Statistics CH4

Xiang Li

2023/10/21

4.2.0.1 Excercise

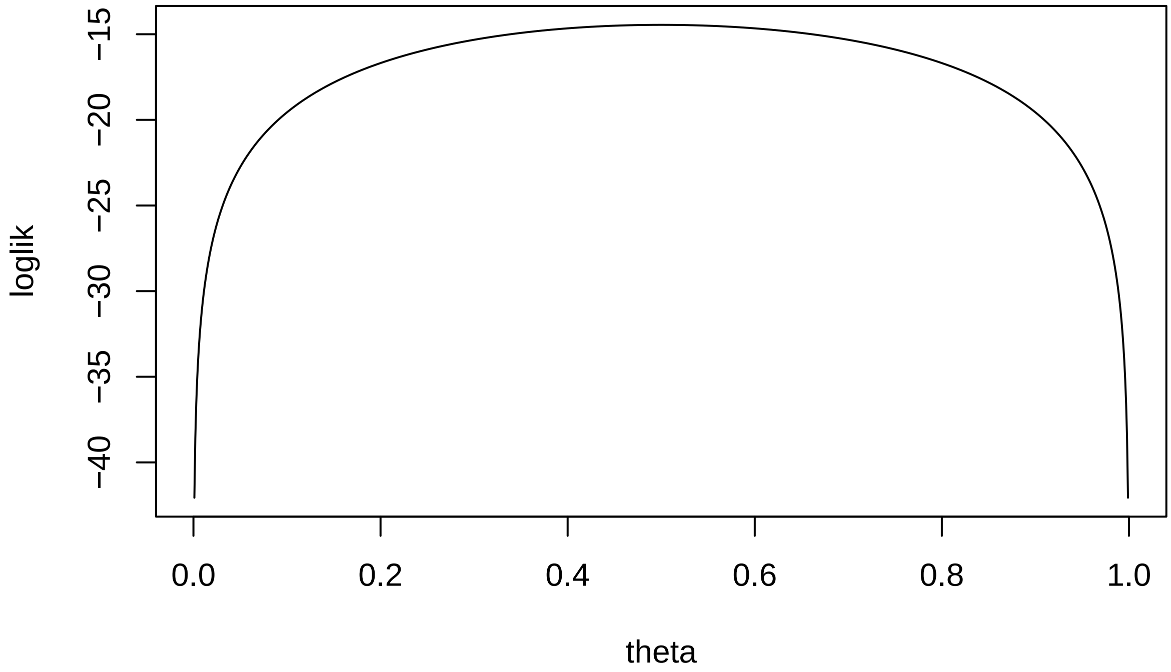
a.

```
X = c(3, 0, 2, 1, 3, 2, 1, 0, 2, 1)
theta_hat_mom = (7/3 - mean(X))/2
theta_hat_mom
```

```
## [1] 0.4166667
```

b.

```
theta = seq(0, 1, 0.001)
loglik = 2*log(2/3*theta)+3*log(1/3*theta)+3*log(2/3*(1-theta))+2*log(1/3*(1-theta))
plot(theta, loglik, type='l')
```



c.

```
theta_hat_mle = theta[which.max(loglik)]
theta_hat_mle
```

```
## [1] 0.5
```

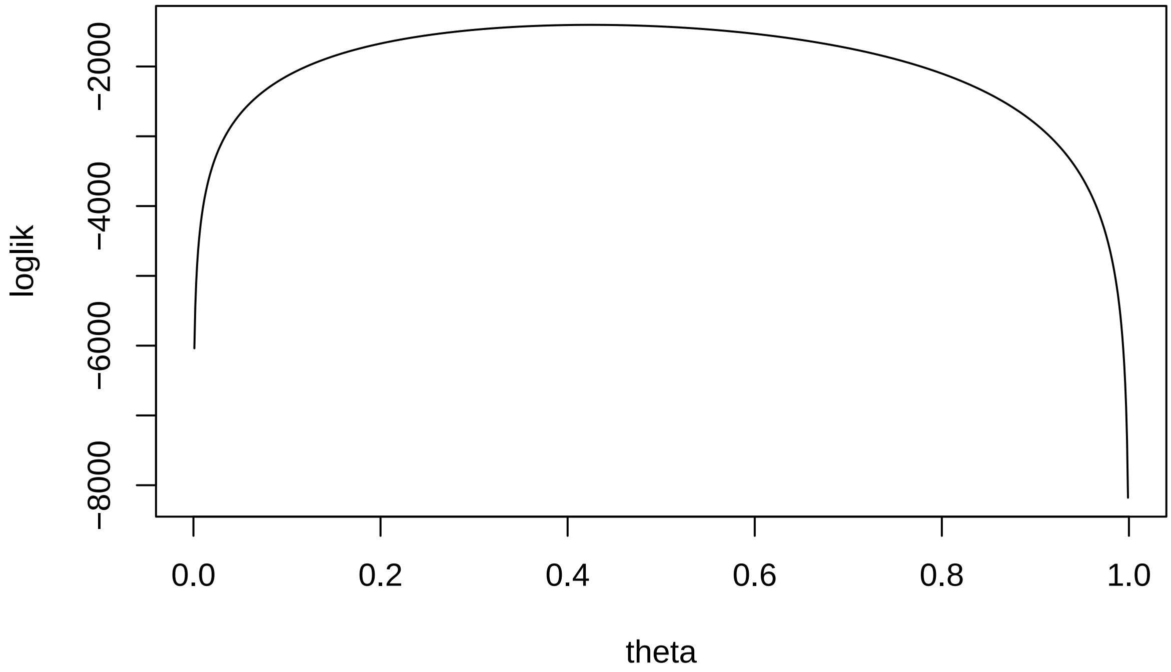
```
mean_hat_mle = 7/3 - 2*theta_hat_mle
mean_hat_mle
```

```
## [1] 1.333333
```

4.2.0.2 Exercise

a.

```
theta = seq(0, 1, 0.001)
loglik = 1184*log(1-theta)+874*log(theta)
plot(theta, loglik, type='l')
```



b.

```
theta_hat_mle = theta[which.max(loglik)]
theta_hat_mle
```

```
## [1] 0.425
```

4.2.0.4 Exercise

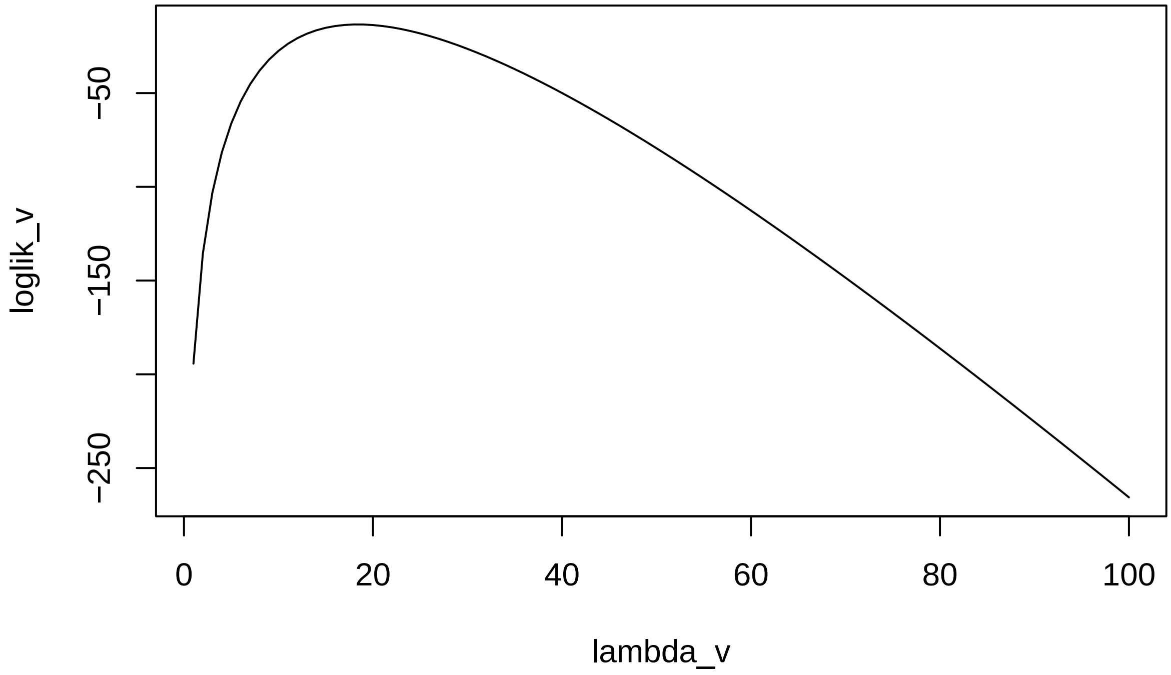
b.

```
X = c(17, 25, 17, 18, 15)
e_hat_mom = mean(X)
e_hat_mom
```

```
## [1] 18.4
```

c.

```r
loglik = function(lambda) {
    X = c(17, 25, 17, 18, 15)
    result = sum(sapply(X, dpois, lambda=lambda, log = TRUE))
    return(result)
}
lambda_v = seq(1, 100, 1)
loglik_v = sapply(lambda_v, loglik)
plot(lambda_v, loglik_v, type='l')
```



d.

```r
lambda_hat_mle = lambda_v[which.max(loglik_v)]
lambda_hat_mle
```

```
## [1] 18
```

e.

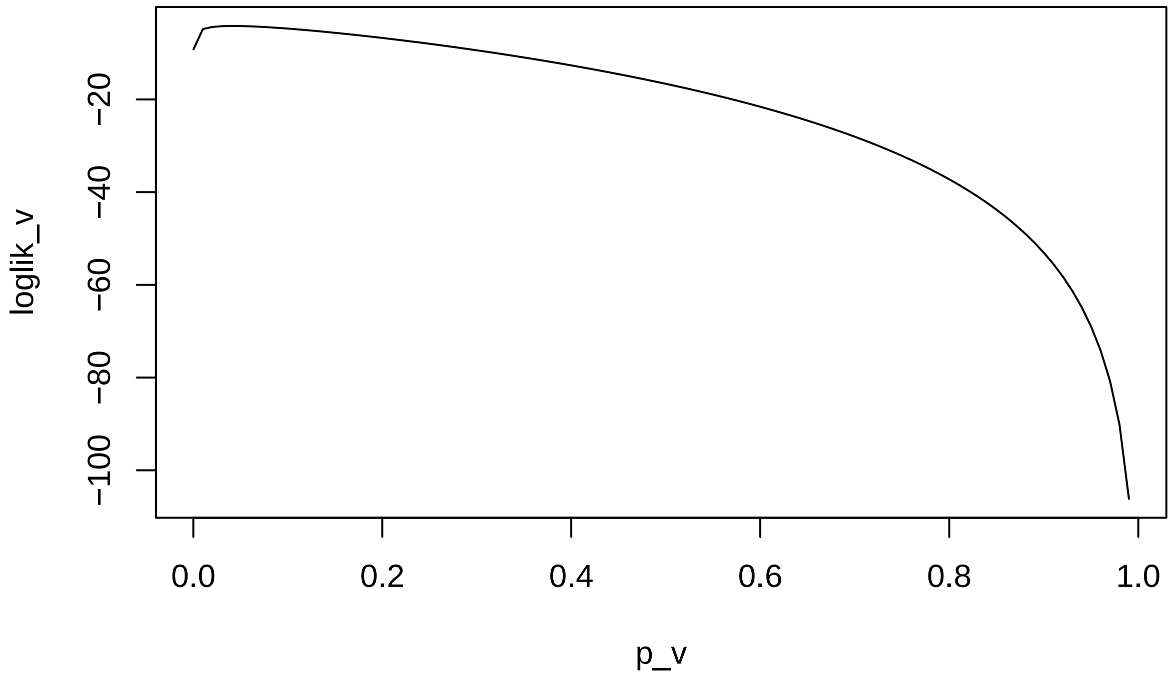
```r
opt = optimize(loglik, c(1, 100), maximum = TRUE)
lambda_hat_mle2 = opt$maximum
lambda_hat_mle2
```

```
## [1] 18.40002
```

4.2.0.6 Exercise

c.

```
loglik = function(p) {
    X = 23
    result = dgeom(X, prob = p, log = TRUE)
    return(result)
}
p_v = seq(0.0001,1,0.01)
loglik_v = loglik(p_v)
plot(p_v, loglik_v, type='l')
```



d.

```
p_hat_mle = p_v[which.max(loglik_v)]
p_hat_mle
```

```
## [1] 0.0401
```

e.

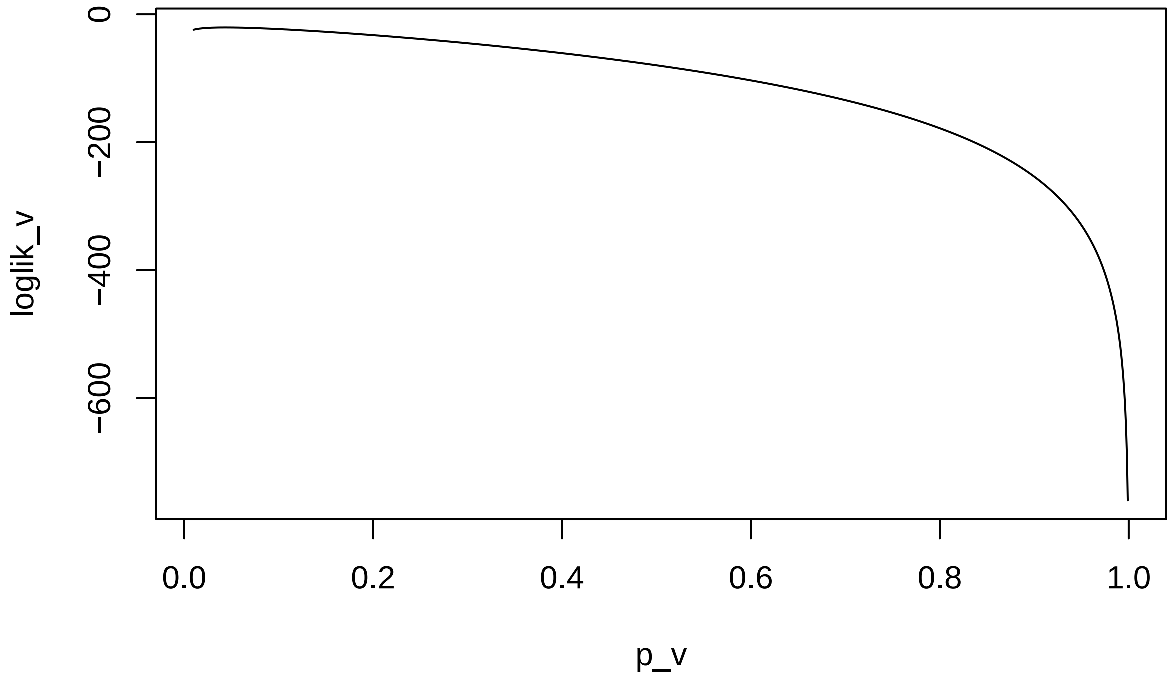
```
opt = optimize(loglik, c(0,1), maximum=TRUE)
p_hat_mle1 = opt$maximum
p_hat_mle1
```

```
## [1] 0.04166196
```

4.2.0.7 Exercise

a.

```
loglik = function(p) {
    X = 20:24
    result = sum(dgeom(X, prob = p, log = TRUE))
    return(result)
}
p_v = seq(0.01,1,0.001)
loglik_v = sapply(p_v, loglik)
plot(p_v, loglik_v, type='l')
```



b.

```
opt = optimize(loglik, c(0,1), maximum=TRUE)
p_hat_mle = opt$maximum
p_hat_mle
```

```
## [1] 0.04347581
```

4.2.0.8 Exercise

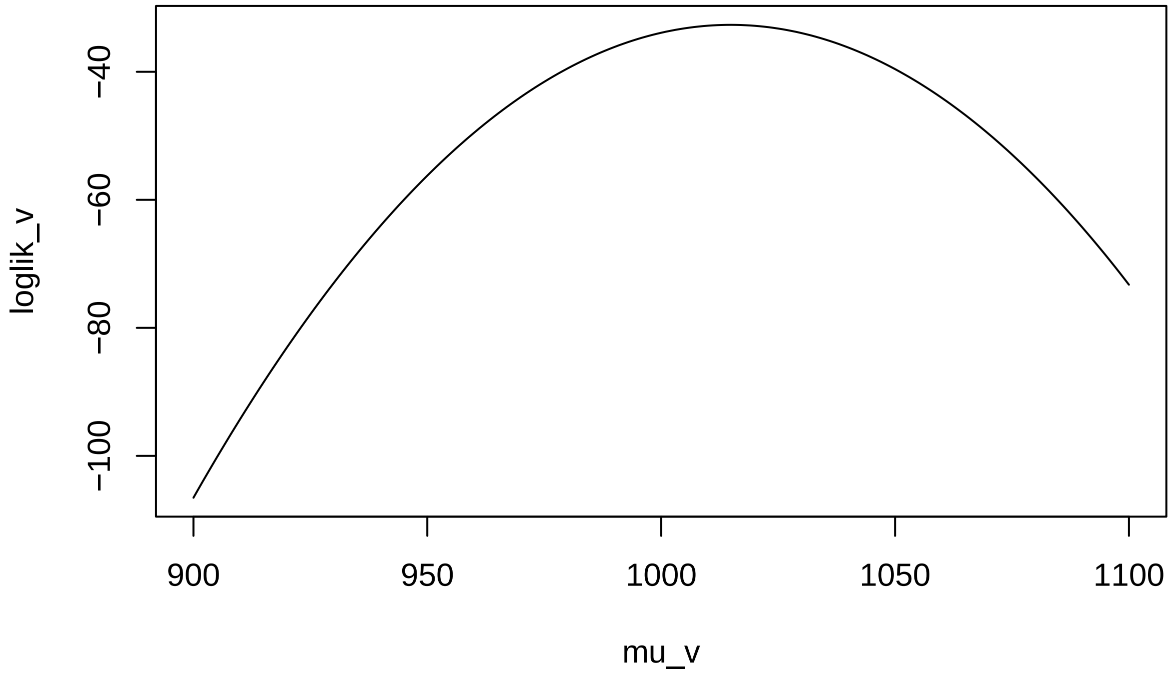
b.

```
X = c(983, 1041, 1026, 1025, 1033, 968, 1028)
mu_hat_mom = mean(X)
mu_hat_mom
```

```
## [1] 1014.857
```

c.

```
loglik = function(mu) {
  X = c(983, 1041, 1026, 1025, 1033, 968, 1028)
  result = sum(dnorm(X, mean = mu, sd = 25, log = TRUE))
  return(result)
}
mu_v = seq(900, 1100, 1)
loglik_v = sapply(mu_v, loglik)
plot(mu_v, loglik_v, type = 'l')
```



```
mu_hat_mle = mu_v[which.max(loglik_v)]
mu_hat_mle
```

```
## [1] 1015
```

d.

```
opt = optimize(loglik, c(900, 1100), maximum = TRUE)
mu_hat_mle2 = opt$maximum
mu_hat_mle2
```

```
## [1] 1014.857
```

4.2.0.9 Exercise

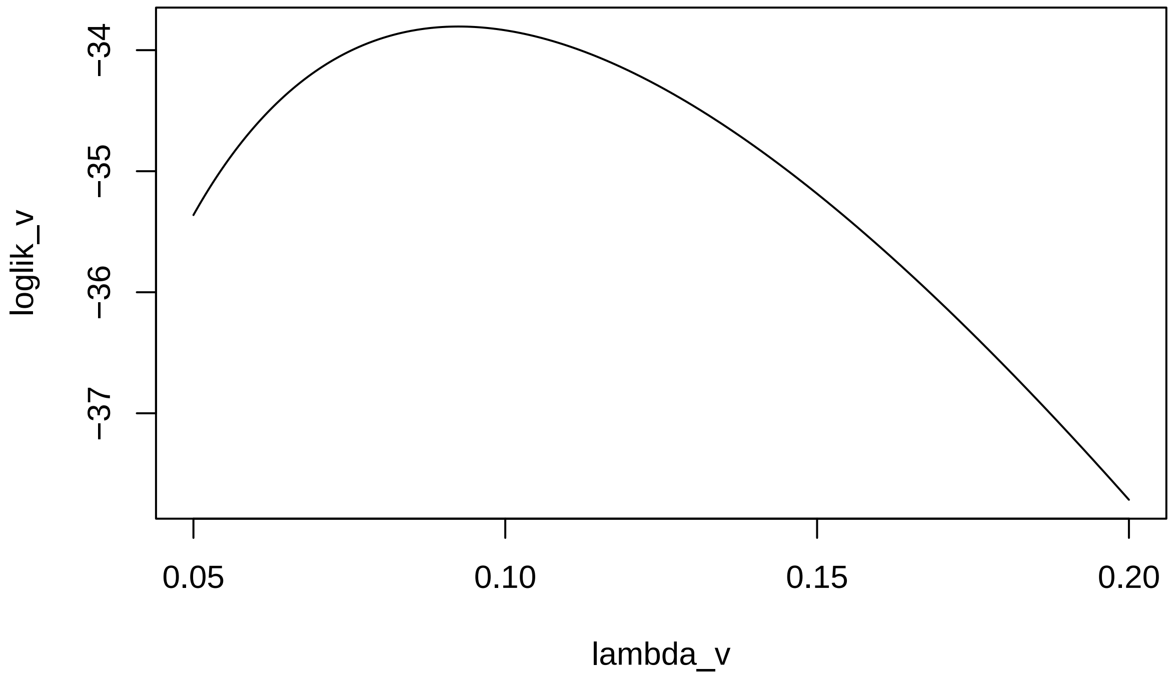
b.

```
X = c(8.5, 2.3, 11.0, 22.5, 13.6, 5.8, 27.3, 13.1, 0.9, 3.1)
mean(X)
```

```
## [1] 10.81
```

c.

```
loglik = function(lambda){
  X = c(8.5, 2.3, 11.0, 22.5, 13.6, 5.8, 27.3, 13.1, 0.9, 3.1)
  result = sum(dexp(X, rate = lambda, log = TRUE))
  return(result)
}
expecta = seq(5, 20, 0.01)
lambda_v = 1/expecta
loglik_v = sapply(lambda_v, loglik)
plot(lambda_v, loglik_v, type = 'l')
```



```
lambda_hat_mle = lambda_v[which.max(loglik_v)]
1/lambda_hat_mle
```

```
## [1] 10.81
```

d.

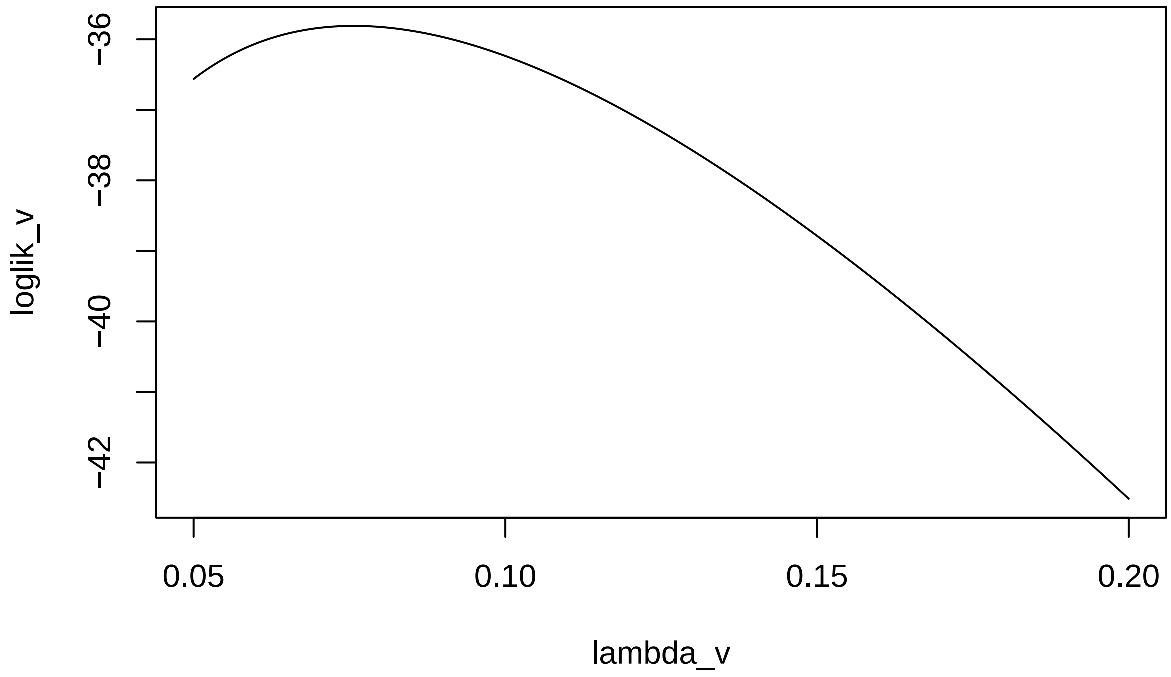
```
opt = optimize(loglik, c(0, 1), maximum = TRUE)
lambda_hat_mle1 = opt$maximum
1/lambda_hat_mle1
```

```
## [1] 10.81054
```

4.2.0.10 Exercise

a.

```
loglik = function(lambda){
  X = c(8.5, 2.3, 11.0, 22.5, 13.6, 5.8, 27.3, 13.1, 0.9, 3.1)
  result = sum(dexp(X, rate = lambda, log = TRUE)) + pexp(24, rate = lambda, lower.tail = FALSE, log.p
  return(result)
}
expecta = seq(5, 20, 0.01)
lambda_v = 1/expecta
loglik_v = sapply(lambda_v, loglik)
plot(lambda_v, loglik_v, type = 'l')
```



```
lambda_hat_mle = lambda_v[which.max(loglik_v)]
1/lambda_hat_mle
```

```
## [1] 13.21
```

b.

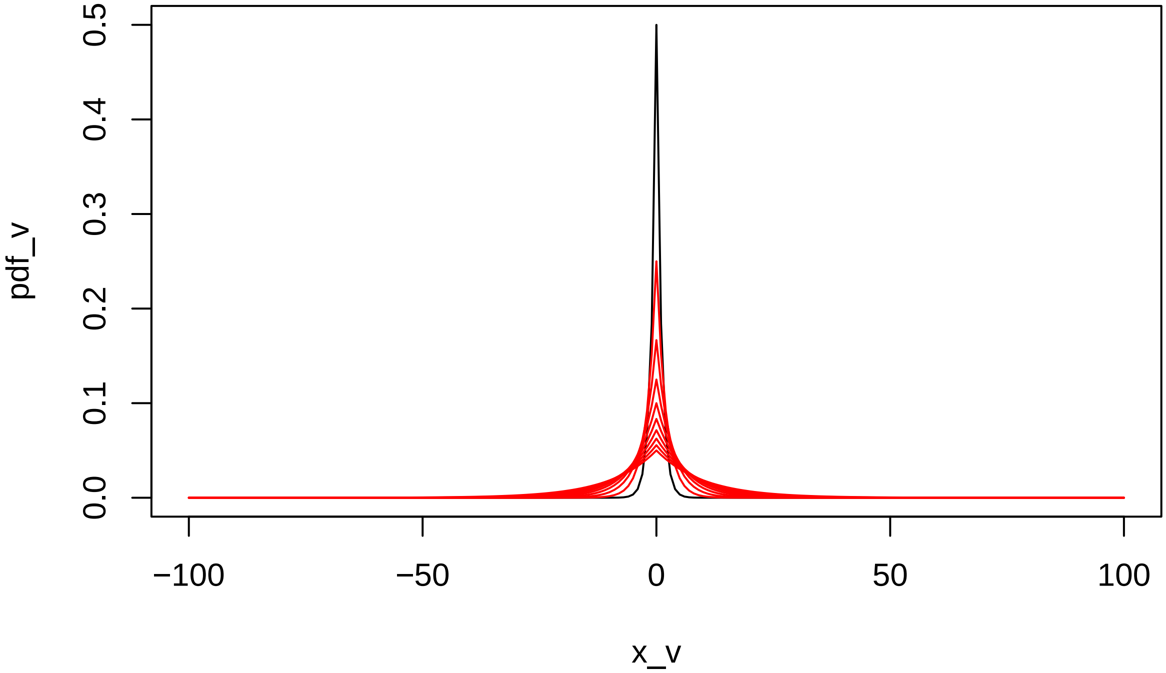
```
opt = optimize(loglik, c(0, 1), maximum = TRUE)
lambda_hat_mle1 = opt$maximum
1/lambda_hat_mle1
```

```
## [1] 13.21264
```

4.2.0.11 Exercise

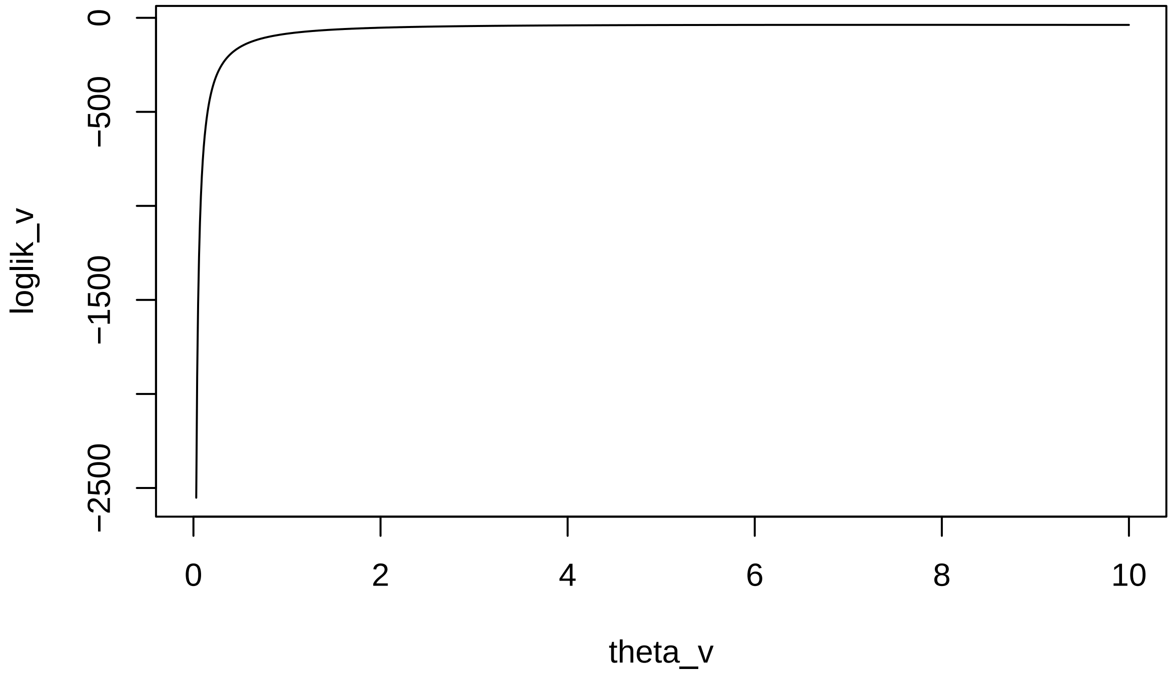
a.

```
pdf = function(x, theta){
  result = 1/(2*theta)*exp(-1*abs(x)/theta)
  return(result)
}
x_v = seq(-100, 100, 1)
pdf_v = pdf(x_v, 1)
plot(x_v, pdf_v, type = 'l')
for (i in 2:10) {
  pdf_v = pdf(x_v, i)
  lines(x_v, pdf_v, type = 'l', col = 'red')
}
```



b.

```
loglik = function(theta){
  X = c(4.77,5.52,0.38,1.13,21.86,13.79,12.61,1.01,13.96,2.37)
  result = sum(log(pdf(X, theta)))
  return(result)
}
theta_v = seq(0, 10, 0.01)
loglik_v = sapply(theta_v, loglik)
plot(theta_v, loglik_v, type = 'l')
```



```
theta_hat_mle = theta_v[which.max(loglik_v)]
theta_hat_mle
```

```
## [1] 7.74
```

c.

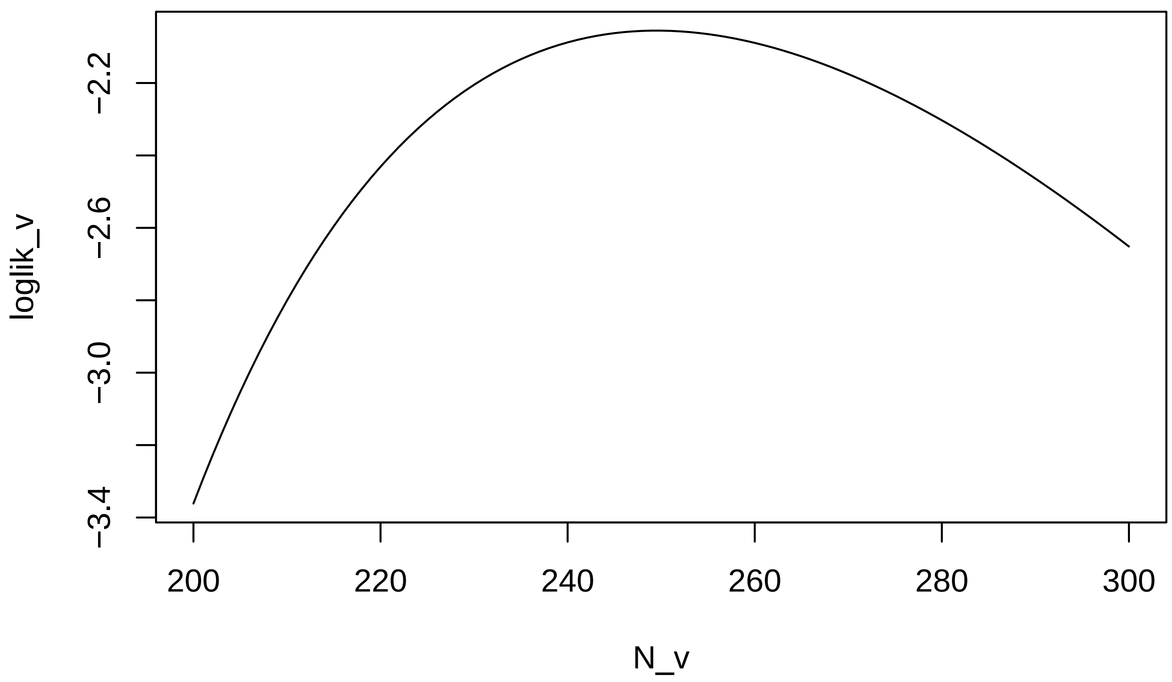
```
opt = optimize(loglik, c(0, 10), maximum = TRUE)
theta_hat_mle1 = opt$maximum
theta_hat_mle1
```

```
## [1] 7.739999
```

4.2.0.12 Exercise

b.

```
loglik = function(N){
  X = 20
  result = dhyper(X, m = 100, n = N-100, k = 50, log = TRUE)
  return(result)
}
N_v = seq(200, 300, 1)
loglik_v = loglik(N_v)
plot(N_v, loglik_v, type = 'l')
```



```
N_hat_mle = N_v[which.max(loglik_v)]
N_hat_mle
```

```
## [1] 249
```

c.

```
opt = optimize(loglik, c(200, 300), maximum = TRUE)
```

```
## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced
```

```
## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value
```

```
## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced
## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value

## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced

## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value

## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced

## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value

## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced

## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value

## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced

## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value

## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced

## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value

## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced

## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value

## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced

## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value

## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced

## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value

## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced

## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value

## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced
## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value

## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced

## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value

## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced

## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value

## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced

## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value

## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced

## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value

## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced

## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value

## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced

## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value

## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced

## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value

## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced

## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value

## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced

## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value

## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced
## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value
```

```
## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced
```

```
## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value
```

```
## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced
```

```
## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value
```

```
## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced
```

```
## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value
```

```
## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced
```

```
## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value
```

```
## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced
```

```
## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value
```

```
## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced
```

```
## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value
```

```
## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced
```

```
## Warning in optimize(loglik, c(200, 300), maximum = TRUE): NA/Inf replaced by
## maximum positive value
```

```
## Warning in dhyper(X, m = 100, n = N - 100, k = 50, log = TRUE): NaNs produced
```

```r
N_hat_mle1 = opt$maximum
N_hat_mle1
```

```
## [1] 299.9999
```

4.2.0.14 Exercise

```
x=c(983,1041,1026,1025,1033,968,1028)
loglik=-Inf
for (m in seq(900,1100,0.1)){
  for (s in seq(10,40,0.1)){
     tmp=sum(log(dnorm(x,m,s)))  # compute the loglikelihood by summing
     if (tmp>loglik){
        loglik=tmp
        mu=m
        sigma=s
     }
  }
}
mu
```

```
## [1] 1014.9
```

```
sigma
```

```
## [1] 25.7
```

4.2.0.15 Exercise

```
x=c(983,1041,1026,1025,1033,968,1028)
ll = function(theta,x) {         # We define an R function that computes the log likelihood
   mu=theta[1]
   sigma=theta[2]
   -sum(log(dnorm(x,mu,sigma)))  # compute loglikelihood by summing
}                                # add a minus sign, because optim *minimizes*.
optim(c(1000,25),ll,x=x,method="BFGS")
```

```
## $par
## [1] 1014.86313   25.69854
##
## $value
## [1] 32.65757
##
## $counts
## function gradient
##       21       18
##
## $convergence
## [1] 0
##
## $message
## NULL
```