(x^{(m-1)})}{f'(x^{(m-1)})}$$

Iterating this algorithm creates a sequence $x^{(m)}$ which approaches x_0.

See here for a nice visualisation on Wikipedia by Ralf Pfeifer.



For the **Weibull distribution** with $\lambda = 2$ we have

$$\mathrm{U} = -\frac{2N}{\theta} + \frac{2\sum_{i=1}^{N} y_i^2}{\theta^3}$$

and the derivative

$$\frac{d\mathrm{U}}{d\theta} = \mathrm{U}' = \frac{2N}{\theta^2} - \frac{2 \times 3 \times \sum_{i=1}^{N} y_i^2}{\theta^4} \tag{1.4.5}$$

Practice: Implement the Newton-Raphson algorithm in R to estimate the scale parameter of the Weibull parameter as a function of $\boldsymbol{y}$ (data), θ_0 (starting value), λ and the number of iterations. See the next slide for RStudio and solution.

Practice using R

Implement the Newton-Raphson algorithm in $\mathbf{R}$ to estimate the scale parameter of the Weibull parameter as a function of $\boldsymbol{y}$ (data), θ_0 (starting value), λ and the number of iterations.

Newton-Raphson algorithm in pD

Assume here that $\boldsymbol{\theta}$ denotes a vector of parameters of length p.

Define $\mathrm{U}_j := \partial\ell/\partial\theta_j$ as the *score with respect to* θ_j, and write $\mathbf{u} := (\mathrm{U}_1, \ldots, \mathrm{U}_p)$. Similarly to $x^{(m)}$, we now construct a vector sequence $\boldsymbol{\theta}^{(m)}$ converging to $\hat{\boldsymbol{\theta}}$, where $\mathbf{u}(\hat{\theta}) = 0$. We write $\mathbf{u}^{(m)} := \mathbf{u}(\theta^{(m)})$. Again, by the mean value theorem, we have

$$\mathbf{u}^{(m)} - \mathbf{u}^{(m-1)} \approx H_\ell^{(m-1)} \left(\boldsymbol{\theta}^{(m)} - \boldsymbol{\theta}^{(m-1)} \right)$$

where the Hessian H_ℓ is the $p \times p$-matrix with entry $\frac{\partial^2 \ell}{\partial\theta_j \partial\theta_k}$ at position (j, k).

Aiming for $\mathbf{u}^{(m)} = 0$ as before, we arrive at a Newton-Raphson step

$$\boldsymbol{\theta}^{(m)} = \boldsymbol{\theta}^{(m-1)} - (H_\ell^{(m-1)})^{-1} \mathbf{u}^{(m-1)}. \tag{1.4.6}$$

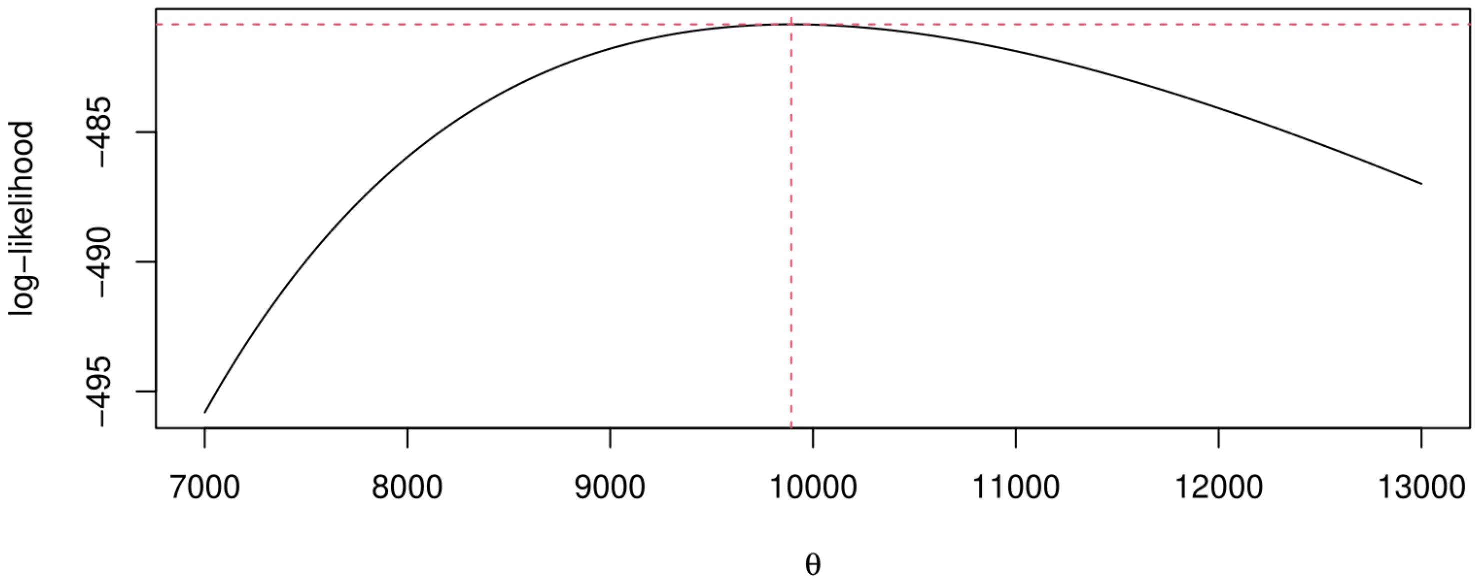
Method of scoring

It is possible to calculate $H_\ell^{(m)}$ at each iteration step $\boldsymbol{\theta}^{(m)}$ via chain rule. However, it has been shown empirically that no significant loss of accuracy is suffered when $H_\ell^{(m)}$ is replaced by minus the information matrix $-\mathcal{I} := \mathbb{E}[\mathrm{U}']$.
That is, instead of $(1.4.6)$ one iterates:

$$\boldsymbol{\theta}^{(m)} = \boldsymbol{\theta}^{(m-1)} + (\mathcal{I}^{(m-1)})^{-1}\mathbf{u}^{(m-1)}; \tag{1.4.7}$$

This is called the **method of scoring**.

Likelihood maximisation



Practice: Write some `R` code to reproduce this plot. See next slide for RStudio and solution.

As we have seen, the MLE is $\hat{\theta} = 9892.17$. The **curvature** of the function ℓ in the vicinity of the maximum gives information about how reliable the MLE is.

The curvature of ℓ is defined by the rate of change of U, i.e. U' or $\mathbb{E}[\mathrm{U}']$:

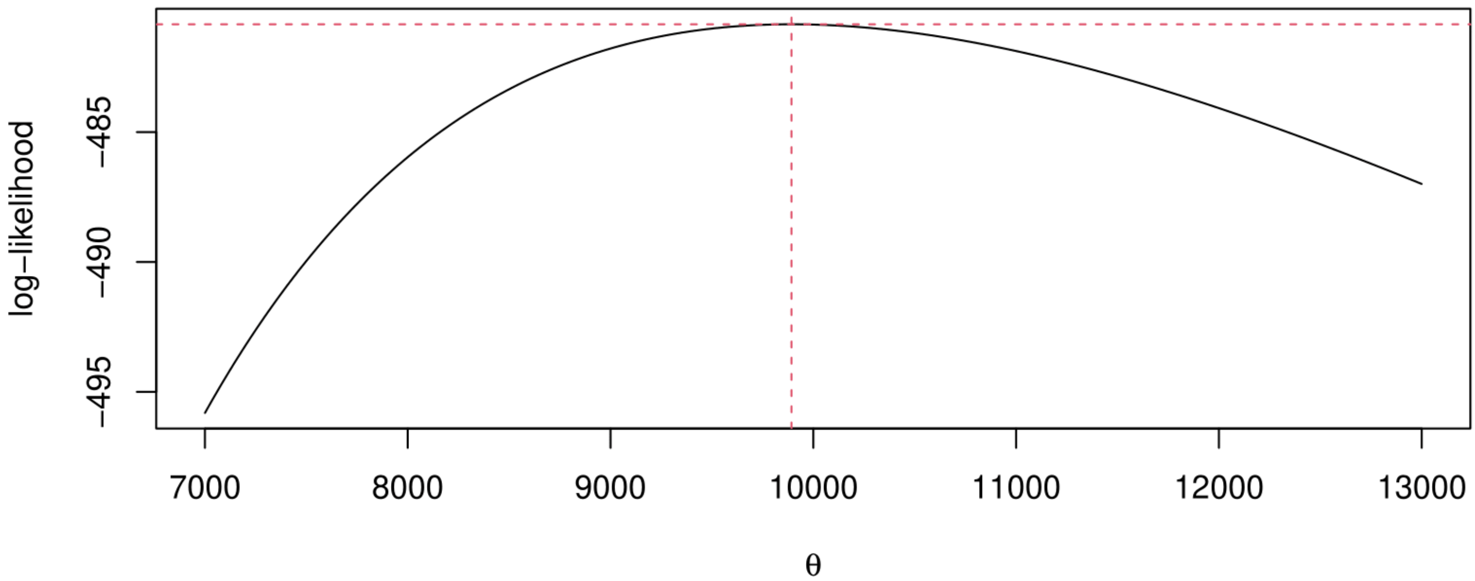
- if U' is small, then ℓ is flat and U is small for a wide interval of values for θ
- if U' is large, then ℓ is concentrated around $\hat{\theta}$

We will see that the variance of $\hat{\theta}$ is inversely related to $\mathcal{I} = \mathbb{E}[-\mathrm{U}']$, i.e.

$$s.e.(\hat{\theta}) = \sqrt{\frac{1}{\mathcal{I}}} \tag{1.4.8}$$

Practice using R

Write some `R` code to reproduce this plot:



Check your understanding

This is a non-assessed self-practice. Attempt the question and then press submit to be able to see the solution.

Question *Submitted Mar 8th 2023 at 5:16:53 pm*

[See slide: Estimation procedure]

Consider the **pressure** dataset used in the Estimation procedure slide and also available in Dobson and Barnett (2018, page 65).

```
library(dobson)
data("failure")
attach(failure)
```

Assume that we are interested in fitting a Weibull distribution to the data with fixed shape $\lambda = 2$.

1. Implement the Newton-Raphson algorithm in **R** to estimate the scale parameter of the Weibull parameter as a function of $\boldsymbol{y}$ (data), θ_0 (starting value), λ and the number of iterations. Run the algorithm with θ_0 set to be the sample mean and reproduce Table 4.2 from Dobson and Barnett (2018) without applying any rounding (the results might be slightly different for that reason).
2. Plot the log-likelihood function as a function of the scale parameter θ. Indicate the maximum likelihood estimator and the maximised value of the log-likelihood

```
library(dobson)
data("failure")
attach(failure)

NR.theta <- function(y, iter, theta0, lambda){

  sum.ylambda <- sum(y^lambda)
  N <- length(y)

  theta.hat <- U <- UPrime <- EUPrime <- vector(length=iter+1)
  theta.hat[1] <- theta0
  U[1] <- -lambda*N/theta0 + lambda * sum.ylambda / theta0^(lambda+1)
  UPrime[1] <- lambda*N/theta0^2 - lambda * (lambda+1) * sum.ylambda / theta0^(lambda+2)
  EUPrime[1] <- - lambda^2 * N / theta0^2

for(i in 2:(iter+1)){

  theta.hat[i] <- theta.hat[i-1] - U[i-1]/UPrime[i-1]
  U[i] <- -lambda*N/theta.hat[i] + lambda * sum.ylambda / theta.hat[i]^(lambda+1)
  UPrime[i] <- lambda*N/theta.hat[i]^2 - lambda * (lambda+1) * sum.ylambda / theta.hat[i]^(lamb
```

```
  EUPrime[i] <- - lambda^2 * N / theta.hat[i]^2
}
return(list(theta=round(theta.hat,1), U=round(U*1e6,2), UPrime=round(UPrime*1e6,2), EUPrime=rou
}
est <- NR.theta(failure$lifetimes, iter=3, theta0=round(mean(failure$lifetimes),1),
                lambda=2)
est
est$U / est$UPrime
est$U / est$EUPrime

## MLE
theta.hat <- function(y, lambda){
  ( sum(y^lambda) / length(y) )^(1/lambda)
}
## Log-likelohood plot
loglik <- function(y, theta, lambda){
  sum((lambda-1)*log(y) + log(lambda) - lambda * log(theta) - (y/theta)^lambda  )
}

y <- failure$lifetimes
theta.values <- seq(from=7000, to=13000, length=1000)
loglik.values <- sapply(theta.values, function(x) loglik(y=y, theta=x,
                        lambda=2) )
plot(theta.values, loglik.values, type="l", ylab="log-likelihood",
     xlab=expression(theta))
abline(v=theta.hat(y=y, lambda=2), lty=2, col=2)
abline(h=loglik( pressure, theta.hat(y=y, lambda=2), lambda=2), lty=2,
       col=2)
```