5361_stat_hw_Qi_You

Question 1b

Equations

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

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

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

We can set x to be tant(x). In this way we can compute the value of I easiliy

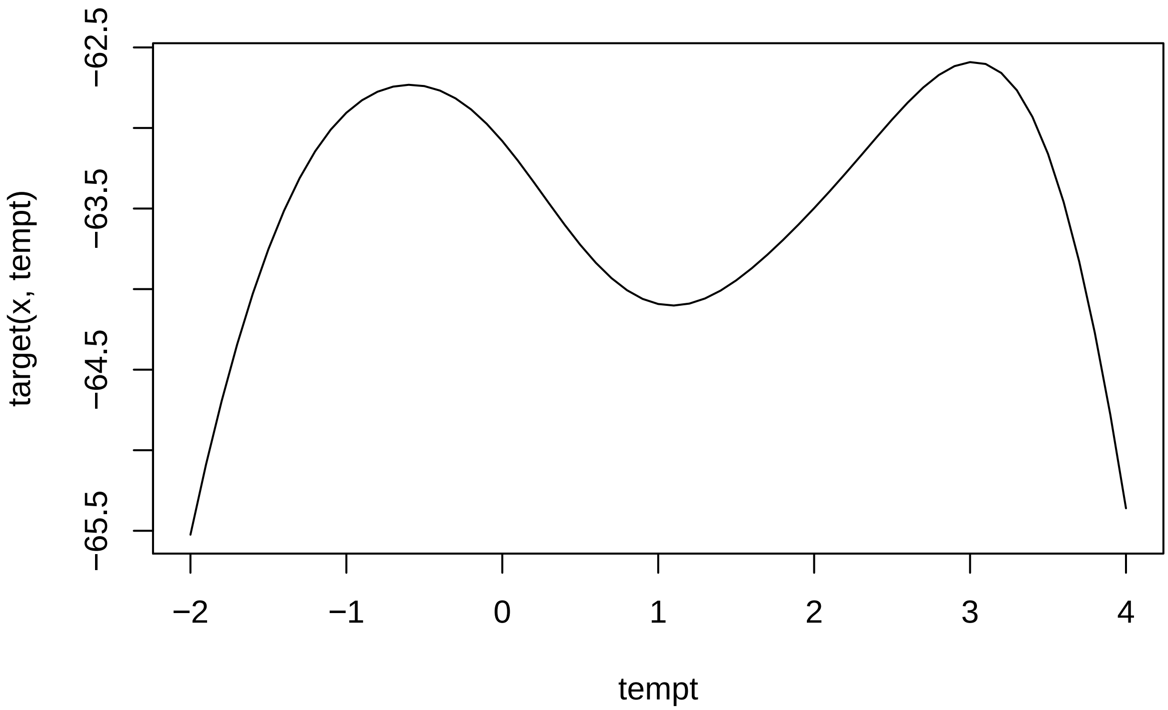
$$
\begin{aligned}
&I(\theta) && (15)\\
&= n\int \frac{\{p'(x)\}^2}{p(x)}dx && (16)\\
&= n\int \frac{4(x-\theta)^2}{\pi[1+(x-\theta)^2]^4} * \pi[1+(x-\theta)^2]dx && (17)\\
&= \frac{4n}{\pi}\int_{-\infty}^{\infty} \frac{(x-\theta)^2}{[1+(x-\theta)^2]^3}dx && (18)\\
&= \frac{4n}{\pi}\int_{-\infty}^{\infty}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

Functions I setup

```
derivitive1 <- function(x,sita){
  value <- 0
  for (i in 1:length(x)){
    value <- value-2*(sita-x[i])/(1+(sita-x[i])^2)
  }
  return(value)
}

derivitive2 <- function(s,sita){
  value <- 0
  for (i in 1:length(x)){
    value <- value-2*((1-(sita-x[i])^2)/(1+(sita-x[i])^2)^2)
  }
  return(value)
}
```

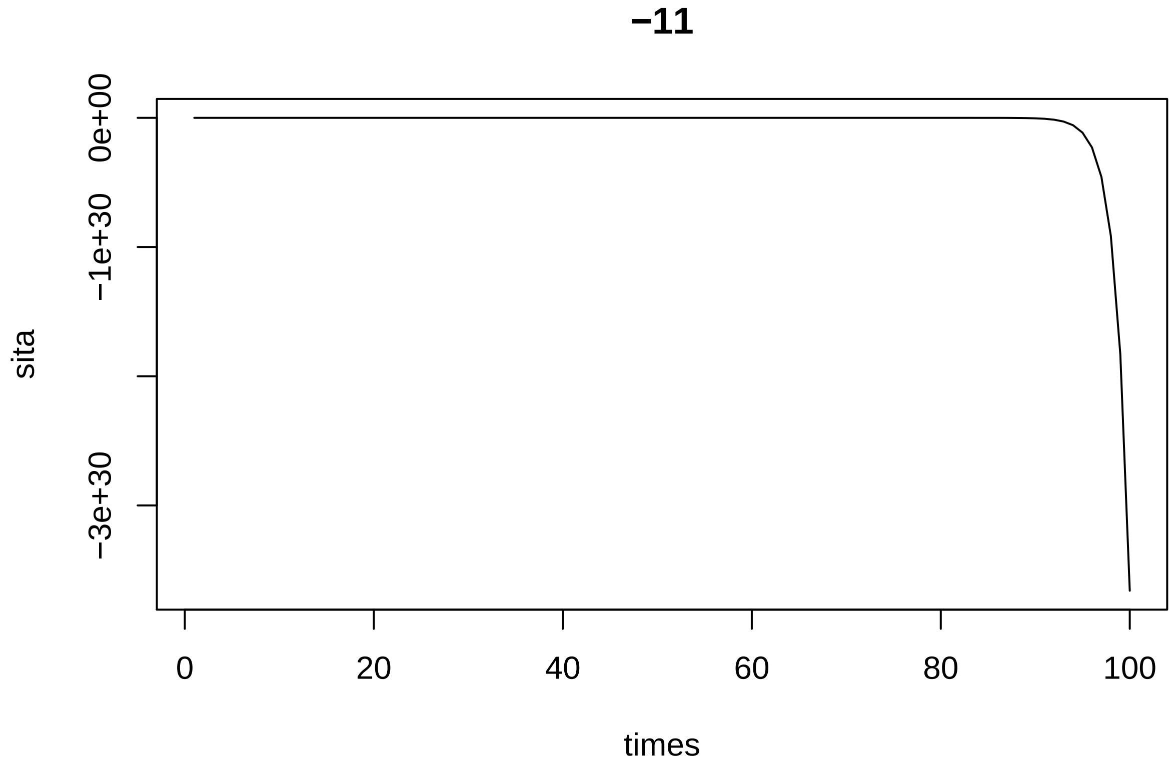
Plots(1b)

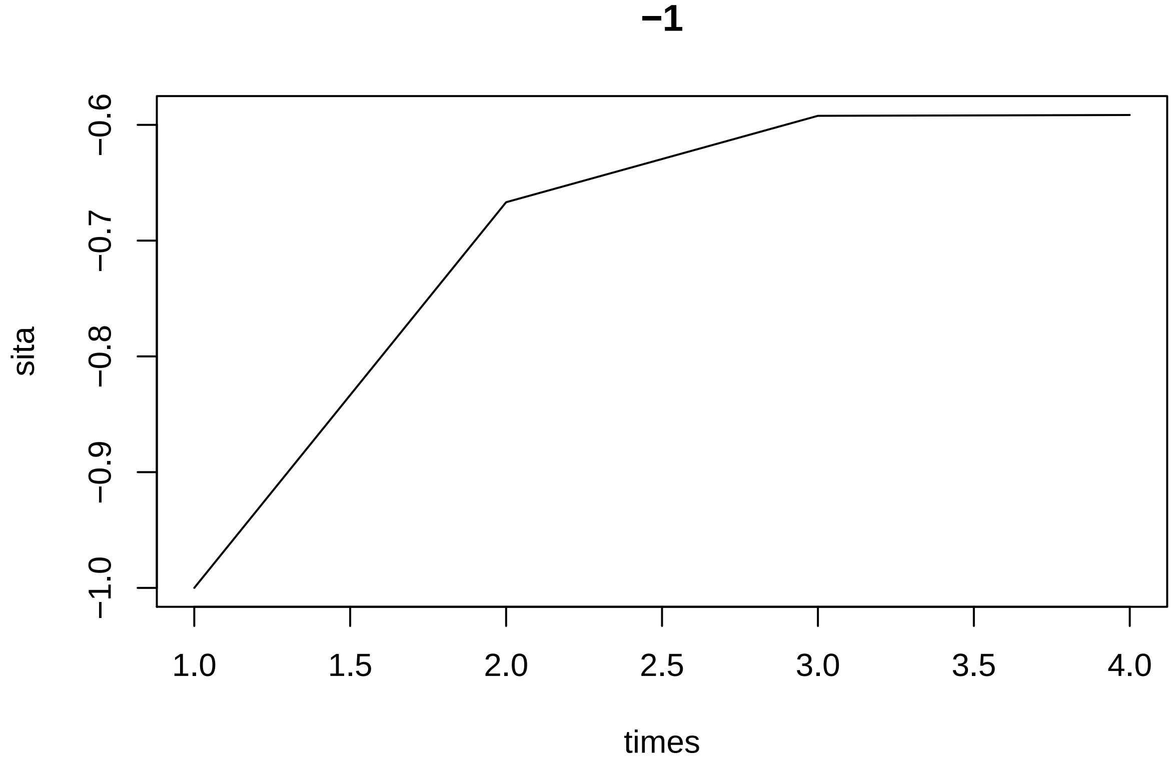
Here are the plots of my result

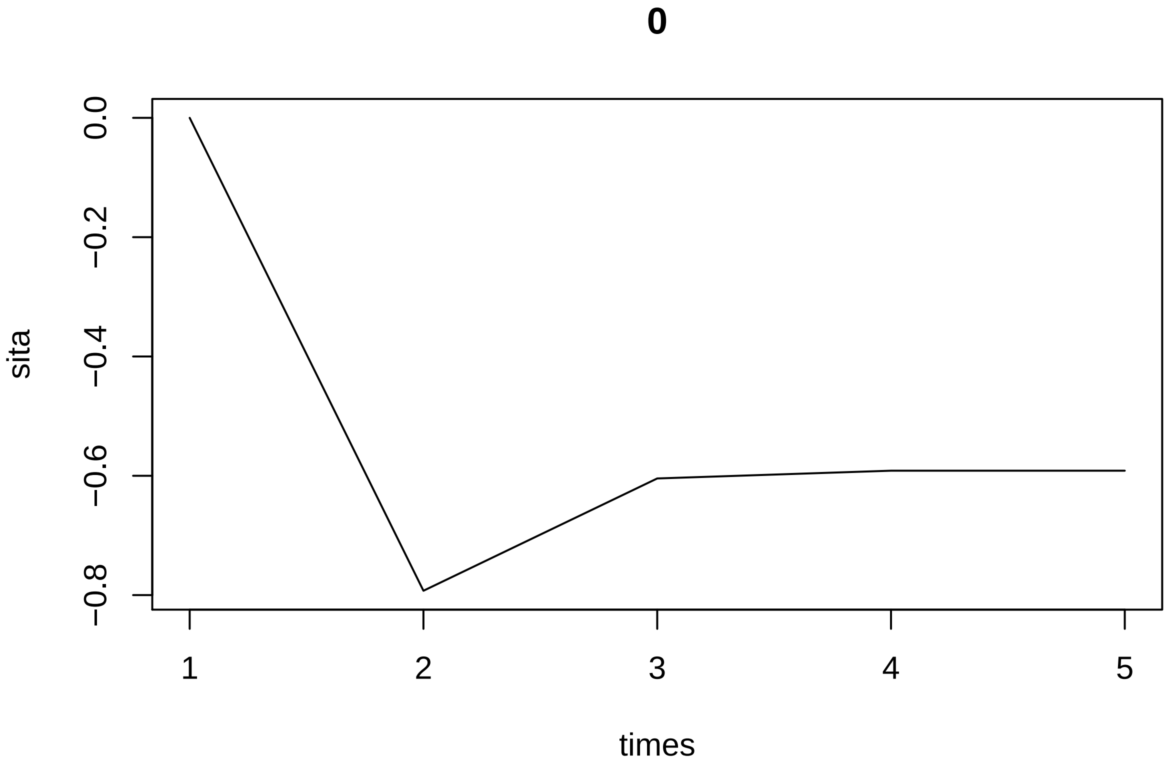


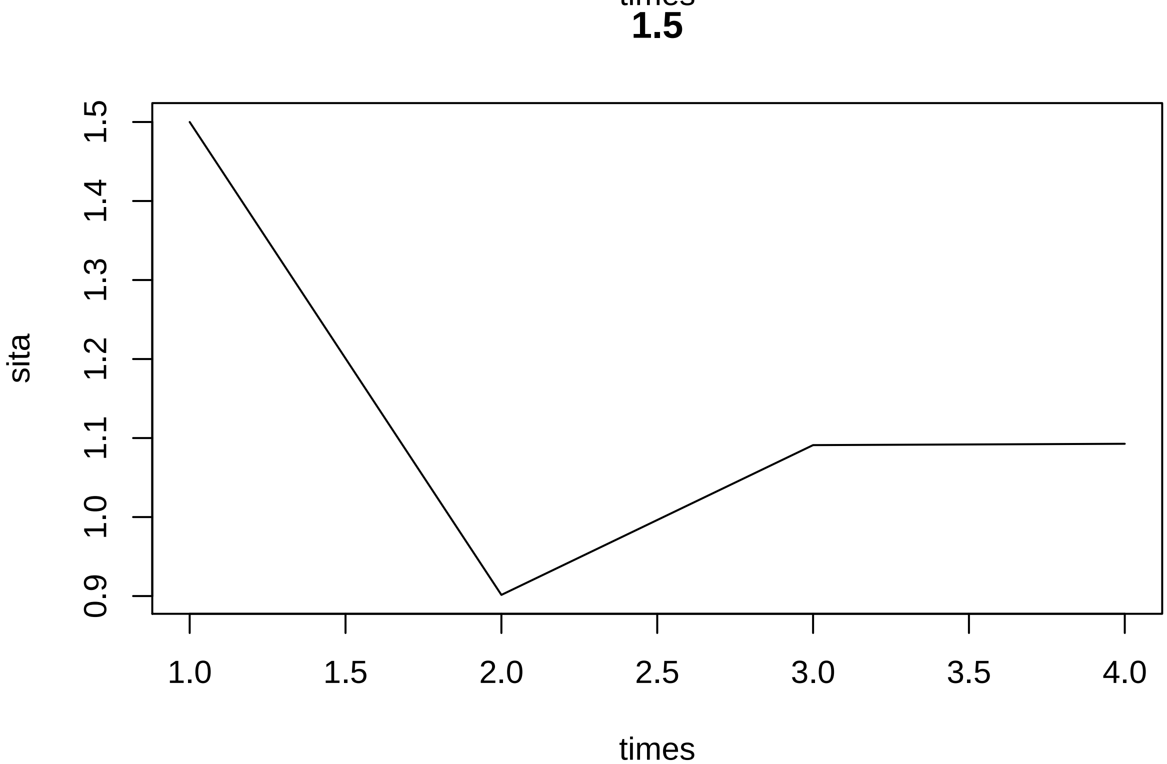
Here are the plots of mutiple start points

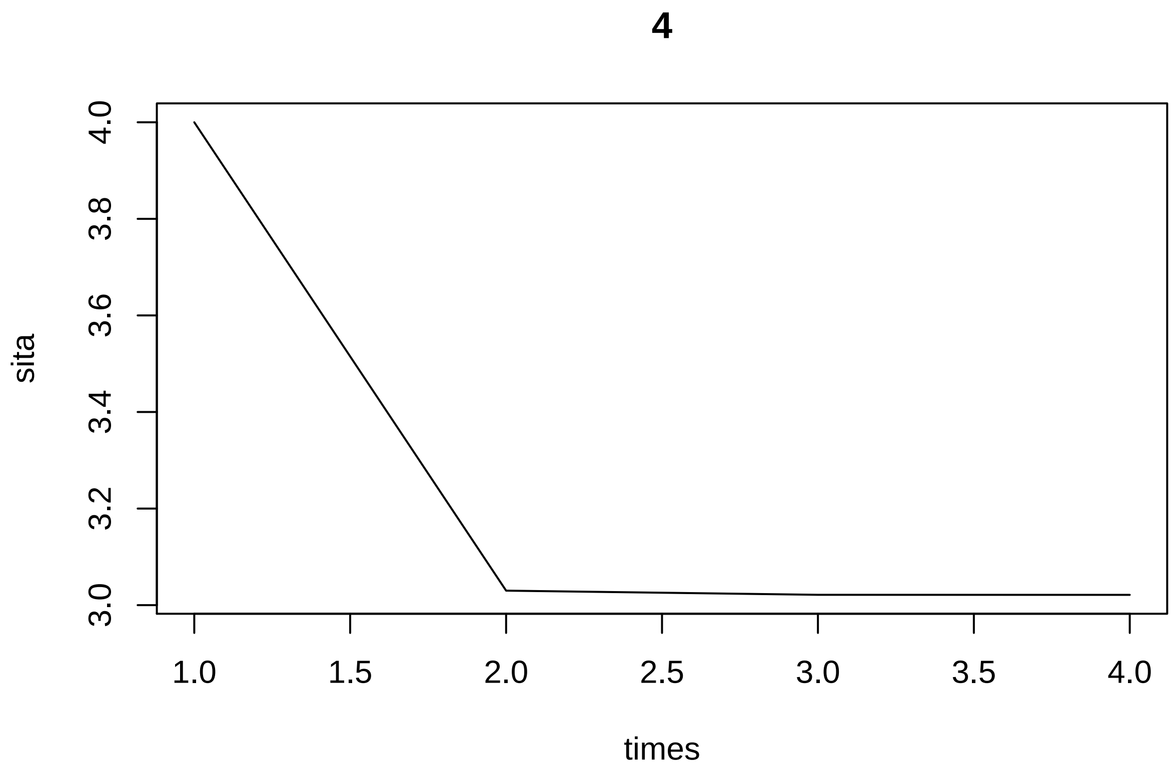
```
for(i in 1:length(start_points)){
  estimate(start_points[i])
}
```

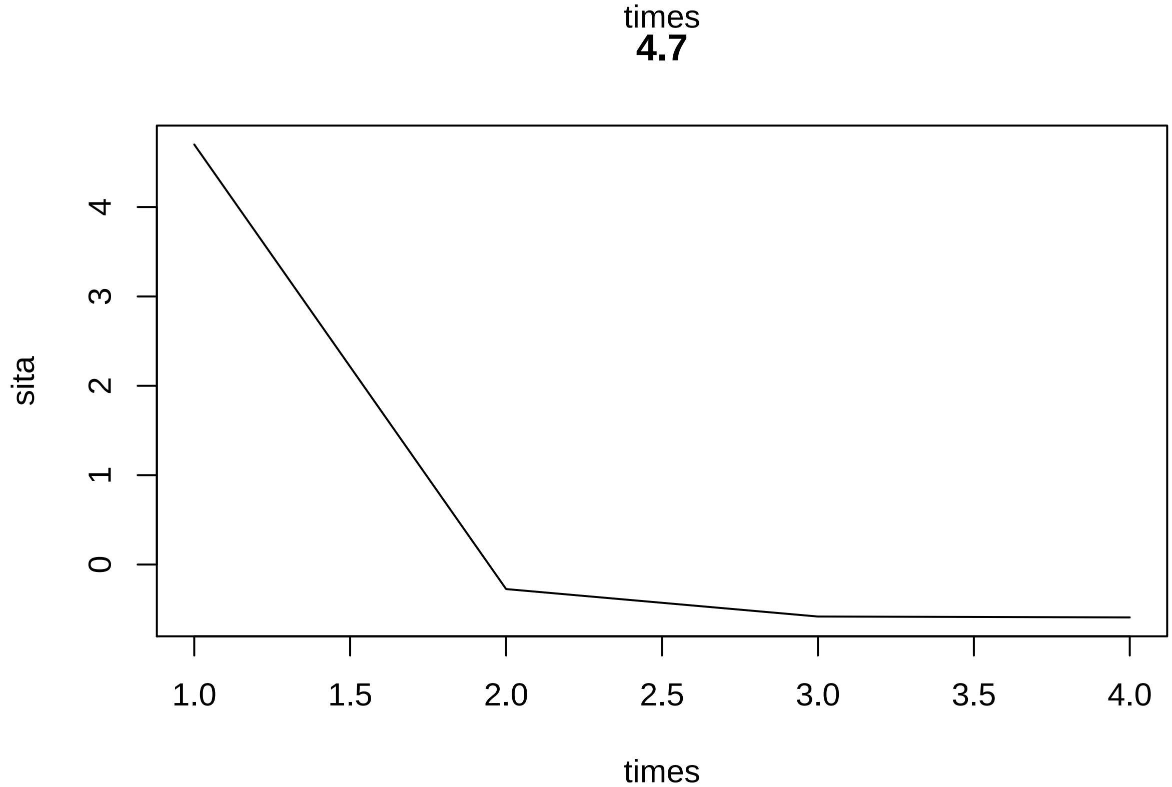

−11
0e+00
−1e+30
−3e+30
sita
0
20
40
60
80
100
times

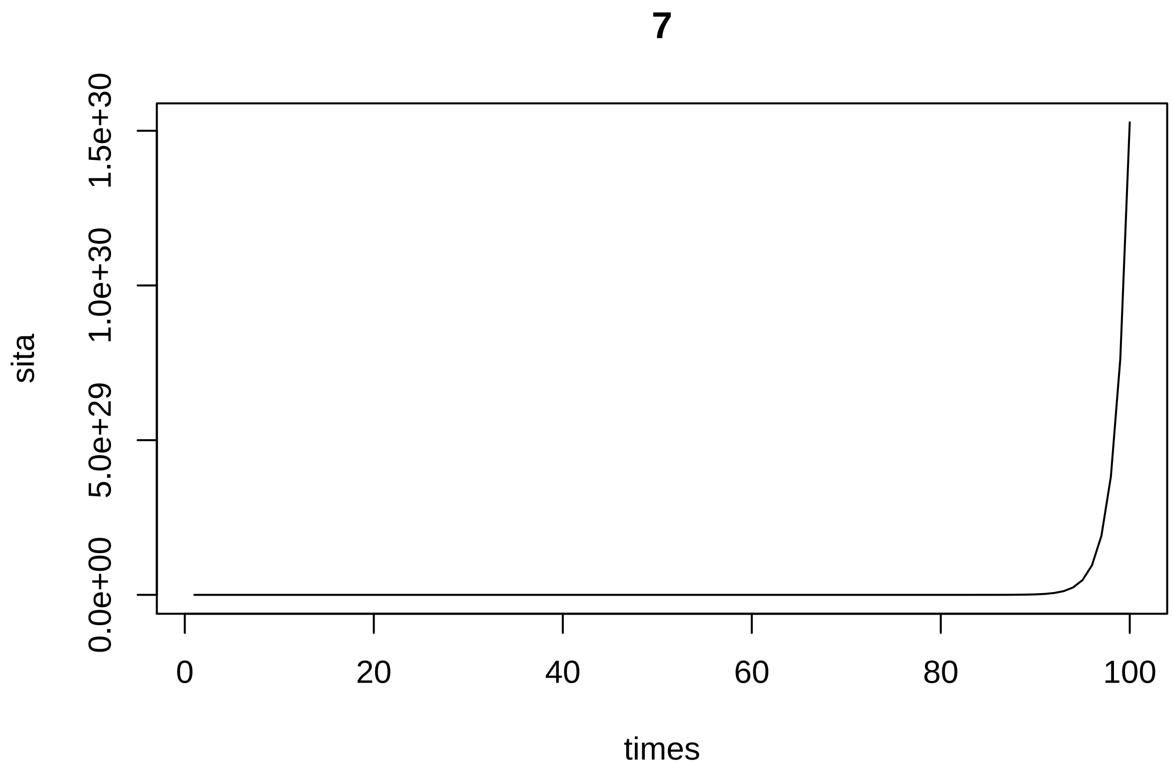

−1
−0.6
−0.7
−0.8
−0.9
−1.0
sita
1.0
1.5
2.0
2.5
3.0
3.5
4.0
times

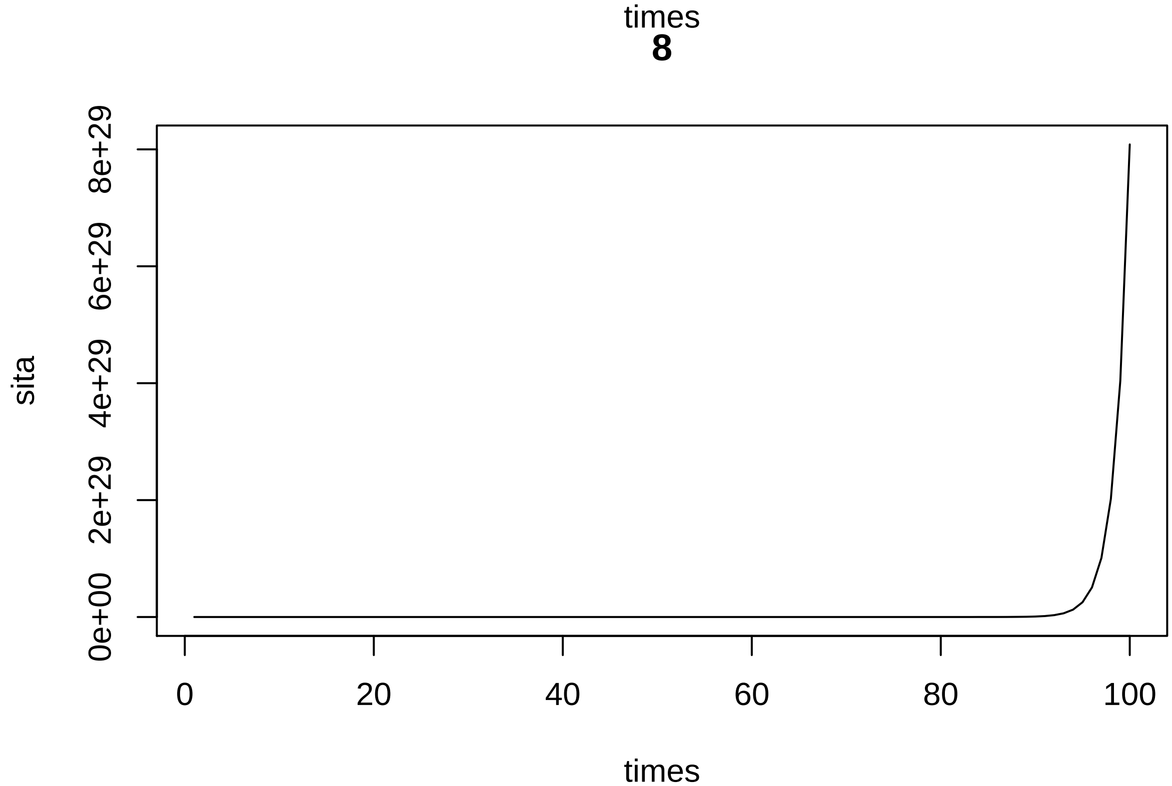

0
sita
0.0
−0.2
−0.4
−0.6
−0.8
1
2
3
4
5
times


1.5
sita
1.5
1.4
1.3
1.2
1.1
1.0
0.9
1.0
1.5
2.0
2.5
3.0
3.5
4.0
times


4
sita
times
4.0
3.8
3.6
3.4
3.2
3.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0

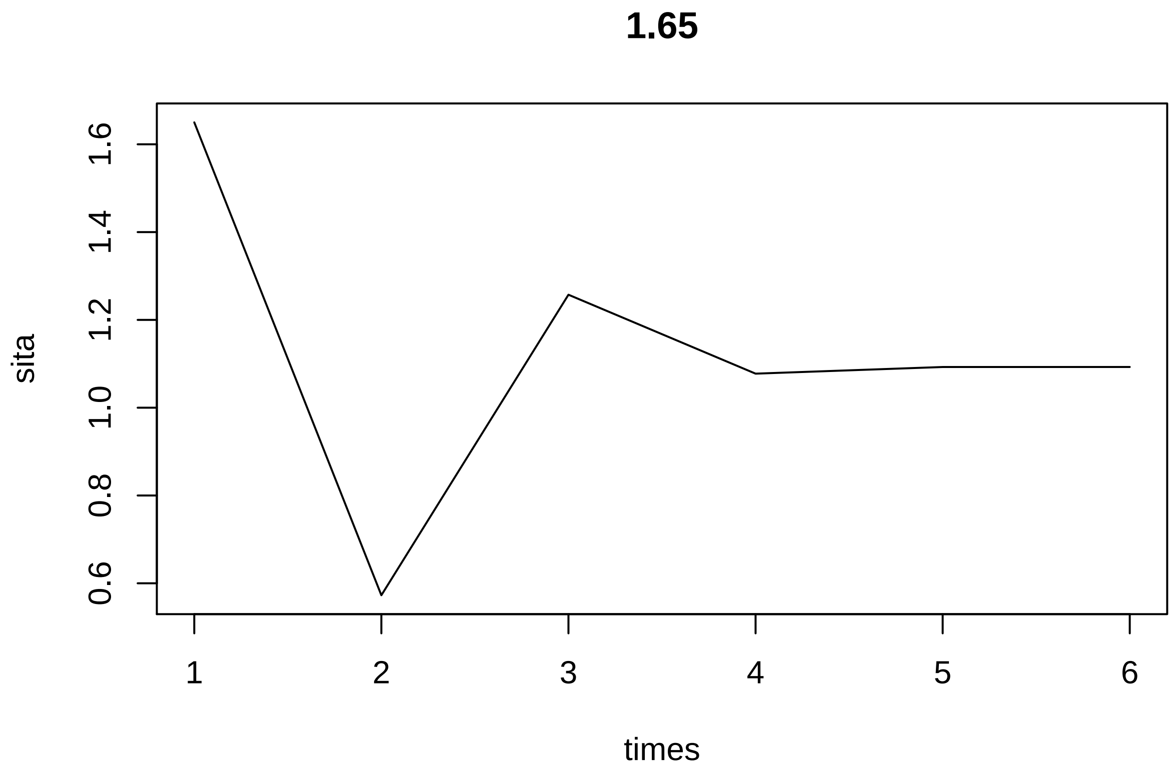

4.7
sita
times
4
3
2
1
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0





This is the plot of mean

```
estimate(mean(start_points))
```



Question 1(c)

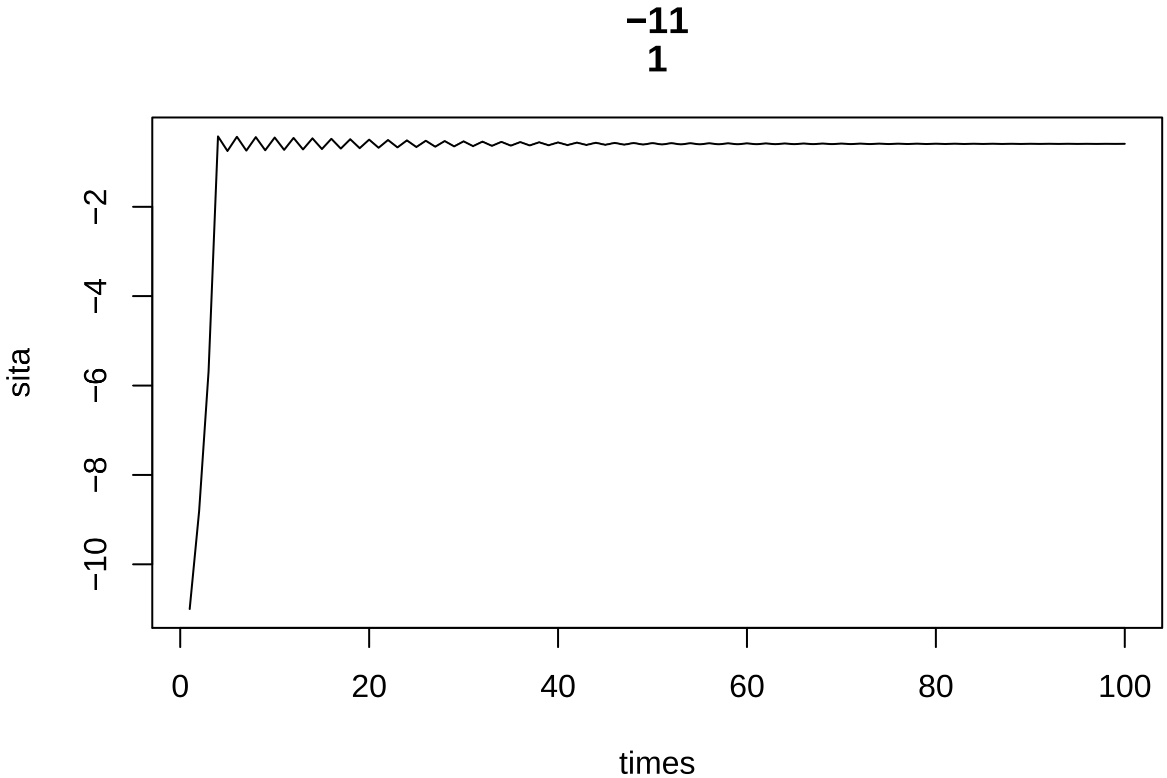
This time we use fixed points

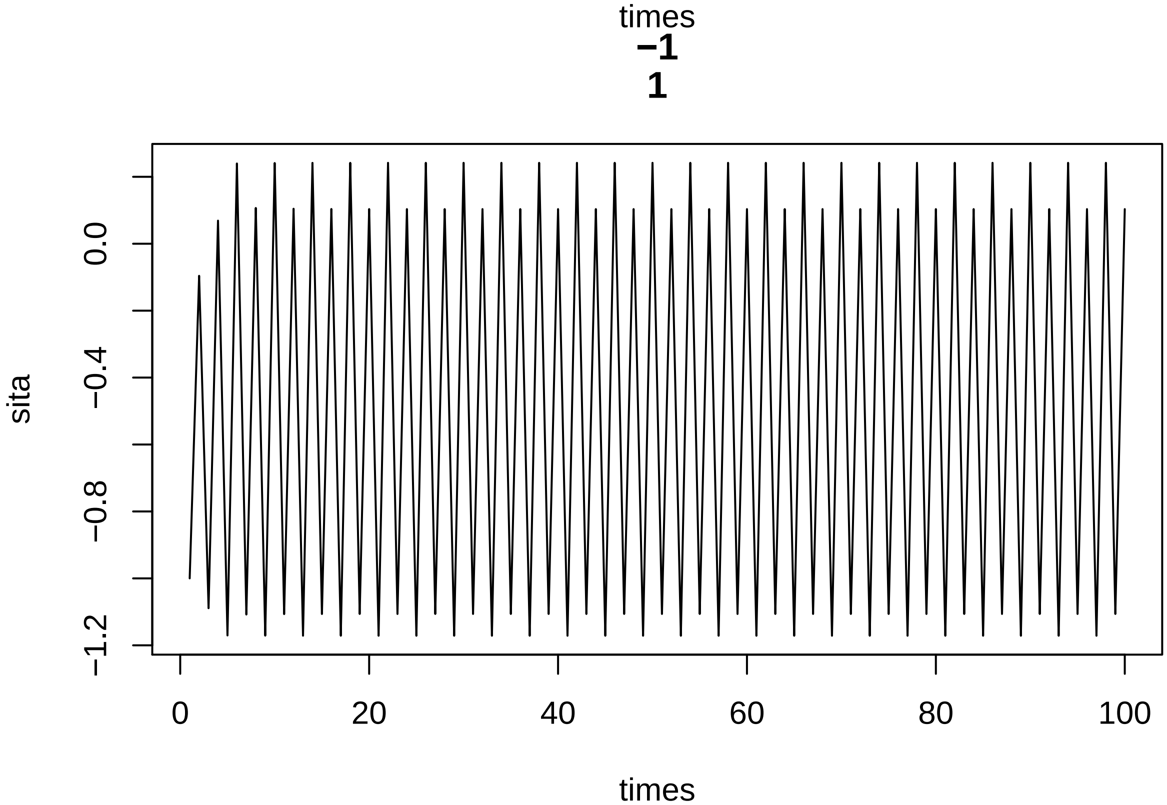
These are functions I established to compute derivition

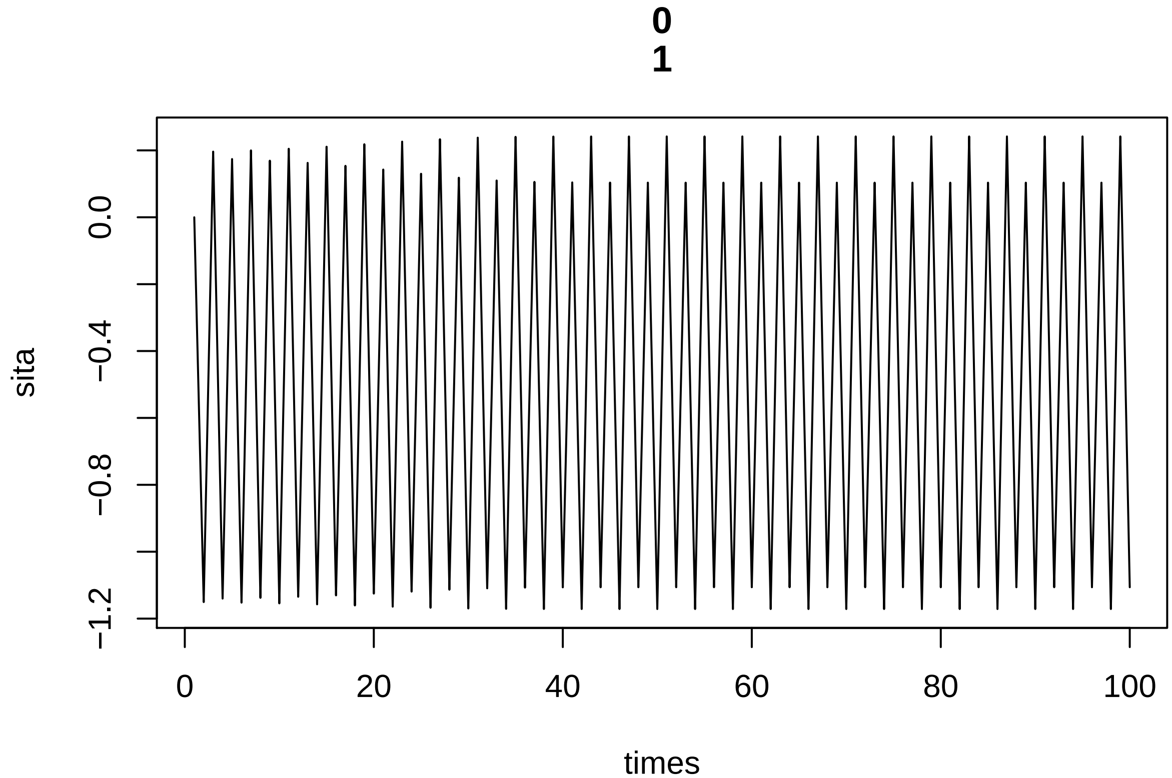
```
derivitive1 <- function(x,sita){
  value <- 0
  for (i in 1:length(x)){
    value <- value-2*(sita-x[i])/(1+(sita-x[i])^2)
  }
  return(value)
}

G <- function(alpha,x,sita){
  value <- alpha*derivitive1(x,sita)+sita
  return(value)
}
```

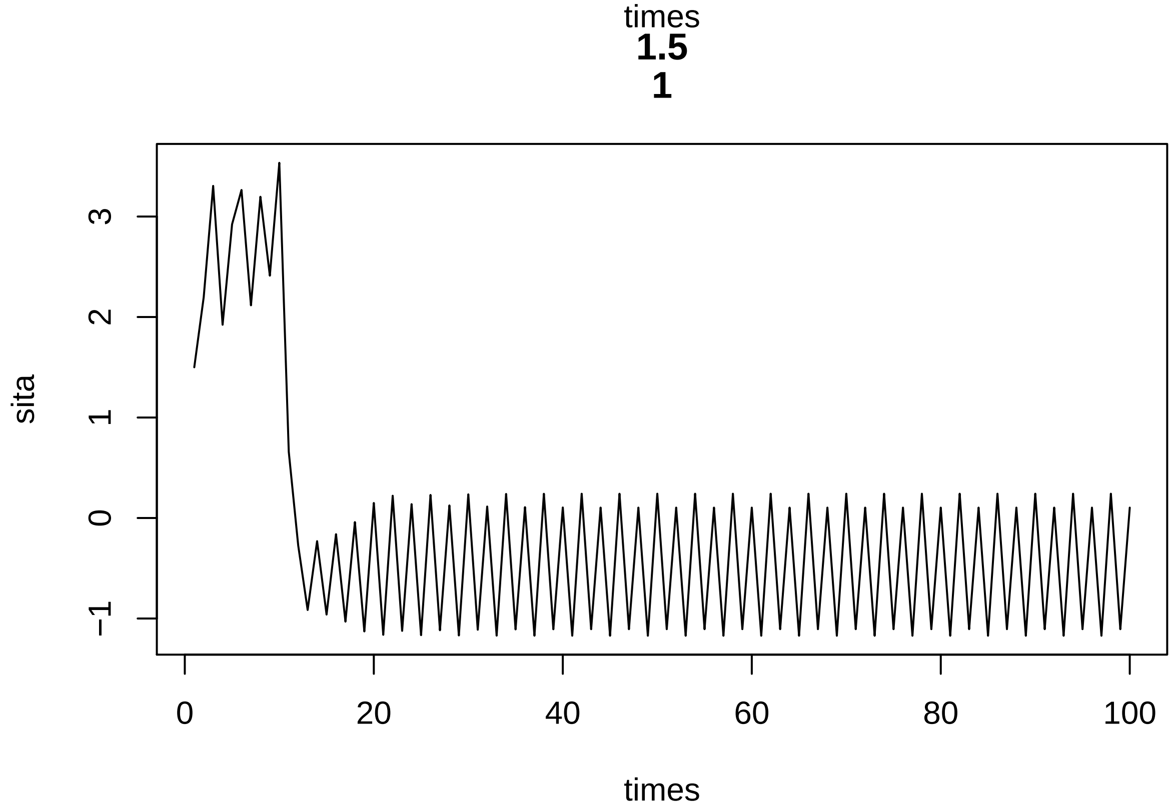
Here are plots1(c)

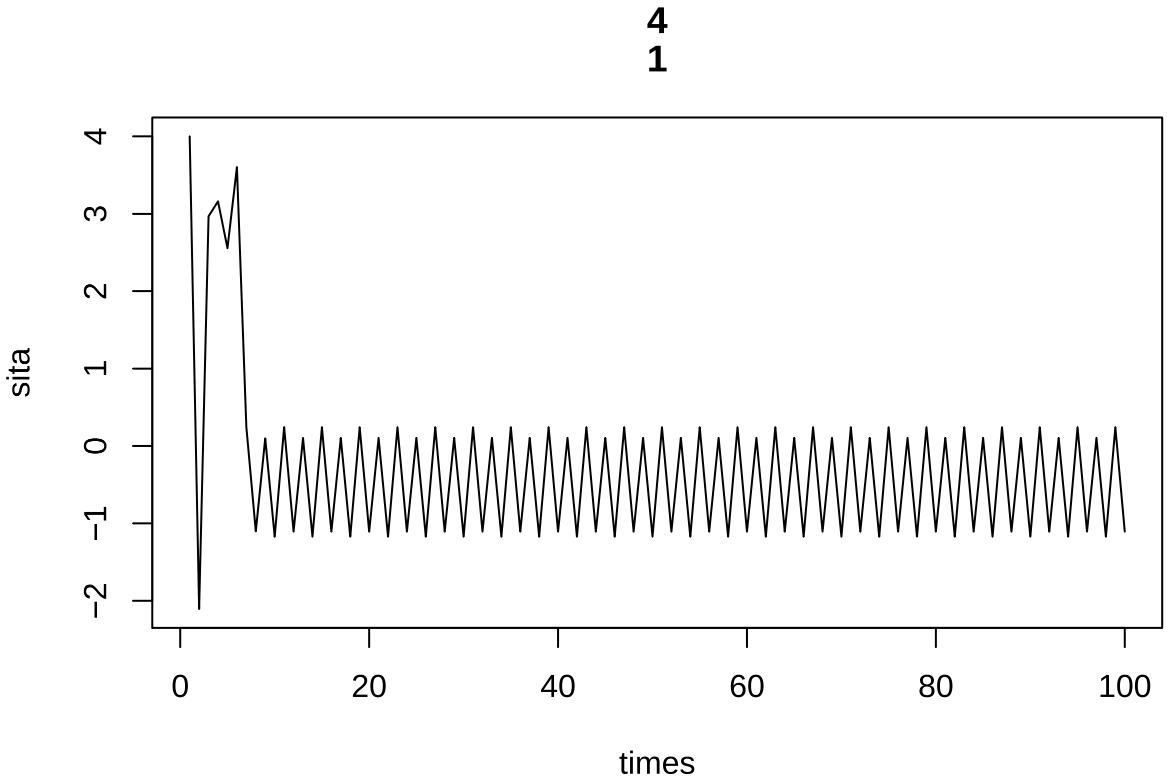
These plots are using different alphas and start points


−11
1
sita
−2
−4
−6
−8
−10
0
20
40
60
80
100
times

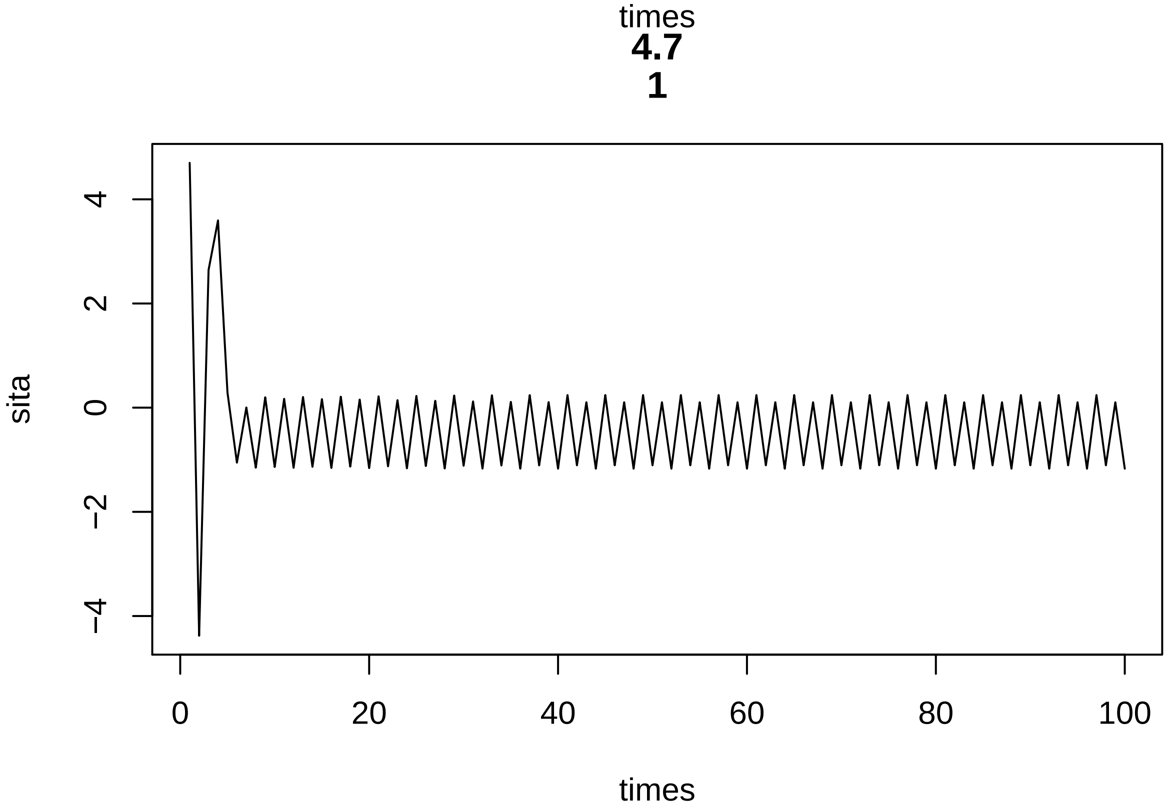

−1
1
sita
0.0
−0.4
−0.8
−1.2
0
20
40
60
80
100
times


0
1
sita
0.0
−0.4
−0.8
−1.2
0
20
40
60
80
100
times

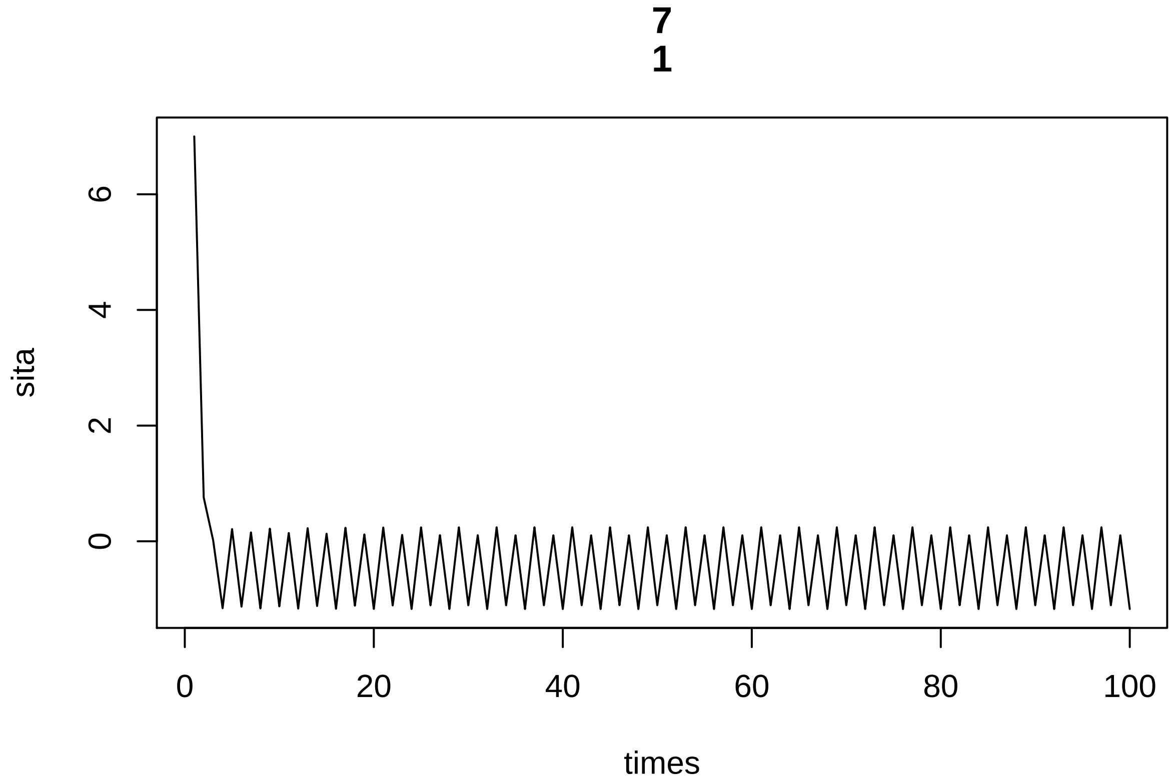

1.5
1
sita
3
2
1
0
−1
0
20
40
60
80
100
times

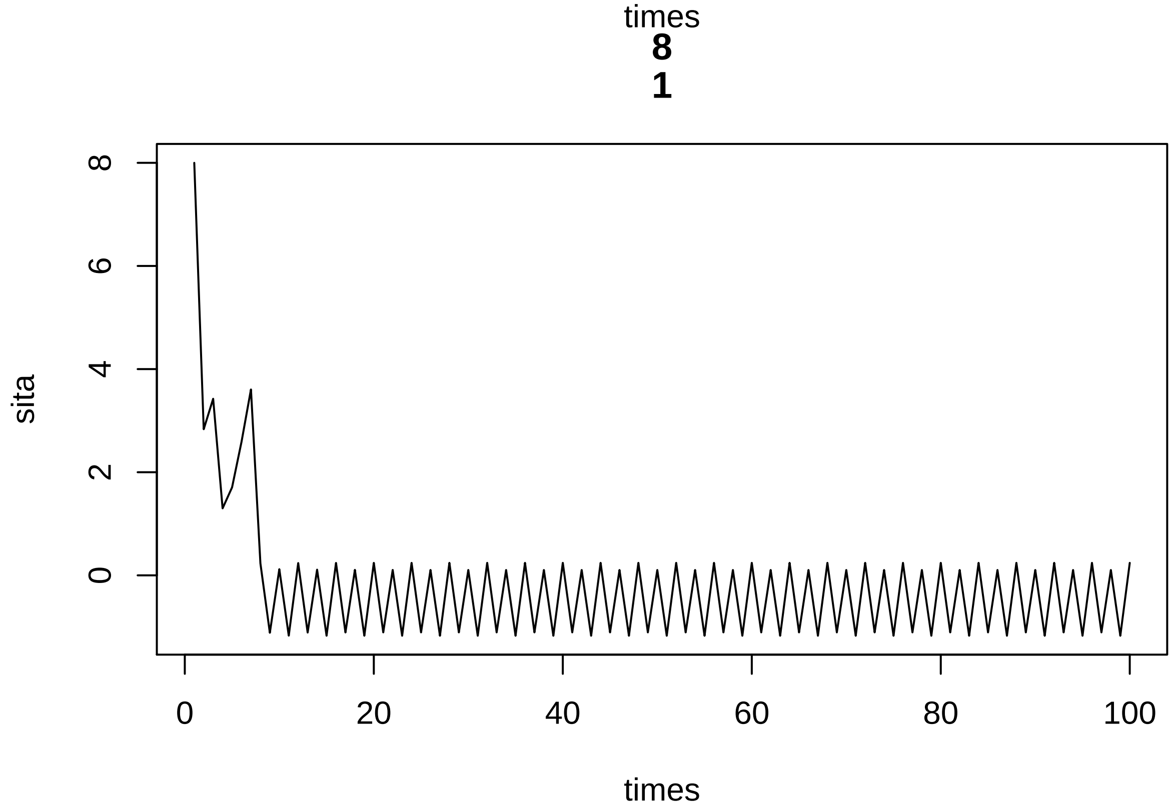


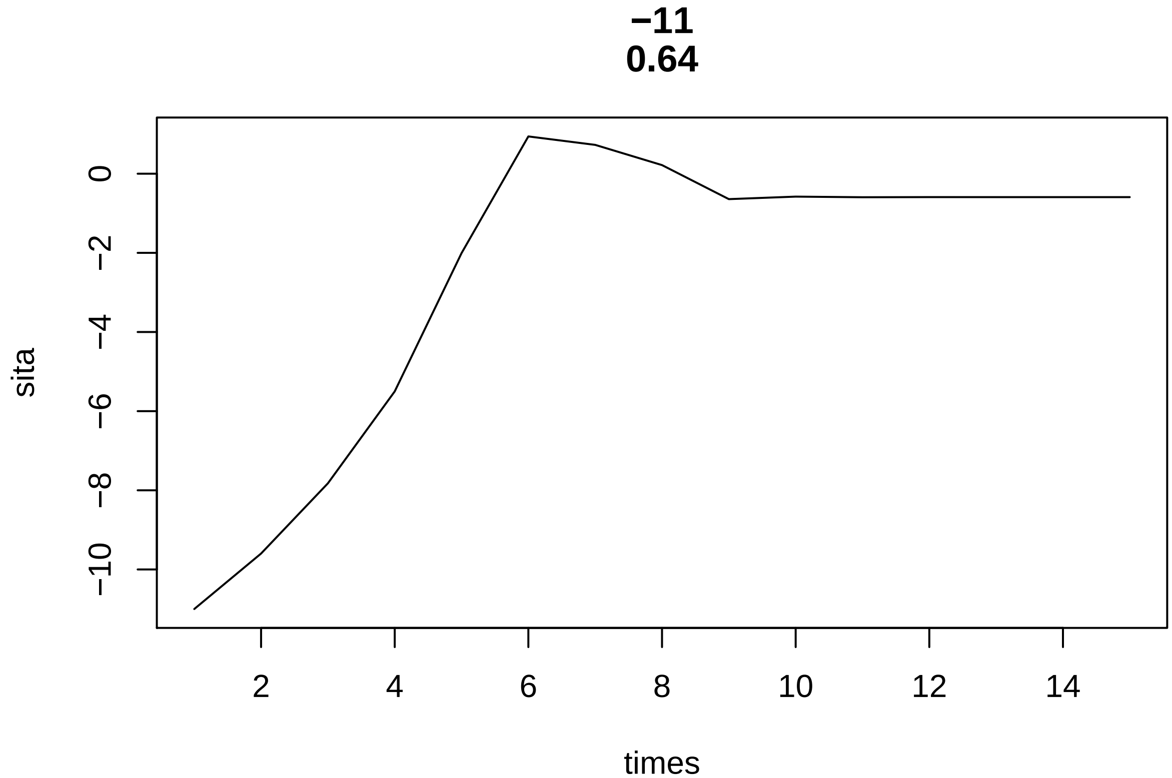
4
1
sita
4
3
2
1
0
−1
−2
0
20
40
60
80
100
times

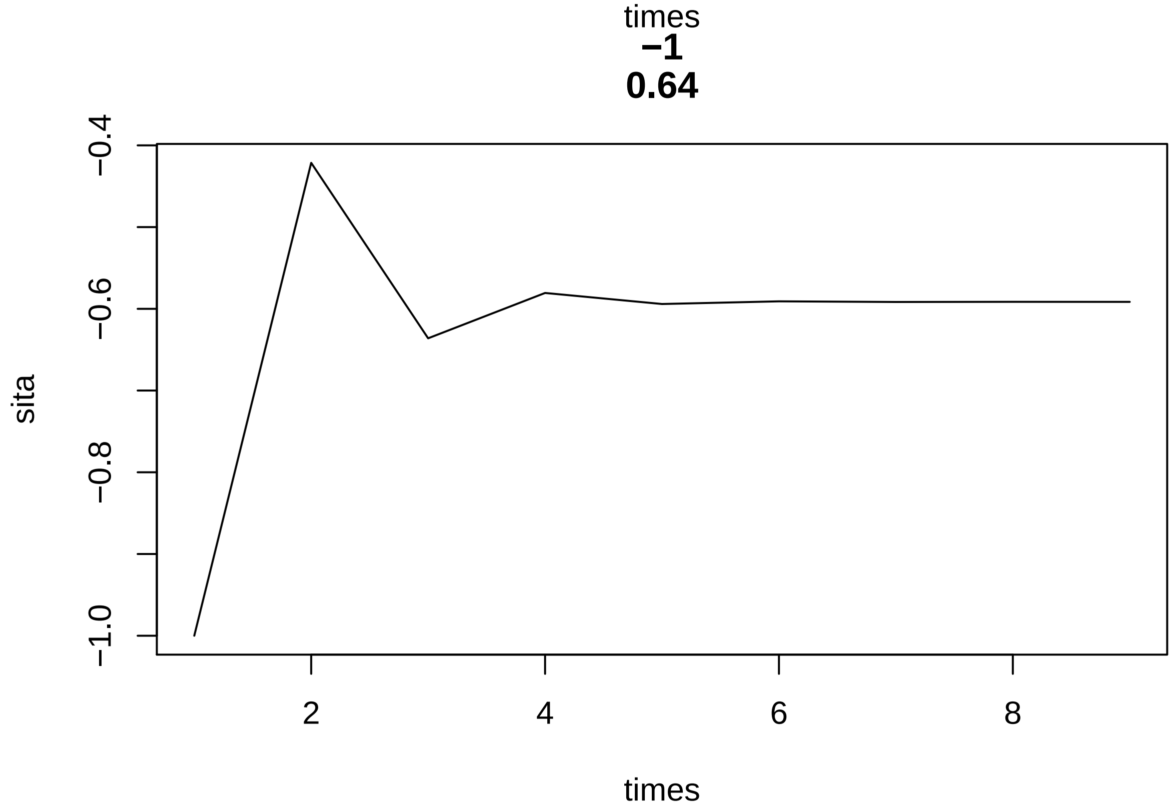


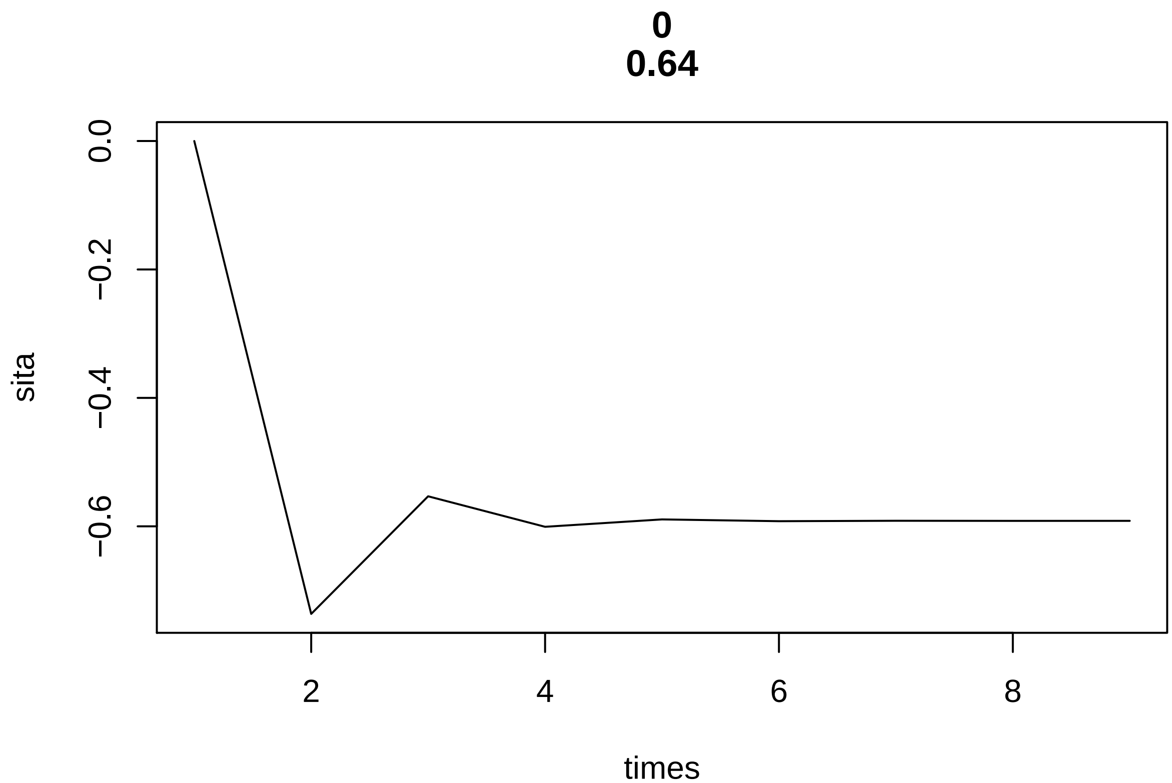
4.7
1
sita
4
2
0
−2
−4
0
20
40
60
80
100
times

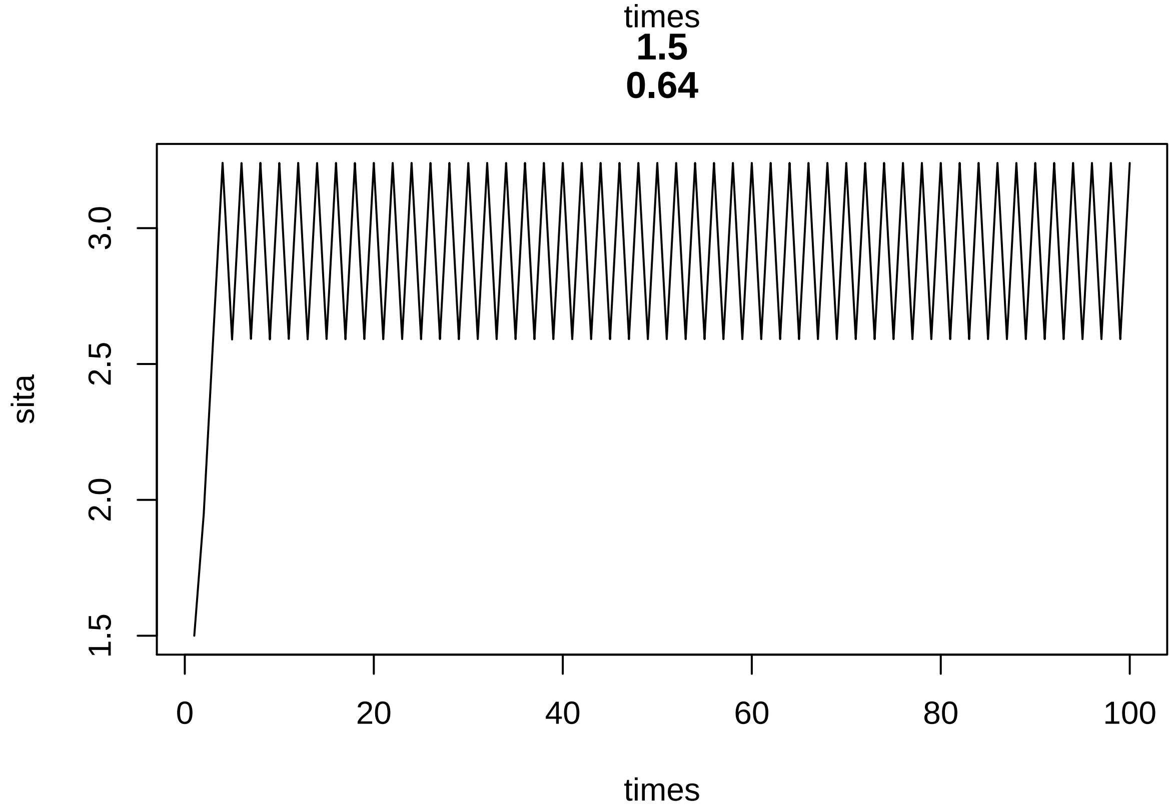

7
1
sita
times
0
2
4
6
0
20
40
60
80
100

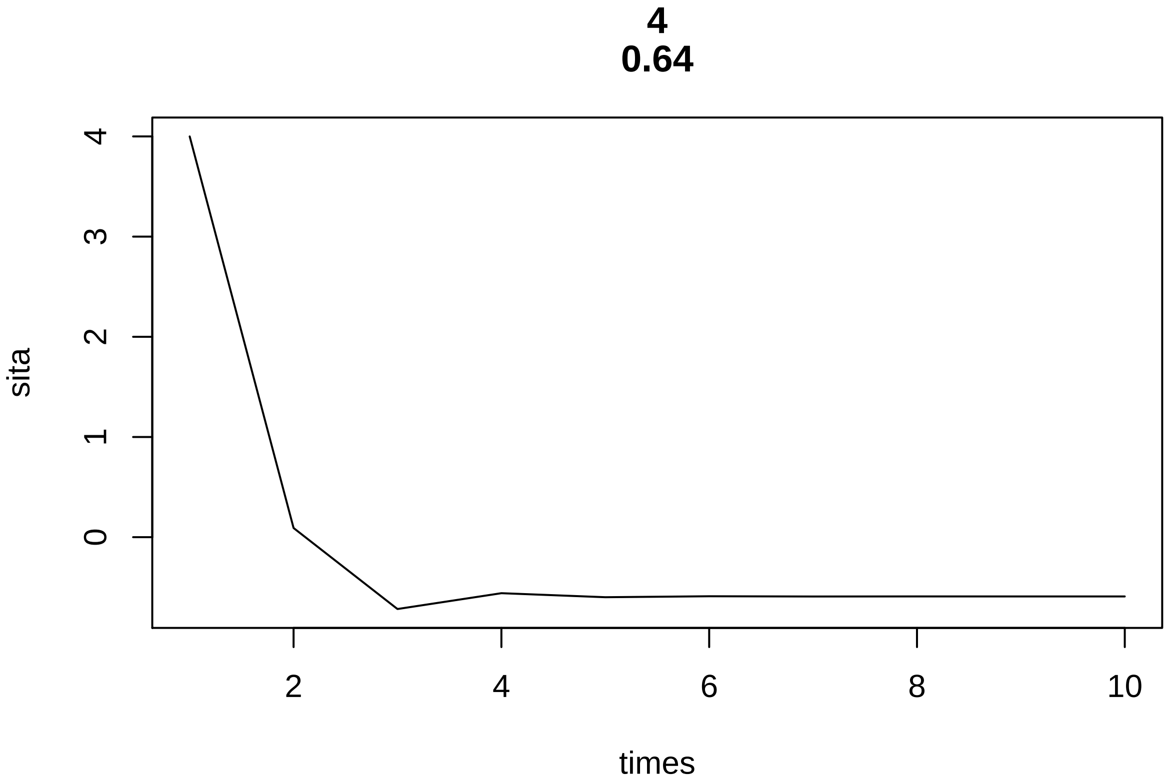

8
1
sita
times
0
2
4
6
8
0
20
40
60
80
100

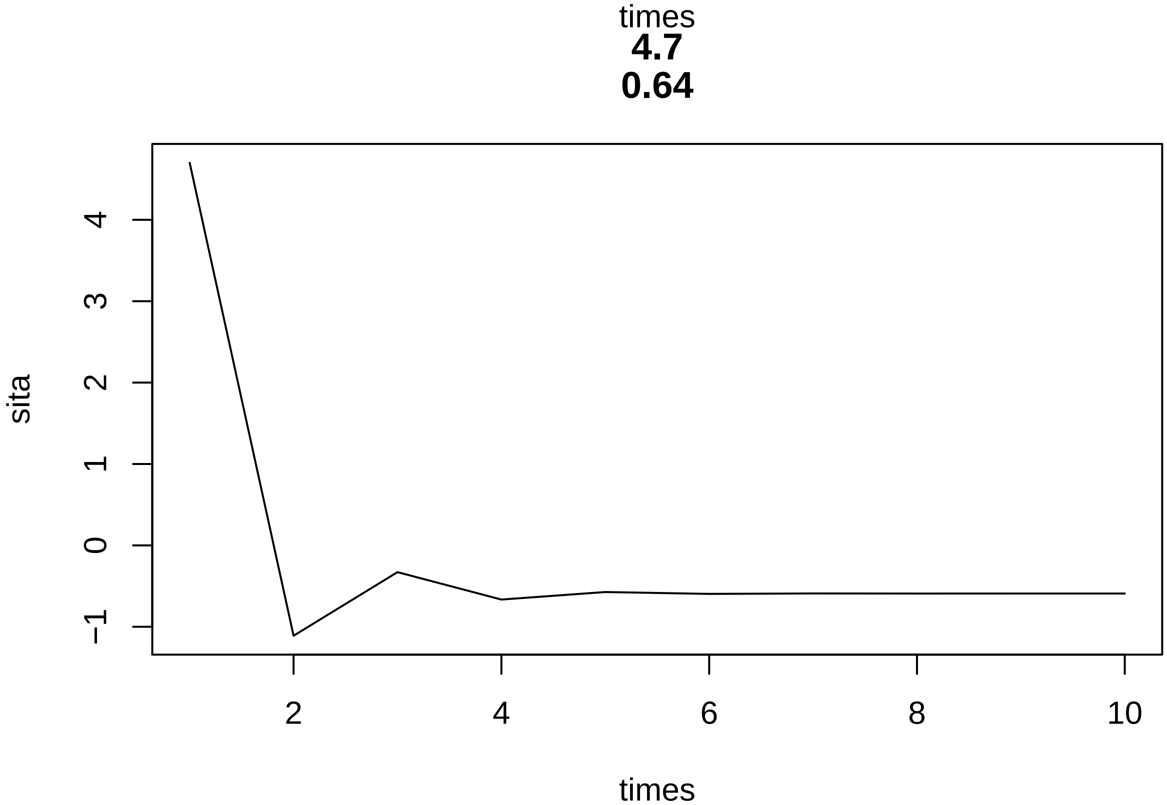

−11
0.64
sita
times
0
−2
−4
−6
−8
−10
2
4
6
8
10
12
14

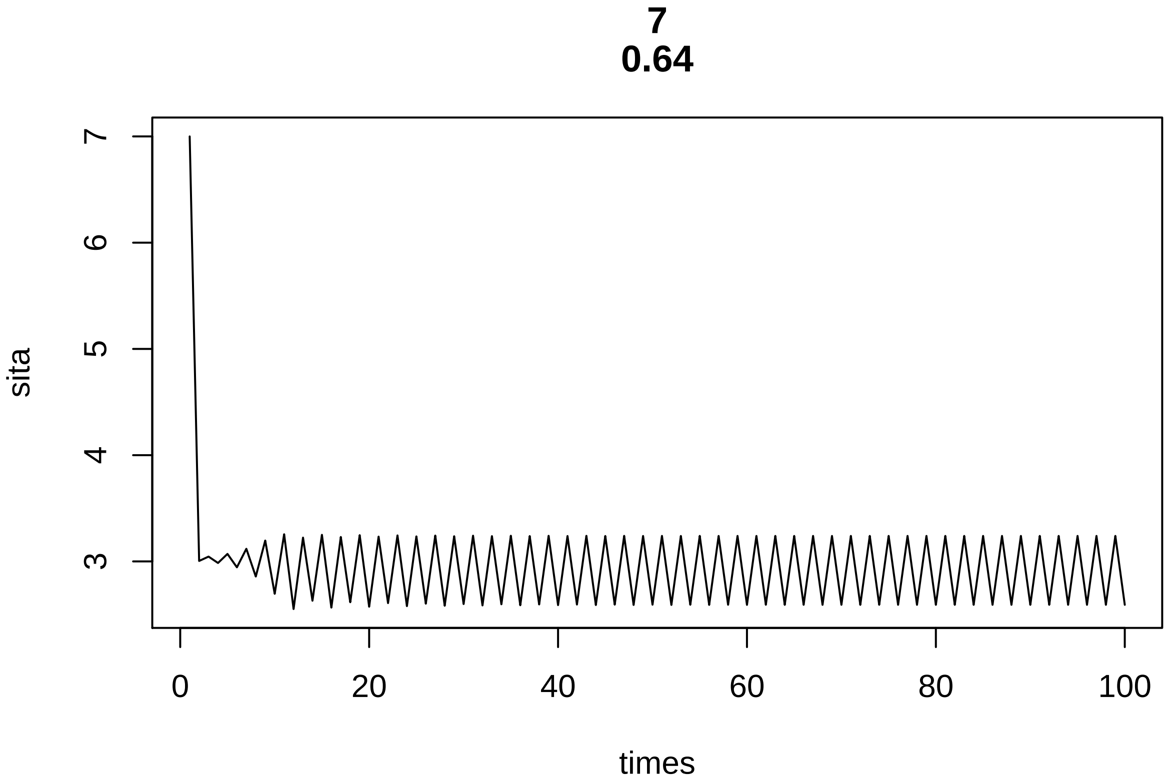

−1
0.64
sita
times
−0.4
−0.6
−0.8
−1.0
2
4
6
8

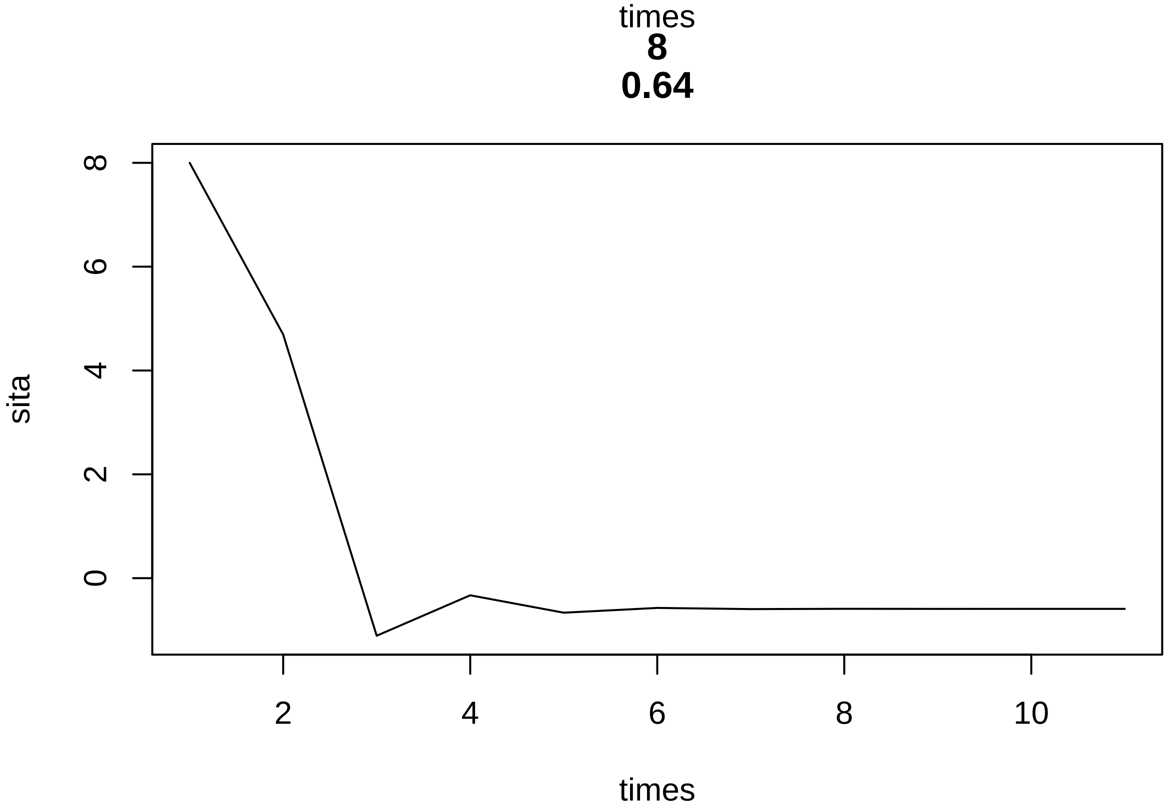

0
0.64
0.0
−0.2
−0.4
−0.6
sita
2
4
6
8
times

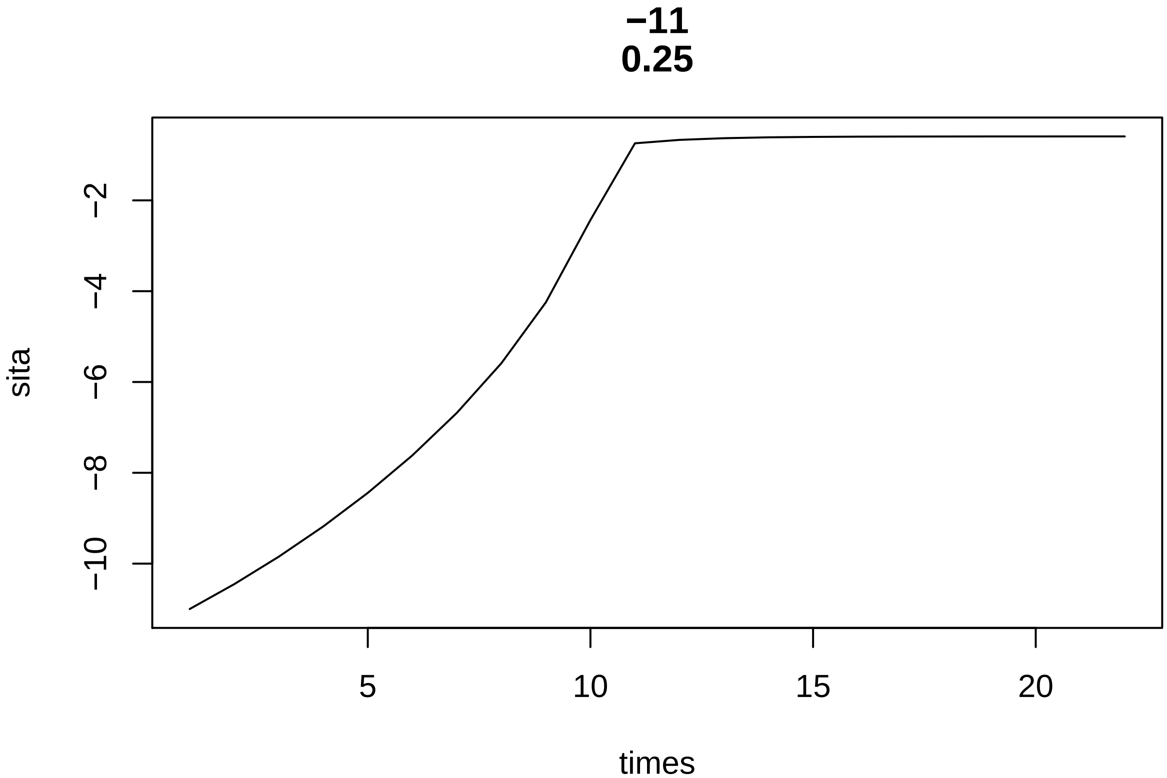

1.5
0.64
3.0
2.5
2.0
1.5
sita
0
20
40
60
80
100
times


4
0.64
sita
4
3
2
1
0
2
4
6
8
10
times

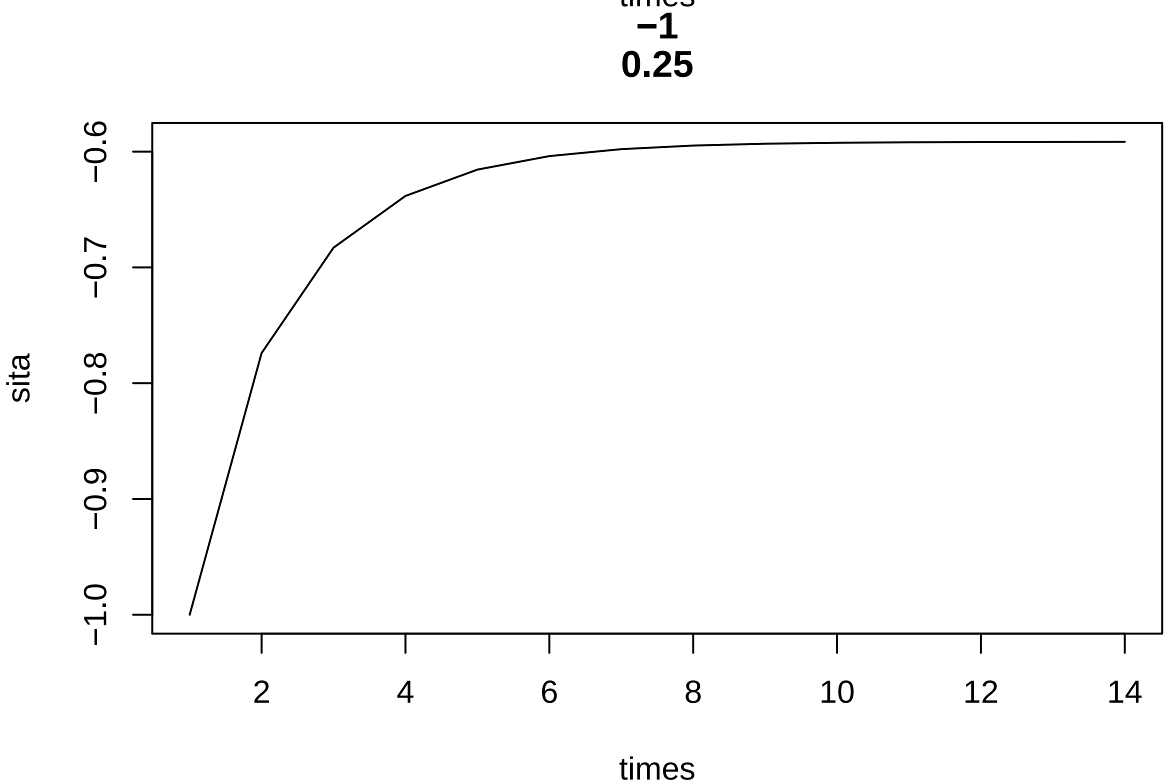

4.7
0.64
sita
4
3
2
1
0
−1
2
4
6
8
10
times


7
0.64
sita
7
6
5
4
3
0
20
40
60
80
100
times

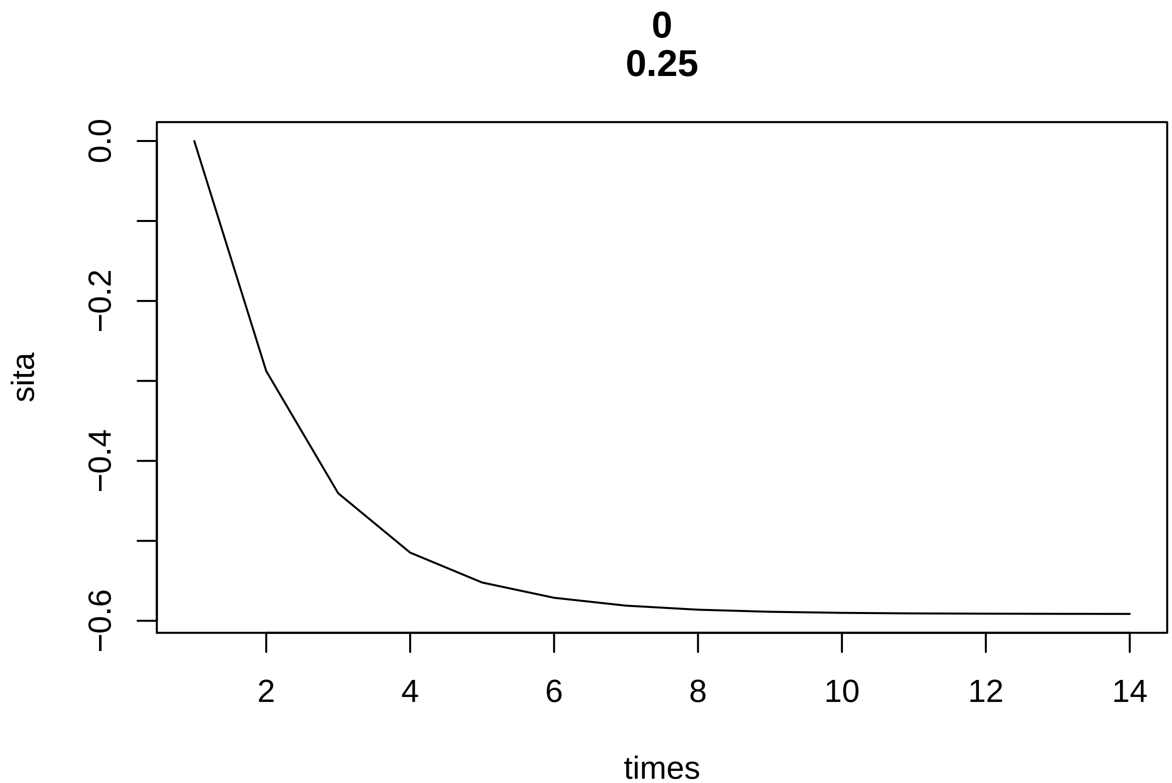

8
0.64
sita
8
6
4
2
0
2
4
6
8
10
times

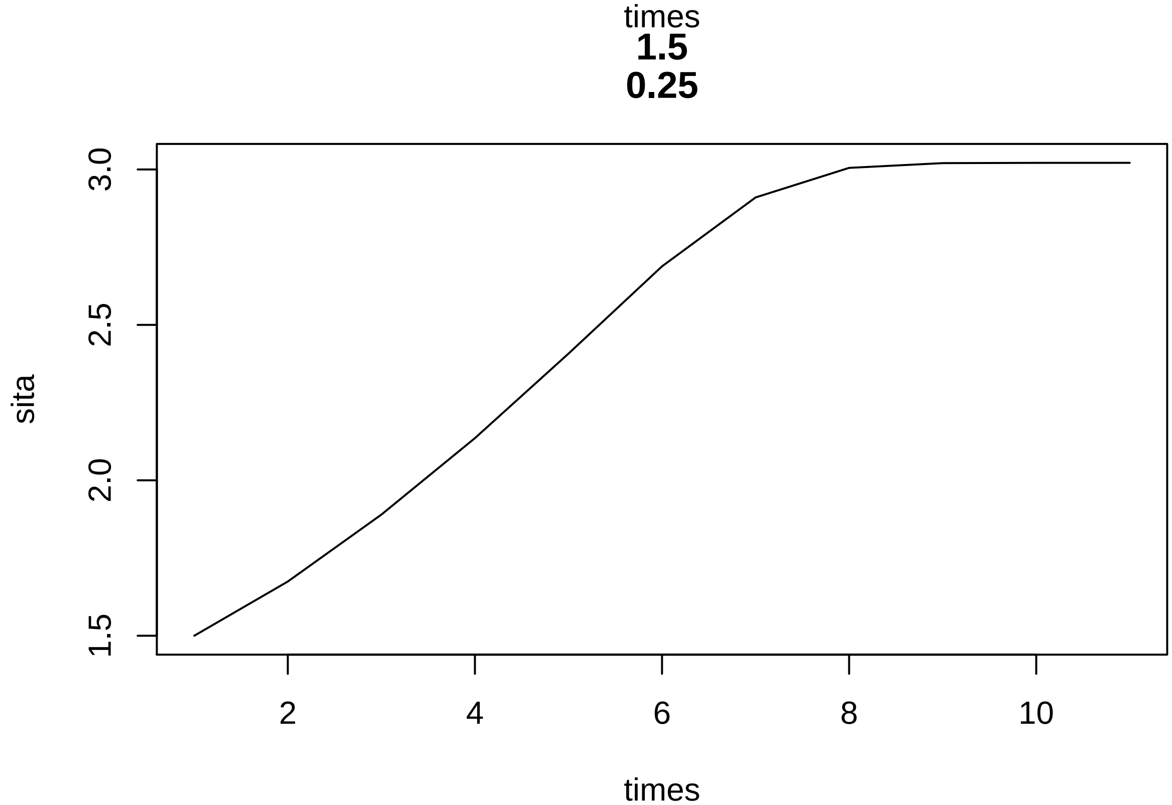


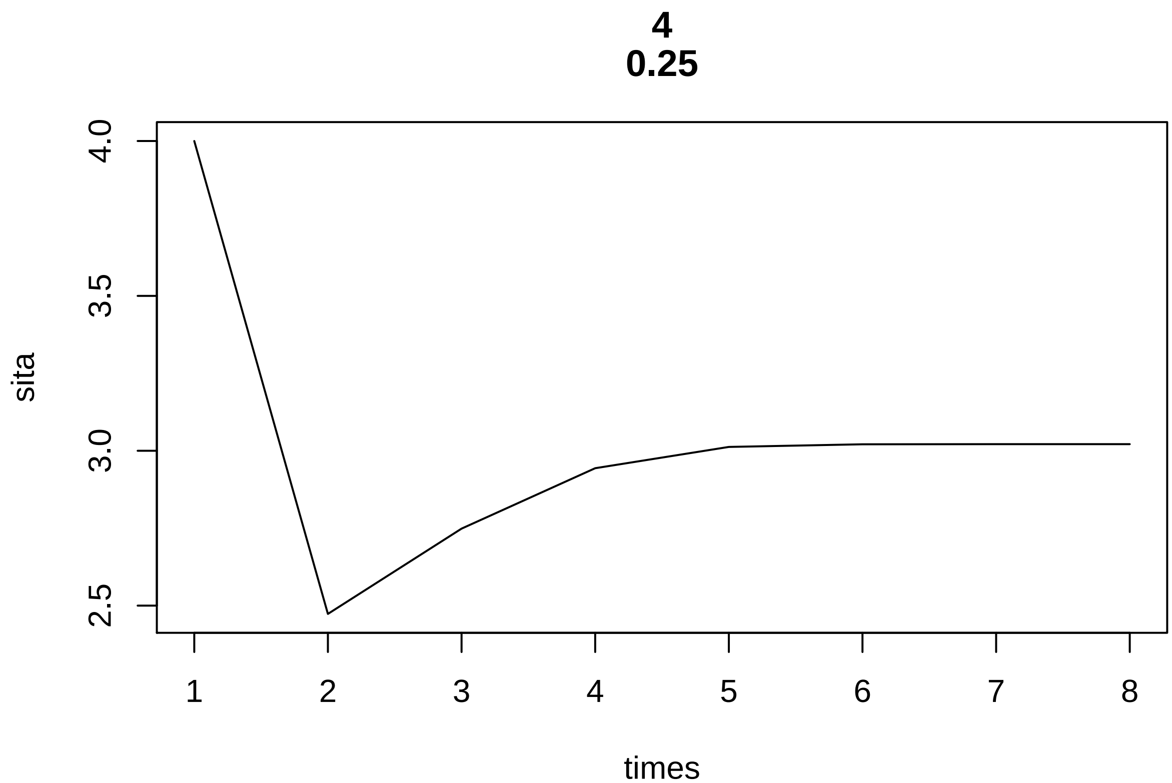
−11
0.25
sita
times
−2
−4
−6
−8
−10
5
10
15
20

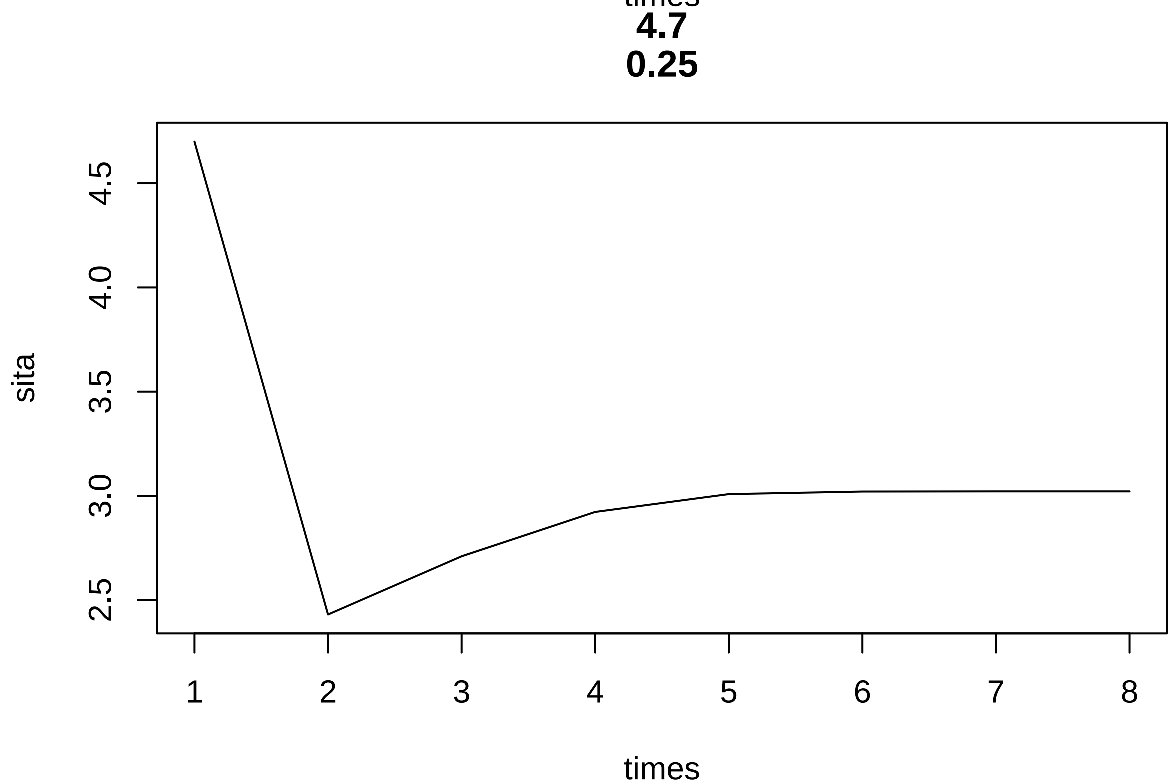


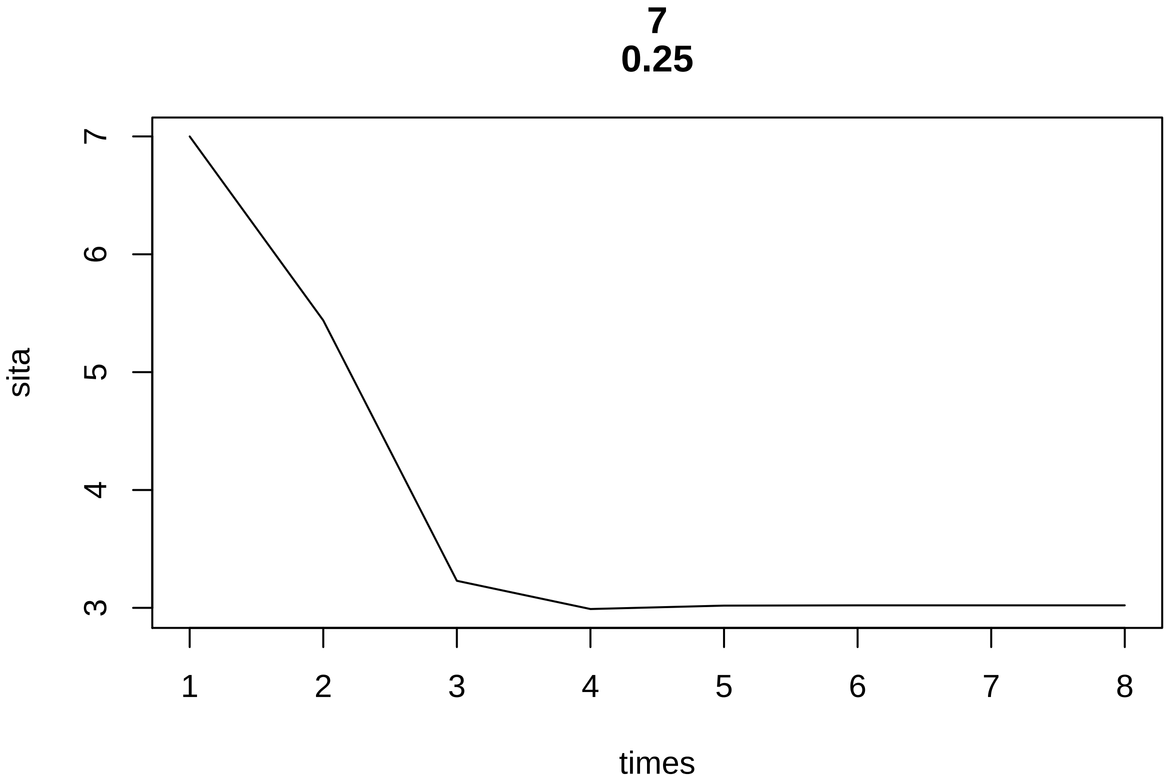
−1
0.25
sita
times
−0.6
−0.7
−0.8
−0.9
−1.0
2
4
6
8
10
12
14

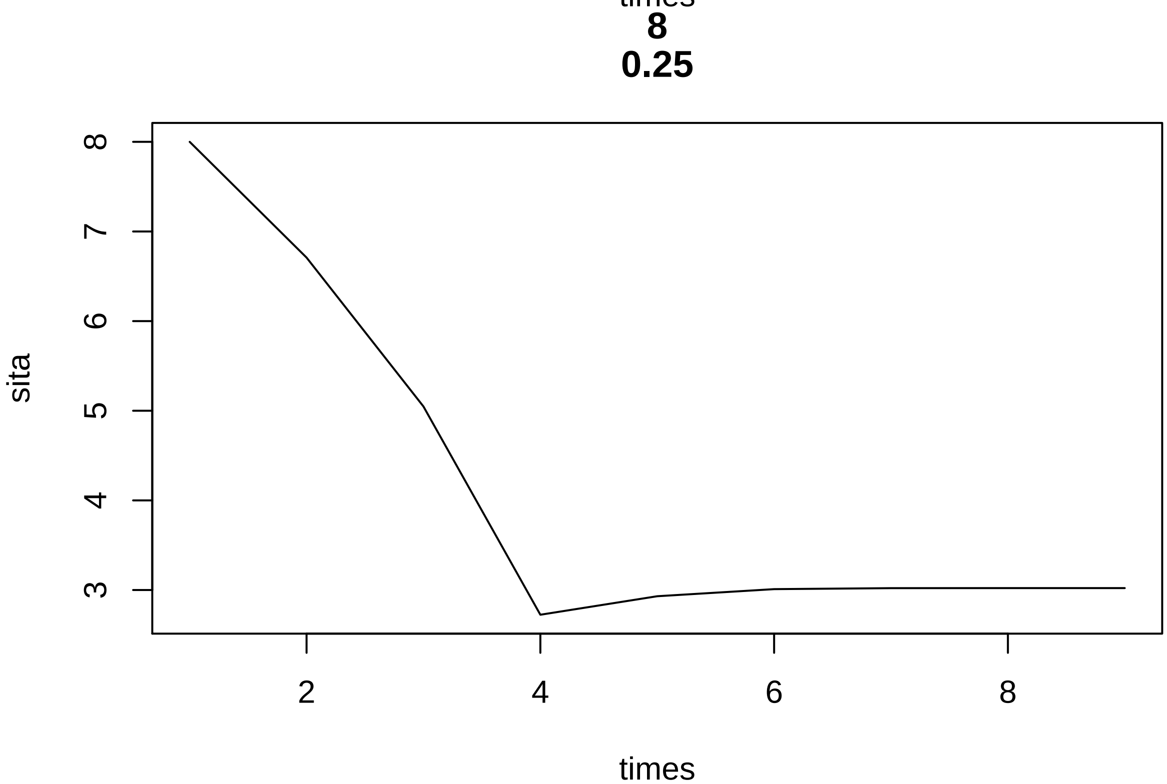

0
0.25
sita
0.0
−0.2
−0.4
−0.6
2
4
6
8
10
12
14
times


1.5
0.25
sita
3.0
2.5
2.0
1.5
2
4
6
8
10
times


4
0.25
sita
4.0
3.5
3.0
2.5
1
2
3
4
5
6
7
8
times


4.7
0.25
sita
4.5
4.0
3.5
3.0
2.5
1
2
3
4
5
6
7
8
times





Question 1(d)

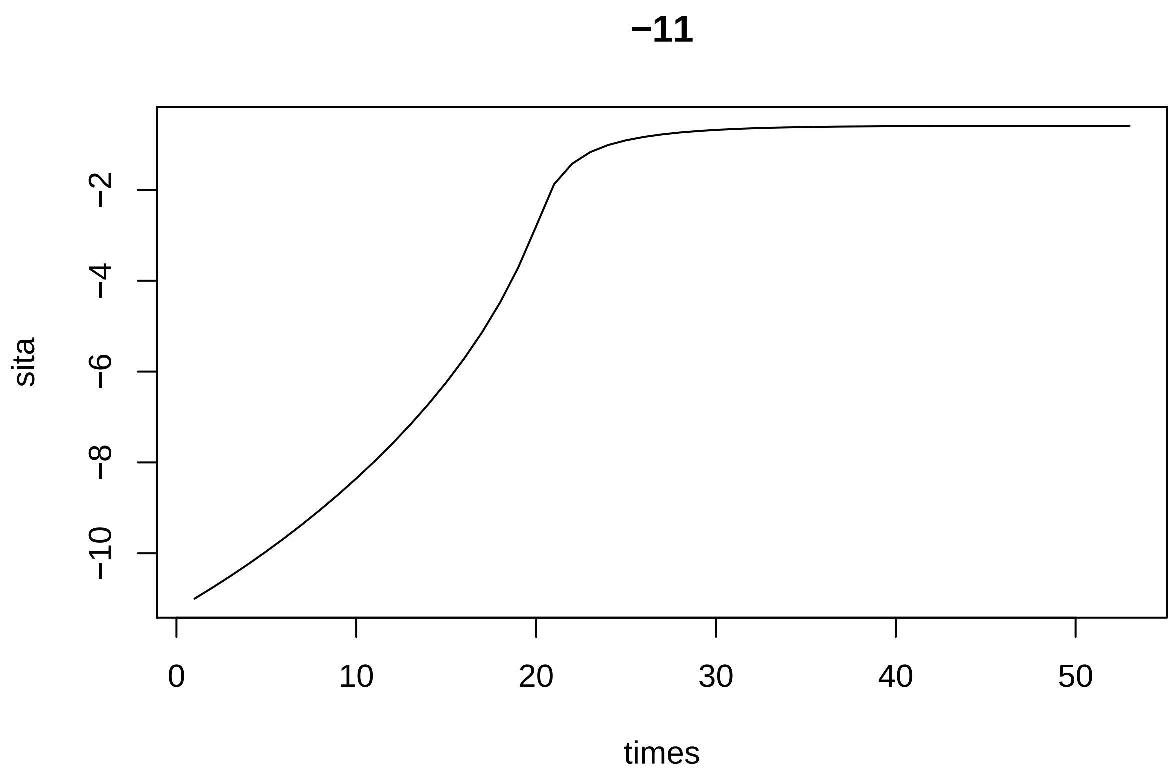
This section we use Fisher Scoring to find MLE, with I(theta) being constant of n/2

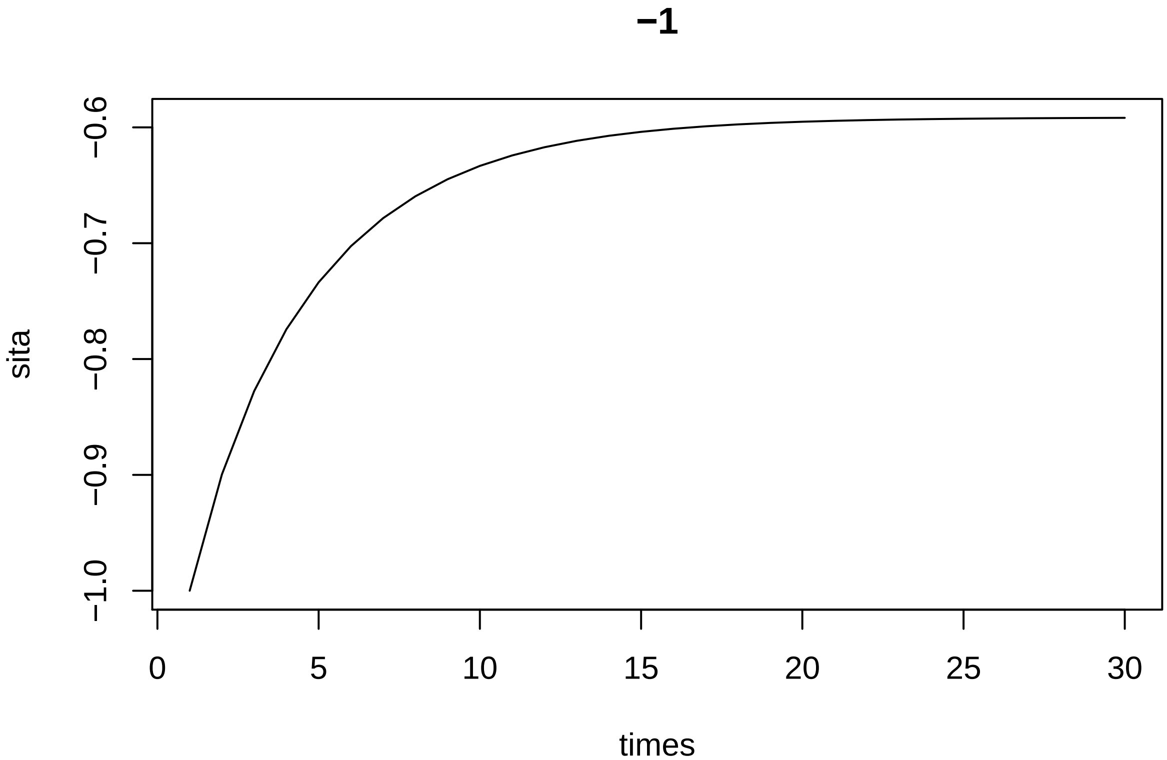
The functions we set up are similar with previous functions, and we compute I

```
derivitive1 <- function(x,sita){
  value <- 0
  for (i in 1:length(x)){
    value <- value-2*(sita-x[i])/(1+(sita-x[i])^2)
  }
  return(value)
}

I <- length(x)/2
```

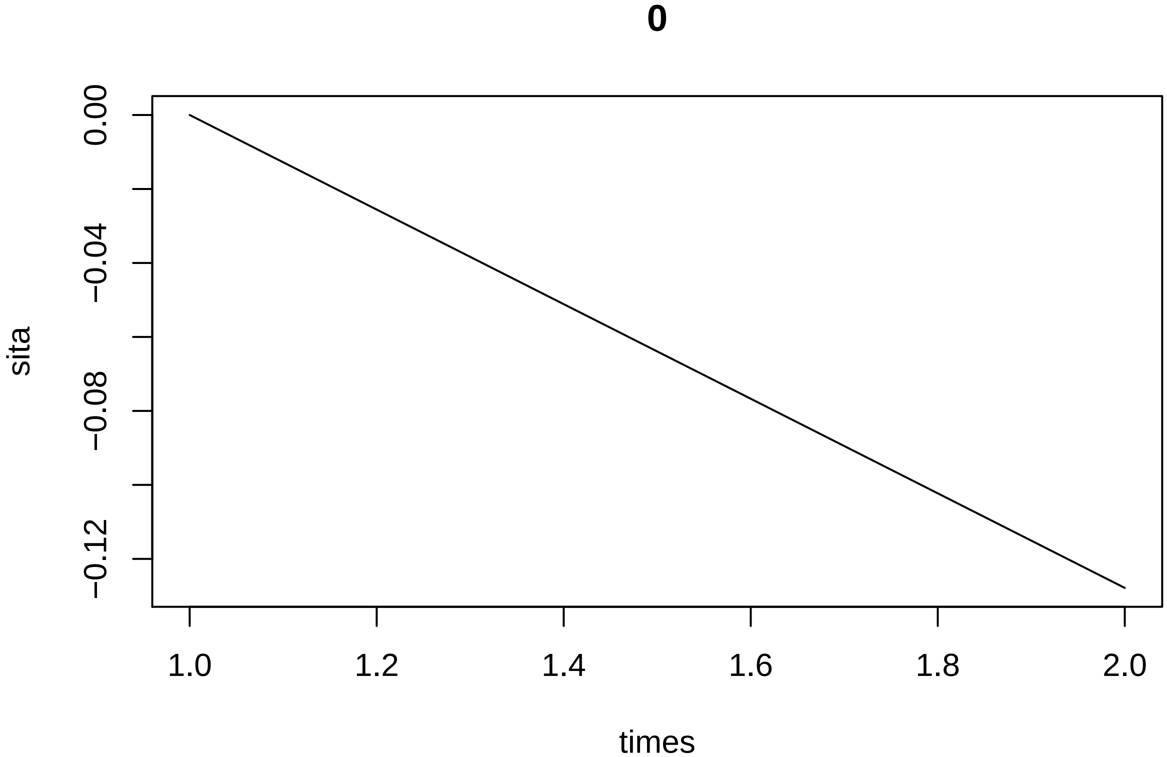
Plots1(d)

These are plots that I use multiple start points

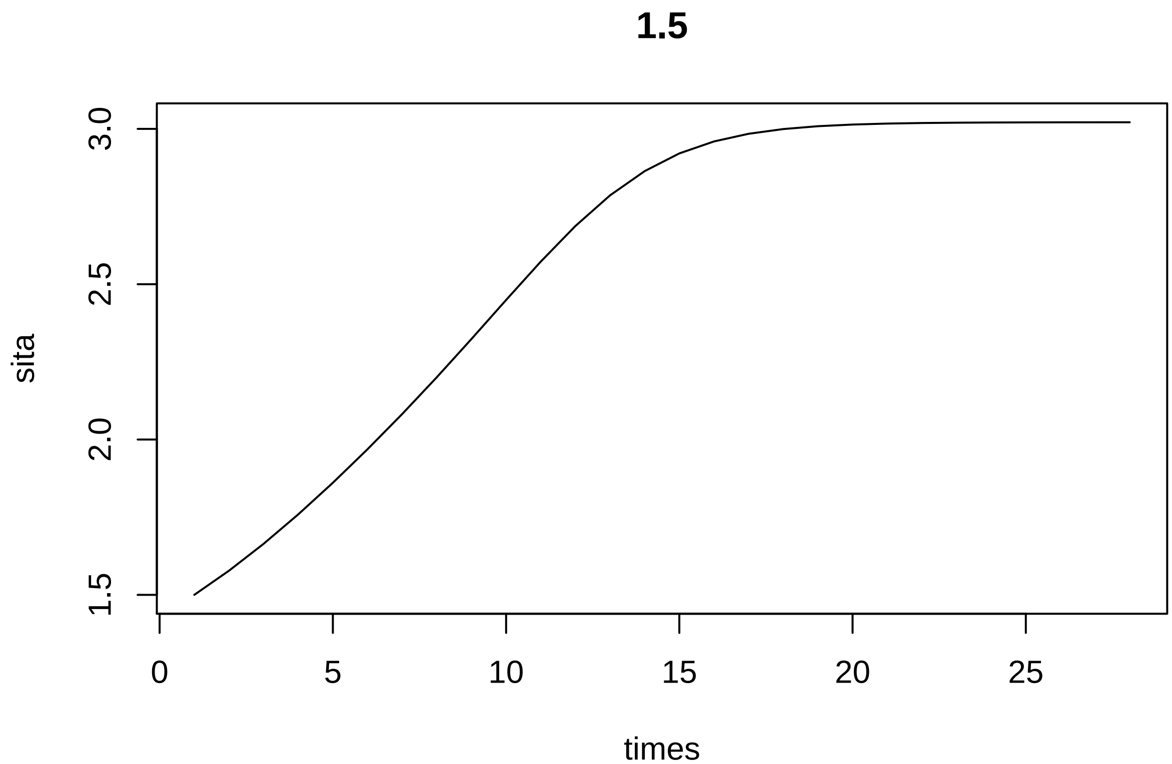


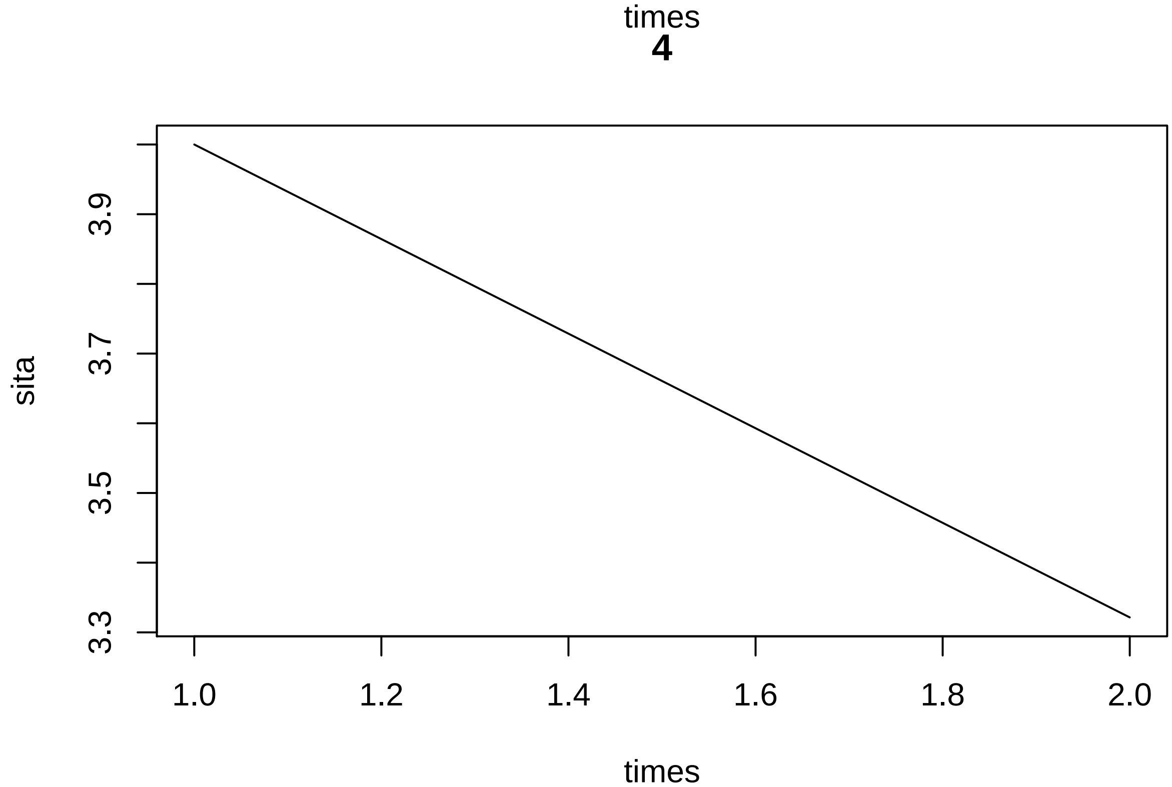


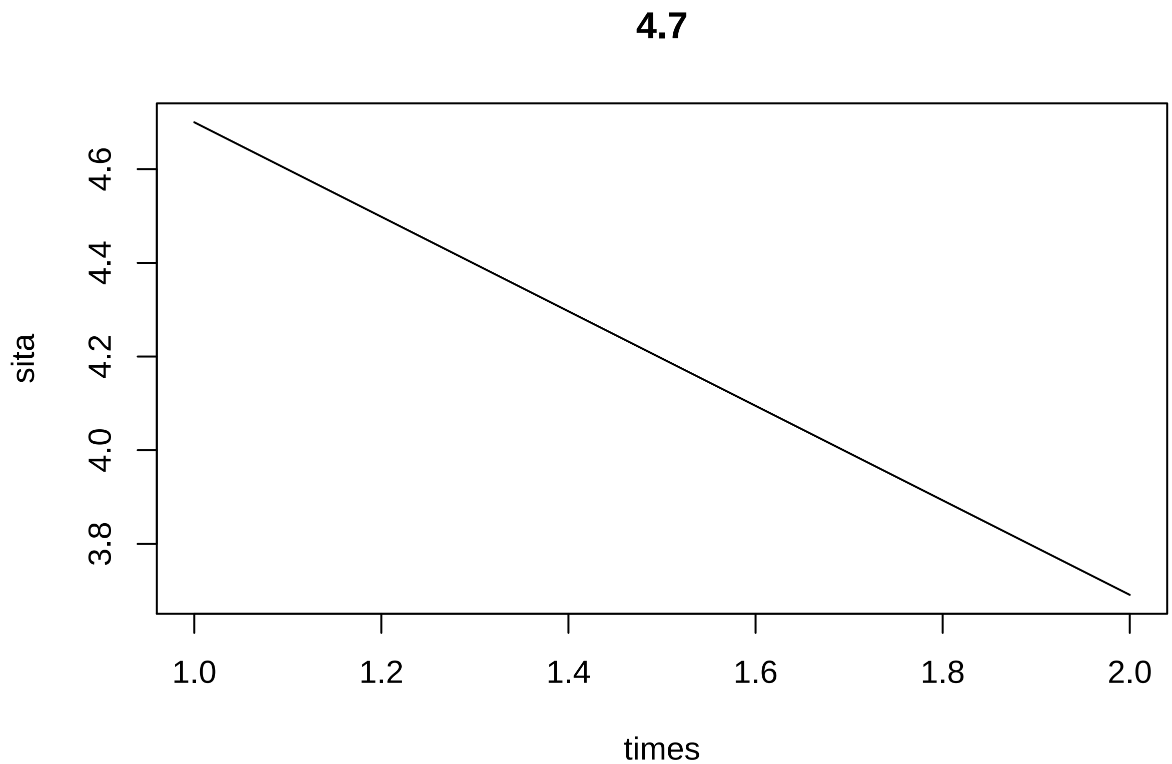
−1
−0.6
−0.7
−0.8
−0.9
−1.0
sita
0
5
10
15
20
25
30
times

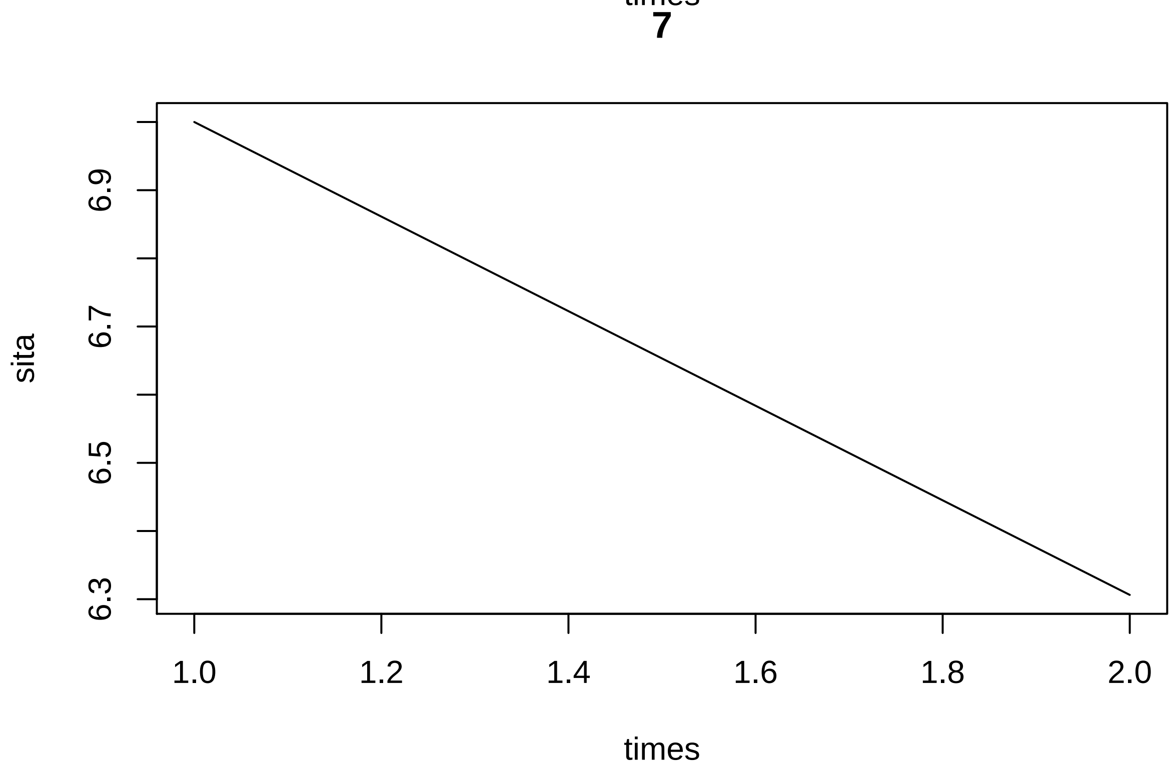


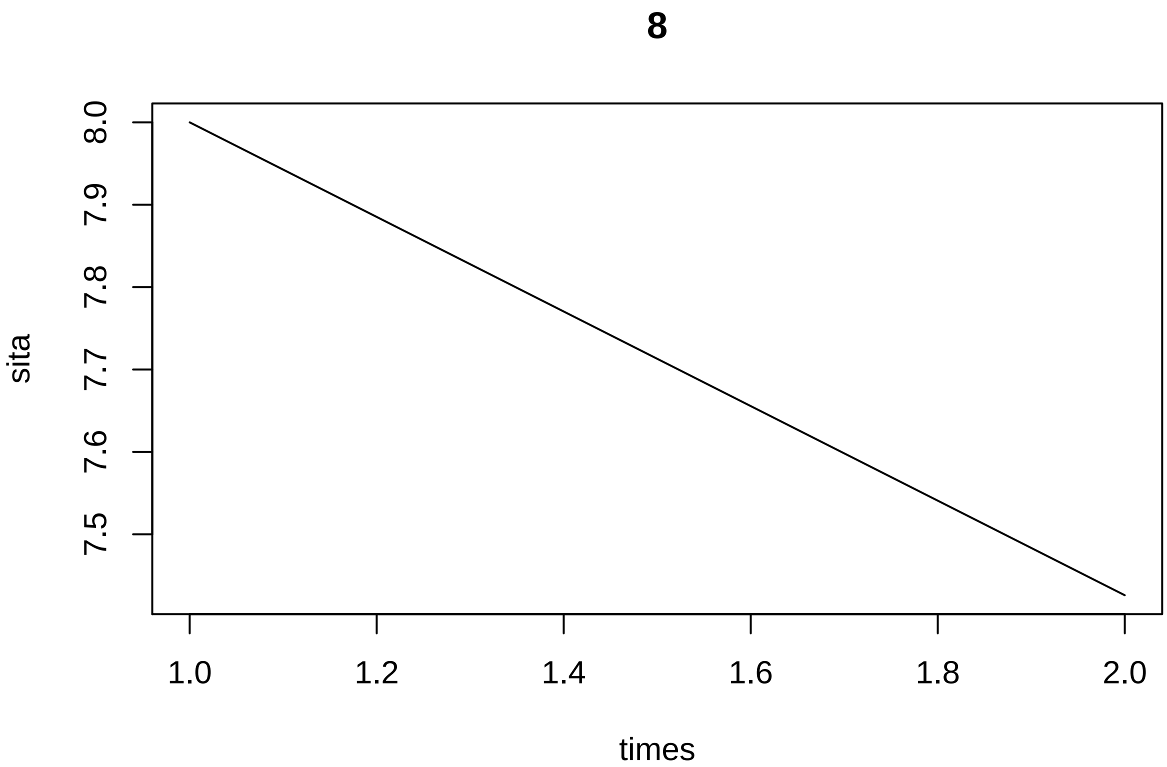
0
0.00
−0.04
−0.08
−0.12
sita
1.0
1.2
1.4
1.6
1.8
2.0
times


1.5
sita
3.0
2.5
2.0
1.5
0
5
10
15
20
25
times


4
sita
3.9
3.7
3.5
3.3
1.0
1.2
1.4
1.6
1.8
2.0
times


4.7
sita
4.6
4.4
4.2
4.0
3.8
1.0
1.2
1.4
1.6
1.8
2.0
times


7
sita
6.9
6.7
6.5
6.3
1.0
1.2
1.4
1.6
1.8
2.0
times



1(e)

With observation of the results and plots, we can see that Newton method has the highest speed and stability. For Fisher scoring, it has lower speed, but we can see from the plots, with alpha being some point the line is fluctuating up and down, so it has lowest stability. And for fixed point method, the speed of it is lowest but is relatively stable. But fixed-point has its disadvantage, not being able to utilize some points.

Qusetion 2

The likelihood function

$$E[X|\theta] = \pi + \sin(\theta) \tag{29}$$

two points

```r
target <- asin(mm-pi)
print(target)
```

```
## [1] 0.09539407
```

```r
target1 <- pi-target
print(target1)
```

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

The sequence of same converge start points

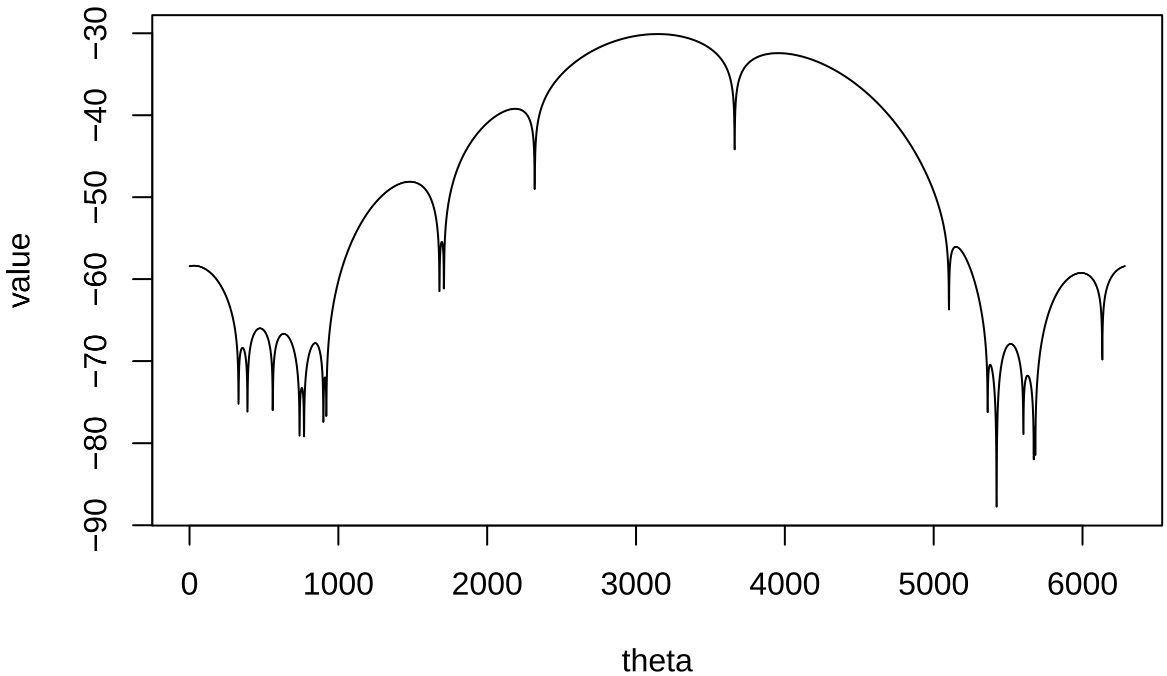
```r
loglikelihood <- function(x,theta){
  value <- 0
  for (i in 1:length(x)){
    value <- value+log((1-cos(x[i]-theta))/(2*pi))
  }
  return(value)
}
for (i in 1:length(result)){
  if (abs(result[i]-result[j])>10^(-4)){
    j <- i
    num <- append(num,j-1)
   }
}
num <- unique(num)
print(unique(num[1:length(num)-1]))
```

```
## [[1]]
## [1] 11
##
## [[2]]
## [1] 13
##
## [[3]]
## [1] 18
##
## [[4]]
## [1] 24
##
## [[5]]
## [1] 25
##
## [[6]]
## [1] 29
##
## [[7]]
## [1] 30
##
## [[8]]
## [1] 54
##
## [[9]]
## [1] 55
##
## [[10]]
## [1] 74
##
## [[11]]
## [1] 116
##
## [[12]]
## [1] 162
##
```

```
## [[13]]
## [1] 170
##
## [[14]]
## [1] 172
##
## [[15]]
## [1] 178
##
## [[16]]
## [1] 180
##
## [[17]]
## [1] 195
```

Plots2

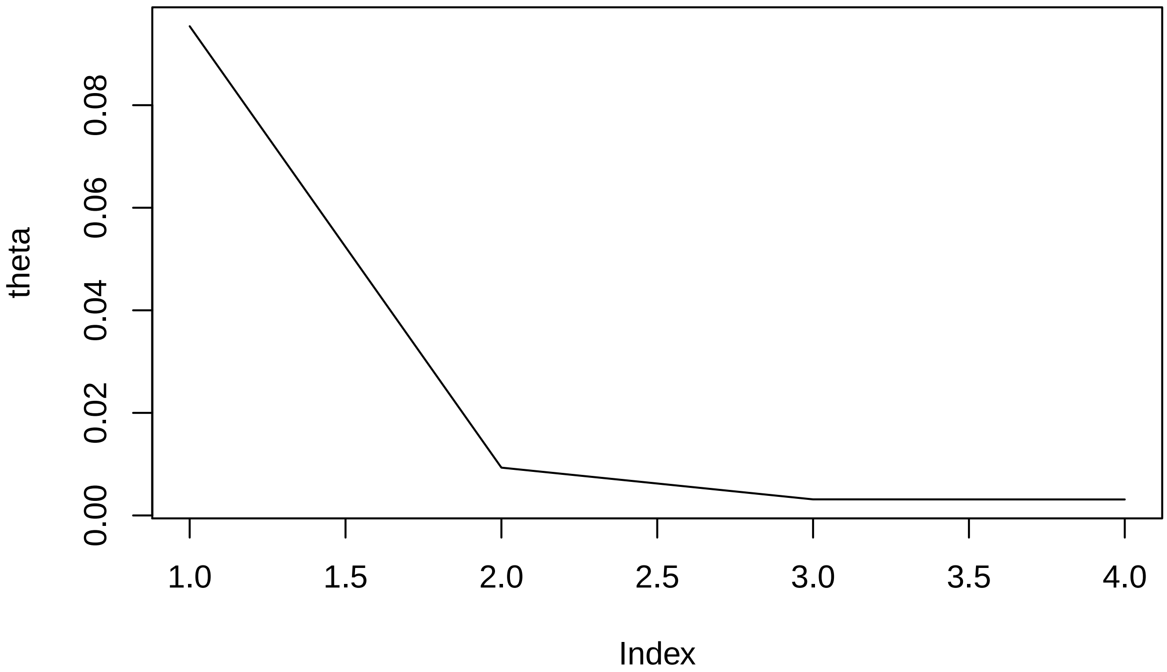
We plot the value of our target



we also plot the picture with strat point being 'target'

```
MLE <- function(theta){
  diff <- 1
  i <- 1
  while(abs(diff)>10^(-4)){
    theta[i+1] <- theta[i]-derivetive1(x,theta[i])/derivetive2(x,theta[i])
    diff <- theta[i+1]-theta[i]
    i <- i+1
  }
  plot(theta,type='l')
  return(theta)
}
```

```
theta <- array()
theta[1] <- target
s <- MLE(theta)
```



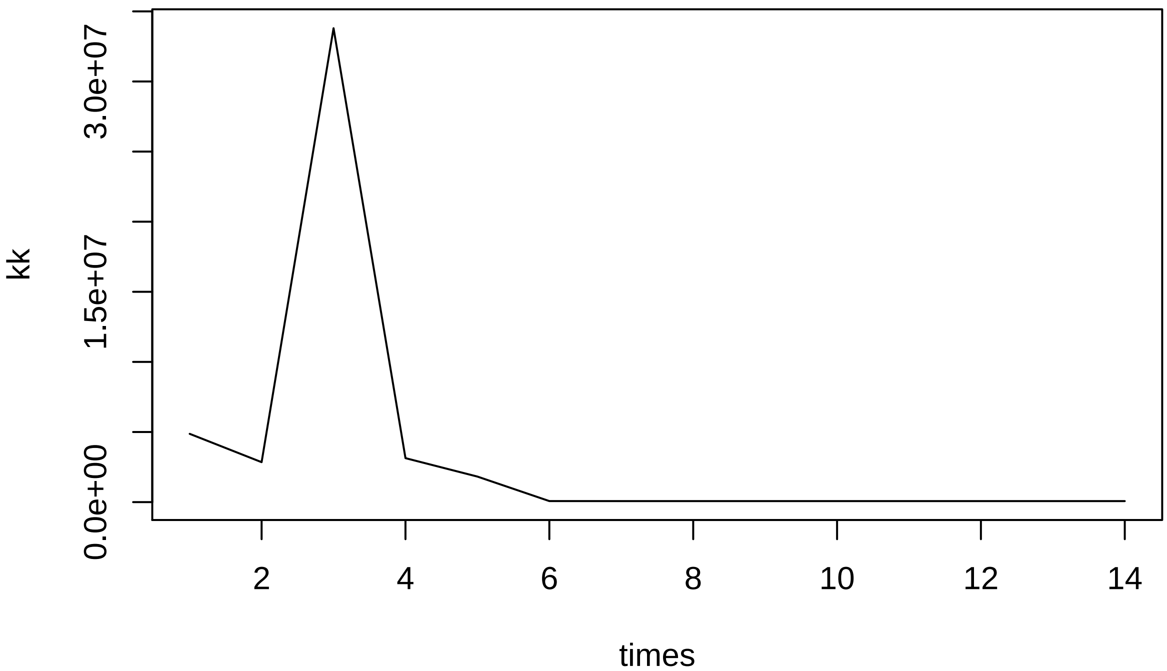
Qustion 3(b)

Functions

Here are matrix we need to establish to build our function

```
A_matrix <- function(r,k,times){
  A <- matrix(nrow=times,ncol=2)
  for (i in 1:times){
    A[i,1] <- (N0^2-N0^2*exp(-1*r*beetles$days[i]))/((N0+(k-N0)*exp(-1*r*beetles$days[i]))^2)
    A[i,2] <- beetles$days[i]*(k-N0)*exp(-1*r*beetles$days[i])*k*N0/((N0+(k-N0)*exp(-1*r*beetles$days[i
  }
  A
}


Z_matrix <- function(r,k,times){
  z <- matrix(nrow=times,ncol=1)
  for (i in 1:times){
    z[i,1] <- beetles$beetles[i]-k*2/(2+(k-2)*exp(-r*beetles$days[i]))
  }
  z
}
```

Sum of squared errors



Question 3(c)

This time we need to optimize the value of r, K, sigma

Partial Derivative

```
d_sigma <- as.expression(D(u,'sigma'))
dd_sigma <- as.expression(D(d_sigma,'sigma'))

d_k <-as.expression(D(u,'k'))
dd_k <- as.expression(D(d_k,'k'))

d_r <- as.expression(D(u,'r'))
dd_r <- as.expression(D(d_r,'r'))
```

3c

I use Newton method to optimize the r,K, and square of sigma. These numbers are

Times we use

```
## [1] 37
```

The results

```
print(kF)
```

```
## [1] 820.38
```

```r
print(rF)
```

```
## [1] 0.1926401
```

```r
print(sigmaF)
```

```
## [1] 0.9108726
```