

Linear Models Statistical Analysis

Hector Mathonsi

01 April 2025

Contents

1	Introduction	2
2	Data Preparation	2
3	Data Exploration	2
4	Probability Density Function	3
4.1	Log-Likelihood Function	3
4.2	Score Function	3
4.3	Derivative of the Score Function	3
4.4	R Implementation	4
5	Newton-Raphson Algorithm	4
6	Method of Scoring	5
7	Comparison of Methods (Newton-Raphson vs. Fisher Scoring)	7
8	95% Confidence Interval for β	7

1 Introduction

This document provides a comprehensive overview of linear models and their application in statistical analysis. It covers the theoretical foundations, practical applications, and interpretation of results from linear models. We will explore estimation techniques involving Newton-Raphson and Fisher scoring methods, as well as the use of R for implementing these models.

A gauge length of 1 mm was utilized to measure the strength of single carbon fibers, and impregnated 1000-carbon fiber tows were measured in GPa, yielding the following data:

2 Data Preparation

```
# import the data
carbon_fiber_data <- c(
  2.247, 2.64, 2.908, 3.099, 3.126, 3.245, 3.328, 3.355, 3.383, 3.572, 3.581, 3.681, 3.726,
  3.727, 3.728, 3.783, 3.785, 3.786, 3.896, 3.912, 3.964, 4.05, 4.063, 4.082, 4.111, 4.118,
  4.141, 4.246, 4.251, 4.262, 4.326, 4.402, 4.457, 4.466, 4.519, 4.542, 4.555, 4.614, 4.632,
  4.634, 4.636, 4.678, 4.698, 4.738, 4.832, 4.924, 5.043, 5.099, 5.134, 5.359, 5.473, 5.571,
  5.684, 5.721, 5.998, 6.06)
knitr::kable(head(carbon_fiber_data), caption = "Carbon Fiber Data Overview")
```

Table 1: Carbon Fiber Data Overview

x
2.247
2.640
2.908
3.099
3.126
3.245

3 Data Exploration

```
summary(carbon_fiber_data)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  2.247   3.728   4.248   4.261   4.683   6.060
```

4 Probability Density Function

The formula for the probability density function is:

$$f(x; \beta) = \frac{2}{\beta} \exp\left(-\frac{x^2}{\beta}\right), 0 \leq x < \infty \quad (1)$$

where:

- x is the random variable
- β is the scale parameter

4.1 Log-Likelihood Function

The log-likelihood is given as:

$$l(\beta; x_i) = n \log \frac{2}{\beta} - \frac{1}{\beta} \sum_{i=1}^n x_i^2 \quad (2)$$

4.2 Score Function

The Score function is given as:

$$\frac{dl}{d\beta} = U = -\frac{n}{\beta} + \frac{1}{\beta^2} \sum_{i=1}^n x_i^2 \quad (3)$$

4.3 Derivative of the Score Function

The derivative of the score function is given as:

$$\frac{dl}{d\beta} = U' = \frac{n}{\beta^2} - \frac{2}{\beta^3} \sum_{i=1}^n x_i^2 \quad (4)$$

4.4 R Implementation

```
# pdf function
pdf_function <- function(x, b) {
  return (2/b * x * exp(-x^2/b))
}

# Log-Likelihood Function
log_Likelihood_function <- function(n, b, x) {
  return (n*log(2/b) - 1/b * sum(x^2))
}

# Score Function
score_function <- function(n, b, x) {
  return (-n/b + 1/b^2 * sum(x^2))
}

# Derivative of the Score Function
d_score_function <- function(n, b, x) {
  return (n/b^2 - 2/b^3 * sum(x^2))
}
```

5 Newton-Raphson Algorithm

The Newton-Raphson algorithm is an iterative method for finding successively better approximations to the roots (or zeroes) of a real-valued function. The formula is given by:

$$\beta^{(m)} = \beta^{(m-1)} - \frac{U^{(m-1)}}{U'^{(m-1)}} \quad (5)$$

```
# Newton-Raphson Algorithm
newton_raphson <- function(b, U, d_U) {
  return (b - U/d_U)
}

b <- mean(carbon_fiber_data)
U <- score_function(length(carbon_fiber_data), b, carbon_fiber_data)
d_U <- d_score_function(length(carbon_fiber_data), b, carbon_fiber_data)

newton_table <- data.frame(
  m = 0,
  b = b,
  U = U,
  dU = d_U,
  U_frac_dU = U/d_U
)

is_converge <- FALSE
iteration_num <- 1
```

```

max_iterations <- 100 # Prevent infinite loops

while (!is_converge && iteration_num <= max_iterations) {
  b <- newton_raphson(b, U, d_U)
  U <- score_function(length(carbon_fiber_data), b, carbon_fiber_data)
  d_U <- d_score_function(length(carbon_fiber_data), b, carbon_fiber_data)
  new_iteration <- c(iteration_num, b, U, d_U, U/d_U)

  if (round(b, 6) == round(newton_table$b[length(newton_table$b)], 6)) {
    is_converge <- TRUE
  }

  newton_table <- rbind(newton_table, new_iteration)
  iteration_num <- iteration_num + 1
}

knitr::kable(
  newton_table,
  caption = "Newton-Raphson Iteration Table",
  col.names = c("Iteration(m)", "b", "U", "U'", "U/U'")
)

```

Table 2: Newton-Raphson Iteration Table

Iteration(m)	b	U	U'	U/U'
0	4.260554	44.8992612	-24.1617322	-1.8582799
1	6.118834	18.9893056	-7.7025601	-2.4653239
2	8.584157	7.7747487	-2.5713814	-3.0235688
3	11.607726	2.9952948	-0.9317038	-3.2148574
4	14.822584	1.0174911	-0.3921719	-2.5945027
5	17.417086	0.2579809	-0.2142262	-1.2042458
6	18.621332	0.0312095	-0.1648498	-0.1893208
7	18.810653	0.0006219	-0.1583294	-0.0039279
8	18.814581	0.0000003	-0.1581973	-0.0000016
9	18.814582	0.0000000	-0.1581972	0.0000000
10	18.814582	0.0000000	-0.1581972	0.0000000

6 Method of Scoring

The method of scoring is an iterative method for finding the maximum likelihood estimates of parameters in statistical models. The formula is given by:

$$\beta^{(m)} = \beta^{(m-1)} + \frac{U^{(m-1)}}{\mathfrak{I}'^{(m-1)}} \quad (6)$$

where:

- $\mathfrak{I} = -E(U')$ is the Fisher information matrix

In our case, the Fisher information matrix is given by:

$$\mathbf{J} = \frac{N}{\beta^2} \quad (7)$$

```
# Fisher Information
fisher_information <- function(N, b) {
  return (N/b^2)
}

b <- mean(carbon_fiber_data)
U <- score_function(length(carbon_fiber_data), b, carbon_fiber_data)
d_U <- d_score_function(length(carbon_fiber_data), b, carbon_fiber_data)
E_dU <- -fisher_information(length(carbon_fiber_data), b)

newton_table_fisher <- data.frame(
  m = 0,
  b = b,
  U = U,
  dU = d_U,
  E_dU = E_dU,
  U_frac_dU = U/d_U,
  U_frac_fisher = U/-E_dU
)

is_converge <- FALSE
iteration_num <- 1
max_iterations <- 100 # Prevent infinite loops

while (!is_converge && iteration_num <= max_iterations) {
  b <- newton_raphson(b, U, E_dU)
  U <- score_function(length(carbon_fiber_data), b, carbon_fiber_data)
  d_U <- d_score_function(length(carbon_fiber_data), b, carbon_fiber_data)
  E_dU <- -fisher_information(length(carbon_fiber_data), b)
  new_iteration <- c(iteration_num, b, U, d_U, E_dU, U/d_U, U/-E_dU)

  if (round(b, 6) == round(newton_table_fisher$b[length(newton_table_fisher$b)], 6)) {
    is_converge <- TRUE
  }

  newton_table_fisher <- rbind(newton_table_fisher, new_iteration)
  iteration_num <- iteration_num + 1
}

knitr::kable(
  newton_table_fisher,
  caption = "Newton-Raphson Iteration Table",
  col.names = c("Iteration(m)", "b", "U", "U'", "E(U')", "U/U'", "U/I'")
)
```

Table 3: Newton-Raphson Iteration Table

Iteration(m)	b	U	U'	E(U')	U/U'	U/I
0	4.260554	44.89926	-24.1617322	-3.0850057	-1.85828	14.55403
1	18.814582	0.00000	-0.1581972	-0.1581972	0.00000	0.00000
2	18.814582	0.00000	-0.1581972	-0.1581972	0.00000	0.00000

7 Comparison of Methods (Newton-Raphson vs. Fisher Scoring)

Newton-Raphson and Fisher scoring are two iterative methods used for finding maximum likelihood estimates. The key differences between the two methods are:

- **Newton-Raphson:** Uses the second derivative (Hessian) of the log-likelihood function to update the parameter estimates. It can converge faster but may be less stable if the Hessian is not positive definite.
- **Fisher Scoring:** Uses the Fisher information matrix (expected value of the second derivative) to update the parameter estimates. It is generally more stable and robust, especially in cases where the Hessian is not well-behaved.
- **Convergence:** Both methods can converge to the maximum likelihood estimates, but the convergence speed and stability may vary depending on the specific problem and data. Fisher scoring is often preferred for its stability, while Newton-Raphson may be faster in some cases. From our case it only took 3 iterations to converge using Fisher scoring, while it took 8 to 9 iterations to converge using Newton-Raphson.

8 95% Confidence Interval for β

The 95% confidence interval for β can be calculated using the formula:

$$CL = \hat{\beta} \pm z_{\alpha/2} \times s.e(\hat{\beta}) \quad (8)$$

where:

- $s.e(\hat{\beta}) = \sqrt{\frac{1}{J}}$

```
# 95% CL
alpha <- 0.05
b <- 18.814582
critical_value <- abs(qnorm((alpha)/2))
st_error <- sqrt(1/fisher_information(length(carbon_fiber_data), b))
cat("CL = (", b-critical_value*st_error, ",", b+critical_value*st_error,")")
```

```
## CL = ( 13.88683 , 23.74233 )
```

The 95% confidence interval for β is approximately (13.88683, 23.74233). It provides a credible range of values β based on observed data. The interval does not include zero, so we reject the null hypothesis at the 5% significance level. This suggests that the carbon fibers have a significant strength, as indicated by the positive estimate of β .