Lesson4

Federico Reali

Introduction

This section demonstrates how to construct a likelihood function and compute its Maximum Likelihood Estimate (MLE).

Exercise 1

- A therapy was tested on **30 patients**.
- **10 patients** experienced success (according to a specific definition of success).
- We assume the success probability (p) is **unknown**.
- Each patient’s outcome is **independent**.

Goal:

- Derive and maximize the **log-likelihood function** for (p).
- Find the **Maximum Likelihood Estimate (MLE)** for (p).

Log-Likelihood Function

The log-likelihood function is given by:

[$\ell(p) = k \cdot \log(p) + (n - k) \cdot \log(1 - p)$]

Where:

- ($k = 10$) (number of successes)
- ($n = 30$) (total patients)

Implementation in R

```r
logLik <- function(p) {
  k <- 10  # Number of successes
  n <- 30  # Total number of patients
  return(k * log(p) + (n - k) * log(1 - p))
}

# Test the function
logLik(0.2)  # Example for p = 0.2
```

```
[1] -20.55725
```

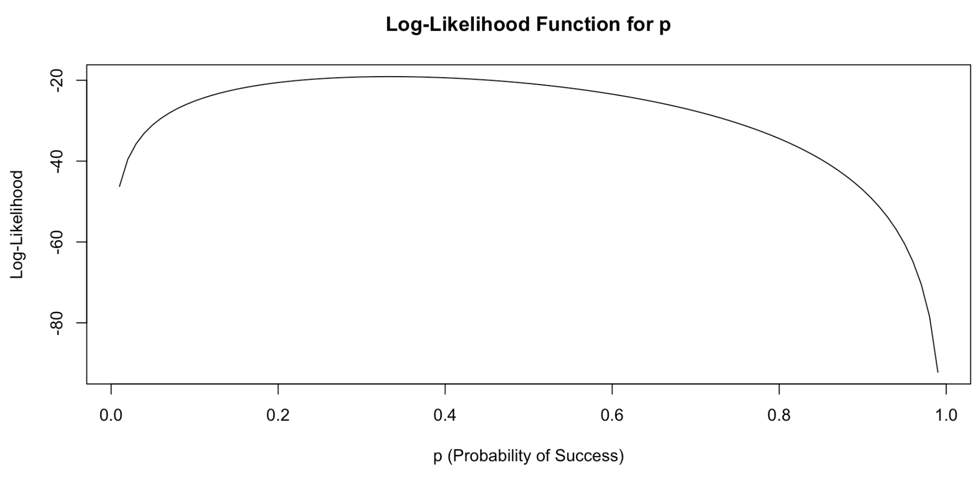
```r
logLik(0.5)  # Example for p = 0.5
```

```
[1] -20.79442
```

Finding the MLE

1. Visualize the Log-Likelihood Function

```r
xx <- seq(0.01, 0.99, 0.01)  # Range of p values
yy <- logLik(xx)             # Compute log-likelihood

# Plot the log-likelihood function
plot(xx, yy, type = "l",
     main = "Log-Likelihood Function for p",
     xlab = "p (Probability of Success)",
     ylab = "Log-Likelihood")
```



2. MLE from the Plot

The MLE corresponds to the value of (p) that maximizes the log-likelihood.

```r
# Find the value of p that maximizes log-likelihood
xx[which.max(yy)]
```

```
[1] 0.33
```

3. Optimize Using R

We can also use R's optimization functions to find the MLE programmatically.

```r
# Define the negative log-likelihood for minimization
mlogLik <- function(p) {
  -logLik(p)  # Negative for minimization
}

# Use optim to find the value of p that minimizes -log-likelihood
optim(0.3, mlogLik)  # Starting value for p
```

```
$par
[1] 0.3332812

$value
[1] 19.09543

$counts
function gradient 
      18       NA 

$convergence
[1] 0
```

```
$message
NULL
```

```
# optim(0.3, mlogLik, method = "Brent", lower = 0, upper = 1)  # Following
```

Using bbmle for MLE

The bbmle package provides convenient tools for performing MLE.

```r
library(bbmle)
# Fit the model using mle2
mle_fit <- mle2(mlogLik, start = list(p = 0.33))
# Summary of the MLE fit
summary(mle_fit)
```

```
Maximum likelihood estimation

Call:
mle2(minuslogl = mlogLik, start = list(p = 0.33))

Coefficients:
  Estimate Std. Error z value     Pr(z)
p 0.333334   0.086066   3.873 0.0001075 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

-2 log L: 38.19085
```

Conclusion

- The **log-likelihood function** describes the likelihood of observing the data for a given (p).
- The **MLE** maximizes this log-likelihood function.
- Based on the methods:
  - **Graphical MLE**: value at the peak of the curve.
  - **Optimization MLE**: Found using numerical methods like `optim` or `mle2`.

Exercise 2

Dataset and Setup

The Geissler dataset refers to a historical dataset that originates from studies conducted by Wilhelm Geissler in the late 19th and early 20th centuries.

The Geissler dataset typically contains the distribution of family sizes and the number of boys in families of a specific size. For example:

Columns in the dataset:

- V1: Number of boys in a family.
- V2: Frequency or count of families with that number of boys.

Importing the Dataset

```r
# Read the Geissler dataset
df_geis <- read.table('./Datasets/geissler.txt', sep = '\t', header = FALSE
head(df_geis, 15) # See the content
```

```
   V1   V2
1   0    7
2   1   45
3   2  181
4   3  478
5   4  829
6   5 1112
7   6 1343
8   7 1033
9   8  670
10  9  286
11 10  104
12 11   24
13 12    3
```

Likelihood Function

Step 1: Define Parameters

```r
# Parameters
p <- 0.52               # Initial probability
n_event <- 12           # Total number of events

# Probability computations
dbinom(6, n_event, p)
```

```
[1] 0.223429
```

```r
dbinom(0:12, n_event, p)
```

```
 [1] 0.0001495873 0.0019446355 0.0115867863 0.0418411727 0.1019878584
 [6] 0.1767789546 0.2234289565 0.2074697454 0.1404743068 0.0676357773
[11] 0.0219816276 0.0043297145 0.0003908770
```

```r
dbinom(1, n_event, p)^45
```

```
[1] 9.947617e-123
```

Step 2: Define Likelihood and Log-Likelihood Functions

Likelihood Function

```
geisL <- function(p) {
  n_event <- 12
  val <- 1
  for (i in 1:13) {
    val <- val * dbinom(df_geis$V1[i], n_event, p)^df_geis$V2[i]
  }
  return(val)
}
geisL(0.3)
```

```
[1] 0
```

```
geisL(0.44)
```

```
[1] 0
```

```
geisL(0.6533)
```

```
[1] 0
```

The values are too small!

Log-Likelihood Function

```
geisLL <- function(p) {
  n_event <- 12
  val <- 0
  for (i in 1:13) {
    val <- val + df_geis$V2[i] * dbinom(df_geis$V1[i], n_event, p, log = TR
  }
  return(val)
}

# Compute log-likelihood for specific probabilities
geisLL(0.418)
```

```
[1] -13122.01
```

```
geisLL(0.45)
```

```
[1] -12674.17
```

```
geisLL(0.52)
```

```
[1] -12759.98
```

Step 3: Plot Log-Likelihood

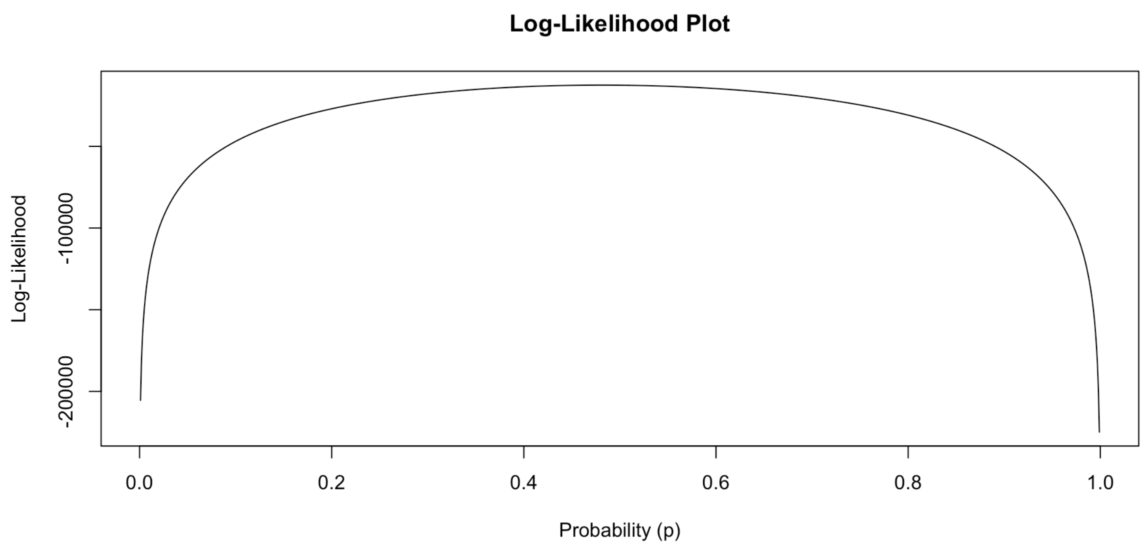
```r
xx <- seq(0, 1, 0.001)
yy <- geisLL(xx)

plot(xx, yy, type = "l", main = "Log-Likelihood Plot", xlab = "Probability
```

```r
which.max(yy) # Find the index of maximum likelihood
```

```
[1] 482
```

```r
xx[which.max(yy)] # Corresponding probability
```

```
[1] 0.481
```



Optimization

Finding the MLE

```r
# Optimize the log-likelihood
p_optimized <- optimize(geisLL, c(0.35, 0.50), maximum = TRUE)
p_optimized
```

```
$maximum
[1] 0.480784

$objective
[1] -12534.17
```

```r
# Verify the optimized value
geisLL(p_optimized$maximum)
```

```
[1] -12534.17
```

BBMLE

```r
library(bbmle)
# Requires the minus log-likelihood

mingeissLL <- function(p){
  return(-1*geisLL(p))
}

p_mle2 <- mle2(mingeissLL, start=list(p=0.4))
p_mle2
```

```
Call:
mle2(minuslogl = mingeissLL, start = list(p = 0.4))

Coefficients:
        p
0.4807842

Log-likelihood: -12534.17
```

```r
summary(p_mle2)
```

```
Maximum likelihood estimation

Call:
mle2(minuslogl = mingeissLL, start = list(p = 0.4))

Coefficients:
   Estimate Std. Error z value     Pr(z)    
p 0.4807842  0.0018444  260.67 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

-2 log L: 25068.34 
```

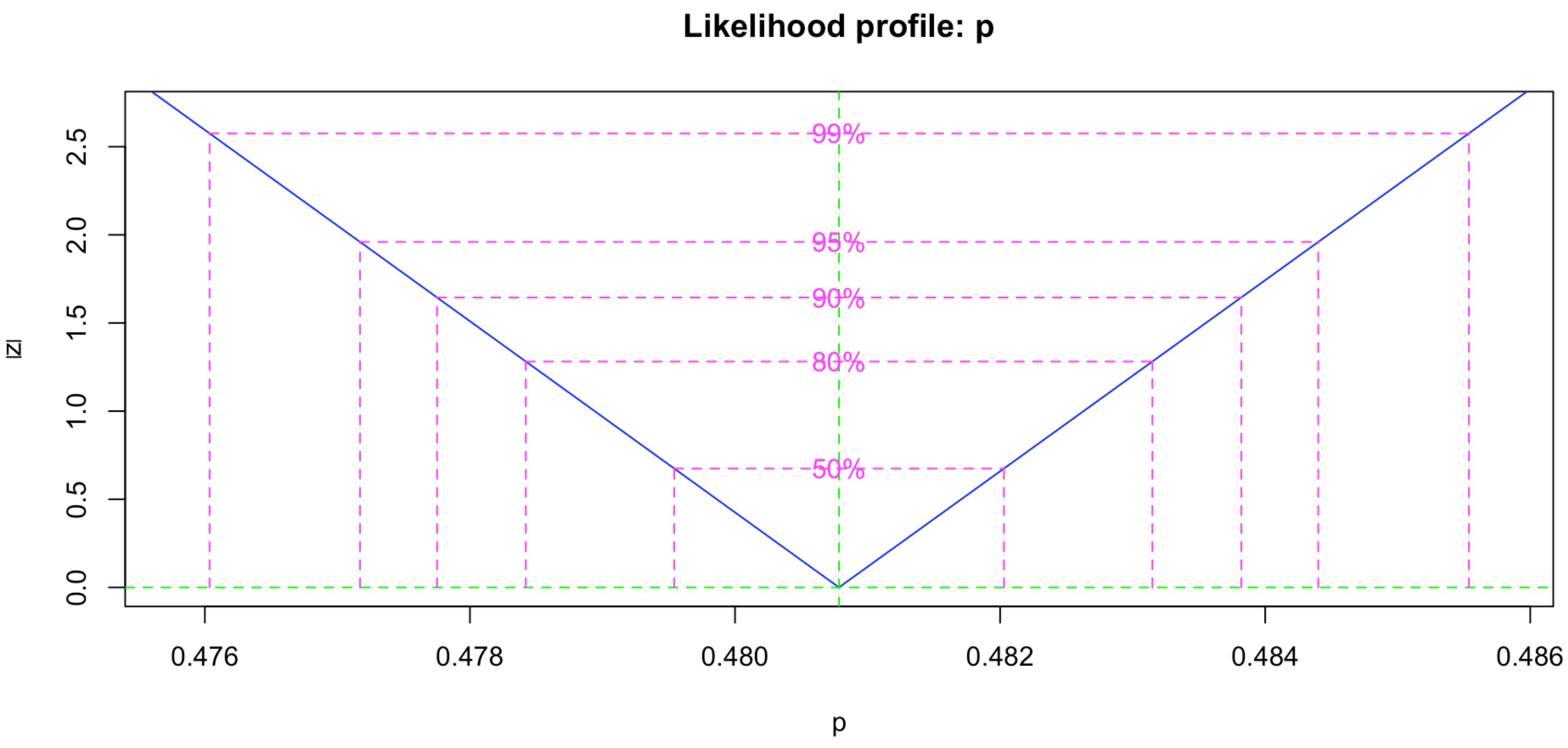
```r
# Verify the optimized value
geisLL(coef(p_mle2))
```

```
[1] -12534.17
```

```
confint(p_mle2)
```

```
    2.5 %    97.5 %
0.4771707 0.4844006
```

```
plot(profile(p_mle2))
```



Exercise 3

We analyze the survival of **100 fruit flies** observed daily, with the following data:

Day	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Alive	91	79	65	58	50	29	16	10	7	4	1	0

Assumptions

1. Survival times follow an **Exponential Distribution** with parameter (λ).
2. Each fly is observed daily until death.

Goal

- Derive the **likelihood function** for (λ).
- Compute the Maximum Likelihood Estimate (MLE) for (λ).

Likelihood Function

Exponential Model

For the exponential distribution: [$f(t;\lambda) = \lambda e^{-\lambda t}$]

where ($\lambda > 0$) is the rate parameter.

Likelihood Function

Let:

- (x_d): Days (1 to 12).
- (z_d): Number of deaths on day (d).

The likelihood function is: [$L(\lambda) = \prod_{d=1}^{12} \left[\lambda e^{-\lambda x_d}\right]^{z_d}$]

Log-Likelihood Function

Taking the log:

[$\ell(\lambda) = \sum_{d=1}^{12} z_d \cdot \left[\log(\lambda) - \lambda x_d\right]$]

Data Preparation

```
# Days and alive counts
xd <- 1:12
yd <- c(91, 79, 65, 58, 50, 29, 16, 10, 7, 4, 1, 0)

# Compute deaths (difference in alive counts)
zd <- -diff(c(100, yd))  # Number of deaths per day

zd
```

```
[1]  9 12 14  7  8 21 13  6  3  3  3  1
```

Likelihood and Log-Likelihood Functions

```
# Likelihood function
DeathLike <- function(lambda) {
  val <- 1
  for (i in xd) {
    val <- val * (lambda * exp(-lambda * xd[i]))^zd[i]
  }
  return(val)
}

# Log-likelihood function
LDeathLike <- function(lambda) {
  sum(zd) * log(lambda) - lambda * sum(xd * zd)
}

LDeathLike(1)
```

```
[1] -510
```

```
LDeathLike(0.6)
```

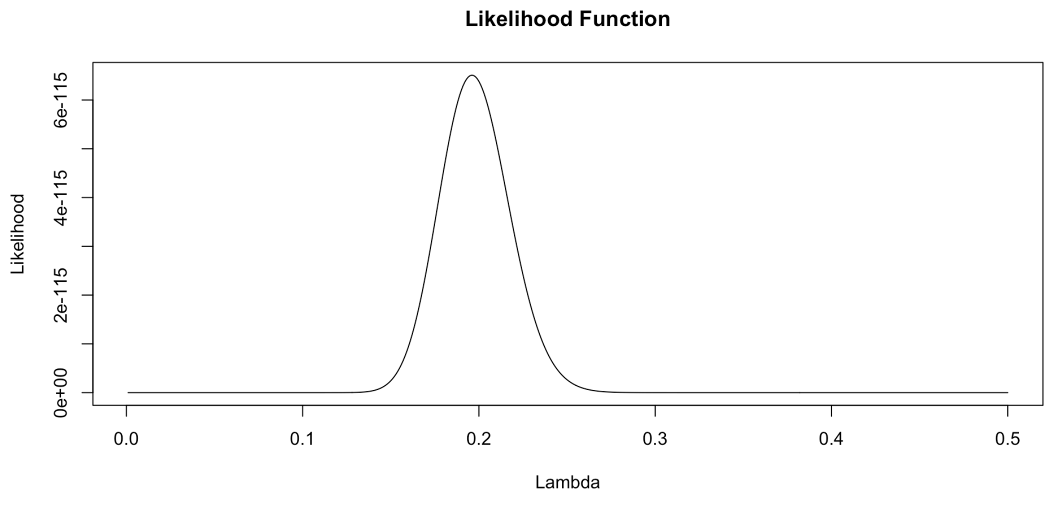
```
[1] -357.0826
```

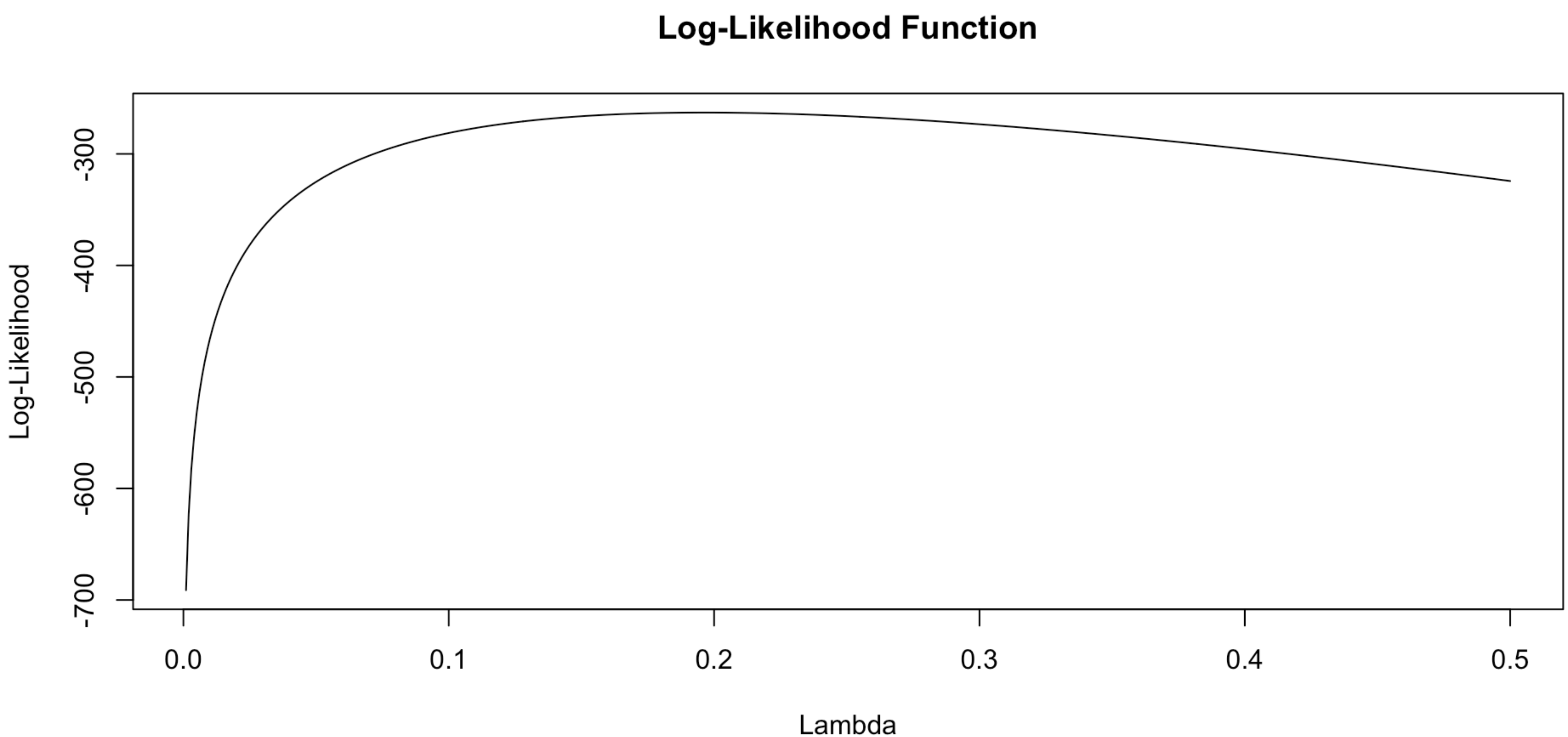
```
LDeathLike(0.3)
```

```
[1] -273.3973
```

Visualizing the Likelihood

```r
# Evaluate the likelihood and log-likelihood
xx <- seq(0.001, 0.5, 0.001)
likelihood <- sapply(xx, DeathLike)
log_likelihood <- sapply(xx, LDeathLike)

# Plot likelihood
plot(xx, likelihood, type = "l",
     main = "Likelihood Function",
     xlab = "Lambda",
     ylab = "Likelihood")
```




Log-Likelihood Function
Log-Likelihood
-300
-400
-500
-600
-700
0.0
0.1
0.2
0.3
0.4
0.5
Lambda

Finding the MLE

Using Visual Inspection

```r
# Maximum from the log-likelihood plot
xx[which.max(log_likelihood)]
```

```
[1] 0.196
```

Using Optimization

```r
# Minimizing the negative log-likelihood
minusL <- function(lambda) -LDeathLike(lambda)

# Optimize
opt_result <- optimize(LDeathLike, c(0, 1), maximum = TRUE)
opt_result
```

```
$maximum
[1] 0.1960841

$objective
[1] -262.9241
```

Conclusion

- The MLE for (λ) is approximately ($\lambda =$ 0.1960841).
- Using the exponential survival model, (λ) quantifies the daily survival rate.

Here's the updated QUARTO presentation with `echo=T` added to all code chunks:

Exercise 4: Likelihood ratio

- Suppose we flip a coin 10 times, and observe 7 heads.
- We want to test:
  - H_0: The coin is fair $p = 0.5$.
  - H_1: The coin is biased $p \neq 0.5$.

Likelihoods Under H_0 and H_1

- **Likelihood under H_0:** *[Math Processing Error]*
- **Likelihood under H_1:**
  - Maximum likelihood estimate (MLE): $\hat{p} = \frac{7}{10}$

[Math Processing Error]

Likelihood Ratio Test

Purpose of the Likelihood Ratio Test

- The likelihood ratio test is a statistical method used to compare two competing models:
  - **Null hypothesis** (H_0): Simpler model, e.g., equal means for two groups.
  - **Alternative hypothesis** (H_1): More complex model, e.g., different means for two groups.
- The LRT evaluates whether the data provide enough evidence to prefer the more complex model (H_1) over the simpler model (H_0).

How the Likelihood Ratio Test Works

1. **Compute the Likelihood Under Each Model**:
   - L_0: Likelihood of the data assuming the null hypothesis H_0.
   - L_1: Likelihood of the data assuming the alternative hypothesis H_1.
2. **Calculate the Test Statistic**:
   - [$\Lambda = -2 \cdot (\log \mathrm{L}_0 - \log \mathrm{L}_1)$]
   - Λ compares the “accuracy” of the two models.

3. **Determine Statistical Significance**:
   - Under the null hypothesis, Λ approximately follows a χ^2 -distribution with degrees of freedom equal to the difference in the number of parameters between H_0 and H_1.
4. **Interpret the Results**:
   - Small p-value ($p < \alpha$) → Reject H_0: Evidence supports the alternative hypothesis.
   - Large p-value ($p \geq \alpha$) → Fail to reject H_0: Insufficient evidence to prefer the alternative hypothesis.

Computing the Log-Likelihoods

```r
# Observed data
n <- 10
k <- 7

# Likelihood under H0
L_H0 <- dbinom(k, n, 0.5)

# Likelihood under H1 (with MLE for p)
p_hat <- k / n
L_H1 <- dbinom(k, n, p_hat)

# Log-likelihood ratio
log_likelihood_ratio <- -2 * (log(L_H0) - log(L_H1))
log_likelihood_ratio
```

```
[1] 1.645658
```

Compare with χ^2

```r
# p-value for the test
pval <- pchisq(log_likelihood_ratio, df=1, lower.tail=FALSE)
pval
```

```
[1] 0.199551
```

So we cannot reject the null hypothesis, we cannot say that the coin is unfair.

Exercise 5: Bike Sharing Data Analysis

Dataset Overview

We analyze a dataset about bike sharing services in Washington to address the following:

1. Checking variable types and data integrity.
2. Visualizing bike counts over time.
3. Investigating whether the mean number of bikes rented in May 2011 and May 2012 are statistically different using:
    - Likelihood ratio test.
    - Comparison with t-test results.

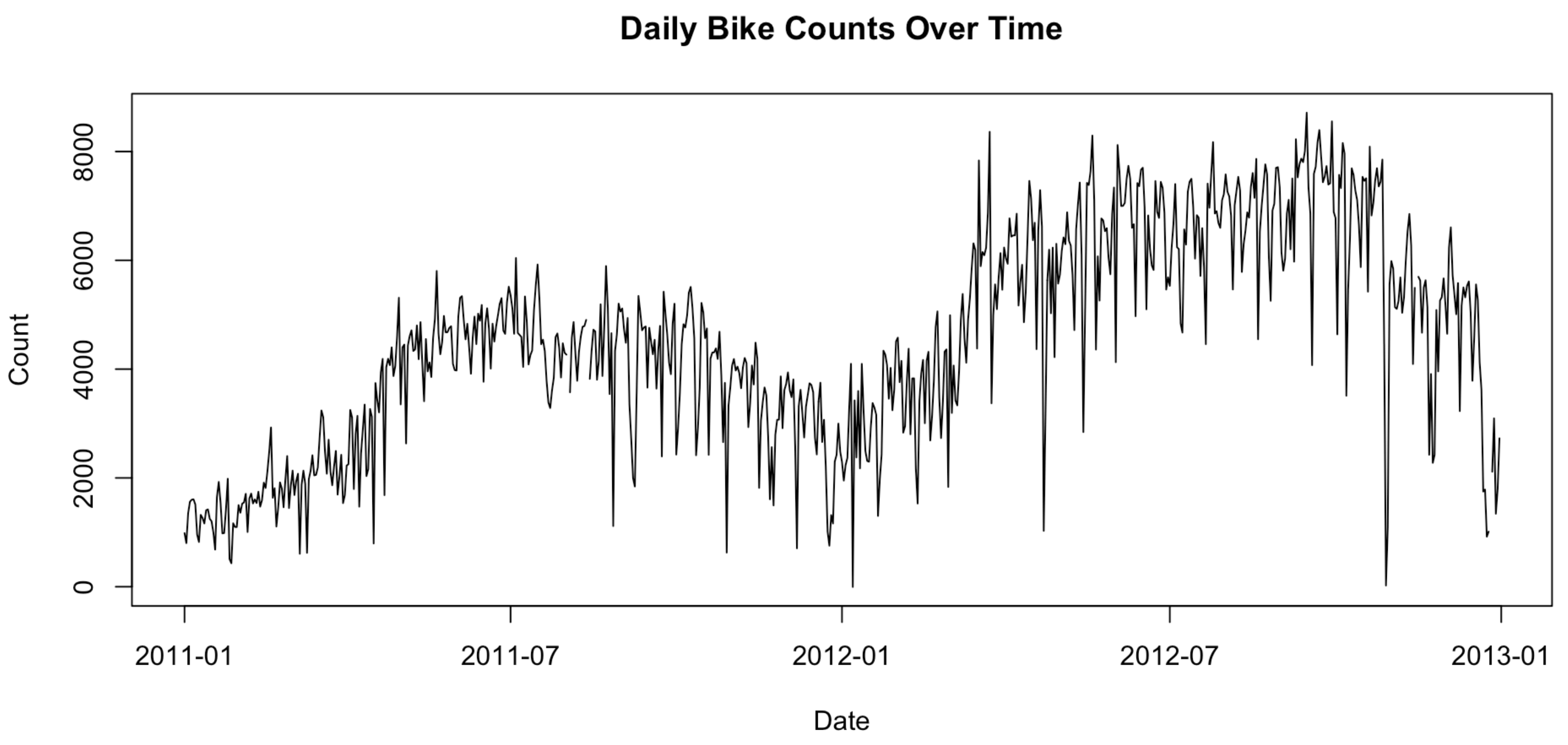
Load and Inspect the Dataset

```r
# Load the dataset
df_b <- read.csv("./Datasets/Count0.csv")

# Convert date column to Date format
df_b$dteday <- as.Date(df_b$dteday)

# View summary of the dataset
summary(df_b)
```

```
       X             dteday               season            yr        
 Min.   :  1.0   Min.   :2011-01-01   Min.   :1.000   Min.   :0.0000  
 1st Qu.:183.5   1st Qu.:2011-07-02   1st Qu.:2.000   1st Qu.:0.0000  
 Median :366.0   Median :2012-01-01   Median :3.000   Median :1.0000  
 Mean   :366.0   Mean   :2012-01-01   Mean   :2.497   Mean   :0.5007  
 3rd Qu.:548.5   3rd Qu.:2012-07-01   3rd Qu.:3.000   3rd Qu.:1.0000  
 Max.   :731.0   Max.   :2012-12-31   Max.   :4.000   Max.   :1.0000  

      mnth          holiday           weekday        workingday   
 Min.   : 1.00   Min.   :0.00000   Min.   :0.000   Min.   :0.000  
 1st Qu.: 4.00   1st Qu.:0.00000   1st Qu.:1.000   1st Qu.:0.000  
 Median : 7.00   Median :0.00000   Median :3.000   Median :1.000  
 Mean   : 6.52   Mean   :0.02873   Mean   :2.997   Mean   :0.684  
 3rd Qu.:10.00   3rd Qu.:0.00000   3rd Qu.:5.000   3rd Qu.:1.000  
 Max.   :12.00   Max.   :1.00000   Max.   :6.000   Max.   :1.000  
```

Visualize Bike Counts Over Time

```r
# Visualize the daily bike counts
plot(df_b$dteday, df_b$cnt, type = "l",
     main = "Daily Bike Counts Over Time",
     xlab = "Date",
     ylab = "Count")
```



Compare May 2011 and May 2012

Extract Data for May 2011 and May 2012

```
# Filter data for May 2011 and May 2012
df_may11 <- df_b$cnt[(df_b$dteday >= '2011-05-01') & (df_b$dteday <= '2011-
df_may12 <- df_b$cnt[(df_b$dteday >= '2012-05-01') & (df_b$dteday <= '2012-
```

Calculate Sample Statistics

```
# Means and variances for May 2011 and May 2012
mu1 <- mean(df_may11)
mu2 <- mean(df_may12)
n1 <- length(df_may11)
n2 <- length(df_may12)
var1 <- var(df_may11)
var2 <- var(df_may12)

# Combined statistics
n <- n1 + n2
```

Comparing Pooled and Separate Variances

- **Pooled Variance H_0:**
  - Assumes both groups have the **same variance** σ^2.
  - Log-likelihood involves a **single variance estimate** shared by both groups.
  - Variance estimate: $\sigma 0 = \frac{\text{pooled variance estimate} \cdot (\text{n}-1)}{\text{n}}$

- **Separate Variances H_1:**
  - Assumes groups have **different variances** σ_1^2, σ_2^2.
  - Log-likelihood incorporates **individual variance estimates** for each group.
  - Variance estimate: $\sigma 1 = \frac{(n_1-1)s_1^2+(n_2-1)s_2^2}{n_1+n_2}$

Likelihood Functions

Under H_0:

- *[Math Processing Error]*
- Single variance σ^2 applied to all observations.

Under H_1:

- $L(\mu_1, \sigma_1^2, \mu_2, \sigma_2^2) \propto$

[Math Processing Error]

Calculate Pooled and Separate Variances

```
# Pooled variance (under H0)
xy <- c(df_may11, df_may12)
ssigma0 <- var(xy) * (n - 1) / n

# Separate variances (under H1)
ssigma1 <- ((n1 - 1) * var1 + (n2 - 1) * var2) / (n1 + n2)
```

Calculate Log-Likelihoods

```
# Log-likelihoods under H0 and H1
l0 <- -0.5 * n * log(ssigma0)
l1 <- -0.5 * n * log(ssigma1)

# Likelihood ratio statistic
LRT_stat <- -2 * (l0 - l1)

# p-value
pval <- pchisq(LRT_stat, df = 1, lower.tail = FALSE)
pval
```

```
[1] 6.68034e-13
```

t-test for Comparison

```r
# Two-sample t-test with equal variances
t.test(df_may11, df_may12, var.equal = TRUE)
```

```
	Two Sample t-test

data:  df_may11 and df_may12
t = -8.8312, df = 60, p-value = 1.9e-12
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2375.618 -1498.189
sample estimates:
mean of x mean of y 
 4381.323  6318.226 
```

```
	Welch Two Sample t-test

data:  df_may11 and df_may12
t = -8.8312, df = 45.686, p-value = 1.927e-11
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2378.463 -1495.344
sample estimates:
mean of x mean of y
 4381.323  6318.226
```

Results and Conclusions

Likelihood Ratio Test

- p-value= 6.6803404^{-13}
- We reject the null hypothesis $\mathrm{H}_0 : \mu_1 = \mu_2$, indicating a significant difference between the mean counts in May 2011 and May 2012.

t-test

- Both equal and unequal variance t-tests yield consistent results with the likelihood ratio test.