HW2

Weixing Gu

February 7, 2018

Abstract

I create several functions by Newton-Raphson Method, Fixed-point Iteration, Fisher scoring, Gauss-Newton Approach and so on to calcualte the required theta and other data asked. Many of them contains stopping or updating rule to get the most ideal value I need.

1(a)

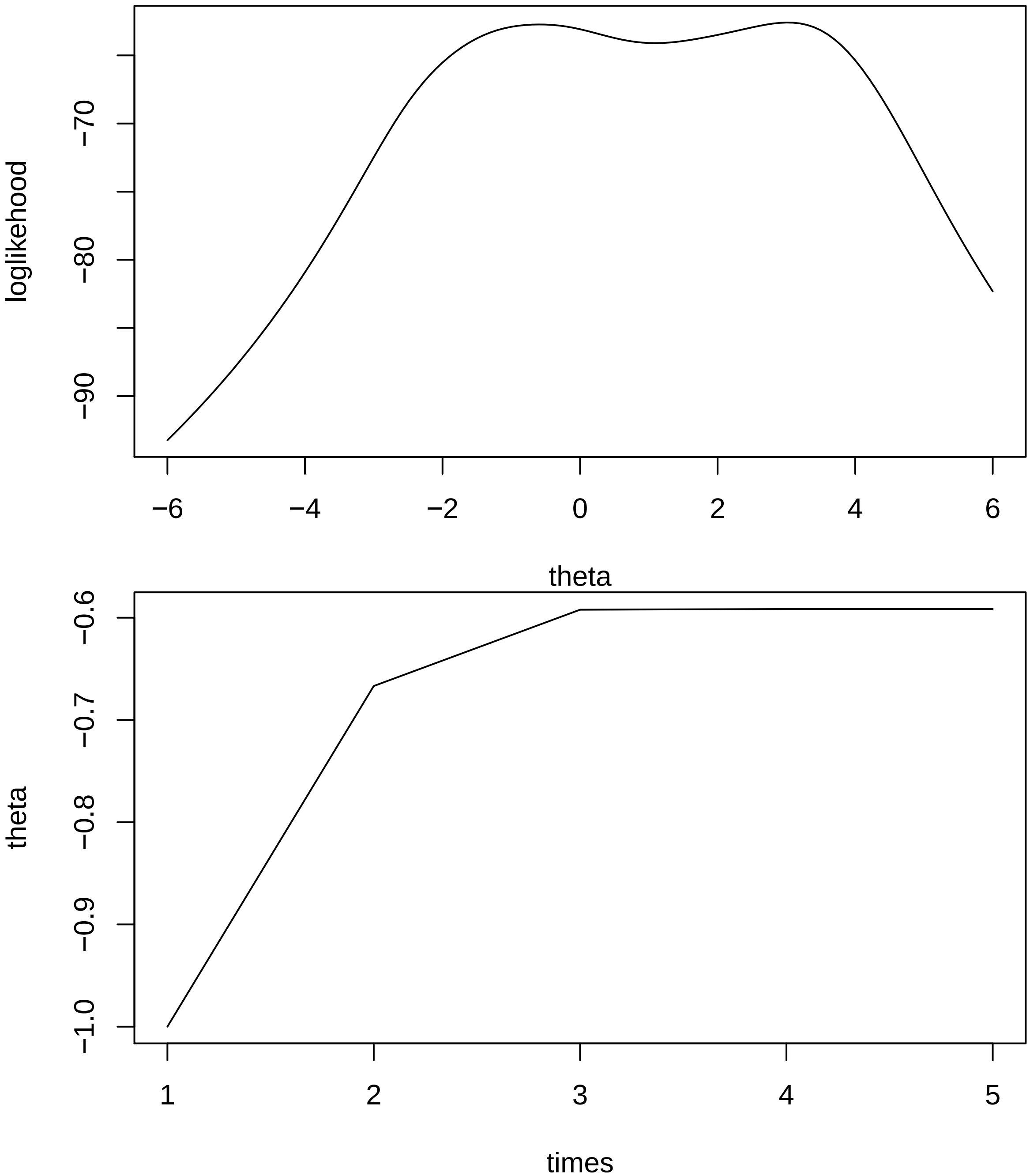
$$
\begin{aligned}
&l(\theta) && (1)\\
&= \ln \prod_{i=1}^{n} \frac{1}{\pi[1+(x-\theta)^2]} && (2)\\
&= \sum_{i=1}^{n} \ln \frac{1}{\pi[1+(x-\theta)^2]} && (3)\\
&= \sum_{i=1}^{n} [\ln \frac{1}{\pi} + \ln \frac{1}{1+(x-\theta)^2}] && (4)\\
&= -n \ln \pi - \sum_{i=1}^{n} \ln[1+(x-\theta)^2] && (5)\\
& && (6)
\end{aligned}
$$

$$
\begin{aligned}
&l'(\theta) && (7)\\
&= 0 - \sum_{i=1}^{n} \frac{2(\theta - x_i)}{1+(\theta-x_i)^2} && (8)\\
&= -2 \sum_{i=1}^{n} \frac{\theta - x_i}{1+(\theta - x_i)} && (9)\\
& && (10)
\end{aligned}
$$

$$
\begin{aligned}
&l''(\theta) && (11)\\
&= -2 \sum_{i=1}^{n} \frac{1+(\theta-x_i)^2 - 2(\theta-x_i)(\theta-x_i)}{[1+(\theta-x_i)^2]^2} && (12)\\
&= -2 \sum_{i=1}^{n} \frac{1-(\theta-x_i)^2}{[1+(\theta-x_i)^2]^2} && (13)\\
& && (14)
\end{aligned}
$$

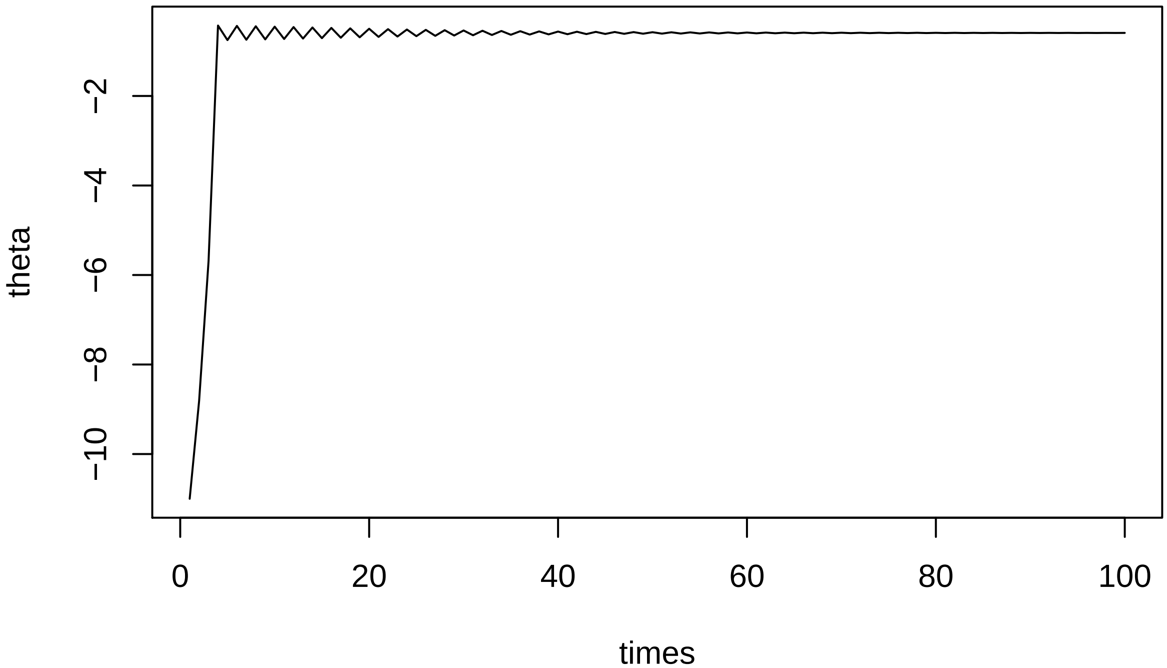
$$
\begin{align}
&I(\theta) \tag{15}\\
&= n\int \frac{\{p'(x)\}^2}{p(x)}dx \tag{16}\\
&= n\int \frac{4(x-\theta)^2}{\pi[1+(x-\theta)^2]^4} * \pi[1+(x-\theta)^2]dx \tag{17}\\
&= \frac{4n}{\pi}\int_{-\infty}^{\infty} \frac{(x-\theta)^2}{[1+(x-\theta)^2]^3}dx \tag{18}\\
&= \frac{4n}{\pi}\int_{-\infty}^{\infty} \frac{x^2}{(1+x^2)^3}dx \tag{19}\\
&= \frac{4n}{\pi}\int_{-\infty}^{\infty}[(\frac{1}{(1+x^2)^2} - \frac{1}{(1+x^2)^3})]dx \tag{20}\\
&= \frac{4n}{\pi}(\int_{-\infty}^{\infty}\frac{1}{(1+x^2)^2}dx - \int_{-\infty}^{\infty}\frac{1}{(1+x^2)^3}dx) \tag{21}\\
&= \frac{4n}{\pi}[\int_{-\infty}^{\infty}\frac{1}{(1+x^2)^2}dx - (\frac{x}{4(x^2+1)^2}|_{-\infty}^{\infty} + \frac{3}{4}\int_{-\infty}^{\infty}\frac{dx}{(x^2+1)^2})] \tag{22}\\
&= \frac{4n}{\pi}(\int_{-\infty}^{\infty}\frac{1}{4(1+x^2)^2}dx - \frac{x}{4(x^2+1)^2}|_{-\infty}^{\infty}) \tag{23}\\
&= \frac{4n}{\pi}[\frac{1}{4}(\frac{x}{2(x^2+1)}|_{-\infty}^{\infty} + \frac{1}{2}\int_{-\infty}^{\infty}\frac{1}{1+x^2}dx) - \frac{x}{4(x^2+1)^2}|_{-\infty}^{\infty}] \tag{24}\\
&= \frac{4n}{\pi}(\frac{x(x^2-1)}{8(x^2+1)^2}|_{-\infty}^{\infty} + \frac{1}{8}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}}\frac{\sec^2 t}{1+\tan^2 t}dt) \tag{25}\\
&= \frac{4n}{\pi}(0+\frac{\pi}{8}) \tag{26}\\
&= \frac{n}{2} \tag{27}\\
& \tag{28}
\end{align}
$$

1(b)

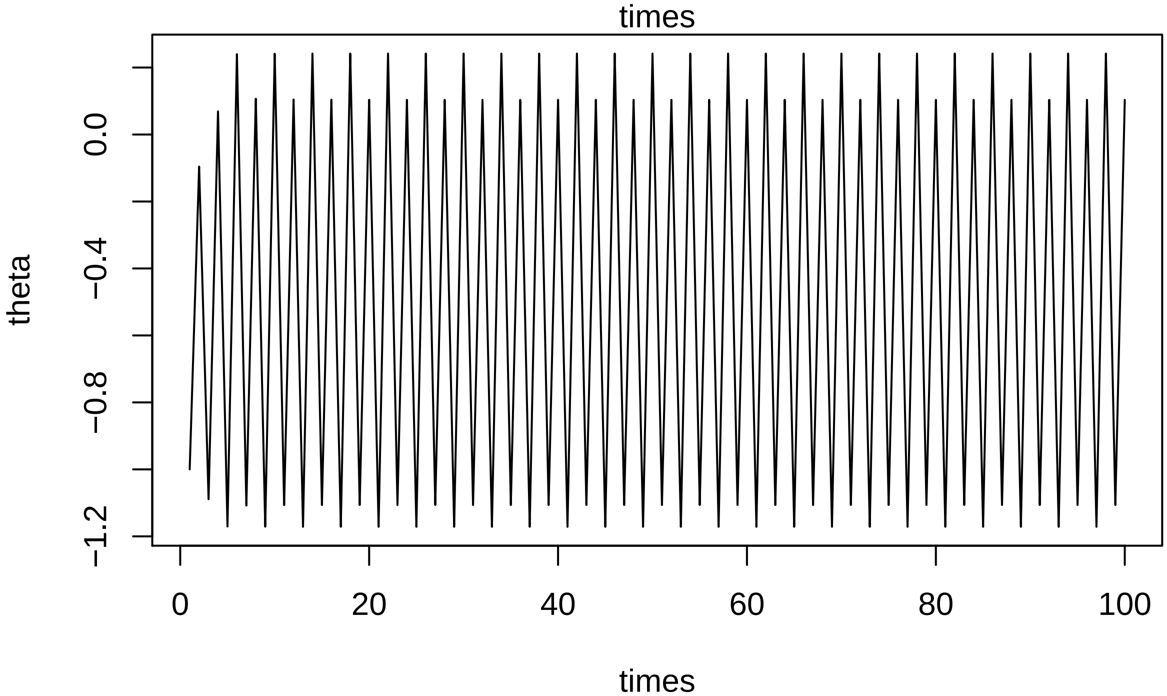


Theta's value of the graph above is -1. I tried all sample starting values, and I found there were no converging value starting from -11, 7, 8 and 38. The mean value is about 3.26, and this is a good starting point since I can find a converging value around 3.

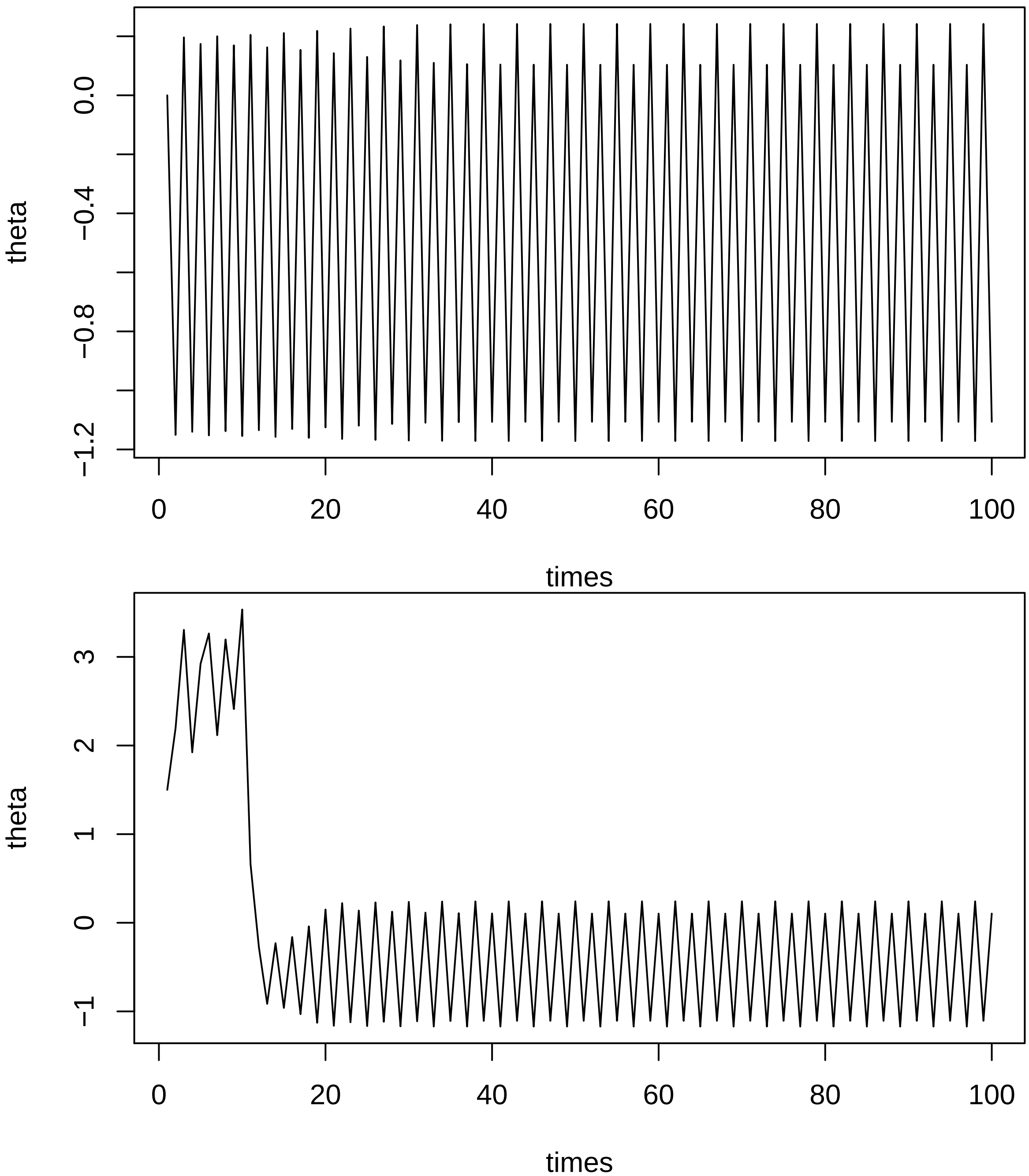
1(c)

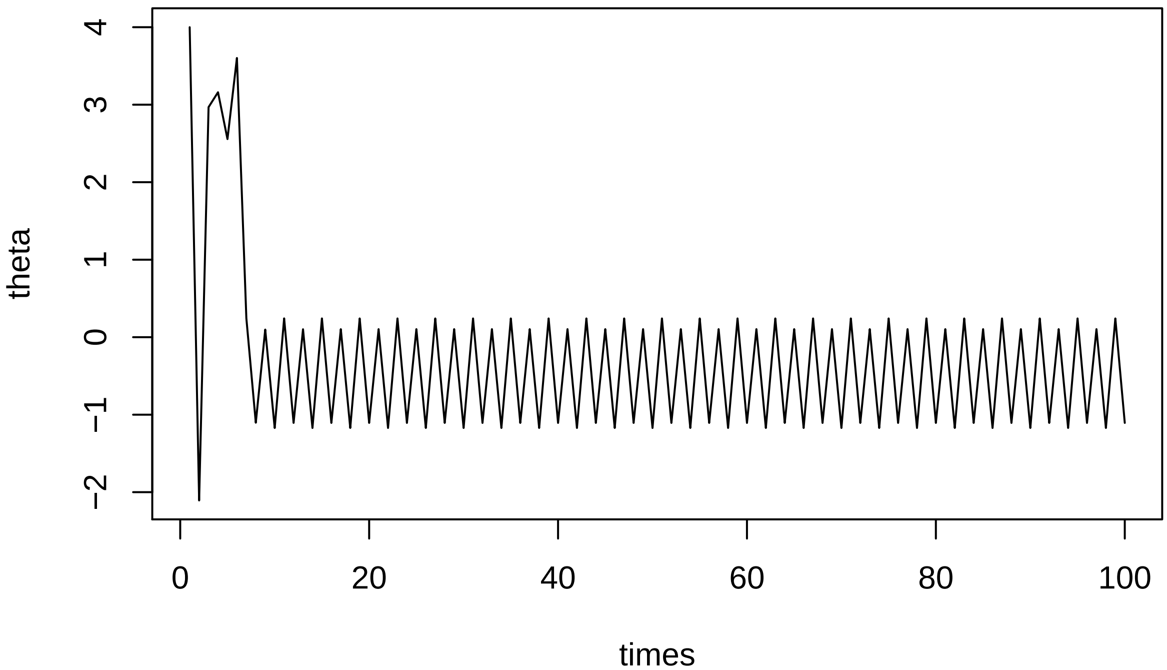


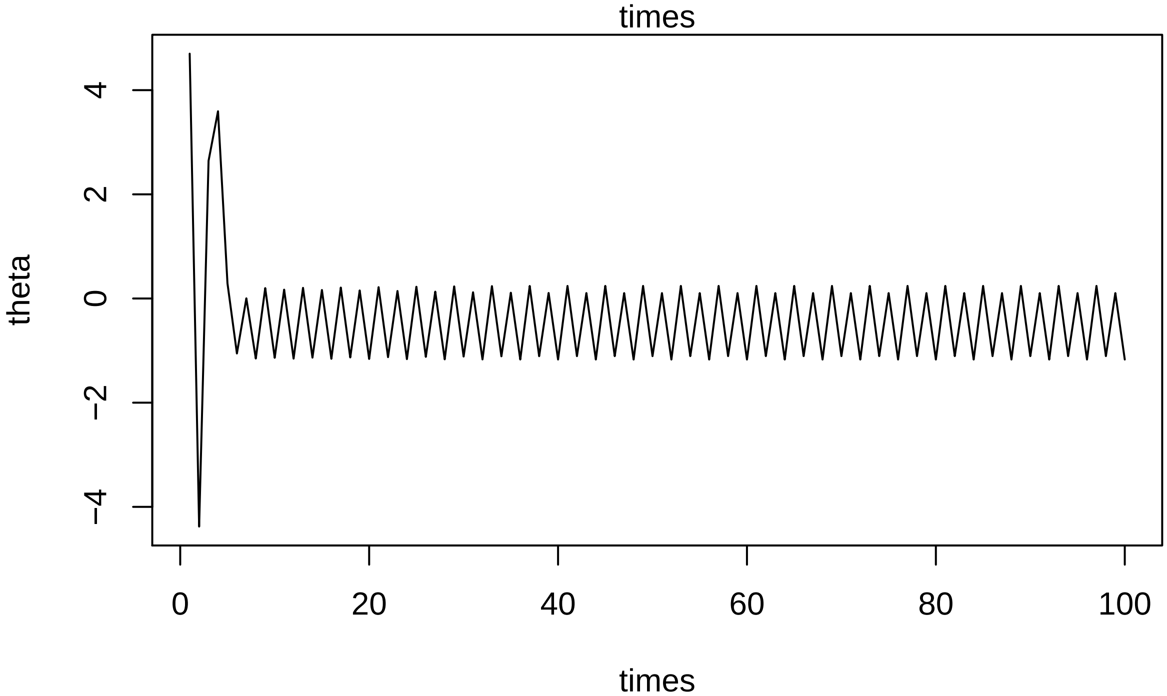
theta
−10
−8
−6
−4
−2
0
20
40
60
80
100
times

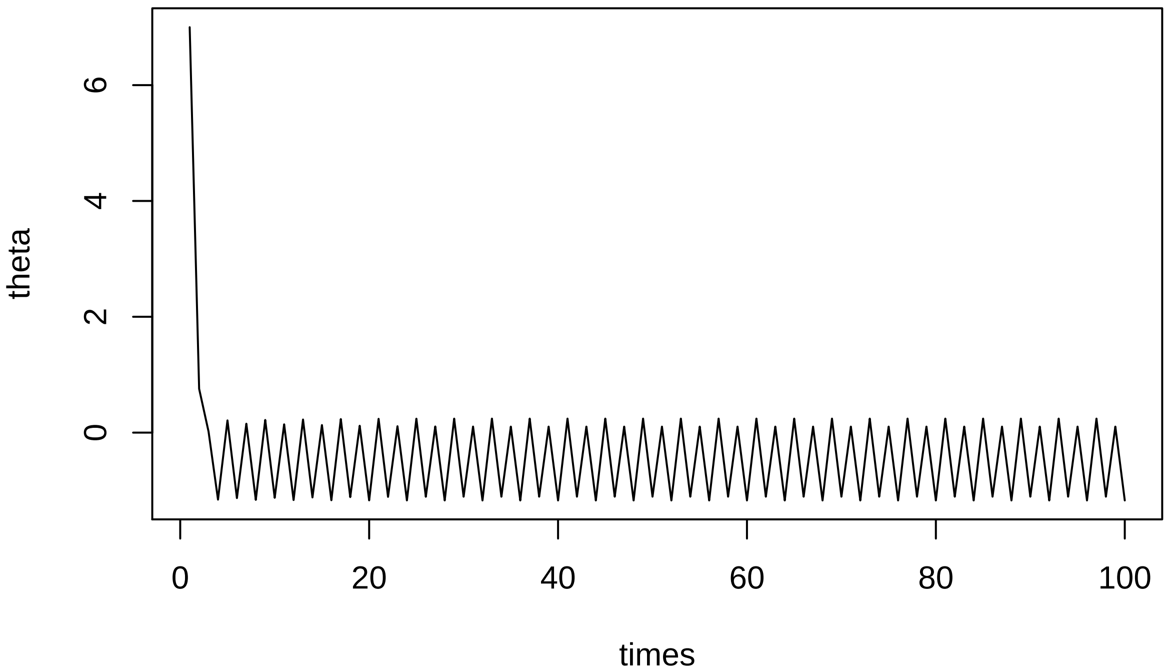


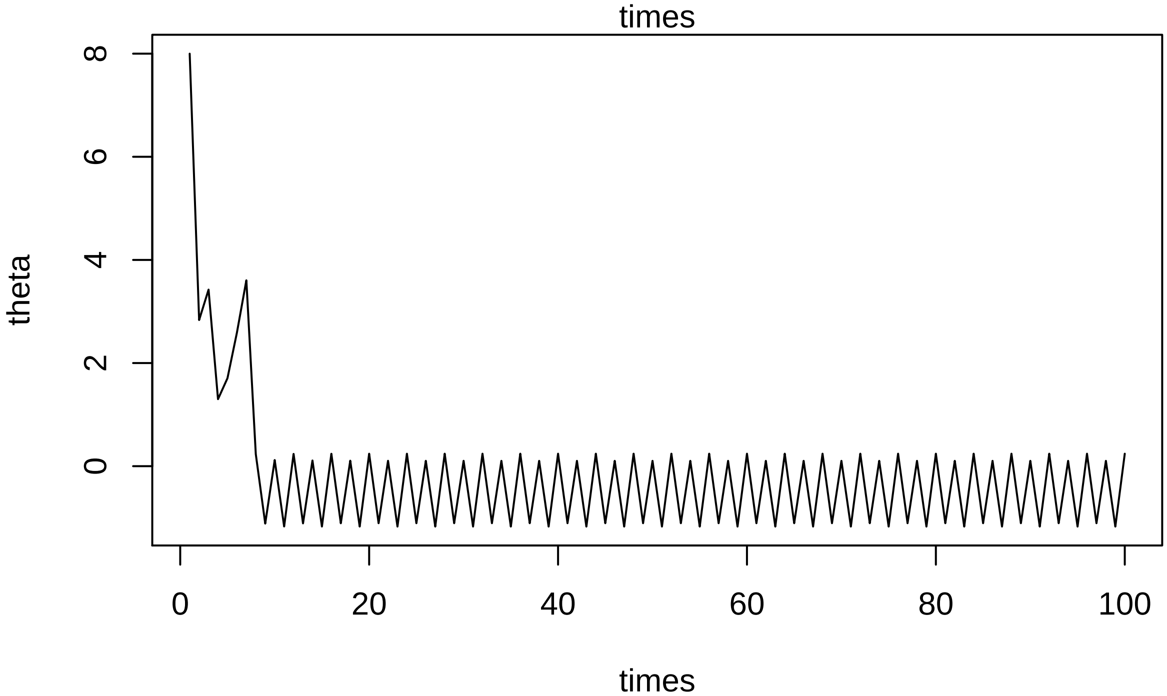
theta
−1.2
−0.8
−0.4
0.0
0
20
40
60
80
100
times

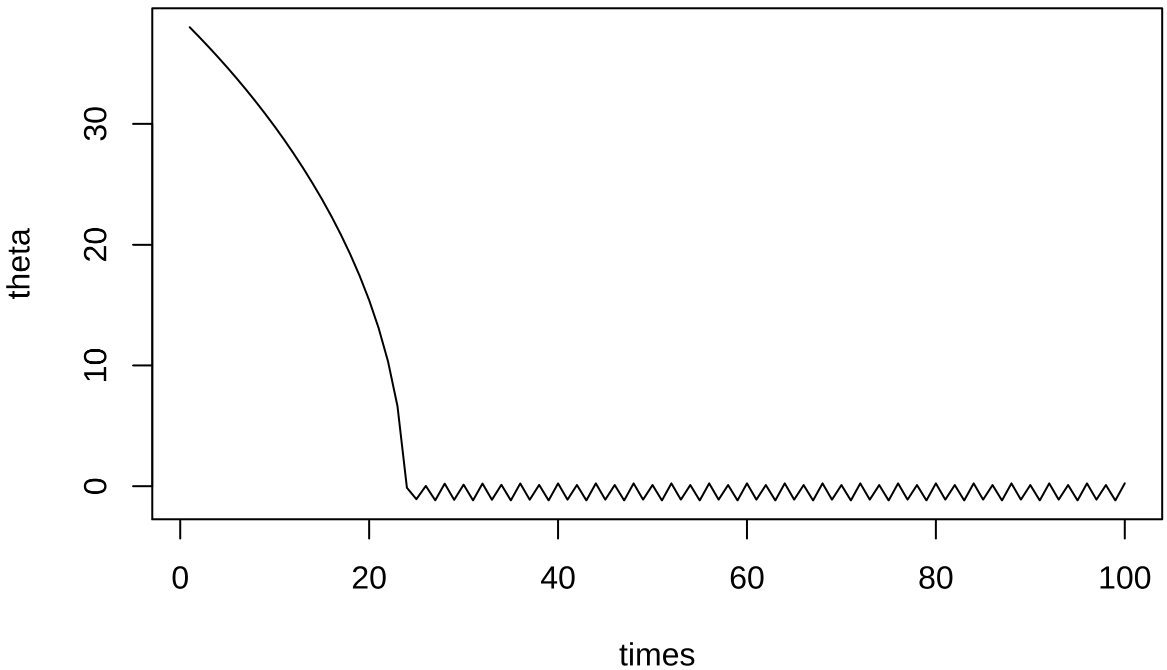

theta
0.0
−0.4
−0.8
−1.2
0
20
40
60
80
100
times
theta
3
2
1
0
−1
0
20
40
60
80
100
times

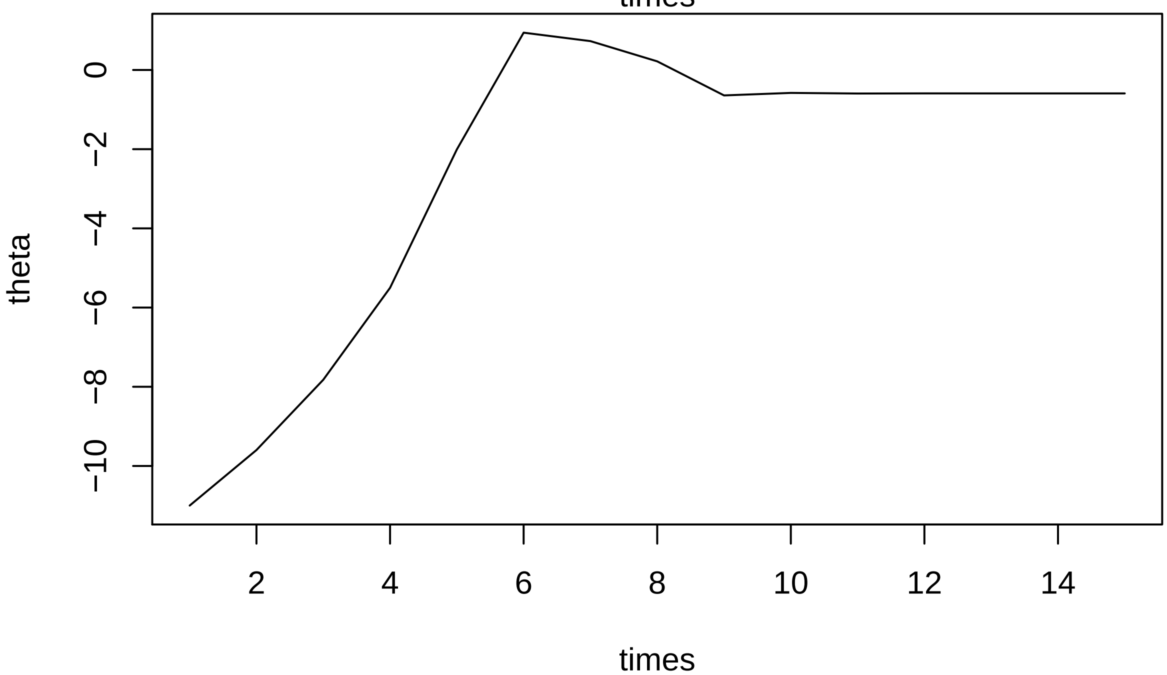

theta
times
0
20
40
60
80
100
−2
−1
0
1
2
3
4

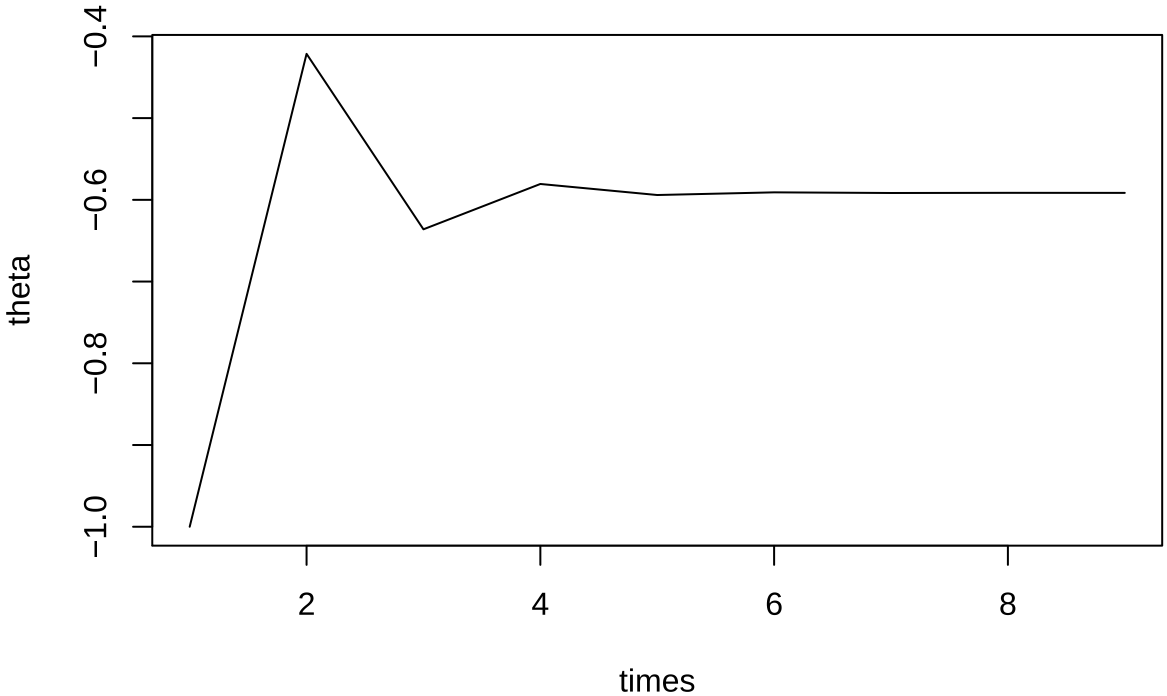

theta
times
0
20
40
60
80
100
−4
−2
0
2
4


theta
times
0
2
4
6
0
20
40
60
80
100

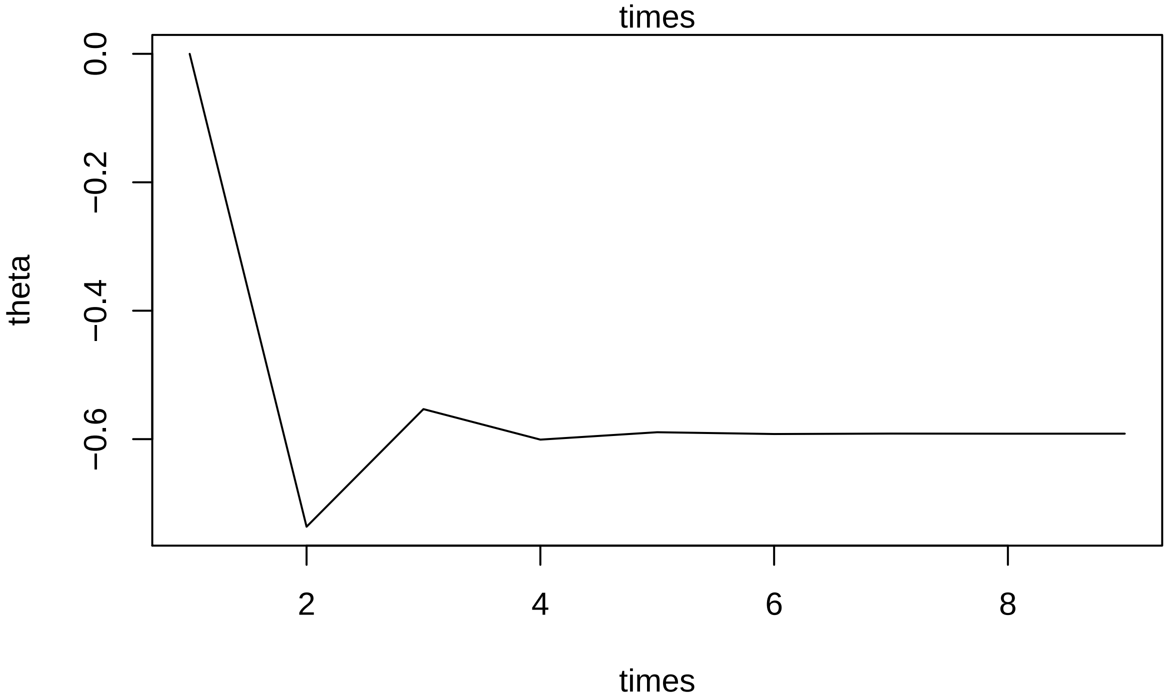

theta
times
0
2
4
6
8
0
20
40
60
80
100


theta
30
20
10
0
0
20
40
60
80
100
times

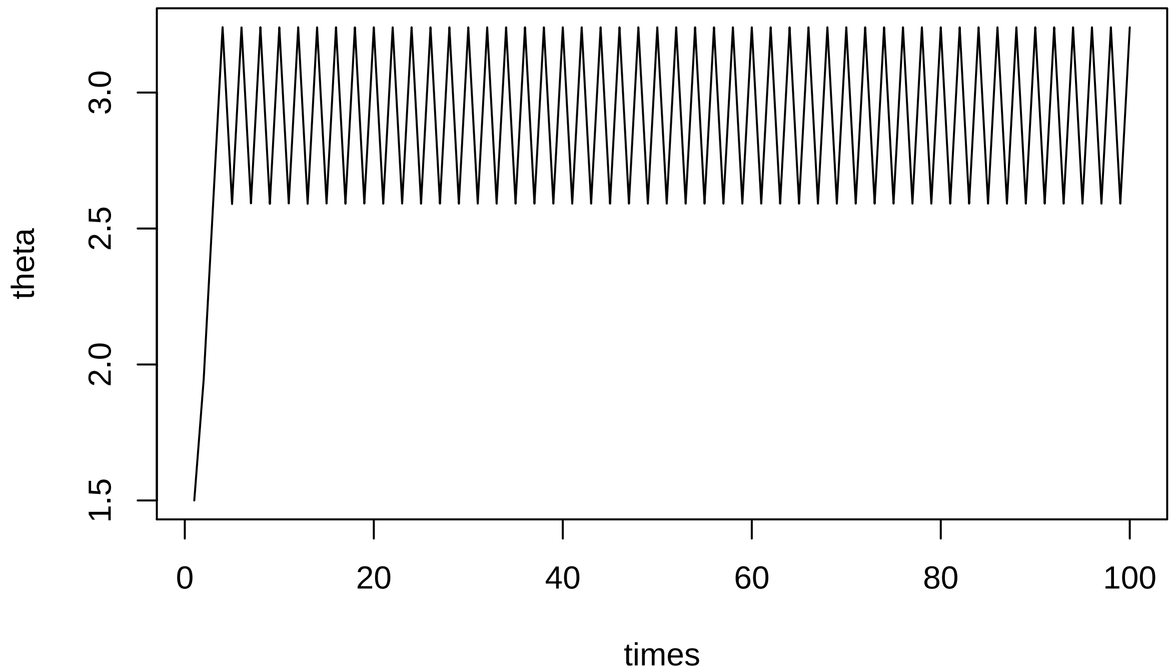

theta
0
−2
−4
−6
−8
−10
2
4
6
8
10
12
14
times

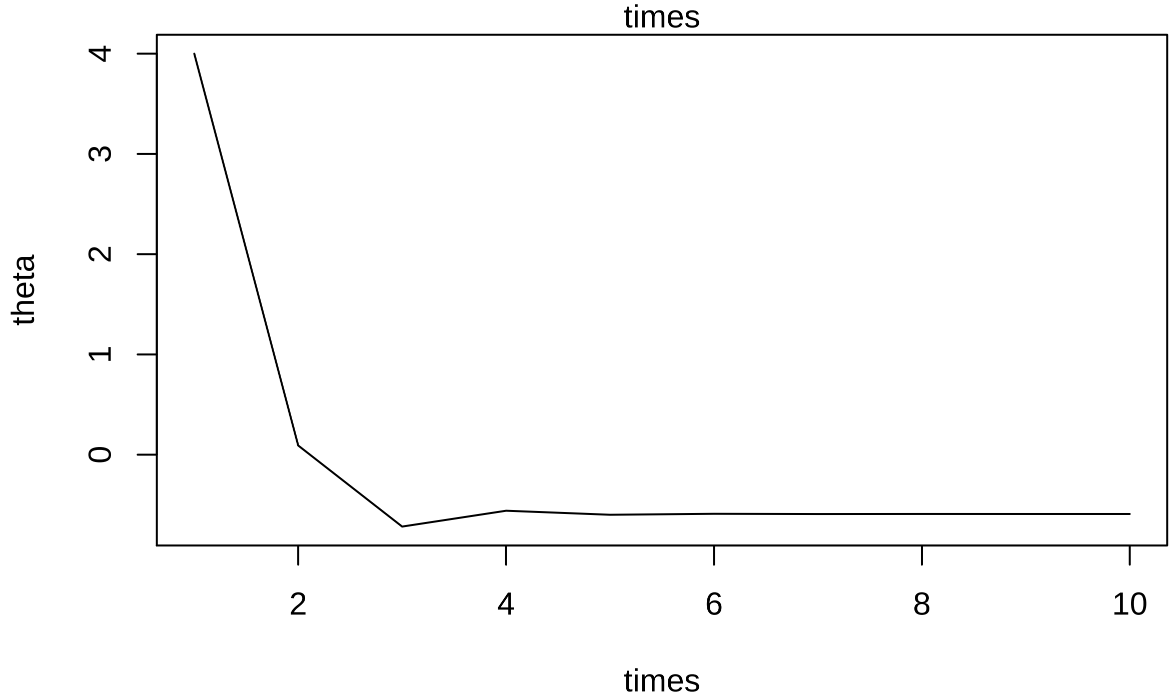


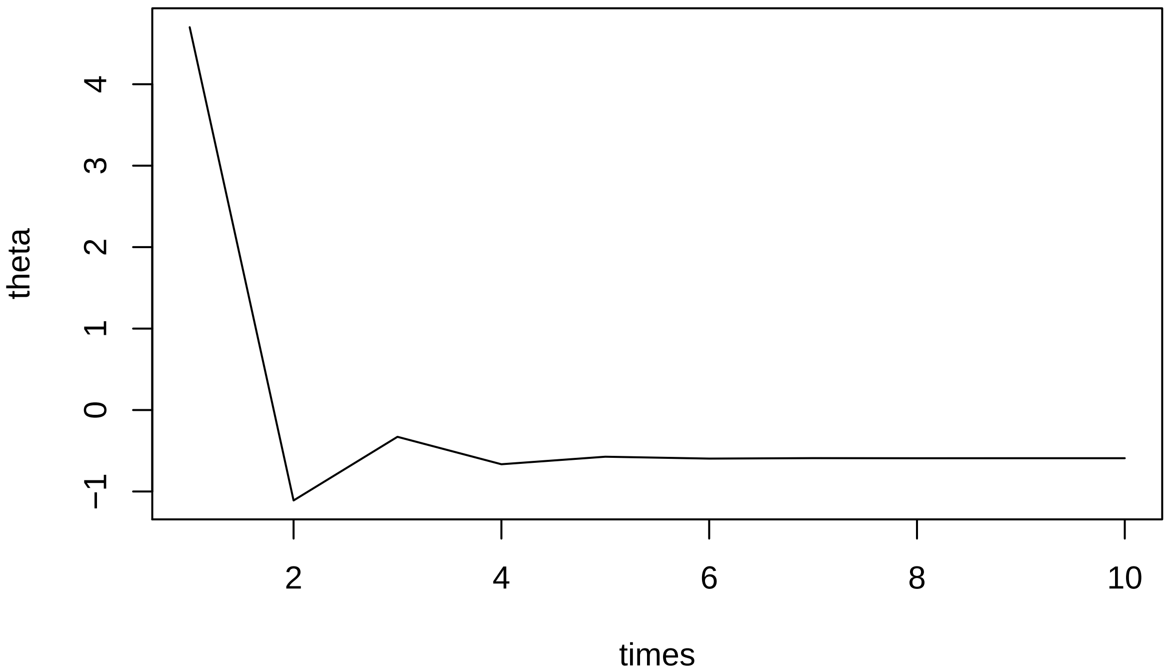
theta
−0.4
−0.6
−0.8
−1.0
2
4
6
8
times

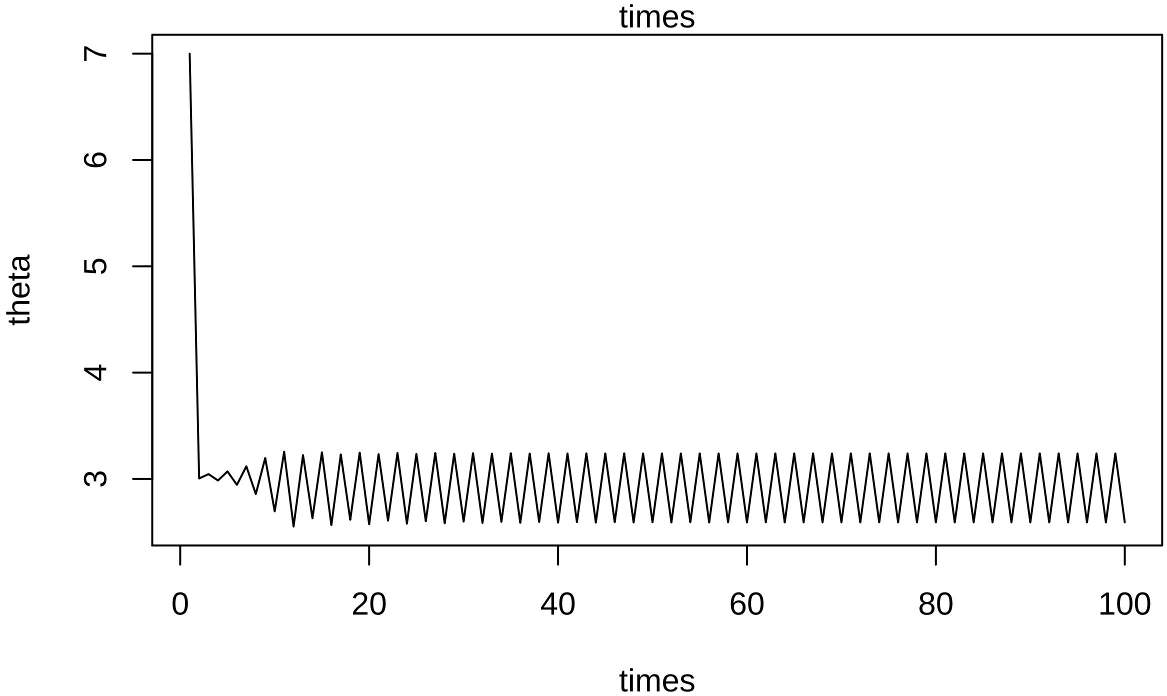


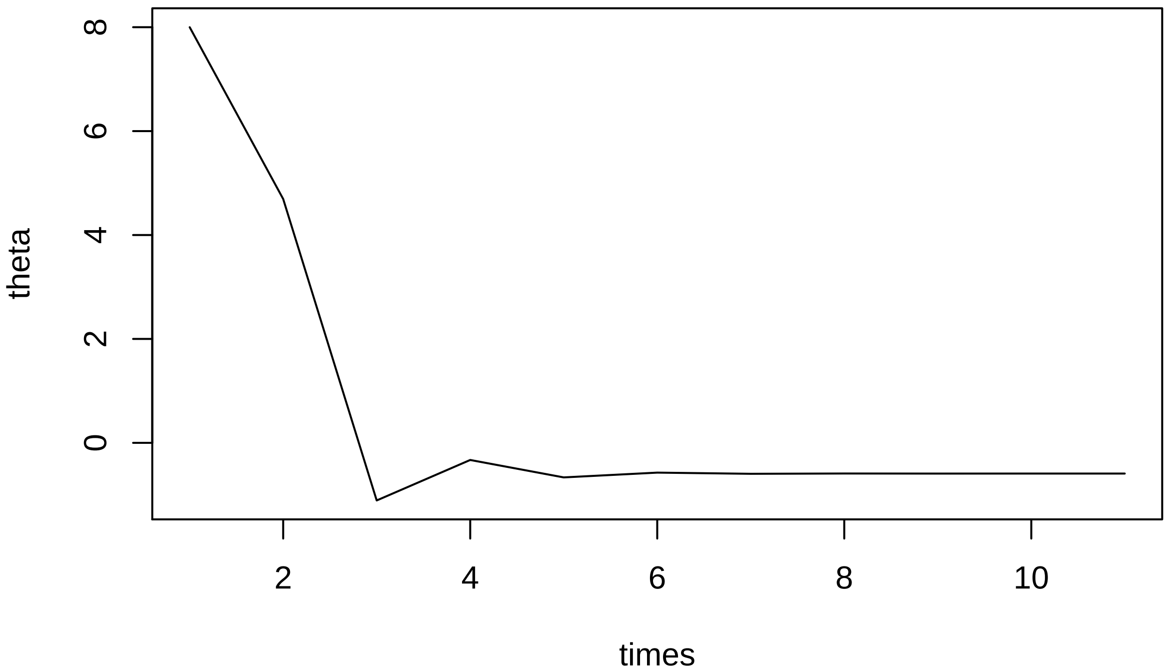
theta
0.0
−0.2
−0.4
−0.6
2
4
6
8
times

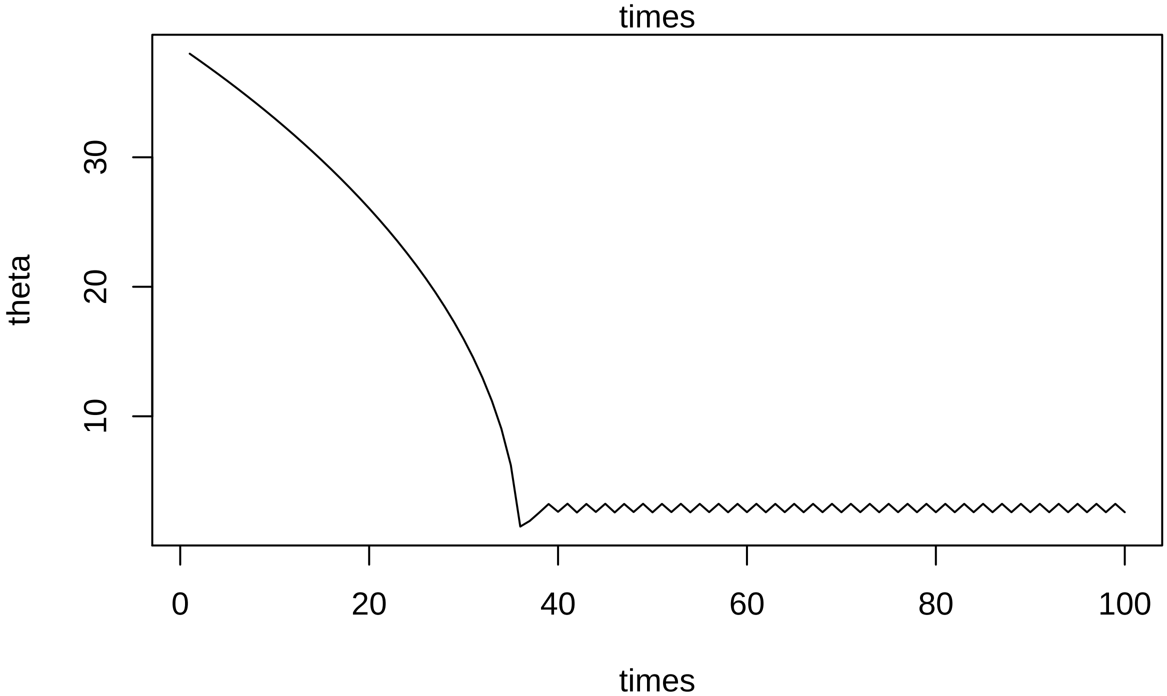

theta
3.0
2.5
2.0
1.5
0
20
40
60
80
100
times

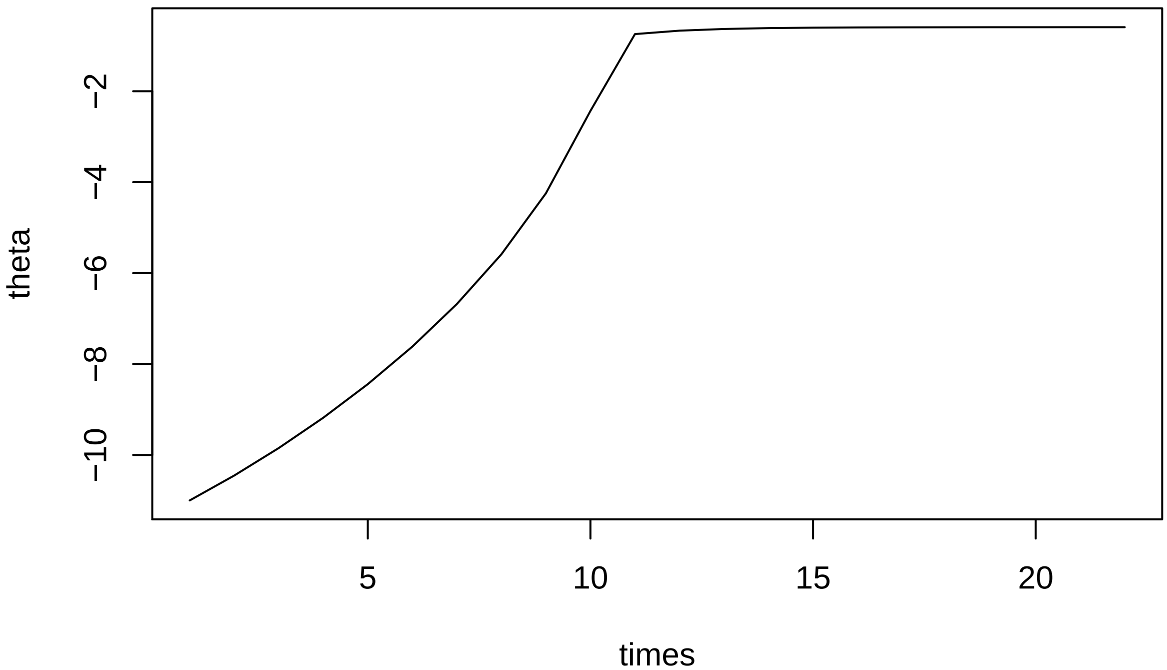

theta
4
3
2
1
0
2
4
6
8
10
times

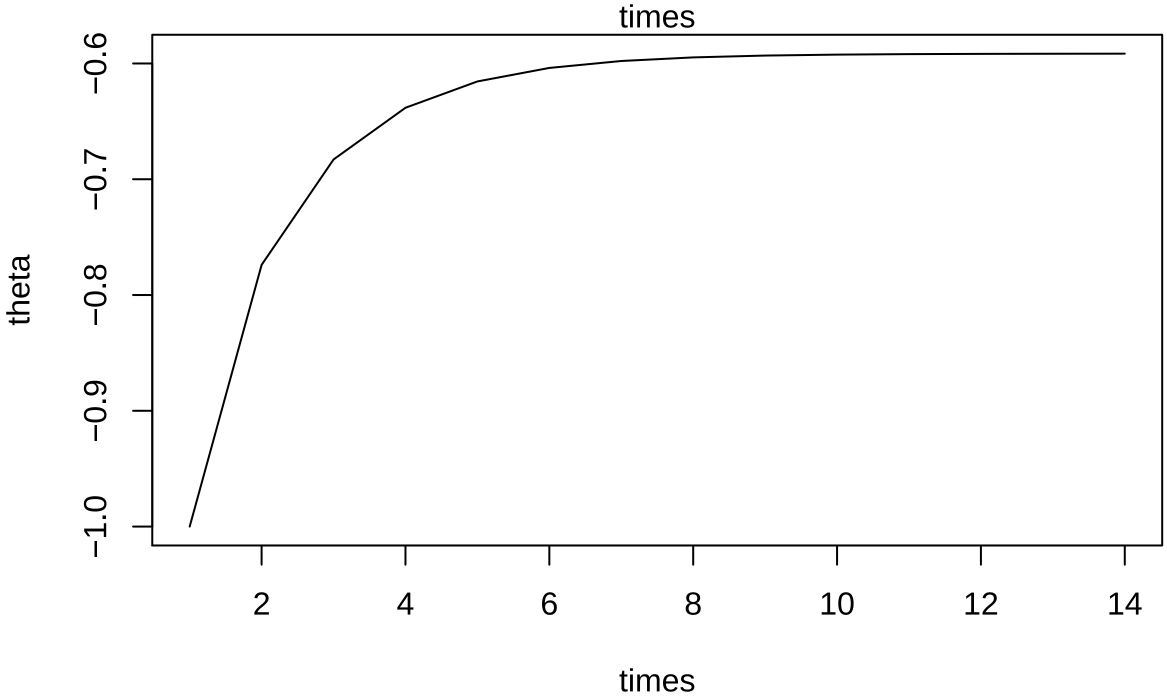

theta
4
3
2
1
0
−1
2
4
6
8
10
times

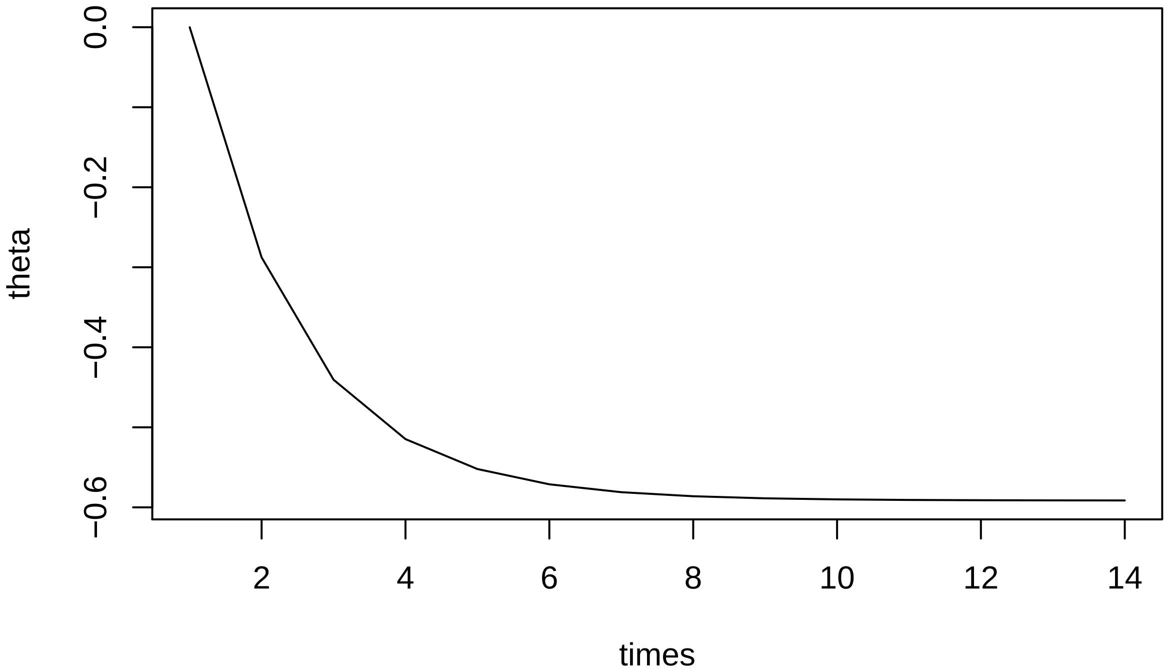

theta
7
6
5
4
3
0
20
40
60
80
100
times

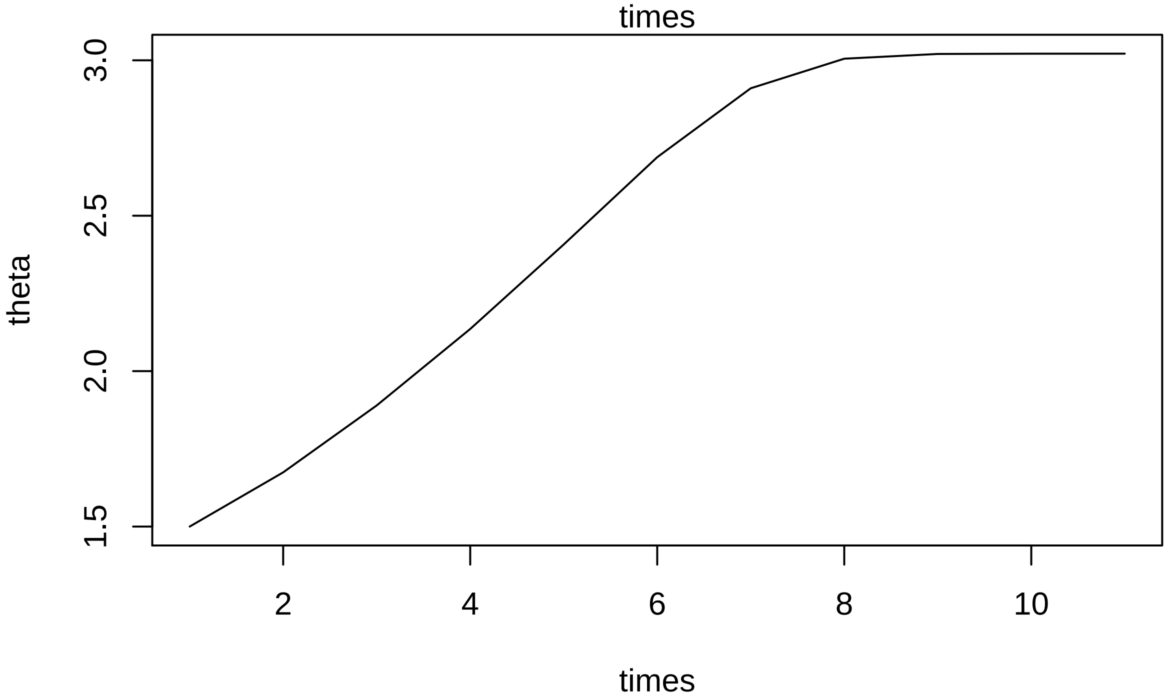

theta
8
6
4
2
0
2
4
6
8
10
times

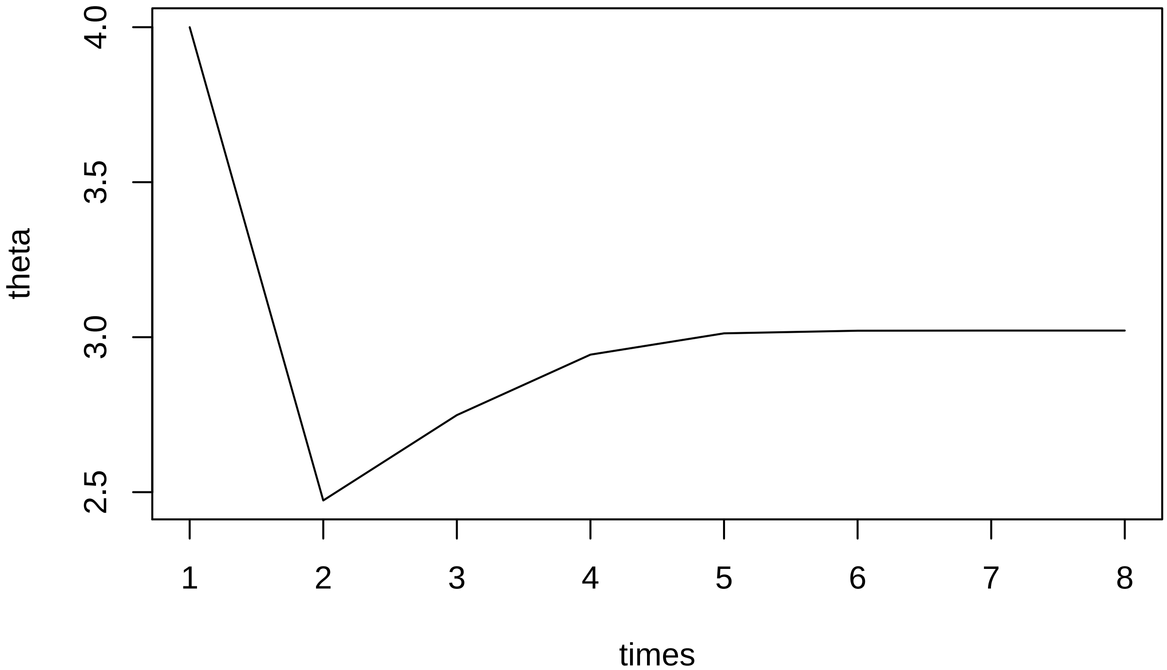

theta
30
20
10
0
20
40
60
80
100
times

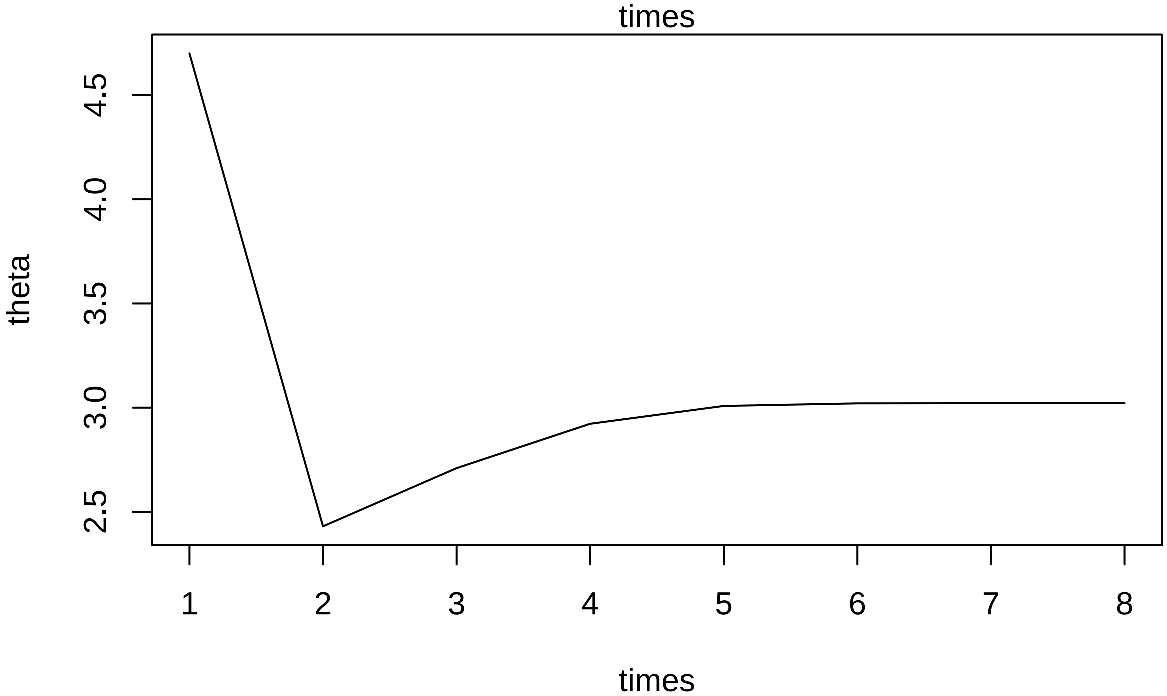

theta
−2
−4
−6
−8
−10
5
10
15
20
times

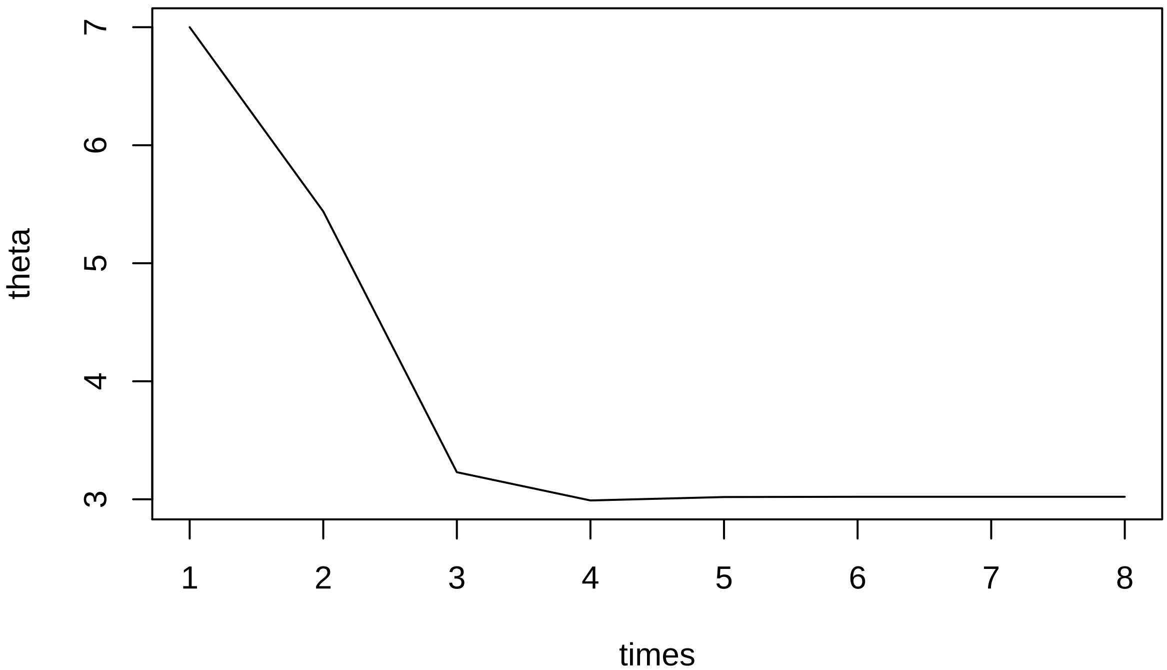

theta
−0.6
−0.7
−0.8
−0.9
−1.0
2
4
6
8
10
12
14
times

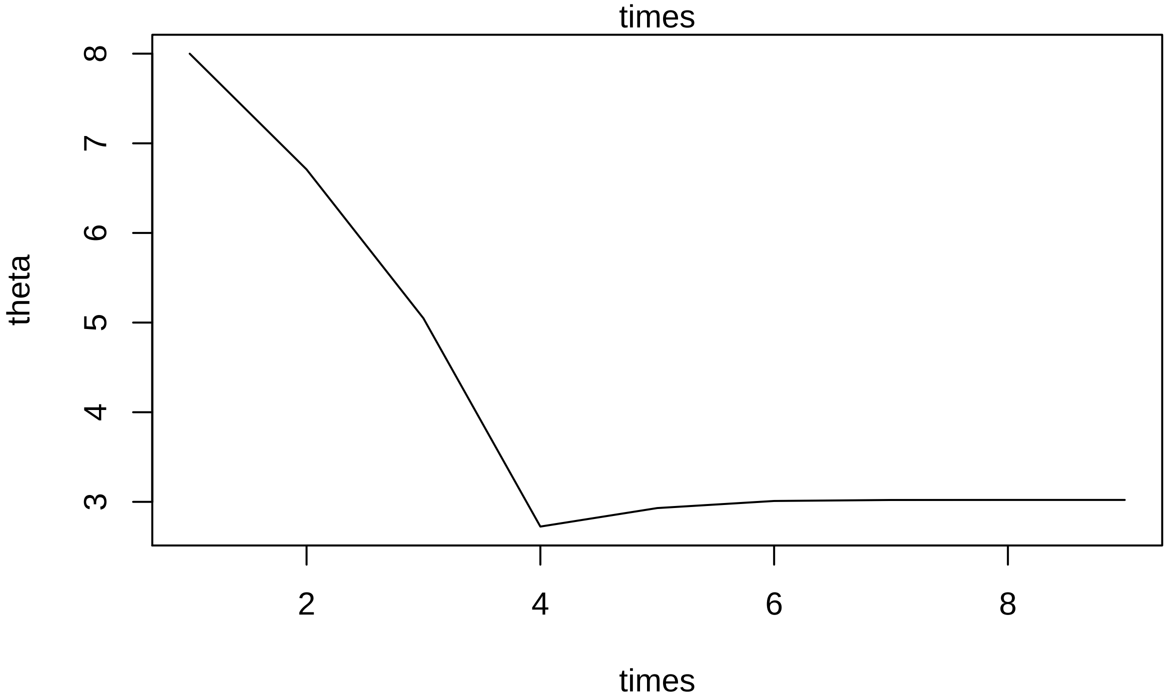

0.0
−0.2
−0.4
−0.6
theta
2
4
6
8
10
12
14
times

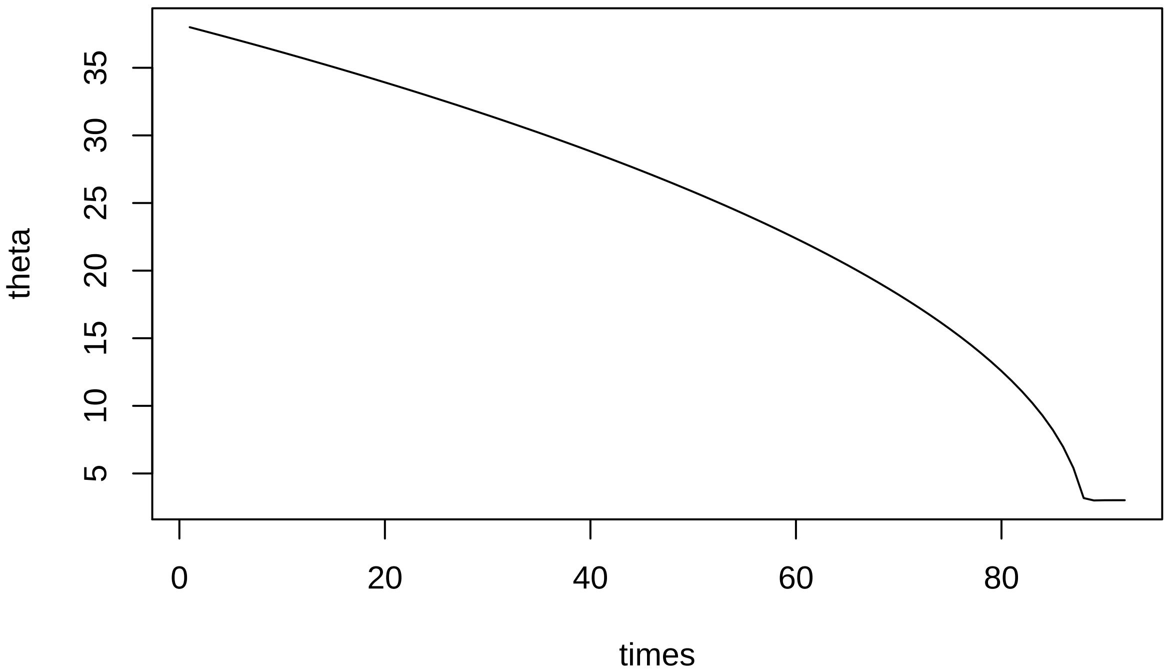

3.0
2.5
2.0
1.5
theta
2
4
6
8
10
times


theta
4.0
3.5
3.0
2.5
1
2
3
4
5
6
7
8
times


theta
4.5
4.0
3.5
3.0
2.5
1
2
3
4
5
6
7
8
times


theta
7
6
5
4
3
1
2
3
4
5
6
7
8
times

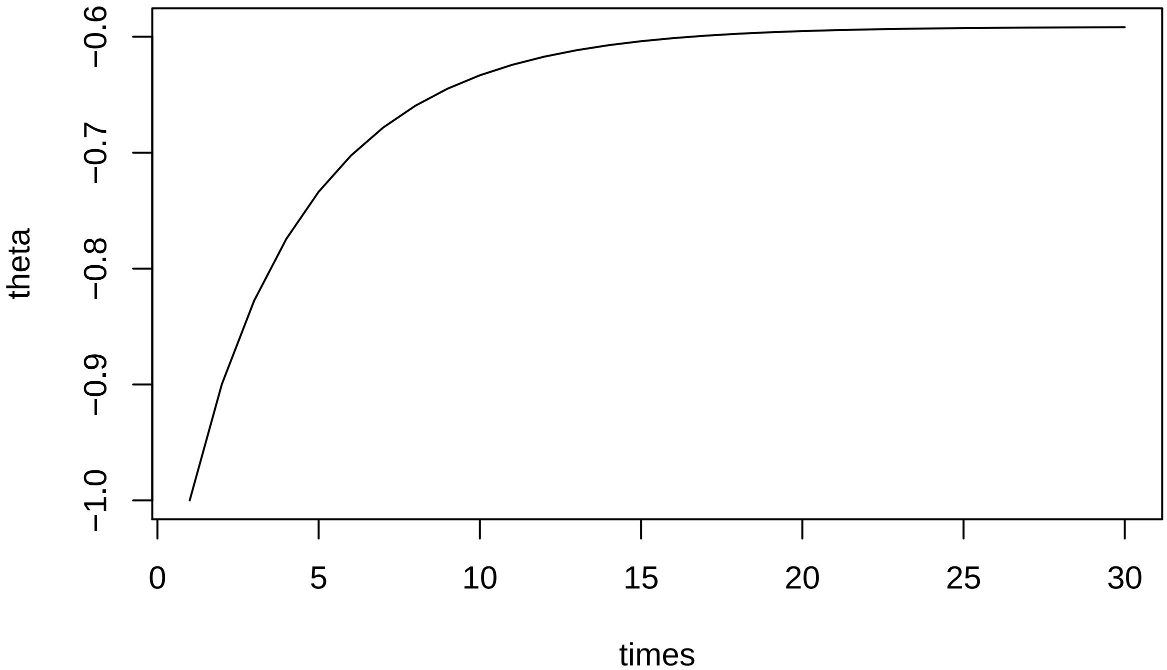

theta
8
7
6
5
4
3
2
4
6
8
times



As we can see graphs above, some starting points have ideal converging values. Some values after iteration are isolating and continues without stopping.

1(d)

```
x <- c(1.77, -0.23, 2.76, 3.80, 3.47, 56.75, -1.34, 4.24, -2.44,
       3.29, 3.71, -2.40, 4.53, -0.07, -1.05, -13.87, -2.53, -1.75)
llh1 <- function(x, theta){
  value <- 0
  for (i in 1:length(x)){
    value <- value - 2*(theta-x[i])/(1+(theta-x[i])^2)
  }
  value
}
theta <- array()
theta[1] <- -1
diff <- 100
i <- 1
while(abs(diff)>10^(-4)){
  theta[i+1] <- theta[i]+llh1(x, theta[i])/(length(x)/2)
  diff <- theta[i+1]-theta[i]
  i <- i+1
}
plot(theta, xlab = 'times', type = 'l')
```



In this point of view, I just print one figure which starting point is -1.

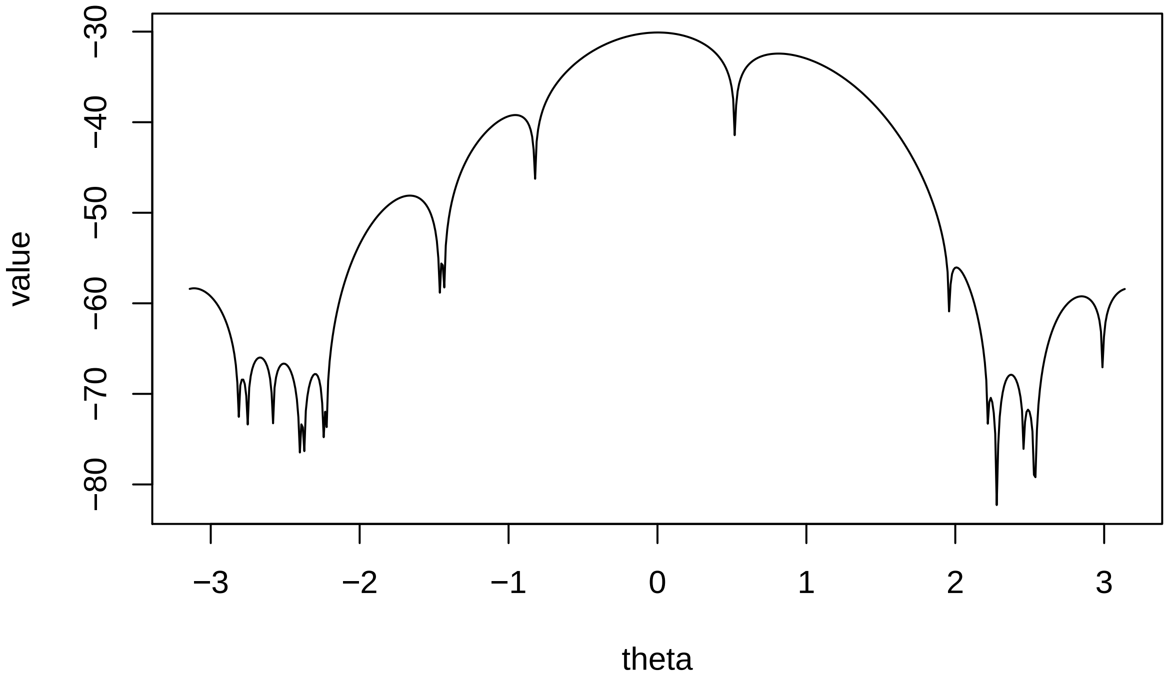
1(e)

After using different methods to measure the value of theta, we can easily see that Newton-Raphson method is the fastest to converge, fixed-point iteration is the second fast, and Fisher scoring is the least. As we can see the graphs above, the fix-point iterations' stability is the worst.

2(a)

$$
\begin{aligned}
& l(\theta) && (29)\\
& = \sum_{i=1}^{n} \ln \sin^2 \frac{x_i - \theta}{2} - n \ln \pi && (30)
\end{aligned}
$$

```
x <- c(3.91, 4.85, 2.28, 4.06, 3.70, 4.04, 5.46, 3.53, 2.28, 1.96,
       2.53, 3.88, 2.22, 3.47, 4.82, 2.46, 2.99, 2.54, 0.52)
theta <- seq(-pi, pi, 0.01)
loglikelihood <- function(x, theta){
  value <- -length(x)*log(pi)
  for (i in 1:length(x)){
    value <- value + log((sin((x[i]-theta)/2)^2))
  }
  value
}
value <- array()
for (i in 1:length(theta)){
  value[i] <- loglikelihood(x, theta[i])
}
plot(theta, value, type = 'l')
```



2(b)

I integrated the function x*p from 0 to 2pi, and get the expression for

$$E[X|\theta] = \pi + \sin(\theta) \tag{31}$$

Here is my two roots:

```
x <- c(3.91, 4.85, 2.28, 4.06, 3.70, 4.04, 5.46, 3.53, 2.28, 1.96,
       2.53, 3.88, 2.22, 3.47, 4.82, 2.46, 2.99, 2.54, 0.52)

theta <- seq(-pi, pi, 0.01)
theta_bar <- function(theta){
  pi+sin(theta)-mean(x)
}

uniroot(theta_bar, c(-pi, 2))$root
```

```
## [1] 0.09539438
```

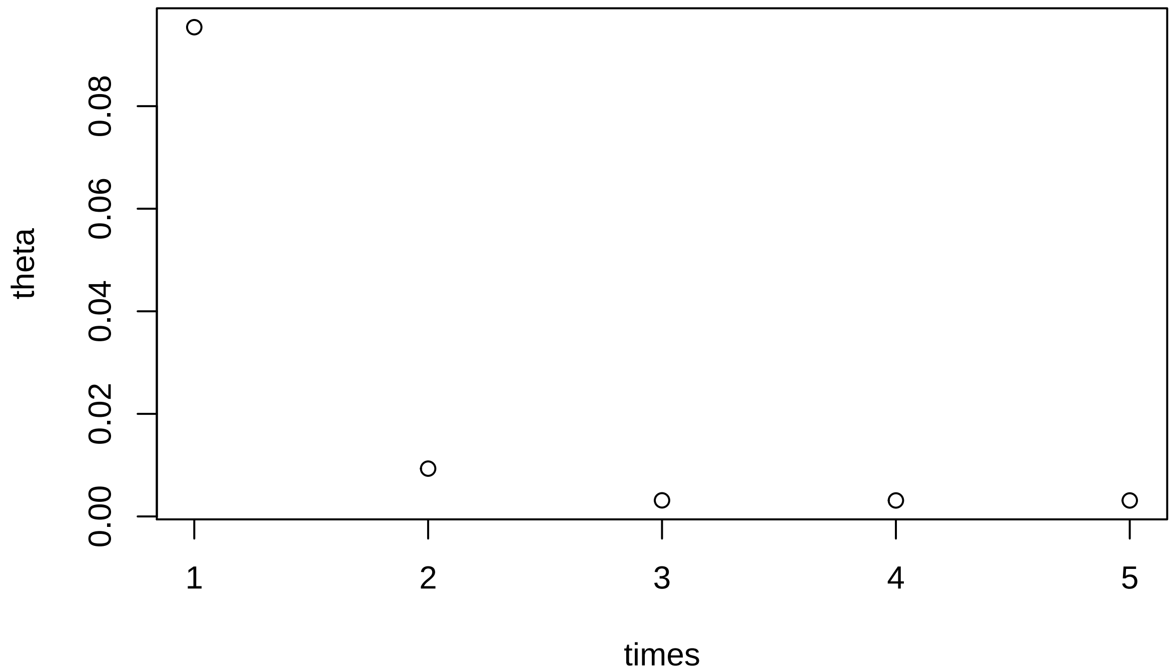
```
uniroot(theta_bar, c(2, pi))$root
```

```
## [1] 3.046199
```

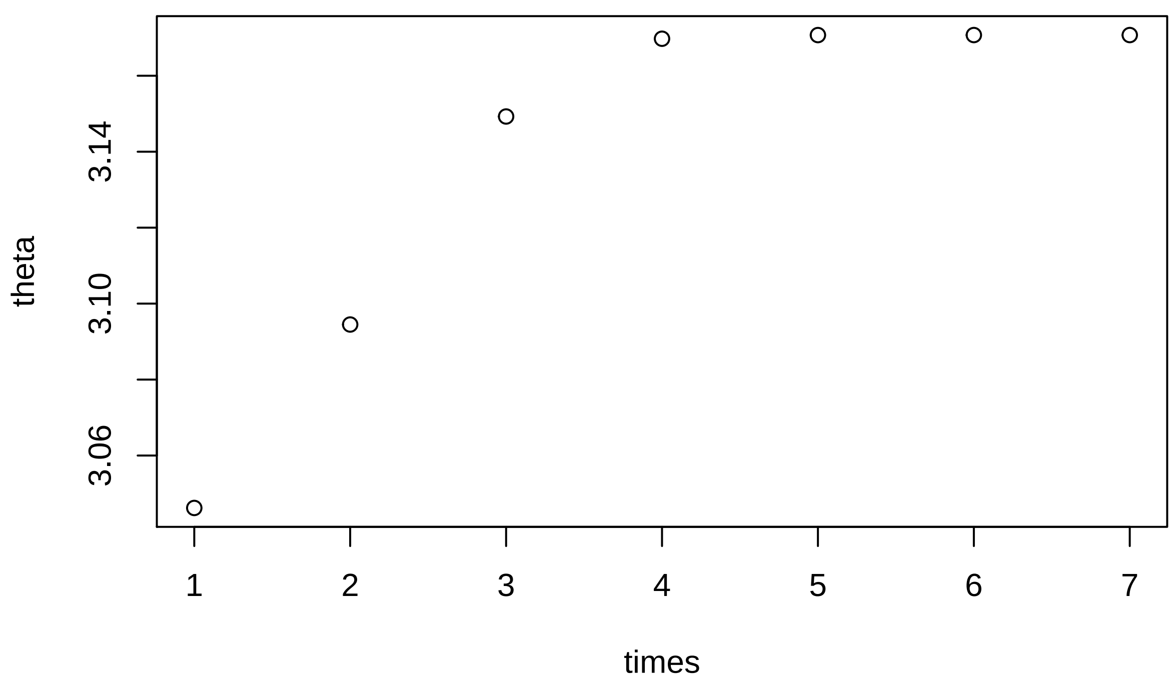
2(c)

```
x <- c(3.91, 4.85, 2.28, 4.06, 3.70, 4.04, 5.46, 3.53, 2.28, 1.96,
       2.53, 3.88, 2.22, 3.47, 4.82, 2.46, 2.99, 2.54, 0.52)
theta <- seq(-pi, pi, 0.01)
theta_bar <- function(theta){
  pi+sin(theta)-mean(x)
```

```
}

theta_moment <- c(uniroot(theta_bar, c(-pi, 2))$root, uniroot(theta_bar, c(2, pi))$root)


### First derivative of loglikelihood
llh1 <- function(x, theta){
  value <- 0
  for (i in 1:length(x)){
    value <- value + sin(x[i]-theta)/(1-cos(x[i]-theta))
  }
  value
}

### Second derivative of loglikelihood
llh2 <- function(x, theta){
  value <- 0
  for (i in 1:length(x)){
    value <- value + 1/(1-cos(x[i]-theta))
  }
  value
}

### Compute theta value when geting MLE
theta <- array()
theta[1] <- theta_moment[1]
difference <- 100
i <- 1
while(abs(difference)>10^(-8)){
  theta[i+1] <- theta[i] - llh1(x, theta[i])/llh2(x, theta[i])
  difference <- theta[i+1] - theta[i]
  i <- i+1
}
plot(theta, xlab = 'times')
```




```
### Compute theta value when geting MLE
theta <- array()
theta[1] <- theta_moment[2]
difference <- 100
i <- 1
while(abs(difference)>10^(-8)){
  theta[i+1] <- theta[i] - llh1(x, theta[i])/llh2(x, theta[i])
  difference <- theta[i+1] - theta[i]
  i <- i+1
}
plot(theta, xlab = 'times')
```



Here is my two plots of two theta moments.

2(d)

When I use the 2(c) functions to calcualte the MLE solutions for theta, I got 2.848415 when starting from 2.7, and -2.668857 when starting from -2.7.

2(e)

This is the group what I have got:

```
##        value Freq
## 1  -3.112471   11
## 2  -2.786557    2
## 3  -2.668857    5
## 4  -2.509356    6
## 5  -2.388267    1
## 6  -2.297926    4
## 7  -2.232192    1
## 8  -1.662712   24
## 9  -1.447503    1
## 10 -0.954406   19
## 11  0.003118   42
## 12  0.812637   46
## 13  2.007223    8
## 14  2.237013    2
## 15  2.374712    6
## 16   2.48845    2
## 17  2.848415   15
## 18  3.170715    5
```

3(a)

In this problem, I both created my own function with Gauss Newton Method and used built-in function called nls to solve the problem and get the result:

```r
Gauss_Newton <- function(r0,k0){
  start <- c(0.5,1000)
  error <- sum(abs(start))
  i <- 1
  while(error>10^(-4)){
    r0 <- r0 + theta_calculate(r0,k0)[1]
    k0 <- k0 + theta_calculate(r0,k0)[2]
    error <- sum(abs(theta_calculate(r0,k0)))
    i <- i + 1
  }
  theta <- c(r0, k0, i)
  theta
}
Gauss_Newton(0.2,2000)
```

```
## [1]    0.1182697 1049.4064234    8.0000000
```

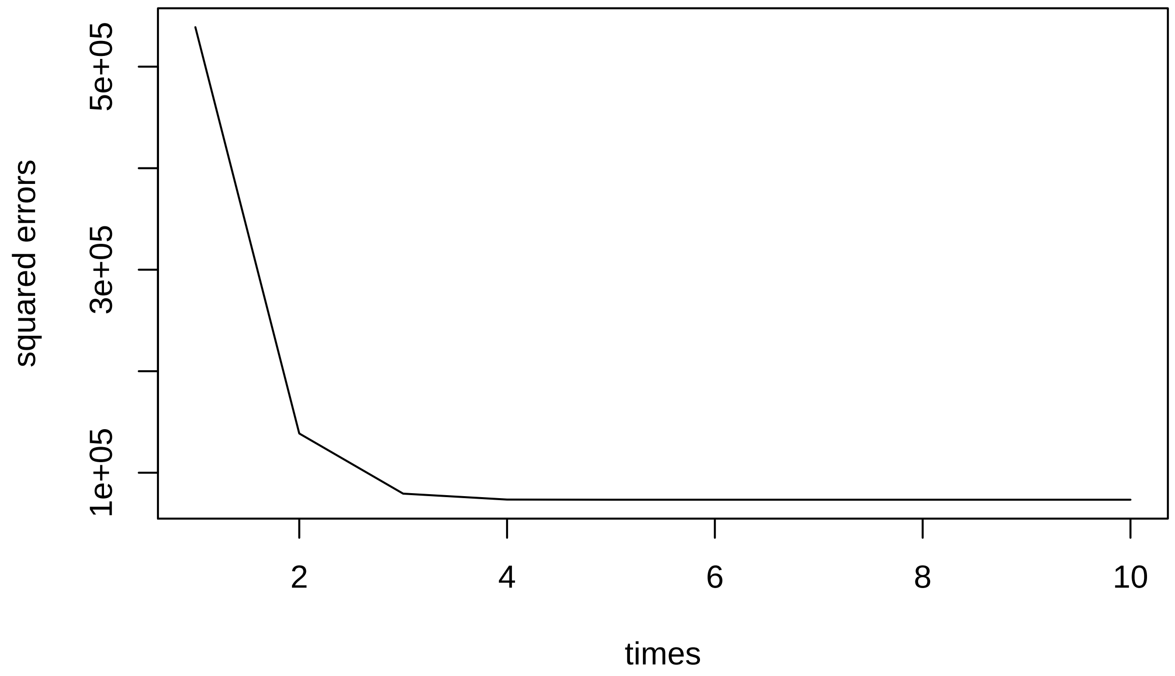
```r
model <- nls(N~(2*k)/(2+(k-2)*exp(-r*times)), start = list(k = 100, r = 0.1))
model
```

```
## Nonlinear regression model
##   model: N ~ (2 * k)/(2 + (k - 2) * exp(-r * times))
##    data: parent.frame()
##         k         r
## 1049.4065    0.1183
##  residual sum-of-squares: 73420
##
## Number of iterations to convergence: 10
## Achieved convergence tolerance: 9.33e-06
```

3(b)

Here is my contour plot of the sum of squared errors:

```r
###Plot the contour of the sum of squared errors
i <- 1
error <- 1
r0 <- array()
k0 <- array()
r0[1] <- 0.2
k0[1] <- 1000
while(error>10^(-4)){
  r0[i+1] <- r0[i] + theta_calculate(r0[i],k0[i])[1]
  k0[i+1] <- k0[i] + theta_calculate(r0[i],k0[i])[2]
  error <- sum(abs(theta_calculate(r0[i+1],k0[i+1])))
  i <- i + 1
}

value <- array()
for (i in 1:length(r0)){
  value[i] <- sum((N - 2*k0[i]/(2+(k0[i]-2)*exp(-r0[i]*times)))^2)
  value
}
plot(value, xlab = 'times', ylab = 'squared errors', type = 'l')
```



squared errors
5e+05
3e+05
1e+05
2
4
6
8
10
times