SISMID Module 6: Stochastic Epidemic Models with Inference – Solutions to Exercise Session 2

July 13, 2021

Exercise 2.1 (Estimation of R0)

Assume that a large outbreak occurs in a homogeneously mixing population. (a) First assuming there is no preventive measures, estimate the R0 if there were 20% infected during the outbreak.

```
R0_a <- function(tau){
-log(1-tau)/tau
}
R0_hat <- R0_a(tau=0.2)
R0_hat
```

```
## [1] 1.115718
```

(b) Suppose now there were a fraction 30% of initially immune. Estimate R0 in this case.

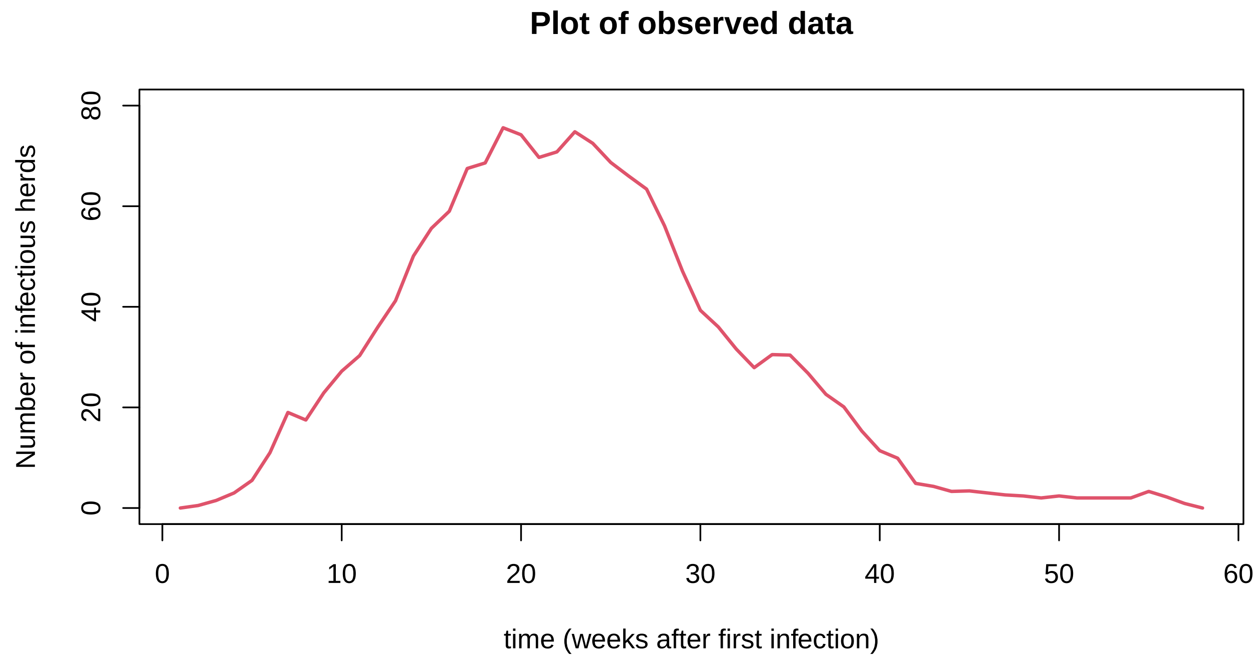
```
# fraction r of initially immune
r <- 0.3
R0_b <- function(tau){
-log(1-tau)/(tau*(1-r))

}
# Note that the over all fraction infected equals tau * (1-r)!

tau_overall <- 0.2
R0_hat <- R0_b(tau=tau_overall/(1-r))
R0_hat
```

```
## [1] 1.682361
```

Exercise 2.2 (Least-squares Fit the SIR model)

(a) First define the log-likelihood function.

```
# plot the observed data
csfv <- read.table("csfv.txt",col.names=c("t","I"))

plot(csfv$t,csfv$I,lwd=2,type="l",lty=1,ylim=c(0,80),
     ylab="Number of infectious herds",xlab="time (weeks after first infection)",col=2)
title("Plot of observed data")
```




```r
# Netherlands has approximately 21 500 pig herds
# we know 429 infected herds detected and stamped out
N <- 21500
sum_y <- 429
f_size <- sum_y/N
R0_hat <- -log(1-f_size)/f_size

######################################################################
# Deterministic SIR Model
######################################################################

deter_sir <- function(t,y, parms) {
  beta <- parms[1]
  gamma <- parms[2]
  S <- y[1]
  I <- y[2]
  return(list(c(S=-(beta*S*I)/N,I=(beta*S*I)/N-gamma*I)))
}

######################################################################
# log-likelihood function
######################################################################
ll.gauss <- function(theta) {
  #Solve ODE using the parameter vector theta
  res <- lsoda(y=c(N-1,1), times=csfv$t, func=deter_sir, parms=exp(theta))
    return(sum(dnorm(csfv$I,mean=(res[,3]),sd=1,log=TRUE)))
}
```

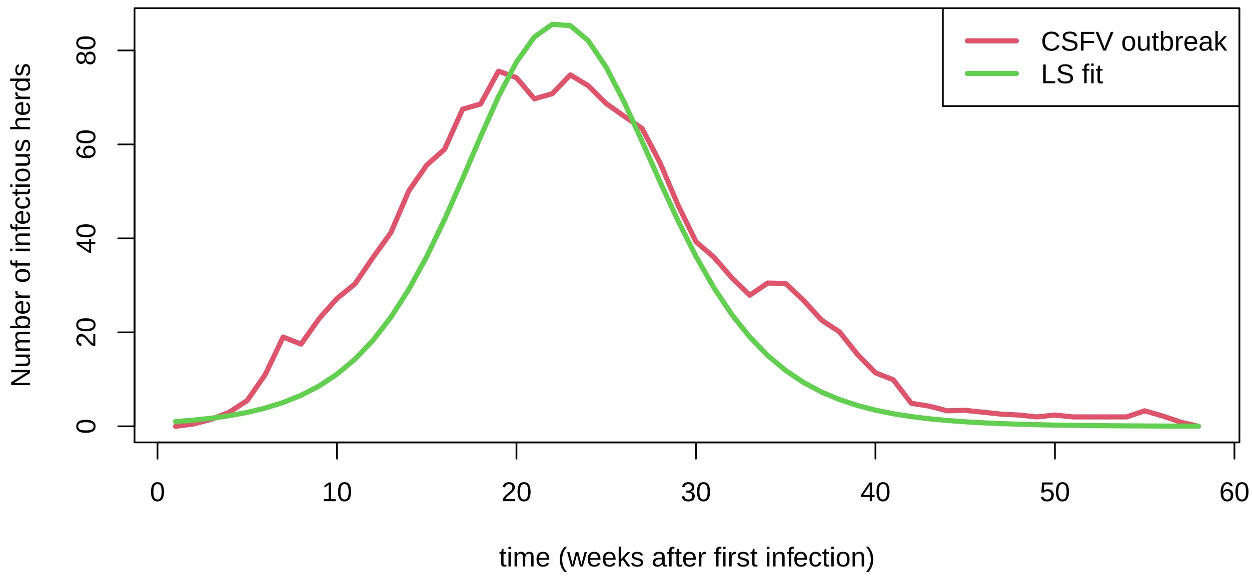
(b) Maximize the log-likelihood using *optim* and compute the MLE.

```r
# Determine MLE
mle <- optim(log(c(3,3)), fn=ll.gauss, control=list(fnscale=-1))

#Show estimates and resulting R0 estimate
```

```
beta.hat <- exp(mle$par)[1]
gamma.hat <- exp(mle$par)[2]
R0.hat <- beta.hat/gamma.hat
```

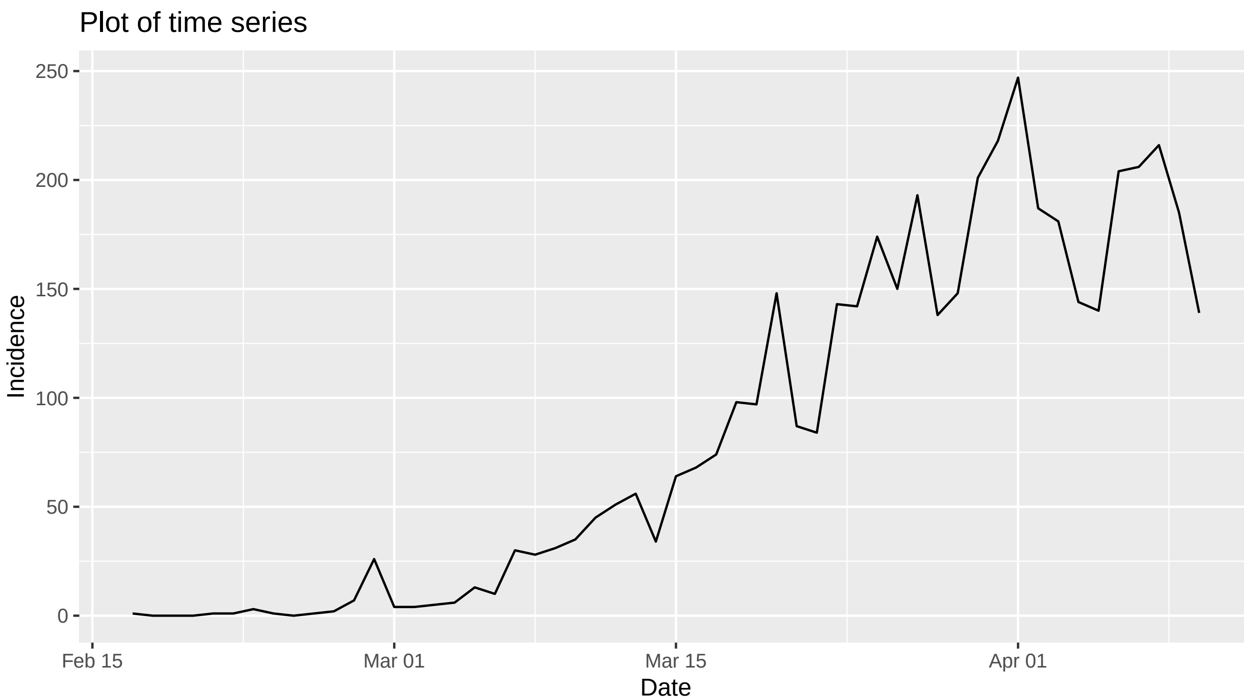
(c) Inserting the values of MLE to find the solution of SIR differential equation system. And plot the fitted curve and the real data of CSFV outbreak together. Does it fit well?

```
mu <- lsoda(y=c(N-1,1), times=csfv$t, func=deter_sir,parms=exp(mle$par))
#head(mu, n=3)
matplot(mu[,1],cbind(csfv$I,mu[,3]),type="l",lwd=3,lty=1,
        ylab="Number of infectious herds",
        xlab="time (weeks after first infection)",col=c(2,3))
legend(x="topright",c("CSFV outbreak","LS fit"), lty=1,col=c(2,3),lwd=3)
```



Exercise 2.3 (Least-squares Fit the SEIR model)

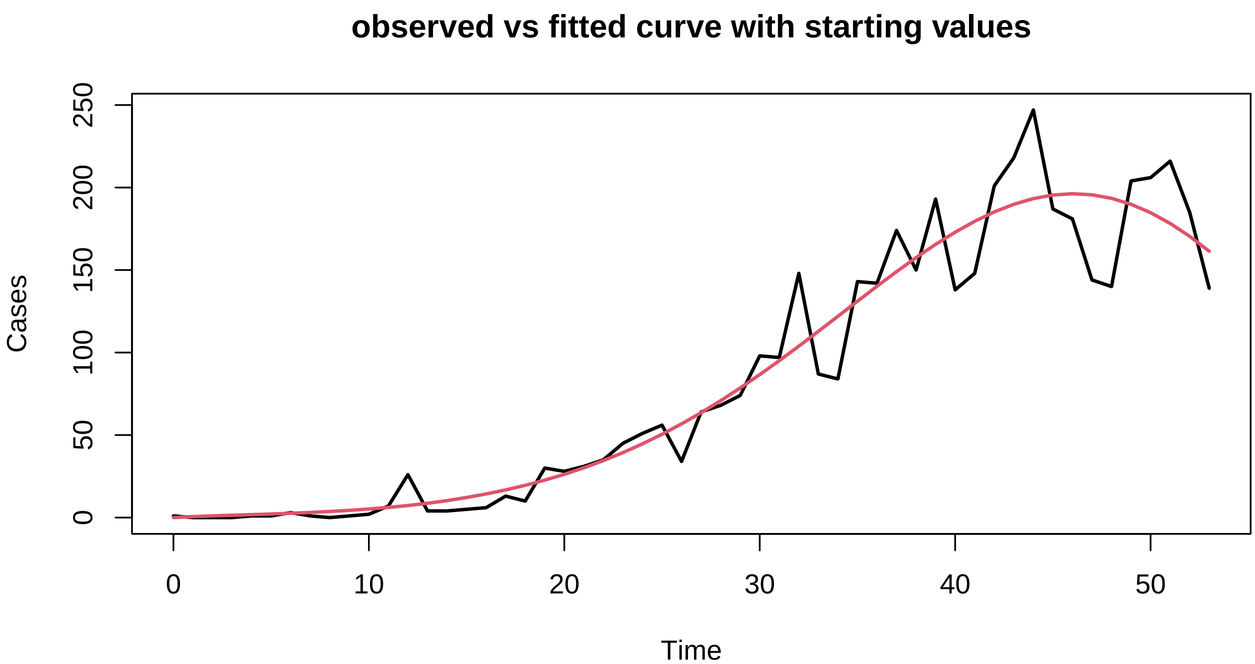
```
N <-  2374550
I0 <- 1
######################################################################
# plot of time series
######################################################################
ts <- read_delim("Data_2020-04-10Ny.txt", delim=" ") %>%
  mutate(t=as.numeric(Date - min(Date)))
ggplot(ts, aes(x=Date, y=Incidence)) + geom_line() +
  ggtitle(expression("Plot of time series"))
```



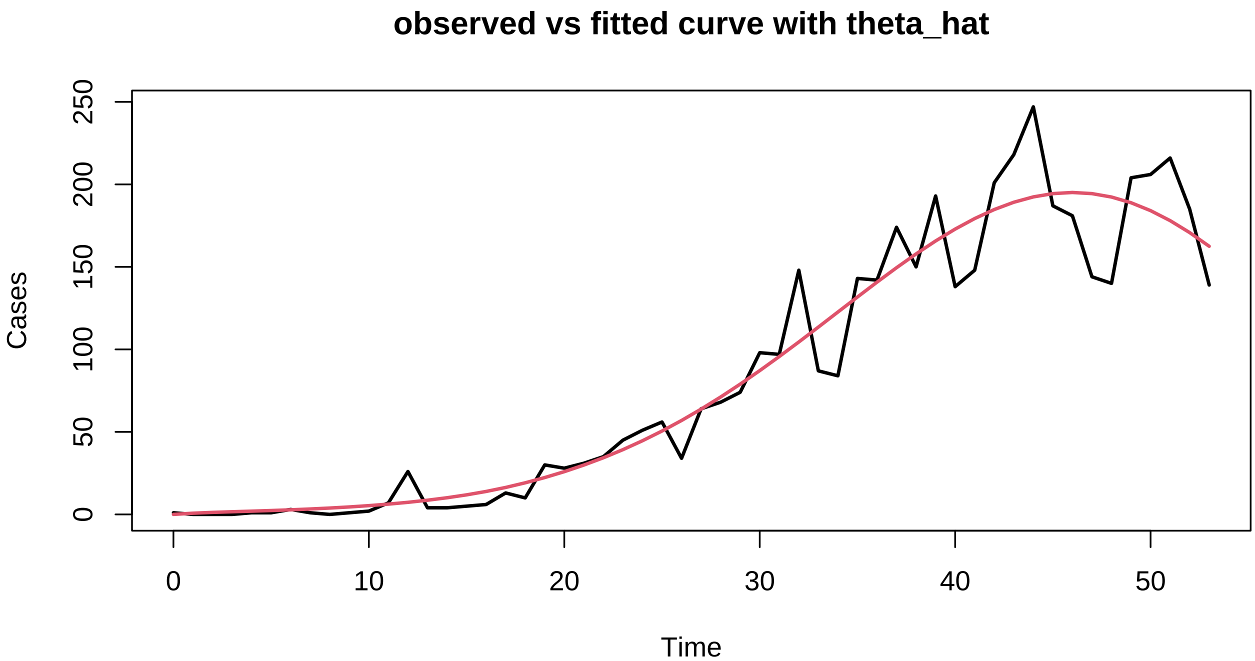

```
#######################################################################
# deterministic SEIR model
#######################################################################

seir_change <- function(t, y, parms) {
  beta0 <- parms[1]
  beta1 <- parms[2]
  t1 <- parms[3]
  w <- parms[4]
  rho <- 1/5
  gamma <- parms[5]

  S <- y[1]
  E <- y[2]
  I <- y[3]

  # time-dependent rate \beta(t):

  beta <- function(t) {ifelse( t <= t1-w, beta0, ifelse(t > t1+w, beta1,
                                    beta0 + (beta1-beta0)/(2*w)*(t-(t1-w))))
  }

  return(list(c(S=-(beta(t)*S*I)/N, E=(beta(t)*S*I)/N-rho*E, I=rho*E - gamma*I)))
}
```

```
#######################################################################
# Least-squares fit
#######################################################################

ll.sq <- function(theta, I0) {
  "
  This function takes a vector theta (consisting of the parameters beta0,
```

```
  beta1, t1, w, gamma) as the input,
  and it solves the ODE system (of the SEIR model) using the exponential
  values of the parameters.
  The function returns the log-likelihood values.
  "
  res <- lsoda(y=c(N-I0,0,I0), times=ts$t, func=seir_change, parms=exp(theta))
  sum( (ts$Incidence - res[,3])^2)
}
# set starting values: (beta_0,beta_1, t_1, w,gamma)
# Note: hand-tuning the starting values of your optimization
# might be necessary in order to get a reasonable fit.
# Try out more starting values.

theta0 <- c(0.710204914, 0.001676203, 38.156570230, 23.796745549,  0.212658688)

mu <- lsoda(y=c(N-I0,0,I0), times=ts$t, func=seir_change, parm=theta0)
matplot(mu[,1],cbind(ts$Incidence,mu[,3]),type="l",lwd=2,lty=1,ylab="Cases", xlab="Time")
title("observed vs fitted curve with starting values")
```




```
theta_hat <- optim(log(theta0), fn=ll.sq,
             method="Nelder-Mead",
             I0=1)

# plug-in of theta_hat:

mu <- lsoda(y=c(N-I0,0,I0), times=ts$t, func=seir_change, parms=exp(theta_hat$par))
matplot(mu[,1],cbind(ts$Incidence,mu[,3]),type="l",lwd=2,lty=1, ylab="Cases", xlab="Time")
title("observed vs fitted curve with theta_hat")
```




```
ts_df <- ts %>% mutate(fitted = mu[,3]) %>%
  pivot_longer(cols=c("fitted","Incidence"), names_to="type", values_to="No.")
ggplot(ts_df, aes(x=Date, y=No., color=type)) +
  geom_line() + ylab("Cases") + xlab("Date") +
  scale_color_brewer(palette="Set2") +
  theme_minimal()
```