

3 Maximum likelihood estimation

Immanuel Klein

```
library("tidyverse")
library("ggplot2")
library("rmutil")
```

This problem is about estimating the parameter μ of a Laplace distribution using MLE and comparing different methods. First, it requires proving that the MLE of μ is the median of the sample and discussing its uniqueness. Then, independent realizations of the Laplace distribution are generated for estimating μ with the R function `quantile` and trying out different types of sample quantiles. We also write a custom function to calculate the MLE using `optimize` and compare this estimator to the quantile-based one. Finally, we look at the distribution of the MLE through Monte Carlo simulations, comparing the results for sample sizes $n = 20$ and $n = 1000$ with the use of histograms, QQ-plots, and variance analysis.

Exercise (a)

$$L(\mu) = \prod_{i=1}^n \frac{1}{2\sigma} \exp\left(-\frac{|x_i - \mu|}{\sigma}\right)$$

Vining the log-likelihood won't change the location of the max

$$\rightarrow l(\mu) = \sum_{i=1}^n \log\left(\frac{1}{2\sigma}\right) - \frac{|x_i - \mu|}{\sigma}$$
$$l(\mu) = n \cdot \log\left(\frac{1}{2\sigma}\right) - \frac{1}{\sigma} \sum_{i=1}^n |x_i - \mu|$$
$$\frac{\partial l}{\partial \mu} = \frac{1}{\sigma} \sum_{i=1}^n \text{sgn}(x_i - \mu) = 0$$

\downarrow

$$\frac{x_i - \mu}{|x_i - \mu|}$$

This expression is zero when there are ~~two~~ ~~as~~ equally as

For both MLE and median calculation, the principle of uniqueness applies similarly:

1. Uneven Sample Size: In the case of an odd sample size, there exists a unique solution.
2. Even Sample Size: When the sample size is even, any value within the range of the two middle values is sufficient.

Exercise (b)

```
# Set parameters
n <- 20
# n <- 1000 # Uncomment to use n = 1000
mu <- 1
sigma <- 1
set.seed(123) # for reproducibility
x <- rlaplace(n, mu, sigma)
t <- 8
#t <- 7 # Uncomment to use type 7
#t <- 4 # Uncomment to use type 4

?quantile #look up the different values of type
est <- quantile(x, probs = 0.5, type = t) # type 8: approximately median
est
```

```
50%
1.083204
```

Type 7 is the most appropriate choice as it approximates the median, while, for instance, type 7 represents the mode. A Laplace distribution with a mean and standard deviation of 1 exhibits symmetry around 1, resulting in the mean and median being equal. The mode will also align with the mean if the exact mean value is present in the sample, although this depends on the specific definition.

As the sample size increases, the estimate becomes more precise, converging towards 1 if the median (type 8) is used.

Exercise (c)

```

likelihood <- function(mu, data) {
  sigma <- sd(data)
  # log-Likelihood:
  # Product over data with a given parameter of mu, then logged
  # -> Same as sum of log of function
  sum(log(exp((-1)*abs(data-mu)/sigma)/(2*sigma)))
}

mle <- function(data) {
  # find the MLE by finding the maximum of the log-Likelihood
  result <- optimize(likelihood,
    interval = range(data),
    data = data,
    maximum = TRUE)
  result$maximum
}

set.seed(123)
sample1 <- rlaplace(20, 1, 1)
sample2 <- rlaplace(1000, 1, 1)

mle1.custom <- mle(sample1)
mle2.custom <- mle(sample2)

mle1.quantile <- quantile(sample1, probs = 0.5, type = 7)
mle2.quantile <- quantile(sample2, probs = 0.5, type = 7)

# Output results
print("MLE for n = 20:")

```

[1] "MLE for n = 20:"

```
print(paste("Custom function:", mle1.custom))
```

[1] "Custom function: 1.06640764592602"

```
print(paste("Quantile function:", mle1.quantile))
```

[1] "Quantile function: 1.08320358078841"

```

print("MLE for n = 1000:")

[1] "MLE for n = 1000:"

print(paste("Custom function:", mle2.custom))

[1] "Custom function: 0.976777272373572"

print(paste("Quantile function:", mle2.quantile))

[1] "Quantile function: 0.976842430313065"

```

The R-Documentation for the function `optimize` states: “The method used is a combination of golden section search and successive parabolic interpolation, and was designed for use with continuous functions.” Within a specified range of values the function can determine the point where the function reaches its minimum or maximum.

For a Newton-Raphson algorithm, we need the derivative of the log-likelihood. In this case, we will end up with a sign function in the first derivative. This leads to the first and second derivative being undefined at points where μ equals x_i . Consequently, the Newton-Raphson can't be applied practically.

Both the custom function and the quantile method show similar results, but for small sample sizes, the differences are more pronounced. In these cases, the custom function appears to perform slightly better in finding the MLE.

Exercise (d)

```

# Function for calculating the 5000 MLEs
generate.mles <- function(n) {
  mles <- numeric(5000)
  for (i in 1:5000) {
    sample <- rlaplace(n, 1, 1)
    mles[i] <- mle(sample)
  }
  mles
}

set.seed(123) # For reproducibility

```

```
mles.n20 <- generate.mles(20) # Generate the MLEs for n = 20
mles.n1000 <- generate.mles(1000) # Generate the MLEs for n = 20

print("Variance of MLE for n = 20:")

[1] "Variance of MLE for n = 20:"

print(var(mles.n20))

[1] 0.06612889

print("Variance of MLE for n = 1000:")

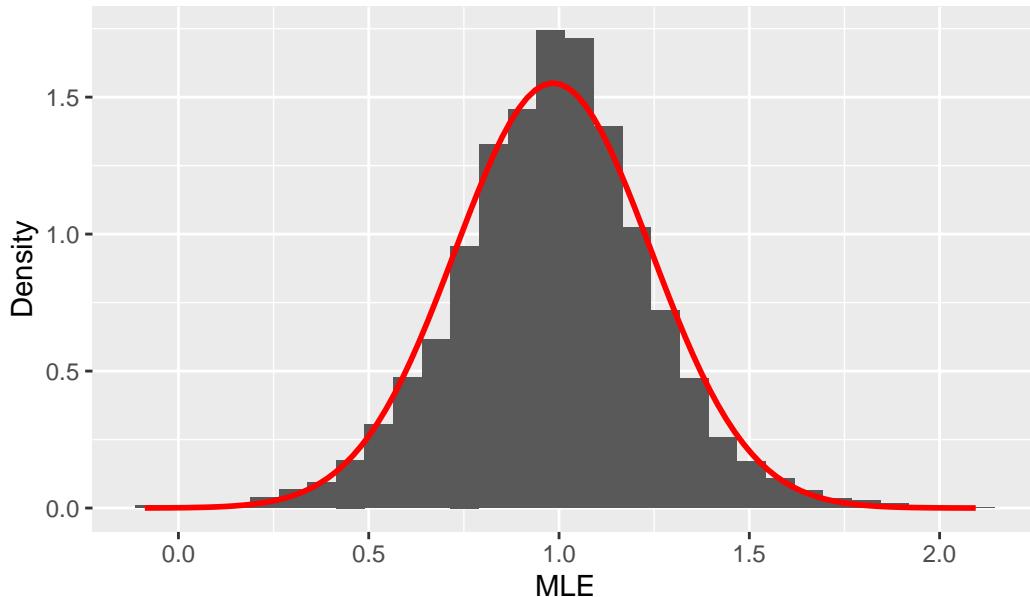
[1] "Variance of MLE for n = 1000:"

print(var(mles.n1000))

[1] 0.001062759

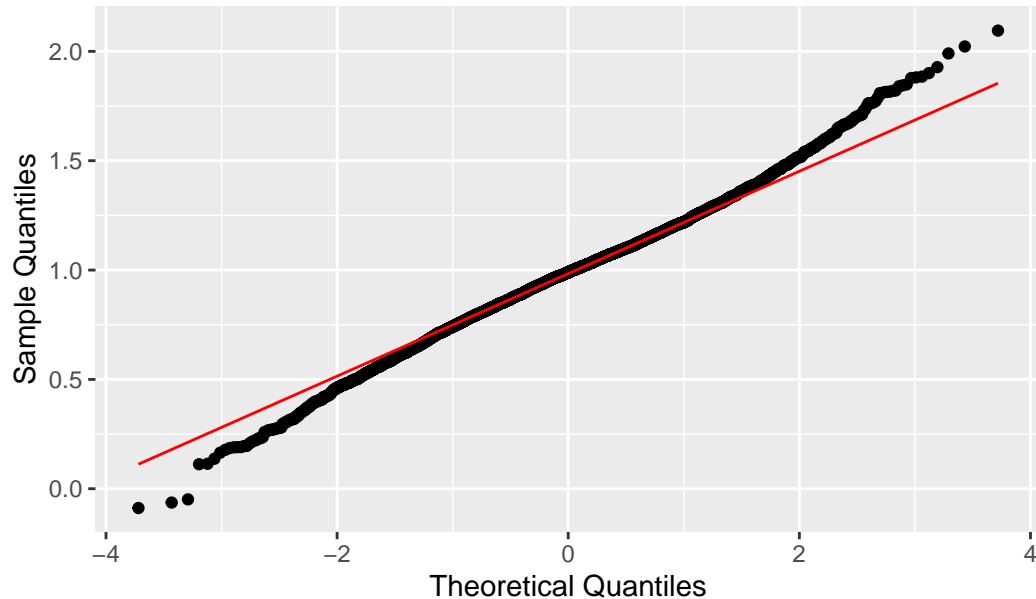
# Dataframe for the plots for n = 20
df.n20 <- data.frame(mle = mles.n20)
# Histogram
ggplot(df.n20, aes(x = mle)) +
  geom_histogram(bins = 30, aes(y = after_stat(density))) +
  stat_function(fun = dnorm,
                args = list(mean = mean(df.n20$mle),
                            sd = sd(df.n20$mle)),
                color = "red", size = 1) +
  labs(title = paste("Histogram of MLE for n =", 20),
       x = "MLE",
       y = "Density")
```

Histogram of MLE for n = 20



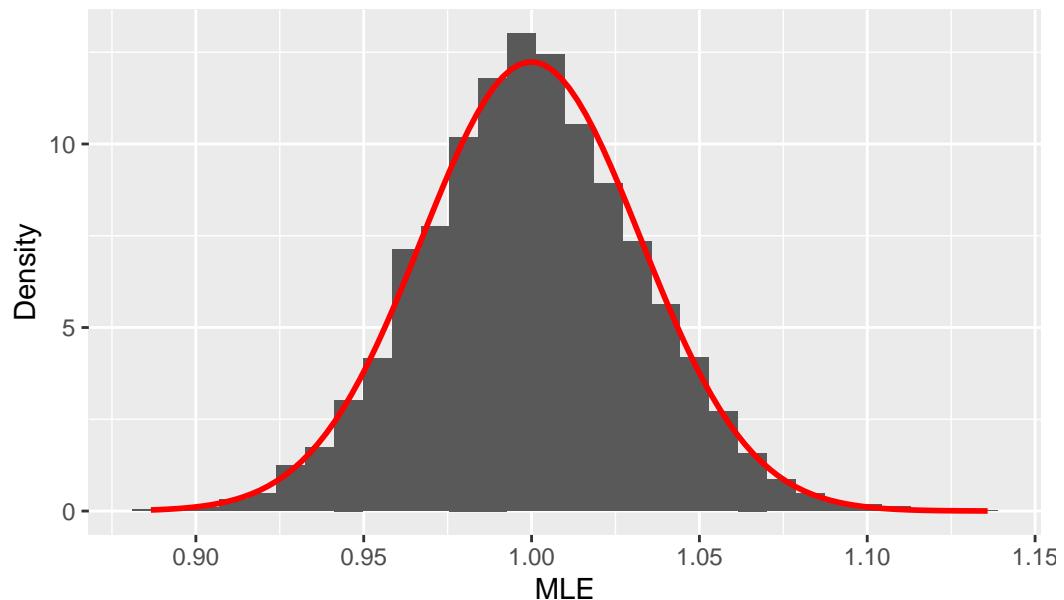
```
# QQ plot
ggplot(df.n20, aes(sample = mle)) +
  geom_qq() +
  geom_qq_line(color = "red") +
  labs(title = paste("QQ Plot of MLE for n =", 20),
       x = "Theoretical Quantiles",
       y = "Sample Quantiles")
```

QQ Plot of MLE for n = 20



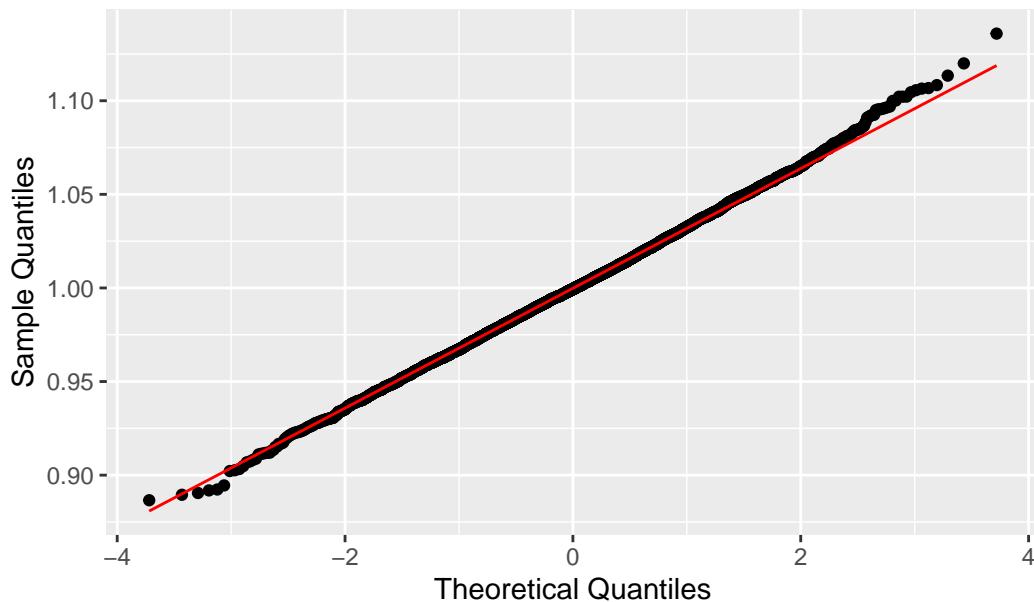
```
# Dataframe for the plots for n = 1000
df.n1000 <- data.frame(mle = mles.n1000)
# Histogram
ggplot(df.n1000, aes(x = mle)) +
  geom_histogram(bins = 30, aes(y = after_stat(density))) +
  stat_function(fun = dnorm,
                args = list(mean = mean(df.n1000$mle),
                            sd = sd(df.n1000$mle)),
                color = "red", size = 1) +
  labs(title = paste("Histogram of MLE for n =", 1000),
       x = "MLE",
       y = "Density")
```

Histogram of MLE for n = 1000



```
# QQ plot
ggplot(df.n1000, aes(sample = mle)) +
  geom_qq() +
  geom_qq_line(color = "red") +
  labs(title = paste("QQ Plot of MLE for n =", 1000),
       x = "Theoretical Quantiles",
       y = "Sample Quantiles")
```

QQ Plot of MLE for n = 1000



From the variances, we can see that for the higher sample size $n = 1000$, the variance reduces drastically, which is expected.

When comparing the two histograms with their respective normal curves, both roughly follow a normal distribution. The sample with the larger size follows the normal distribution even more closely, which is consistent with the reduced variance. In the QQ plots, it can be seen that $n = 20$ has a heavy upper tail and a light lower tail, indicating more extreme values than the normal distribution in the upper end and fewer extreme values in the lower end. For $n = 1000$, this effect is reduced, and the sample closely matches its normal distribution.