Homework4(a)

2023-04-02

Read the lifetime data:

```
shock.absorber <- read.table("ShockAbsorber.txt", header = T)
shock.absorber$Failure <- ifelse(shock.absorber$Status == "Failure", 1, 0)
head(shock.absorber)
```

```
##   Distance     Mode   Status Failure
## 1     6700    Mode1  Failure       1
## 2     6950 Censored Censored       0
## 3     7820 Censored Censored       0
## 4     8790 Censored Censored       0
## 5     9120    Mode2  Failure       1
## 6     9660 Censored Censored       0
```

Using the location and scale parameters

```
neg.log.likelihood.lnorm <- function(pars, data){
  mu <- pars[1]
  sigma <- pars[2]
  n <- length(data[,1])
  time <- data[,1]
  ind.F <- data[,4]

  if (sigma > 0){
    lj <- ind.F * dlnorm(time, mu, sigma, log = TRUE) +
      (1-ind.F) * log(1-plnorm(time, mu, sigma))
    return(-sum(lj))
  } else {
    return(10^10)
  }
}
```

Initial Guess and Estimation

```
mu0 <- mean(log(shock.absorber$Distance))
sig0 <- var(log(shock.absorber$Distance))^0.5

initial <- c(mu0, sig0)
par.hat.lnorm <- optim(initial, neg.log.likelihood.lnorm,
                         data = shock.absorber,
                         hessian = TRUE)
par.hat.lnorm$par
```

```
## [1] 10.1445504  0.5298805
```

```
par.hat.lnorm$value
```

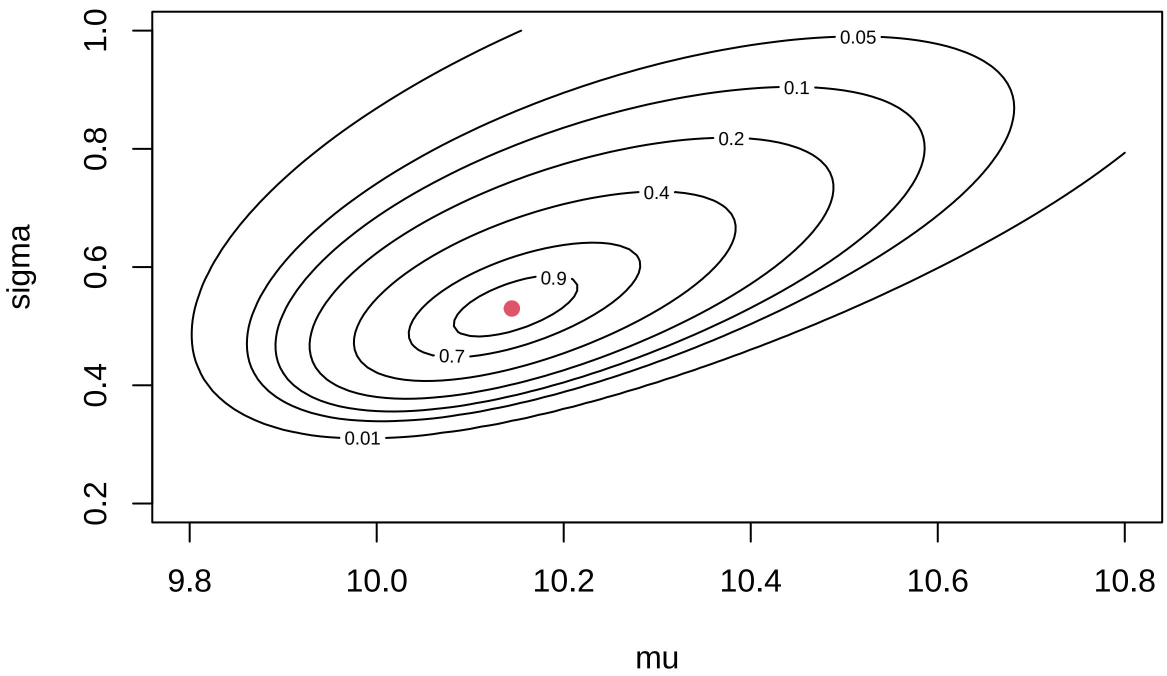
```
## [1] 124.6086
```

Check contour plot

```
mu0seq <- seq(9.8, 10.8, 0.01)
sig0seq <- seq(0.2, 1.0, 0.01)

l.value <- matrix(NA, length(mu0seq), length(sig0seq))
for(i in 1:length(mu0seq)){
  for(j in 1:length(sig0seq)){
    l.value[i,j] <- exp(-neg.log.likelihood.lnorm(c(mu0seq[i], sig0seq[j]),
                                                    shock.absorber))
  }
}

contour(mu0seq, sig0seq, l.value/exp(-par.hat.lnorm$value),
        levels = c(0.9, 0.7, 0.4, 0.2, 0.1, 0.05, 0.01),
        xlab = "mu", ylab = "sigma")
points(par.hat.lnorm$par[1], par.hat.lnorm$par[2], col = 2, pch = 19)
```



Observed information

```
hessian.lnorm <- par.hat.lnorm$hessian
cov.lnorm <- solve(hessian.lnorm)
corr.lnorm <- cov2cor(cov.lnorm)
rho.lnorm <- corr.lnorm[1,2]
cov.lnorm
```

```
##             [,1]        [,2]
## [1,] 0.020756975 0.009718164
## [2,] 0.009718164 0.012674546
```

Confidence Intervals for the Parameters

```
est <- round(c(par.hat.lnorm$par[1], par.hat.lnorm$par[2]), 4)
se <- round(c(cov.lnorm[1,1]^0.5, cov.lnorm[2,2]^0.5), 4)
CIs <- data.frame(matrix(NA, 2, 6))
colnames(CIs) <- c("parameter", "est", "se", "trans", "lower", "upper")

# Normal-Approximate CI for mu
log.tr.CIs <- function(est, se.est, alpha = 0.05, parameter = "theta"){
  w <- exp(qnorm(1-alpha/2)*se.est/est)
  lower <- round(est/w, 4)
  upper <- round(est*w, 4)
  res <- data.frame(parameter = parameter,
                    est = est,
                    se = se.est,
                    trans = "log-trans",
                    lower = lower,
                    upper = upper)
  return(res)
}
no.tr.CIs <- function(est, se.est, alpha = 0.05, parameter = "theta"){
  lower <- est - qnorm(1-alpha/2)*se.est
  upper <- est + qnorm(1-alpha/2)*se.est
  res <- data.frame(parameter = parameter,
                    est = est,
                    se = se.est,
                    trans = "no-trans",
                    lower = lower,
                    upper = upper)
  return(res)
}

CIs[1, ]<- no.tr.CIs(est[1], se[1], parameter = "mu")
CIs[2, ]<- log.tr.CIs(est[2], se[2], parameter = "sigma")
```

Matrix formulation

```
delta.M <- function(c, cov.M){
  matrix(c, nrow = 1) %*% cov.M %*% t(matrix(c, nrow = 1))
}
```

Estimation of tp

```
tp <- function(mu, sig, p){
  Phi.inv <- round(qnorm(p), 4)
  return(exp(mu + sig * Phi.inv))
```

```
}
est.tp.lnorm <- tp(CIs$est[1], CIs$est[2], p = 0.1)
```

Standard error of tp

```
c <- est.tp.lnorm * c(1, qnorm(0.1))
var.tp.lnorm <- delta.M(c, cov.lnorm)
se.tp.lnorm <- sqrt(var.tp.lnorm)
```

CI of tp

```
log.tr.CIs(est.tp.lnorm, se.tp.lnorm, parameter = "t_0.1")
```

```
##   parameter      est       se     trans    lower    upper
## 1     t_0.1 12906.42 1666.112 log-trans 10021.27 16622.21
```

Estimation of F(t0)

```
Ft0 <- function(mu, sig, t0){
  return(round(plnorm(t0, mu, sig), 4))
}
est.F.lnorm <- Ft0(CIs$est[1], CIs$est[2], t0 = 10000)
```

Standard error of F(t0)

```
t0 <- 10000
z <- (log(t0) - CIs$est[1])/CIs$est[2]
c <- -t0*dlnorm(t0, CIs$est[1], CIs$est[2])*c(1, z)
var.Ft0.lnorm <- delta.M(c, cov.lnorm)
se.Ft0.lnorm <- sqrt(var.Ft0.lnorm)
```

CI of F(t0)

```
logit.tr.CIs <- function(est, se.est, alpha = 0.05, parameter = "theta"){
  w <- exp(qnorm(1-alpha/2)*se.est/(est*(1-est)))
  lower <- round(est/(est+(1-est)*w), 4)
  upper <- round(est/(est+(1-est)/w), 4)
  res <- data.frame(parameter = parameter,
                    est = est,
                    se = se.est,
                    trans = "logit-trans",
                    lower = lower,
                    upper = upper)
  return(res)
}
CIs[3, ]<- log.tr.CIs(est.tp.lnorm, se.tp.lnorm, parameter = "t_0.1")
CIs[4, ]<- logit.tr.CIs(est.F.lnorm, se.Ft0.lnorm, parameter = "Ft10000")
```

Homework(a)

```
CIs
```

```
##   parameter        est           se       trans        lower       upper
## 1        mu    10.1446 1.441000e-01    no-trans     9.862169    10.42703
## 2     sigma     0.5299 1.126000e-01   log-trans     0.349400     0.80370
## 3     t_0.1 12906.4202 1.666112e+03   log-trans 10021.269500 16622.21370
## 4   Ft10000     0.0389 2.560147e-02 logit-trans     0.010500     0.13410
```