

A3

Zhengyang Fei

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.4      v readr      2.1.5
v forcats    1.0.0      v stringr    1.5.1
v ggplot2    3.5.1      v tibble     3.2.1
v lubridate  1.9.3      v tidyr      1.3.1
v purrr      1.0.2
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()     masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
library(lattice)
library(ggplot2)
```

Problem 2

Setup

```
# Given data
x <- c(1, 20, 13, 25, 4, 7, 5, 26, 36, 32)
n <- length(x)

# Estimators approximate and exact case
thetahat_apprx <- mean(x)
thetahat_exact <- 1/log((mean(x)+0.5)/(mean(x)-0.5))
```

```

# Information function
expI_approx <- function(theta, x, n) {
  -(n / theta^2) + (2 / theta^3) * sum(x)
}

# log likelihood function
expll <- function(theta, x, n) {
  n * log(exp(0.5/theta) - exp(-0.5/theta)) - (1/theta) * sum(x)
}

# relative log likelihood function for exact case
exprll_exact <- function(theta, thhat, x, n) {
  n * log((exp(0.5/theta) - exp(-0.5/theta))/(exp(0.5/thhat) - exp(-0.5/thhat))) +
    sum(x) * (1/thhat - 1/theta)
}

# relative log likelihood function for approximate case
exprll_approx <- function(theta, thhat, x, n) {
  n*log(thhat / theta) + (1/thhat - 1/theta) * sum(x)
}

```

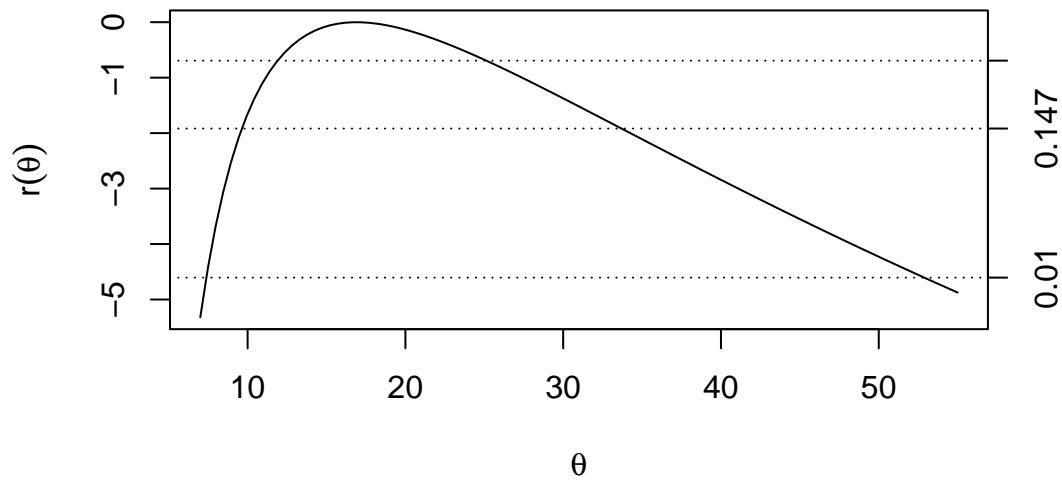
Part c

```

# Plot the relative log likelihood
th <- seq(7, 55, length.out = 100)
plot(th, exprll_exact(th, thetahat_exact, x, n), type = 'l',
      xlab = expression(theta), ylab = expression(r(theta)))
abline(h=log(c(0.5,0.147,0.01)), lty = 3)
axis(4, at = log(c(0.5,0.147,0.01)), labels = c(0.5,0.147,0.01))
title(main = "Exponential Relative Log Likelihood for exact case")

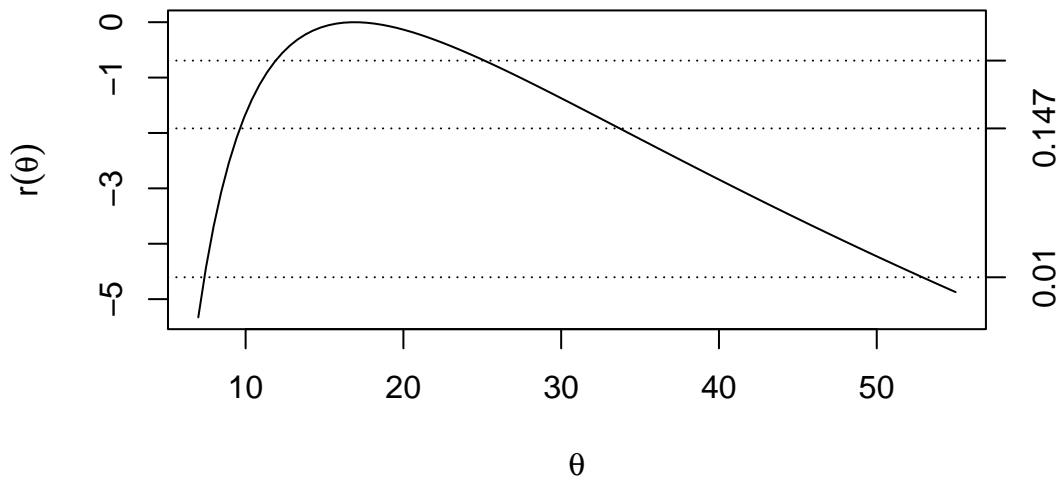
```

Exponential Relative Log Likelihood for exact case



```
# Plot the relative log likelihood
th <- seq(7, 55, length.out = 100)
plot(th, exprll_approx(th, thetahat_apprx, x, n), type = 'l', xlab = expression(theta), ylab = r(theta))
abline(h=log(c(0.5,0.147,0.01)), lty = 3)
axis(4, at = log(c(0.5,0.147,0.01)), labels = c(0.5,0.147,0.01))
title(main = "Exponential Relative Log Likelihood for approximate case")
```

Exponential Relative Log Likelihood for approximate cas



```
# function for likelihood intervals
expLI_exact <- function(theta, x, n, p) {
  -n*(sum(x)/(n * theta) - 1 - log((exp(0.5/theta) - exp(-0.5/theta))/(exp(0.5*n/sum(x))
  - exp(-0.5*n/sum(x))))) - log(p)
}

expLI_approx <- function(theta, x, n, p) {
  -n * (sum(x) / (n * theta) - 1 - log(sum(x) / (n * theta))) - log(p)
}

# 95% CI -- 14.7% LI
(cilik.exact <- c(uniroot(expLI_exact, c(0.1,thetahat_exact),
                        x = x, n = n, p = 0.147)$root,
                uniroot(expLI_exact, c(thetahat_exact,100), x = x, n = n, p = 0.147)$root))
```

```
[1] 9.63598 33.69697
```

```
(cilik.approx <- c(uniroot(expLI_approx, c(0.1,thetahat_aprx),
                        x = x, n = n, p = 0.147)$root,
                uniroot(expLI_approx, c(thetahat_aprx,100), x = x, n = n, p = 0.147)$root))
```

```
[1] 9.639853 33.704352
```

Part e

```
# LRT for testing mean
cat("Approximate case: \n")
```

Approximate case:

```
# H: theta = 14
# LRT for approximate case
cat(sprintf("D: %f\n", -2 * exprll_approx(14,thetahat_aprx,x,n)))
```

D: 0.377731

```
cat(sprintf("p-value: %f\n", pchisq(-2 * exprll_approx(14,thetahat_aprx,x,n), df = 1, lower.tail = FALSE)))
```

p-value: 0.538820

```
cat("Exact case: \n")
```

Exact case:

```
# H: theta = 14
# LRT for exact case
cat(sprintf("D: %f\n", -2 * exprll_exact(14,thetahat_exact,x,n)))
```

D: 0.376398

```
cat(sprintf("p-value: %f\n", pchisq(-2 * exprll_exact(14,thetahat_exact,x,n), df = 1, lower.tail = FALSE)))
```

p-value: 0.539537

Part f

```

# Check for 95% coverage probability
# Set hyperparameters for the simulation
set.seed(123) # Setting the random seed for reproducibility

nobs <- 10 # Number of observations
m <- 1000 # Number of simulation runs
theta <- 14 # True value of theta

# Initialize matrix to store results for the three coverage probabilities
cps <- matrix(NA, m, 3)

# function for likelihood intervals
expLI_exact <- function(theta, x, n, p) {
  -n*(sum(x)/(n * theta) - 1 - log((exp(0.5/theta)
  - exp(-0.5/theta)) / (exp(0.5*n/sum(x)) - exp(-0.5*n/sum(x))))) - log(p)
}

expLI_approx <- function(theta, x, n, p) {
  -n*(sum(x)/(n * theta) - 1 - log(sum(x)/(n * theta))) - log(p)
}

# Information function
expI_approx <- function(theta, x, n) {
  -(n / theta^2) + (2 / theta^3) * sum(x)
}

# Simulation loop over m iterations
for (j in 1:m) {
  # Simulate data
  x <- rexp(nobs, rate=1.0/theta)

  thhat_approx <- mean(x)
  thetahat_exact <- 1/log((mean(x)+0.5)/(mean(x)-0.5))

  # 95% CI (using r()) for exact approach
  if (uniroot(expLI_exact, c(0.01,thetahat_exact), x = x, n = nobs,
    p = 0.147)$root < theta &
    uniroot(expLI_exact, c(thetahat_exact,100), x = x, n = nobs,
    p = 0.147)$root > theta)
  {
    cps[j,1] <- 1
  }
}

```

```

else{cps[j,1] <- 0}
# 95% CI (using r()) for the approximate approach
if (uniroot(expLI_approx, c(0.01,thhat_approx), x = x, n = nobs,
      p = 0.147)$root < theta &
      uniroot(expLI_approx, c(thhat_approx,100), x = x, n = nobs,
      p = 0.147)$root > theta)
{
  cps[j,2] <- 1
}
else{cps[j,2] <- 0}
# 95% CI using the normal approximation
if ((thhat_approx + qnorm(0.975) / sqrt(expI_approx(thhat_approx,x,nobs))) > theta
    & (thhat_approx + qnorm(0.025) / sqrt(expI_approx(thhat_approx,x,nobs))) < theta)
{
  cps[j,3] <- 1
}
else{cps[j,3] <- 0}

if (j %% 100 == 0)
  print(j)
}

```

```

# Calculate coverage probabilities
results <- cbind(
  m,
  nobs,
  colMeans(cps[,1, drop = FALSE]),
  colMeans(cps[,2, drop = FALSE]),
  colMeans(cps[,3, drop = FALSE])
)

colnames(results) <- c('#runs', 'nobs', 'LI for exact', 'LI for approximate', 'Normal CI')

results <- round(results, 4)
results

```

```

      #runs nobs LI for exact LI for approximate Normal CI
[1,] 1000   10      0.955      0.955      0.902

```

Problem 3

```
# Score function for approximate case
expS <- function(theta, x, n) {
  -(n/theta) + (1/theta^2) * sum(x)
}

# Information function
expI <- function(theta, x, n) {
  (n / theta^2) - (2 / theta^3) * sum(x)
}

# The data
x <- c(1, 20, 13, 25, 4, 7, 5, 26, 36, 32)
n <- length(x) # Number of observations
max_iter <- 100 # To avoid infinite loop in case of non-convergence
iter <- 0 # Counter
set.seed(1030)

th0=runif(1,1,24)
cat("initial theta:", th0)
```

initial theta: 20.97108

```
eps = 1e-7 # threshold
for (i in 1:max_iter){
  th.i <- th0 - (expS(th0, x, n) / expI(th0, x, n))
  iter <- iter + 1
  if ((abs(th.i - th0) < eps || iter >= max_iter)) {
    th0 = th.i
    break
  }
  th0 <- th.i
}
cat("\n#iteration:", iter, "\ntheta_hat:", th0)
```

#iteration: 6
theta_hat: 16.9


```

# Score function for exact case
expS.exact <- function(theta, x, n) {
  -n/theta^2 * (exp(-1/theta)/(1-exp(-1/theta))) + sum(x-0.5)/theta^2
}

# Information function for exact case
expI.exact <- function(theta, x, n) {
  t1 <- -2*n*exp(-1/theta)/(theta^3*(1-exp(-1/theta)))
  t2 <- +n/theta^4 * (exp(1/theta)/(exp(1/theta)-1)^2)
  t3 <- + 2*sum(x-0.5)/theta^3
}

# The data
x <- c(1, 20, 13, 25, 4, 7, 5, 26, 36, 32)
n <- length(x) # Number of observations
max_iter <- 100 # To avoid infinite loop in case of non-convergence
iter <- 0 # Counter
set.seed(1030)
th0.exact = runif(1, 1, 24)
cat("initial theta:", th0.exact)

```

initial theta: 20.97108

```

eps = 1e-7
for (i in 1:max_iter){
  th.i <- th0.exact + expS.exact(th0.exact, x, n) / expI.exact(th0.exact, x, n)
  iter <- iter + 1
  if (abs(th.i - th0.exact) < eps || iter >= max_iter) {
    th0.exact = th.i
    break
  }
  th0.exact = th.i
}
cat("\n#iteration:", iter, "\ntheta_hat:", th0.exact)

```

#iteration: 24
theta_hat: 16.89507

Problem 4

Part b

```
A = c(9, 186, 25, 6, 44, 115)
B = c(1, 18, 6, 25, 14, 45)
thhat <- mean(A) # MLE theta
lhat <- sum(B)/sum(A) # MLE lambda

cat(sprintf("theta_hat: %f\n", thhat))
```

theta_hat: 64.166667

```
cat(sprintf("lambda_hat: %f\n", lhat))
```

lambda_hat: 0.283117

Part c

```
expll <- function(lambda, theta, x, y) {
  n <- length(x)
  -2*n*log(theta) - n*log(lambda) - sum(x) / theta - sum(y) / (lambda * theta)
}

exprll <- function(lambda, theta, lambda.hat, theta.hat, x, y) {
  expll(lambda, theta, x, y) - expll(lambda.hat, theta.hat, x, y)
}

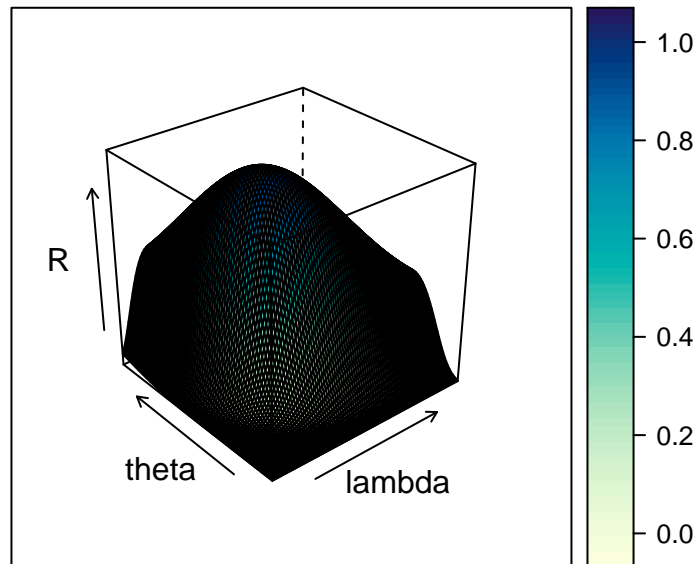
x = c(9, 186, 25, 6, 44, 115)
y = c(1, 18, 6, 25, 14, 45)

theta.hat <- mean(x)
lambda.hat <- sum(y)/sum(x)

RLL <- expand.grid(lambda = seq(0.08, 0.62, length.out = 100),
                  theta = seq(22.08, 106.25, length.out=100))
RLL$r <- sapply(1:nrow(RLL), function(i) exprll(RLL$lambda[i], RLL$theta[i],
                                                lambda.hat, theta.hat, x, y))
```

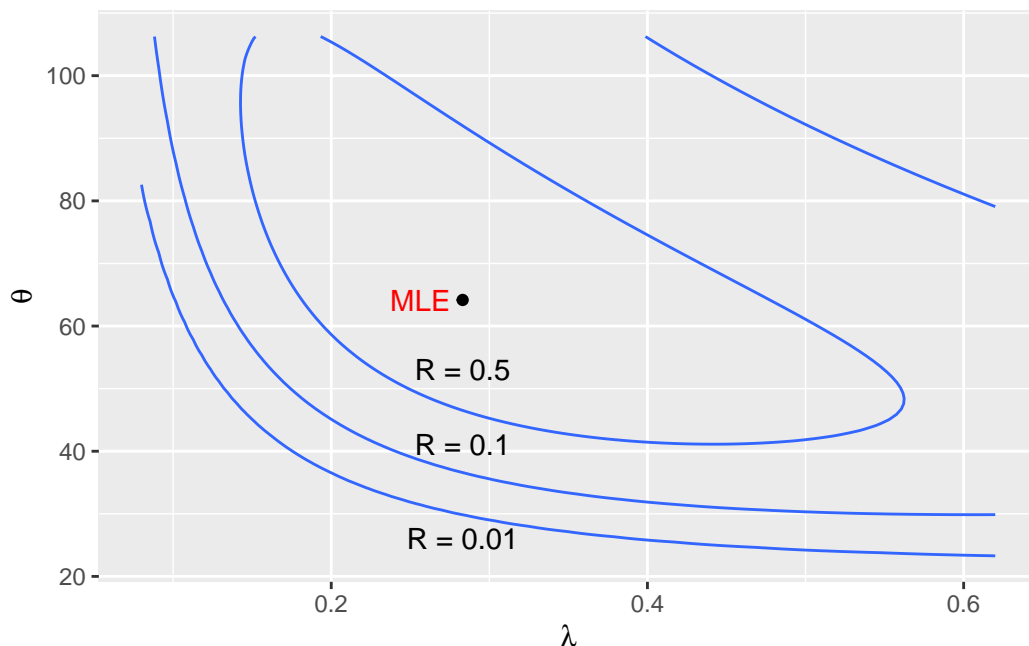
```
RLL$R <- exp(RLL$r)
```

```
wireframe(R~lambda+theta, data = RLL, drape = TRUE)
```



```
gg <- ggplot(RLL, aes(x=lambda, y=theta, z=R)) +
  geom_contour(breaks=c(0.01, 0.1, 0.5)) +
  geom_point(data=data.frame(lambda=lambda.hat, theta=theta.hat,
                             R=exprll(lambda.hat, theta.hat, lambda.hat,
                                       theta.hat, x, y))) +
  annotate("text", x=c(0.2831169, 0.2831169, 0.2831169), y=c(53, 41, 26),
           label = c("R = 0.5", "R = 0.1", "R = 0.01")) +
  annotate("text", x = lambda.hat, y = theta.hat, label = "MLE",
           hjust = 1.2, color = "red")+
  xlab(expression(lambda)) + ylab(expression(theta))
```

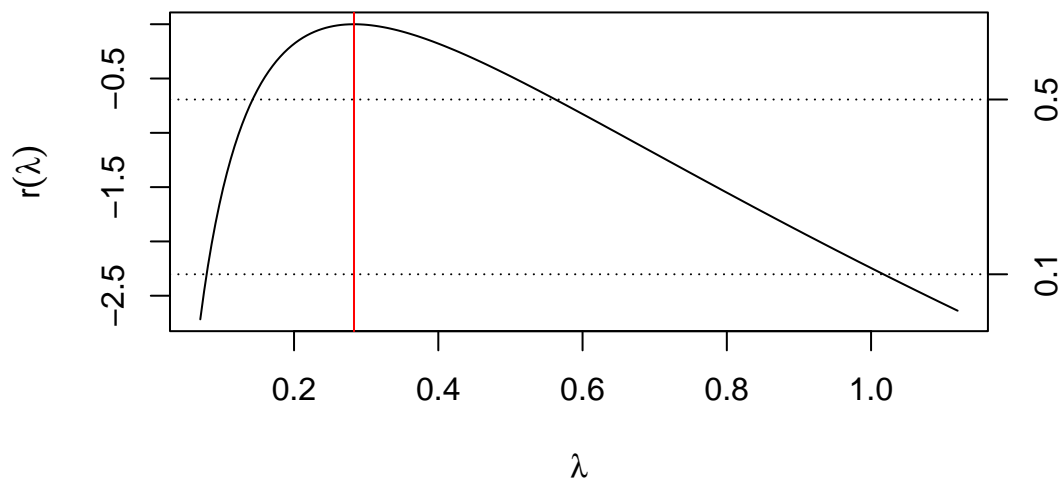
```
gg
```



Part d

```
# Maximum Relative Likelihood function for
expMAXrll <- function(lambda, lambda.hat, theta.hat, x, y) {
  n <- length(x)
  theta.lambda <- (sum(x)+sum(y)/lambda) / (2*n)
  - 2*n*log(theta.lambda) - n*log(lambda) - 1/(theta.lambda)*sum(x) -
    1/(lambda*theta.lambda)*sum(y) -
    expll(lambda.hat, theta.hat, x, y)
}

# Sequence of lambda values
lambda <- seq(0.07, 1.12, length.out = 200)
maxrlambda <- sapply(lambda, expMAXrll, lambda.hat=lambda.hat,
                     theta.hat=theta.hat, x = x, y = y)
plot(lambda, maxrlambda, type = 'l', xlab = expression(lambda),
      ylab = expression(r(lambda)))
abline(v = lambda.hat, col = "red") # MLE line for lambda.hat
abline(h = log(c(0.5, 0.1)), lty = 3) # Horizontal lines for 50% and 10% likelihoods
axis(4, at = log(c(0.5, 0.1)), labels = c(0.5, 0.1))
```



```
Llilambda <- function(lambda, lambda.hat, theta.hat, x, y, p) {
  expMAXrll(lambda, lambda.hat, theta.hat, x, y) - log(p)
}

# Finding the interval bounds at 10% likelihood level
lambdaLow <- uniroot(Llilambda, c(0.01,lambda.hat), lambda.hat=lambda.hat,
  theta.hat=theta.hat, x=x, y=y, p=0.1)$root
lambdaHigh <- uniroot(Llilambda, c(lambda.hat,2), lambda.hat=lambda.hat,
  theta.hat=theta.hat, x=x, y=y, p=0.1)$root
cat("10% likelihood interval for lambda:\n")
```

10% likelihood interval for lambda:

```
cat("(" ,lambdaLow, ", ", lambdaHigh, ")")
```

(0.07879257 , 1.017231)

Part e

```
# Hypothesis testing with the LRT
```

```
# H0: lambda = 1
```

```
cat(sprintf("D: %f\n", -2 * expMAXrll(1,lambda.hat, theta.hat, x, y)))
```

D: 4.490225

```
cat(sprintf("p-value: %f\n", pchisq(-2 * expMAXrll(1,lambda.hat, theta.hat, x, y), df = 1, l
```

p-value: 0.034089

Part c (Again)

```
library(ggplot2)
```

```
library(lattice)
```

```
# Define functions
```

```
expll <- function(lambda, theta, x, y) {
```

```
  n <- length(x)
```

```
  -2 * n * log(theta) - n * log(lambda) - 1 / theta * sum(x) - 1 / (lambda * theta) * sum(y)
```

```
}
```

```
exprll <- function(lambda, theta, lambda.hat, theta.hat, x, y) {
```

```
  expll(lambda, theta, x, y) - expll(lambda.hat, theta.hat, x, y)
```

```
}
```

```
# Data
```

```
x <- c(9, 186, 25, 6, 44, 115)
```

```
y <- c(1, 18, 6, 25, 14, 45)
```

```
# MLE estimates
```

```
theta.hat <- mean(x)
```

```
lambda.hat <- sum(y) / sum(x)
```

```
# Grid with expanded ranges for zoomed-out view
```

```
RLL <- expand.grid(lambda = seq(0.02, 1.0, length.out = 100),  
                  theta = seq(5, 200, length.out = 100))
```

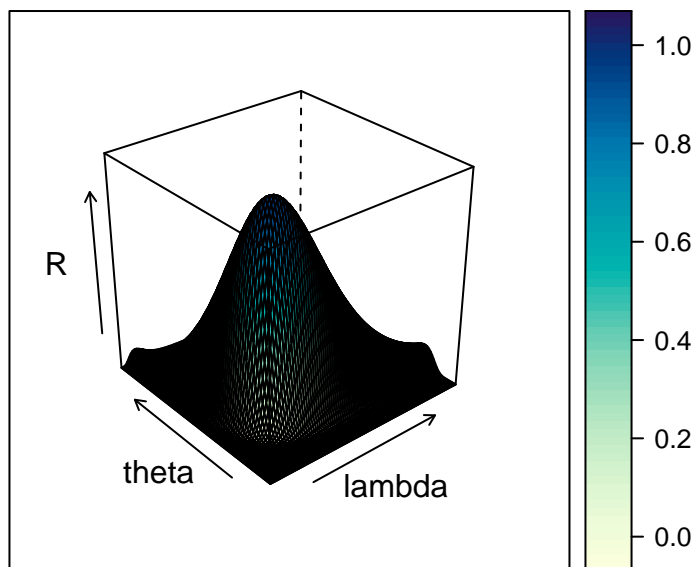
```
RLL$r <- sapply(1:nrow(RLL), function(i)
```

```

  exprll(RLL$lambda[i], RLL$theta[i], lambda.hat, theta.hat, x, y))
RLL$R <- exp(RLL$r)

# 3D wireframe plot (optional)
wireframe(R ~ lambda + theta, data = RLL, drape = TRUE)

```



```

# Contour plot with adjusted axis limits for further zoom-out
gg <- ggplot(RLL, aes(x = lambda, y = theta, z = R)) +
  geom_contour(breaks = c(0.01, 0.1, 0.5), color = "blue") +
  geom_point(data = data.frame(lambda = lambda.hat, theta = theta.hat,
                                R = exprll(lambda.hat, theta.hat, lambda.hat, theta.hat, x, y)),
             aes(x = lambda, y = theta), color = "red") +
  annotate("text", x = c(0.2831169, 0.2831169, 0.2831169), y = c(53, 41, 26),
          label = c("R = 0.5", "R = 0.1", "R = 0.01")) +
  annotate("text", x = lambda.hat, y = theta.hat, label = "MLE", hjust = 1.2, color = "red") +
  xlab(expression(lambda)) + ylab(expression(theta)) +
  xlim(0.02, 1.0) + ylim(5, 200) + # Further expanded axis limits
  ggtitle("Further Expanded Contour Plot with Likelihood Intervals")

# Display the plot
gg

```

Further Expanded Contour Plot with Likelihood Intervals

