MLE&MAP3

LinyiGuo

2019/10/6

Contents


```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang
```

Grid search

Grid search is to calculate the MLE or MAP, which could avoid local minimum value.

Likelihood & Loglikelihood

```
# define likelihood function

likelihood <- function(theta, Trend, Season, Obs){
  n <- length(Obs)
  a <- 0
  for (i in 12:n)  a <- a + (sum(Season[(i-11):i]))^2
  L <- (1/sqrt(2*pi*prod(theta)))^n *
          exp(-sum((Obs - Trend - Season)^2)/(2*theta[1])) *
          exp(-sum((Trend[-1]-Trend[-n])^2)/(2*theta[2])) *
          exp(- a/(2*theta[3]))
  return(L)
}


# loglikelihood
loglikelihood <- function(theta, Trend, Season, Obs ){
```

```r
  n <- length(Obs)
  a <- 0
  for (i in 12:n)  a <- a + (sum(Season[(i-11):i]))^2
  l <- (-n/2) * log(theta[1]) - n/2 * log(theta[2]) - n/2 * log(theta[3]) -
    sum((Obs-Trend-Season)^2)/(2*theta[1]) -
    sum((Trend[-1]-Trend[-n])^2)/(2*theta[2]) -
    a / (2*theta[3])
  return(l)
}
```

Grid Search for Single Parameter

Here, I fix two variances at **1** and let the other one is a sequence with 10000 points.

1. variances in observation function are $0.001, 0.002, \ldots, 10$
2. variances in trend function are $0.01, 0.02, \ldots, 100$
3. variances in seasonal function are $0.01, 0.02, \ldots, 100$

MLE

This is my code for related function:

```r
# define a 'large' function to return the loglikelihood series of one single parameter's sequence
loglikelihood_single_gs <- function(data, type){
  if(type == 1) {
    l1 <- c()
    for (i in 1:10000){
      ssm <- SSModel(data ~ SSMtrend(1, Q=list(1)) +
                     SSMseasonal(12, sea.type = 'dummy', Q=1), H=i*0.001)
      mod <- KFS(ssm)
      Trend <- coef(mod, states = 'trend')
      Season <- rowSums(coef(mod, states = 'seasonal'))
      theta <- c(i*0.001,1,1)
      l1 <- c(l1, loglikelihood(theta=theta, Trend=Trend, Season=Season, Obs = data))
    }
    return(l1)
  }

  if(type == 2){
    l2 <- c()
    for (i in 1:10000){
      ssm <- SSModel(data ~ SSMtrend(1, Q=list(i*0.01)) +
                     SSMseasonal(12, sea.type = 'dummy', Q=1), H=1)
      mod <- KFS(ssm)
      Trend <- coef(mod, states = 'trend')
      Season <- rowSums(coef(mod, states = 'seasonal'))
      theta <- c(1,i*0.01,1)
      l2 <- c(l2, loglikelihood(theta=theta, Trend=Trend, Season=Season, Obs = data))
    }
    return(l2)
```

```
  }

  if(type == 3){
    l3 <- c()
    for (i in 1:10000){
      ssm <- SSModel(data ~ SSMtrend(1, Q=list(1)) +
                       SSMseasonal(12, sea.type = 'dummy', Q=i*0.01), H=1)
      mod <- KFS(ssm)
      Trend <- coef(mod, states = 'trend')
      Season <- rowSums(coef(mod, states = 'seasonal'))
      theta <- c(1,1,i*0.01)
      l3 <- c(l3, loglikelihood(theta=theta, Trend=Trend, Season=Season, Obs = data))
    }
    return(l3)
  }
}
```

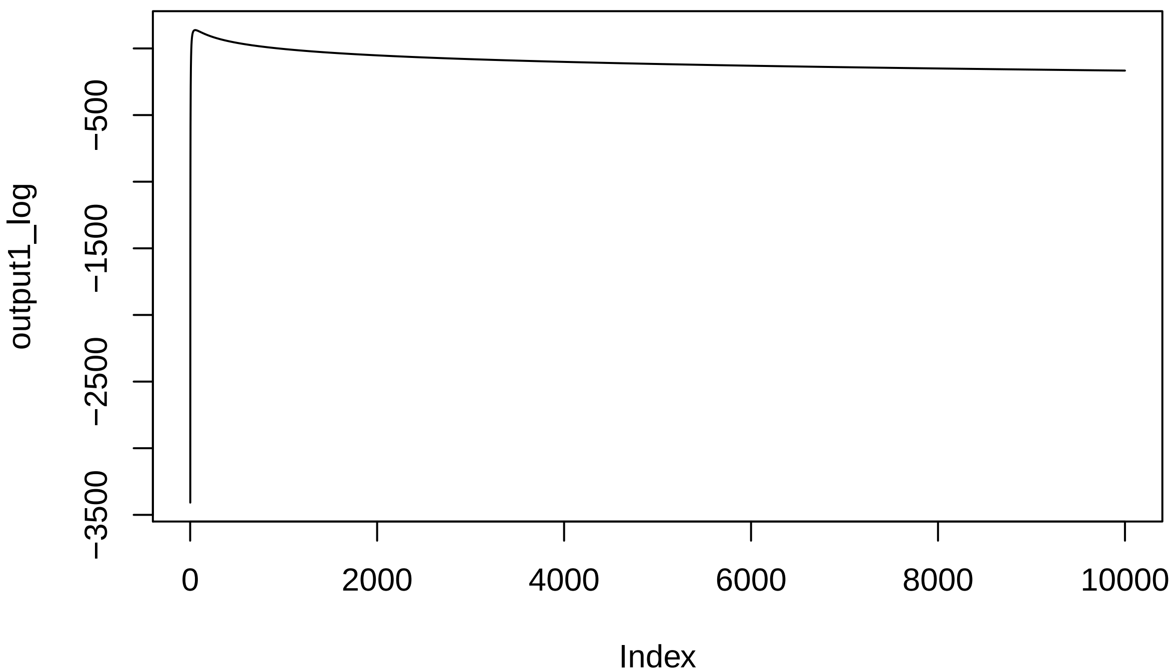
For the likelihood curve

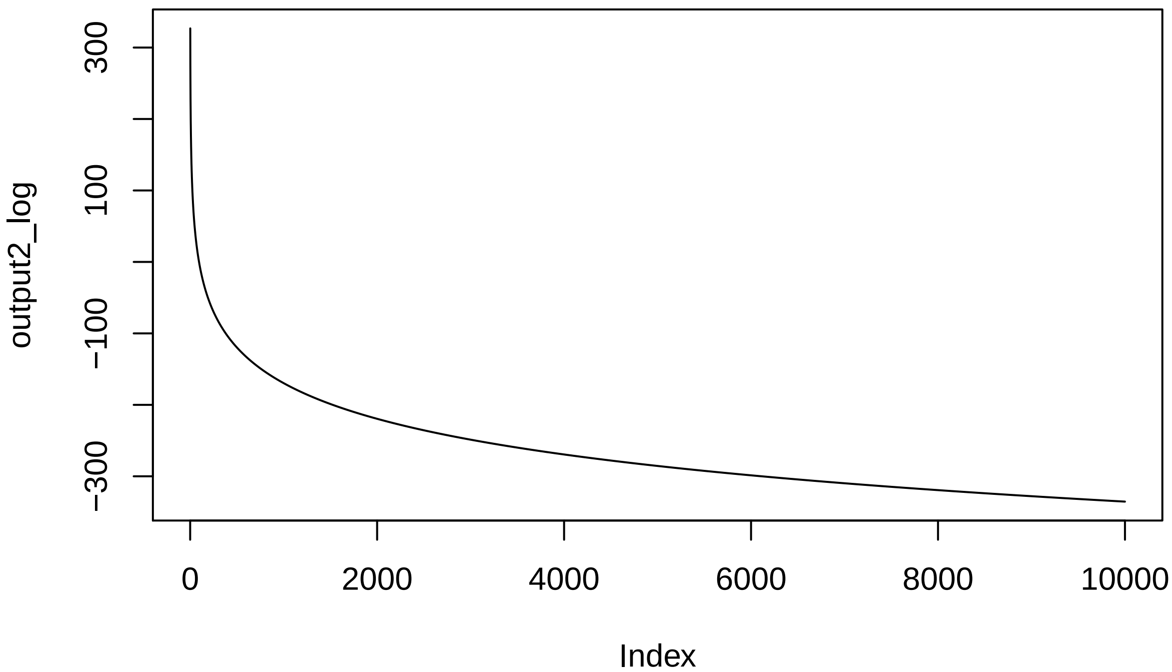
Here, the only difference from the function *loglikelihood_single_gs* is that we apply likelihood instead of loglikelihood defined at beginning.

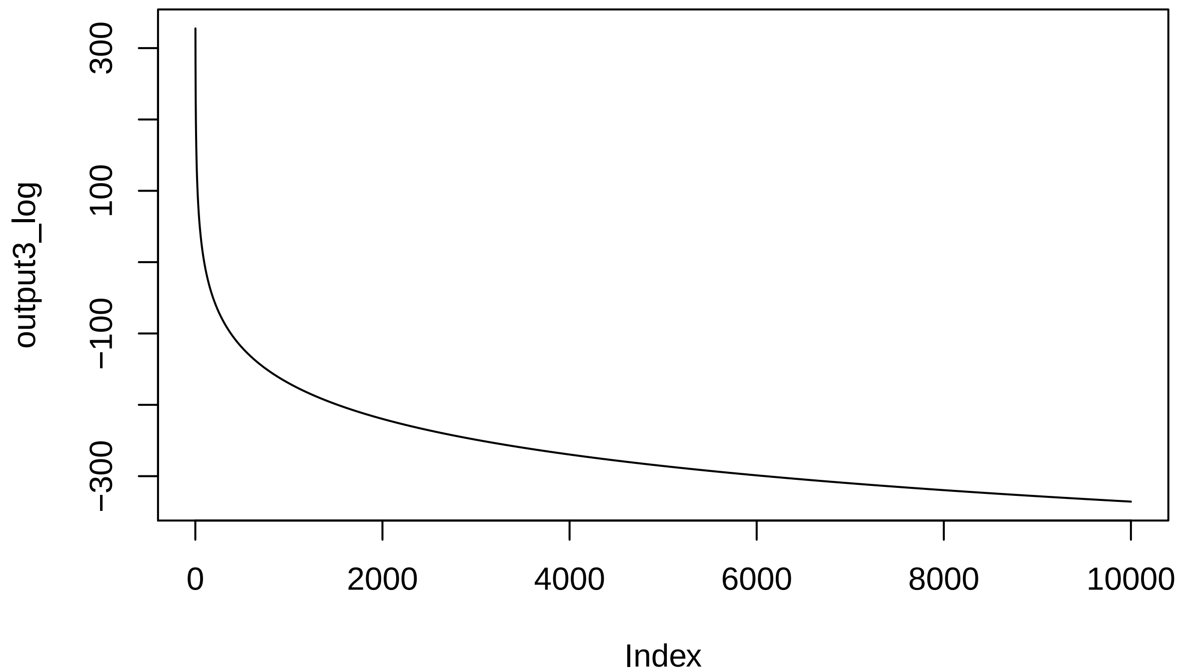
```
# define a 'large' function to return the loglikelihood series of one single parameter's sequence
likelihood_single_gs <- function(data, type){
  if(type == 1) {
    L1 <- c()
    for (i in 1:10000){
      ssm <- SSModel(data ~ SSMtrend(1, Q=list(1)) +
                       SSMseasonal(12, sea.type = 'dummy', Q=1), H=i*0.001)
      mod <- KFS(ssm)
      Trend <- coef(mod, states = 'trend')
      Season <- rowSums(coef(mod, states = 'seasonal'))
      theta <- c(i*0.001,1,1)
      L1 <- c(L1, likelihood(theta=theta, Trend=Trend, Season=Season, Obs = data))
    }
    return(L1)
  }

  if(type == 2){
    L2 <- c()
    for (i in 1:10000){
      ssm <- SSModel(data ~ SSMtrend(1, Q=list(i*0.001)) +
                       SSMseasonal(12, sea.type = 'dummy', Q=1), H=1)
      mod <- KFS(ssm)
      Trend <- coef(mod, states = 'trend')
      Season <- rowSums(coef(mod, states = 'seasonal'))
      theta <- c(1,i*0.001,1)
      L2 <- c(L2, likelihood(theta=theta, Trend=Trend, Season=Season, Obs = data))
    }
    return(L2)
  }

  if(type == 3){
    L3 <- c()
```

```r
    for (i in 1:10000){
      ssm <- SSModel(data ~ SSMtrend(1, Q=list(1)) +
                       SSMseasonal(12, sea.type = 'dummy', Q=i*0.001), H=1)
      mod <- KFS(ssm)
      Trend <- coef(mod, states = 'trend')
      Season <- rowSums(coef(mod, states = 'seasonal'))
      theta <- c(1,1,i*0.001)
      L3 <- c(L3, likelihood(theta=theta, Trend=Trend, Season=Season, Obs = data))
    }
    return(L3)
  }
}
```

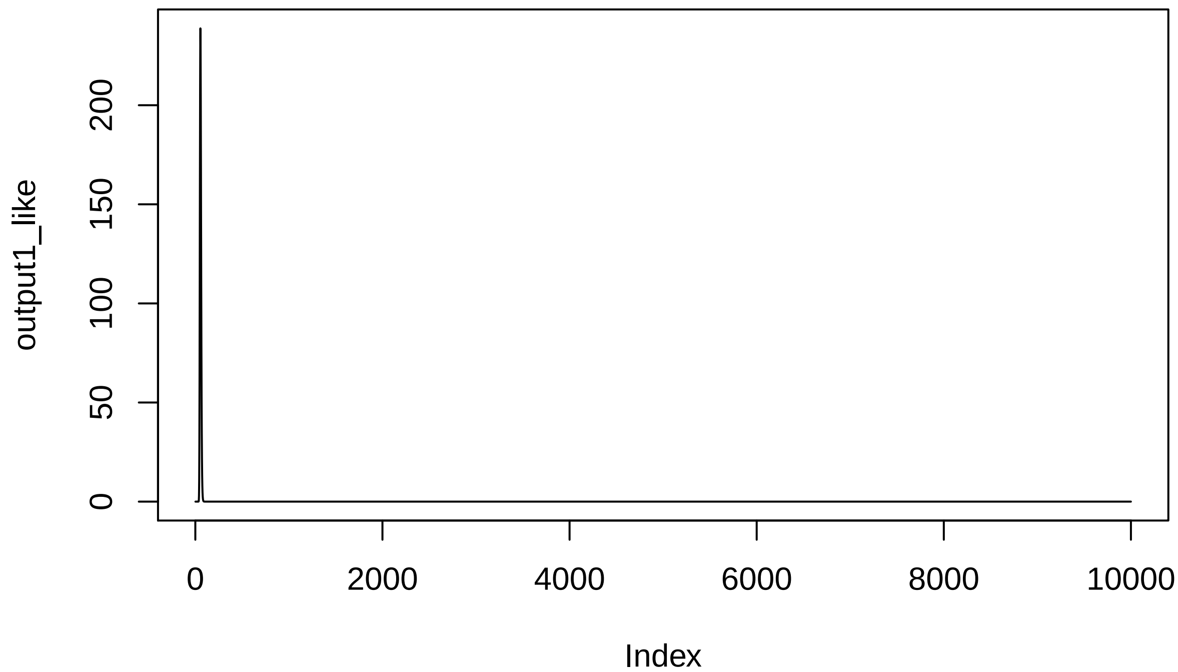
The loglikelihood curves:

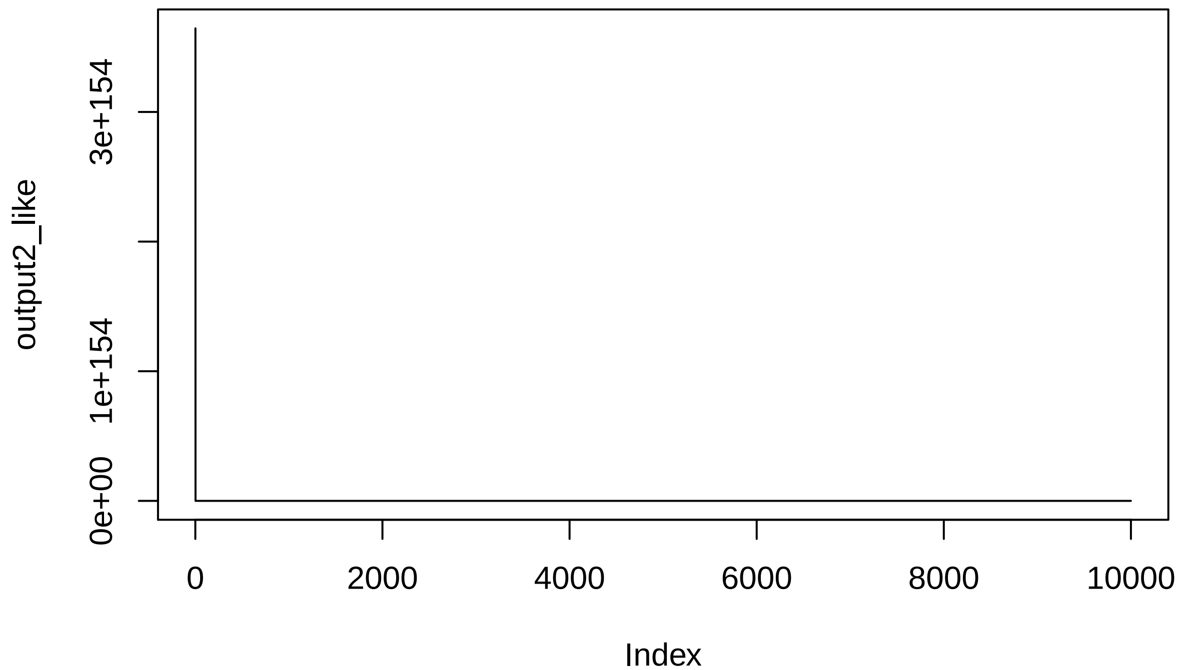



output2_log
300
100
−100
−300
0
2000
4000
6000
8000
10000
Index

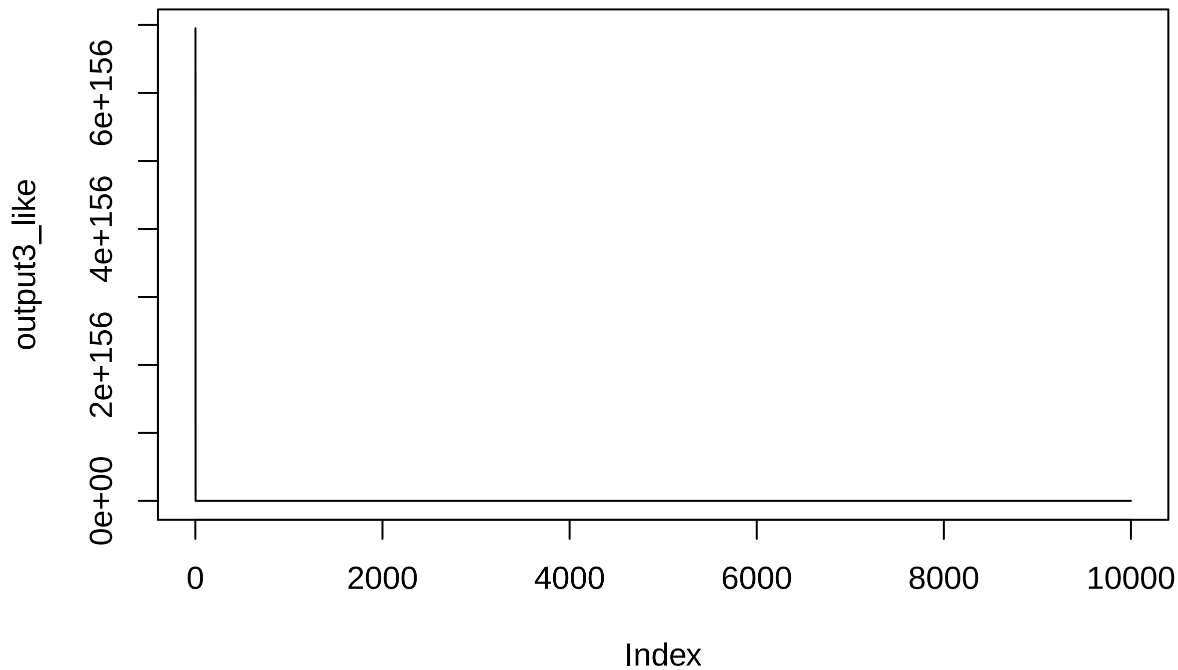


The likelihood curves:


output1_like
0
50
100
150
200
0
2000
4000
6000
8000
10000
Index

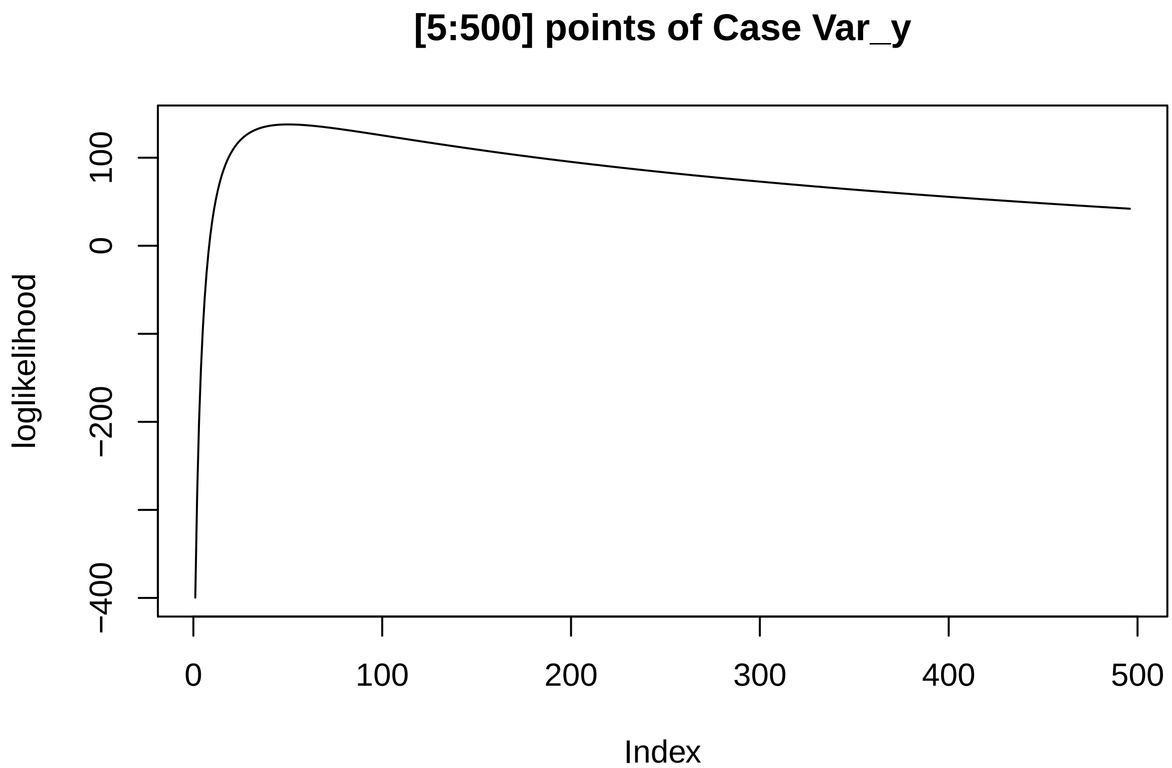


output2_like
3e+154
1e+154
0e+00
0
2000
4000
6000
8000
10000
Index

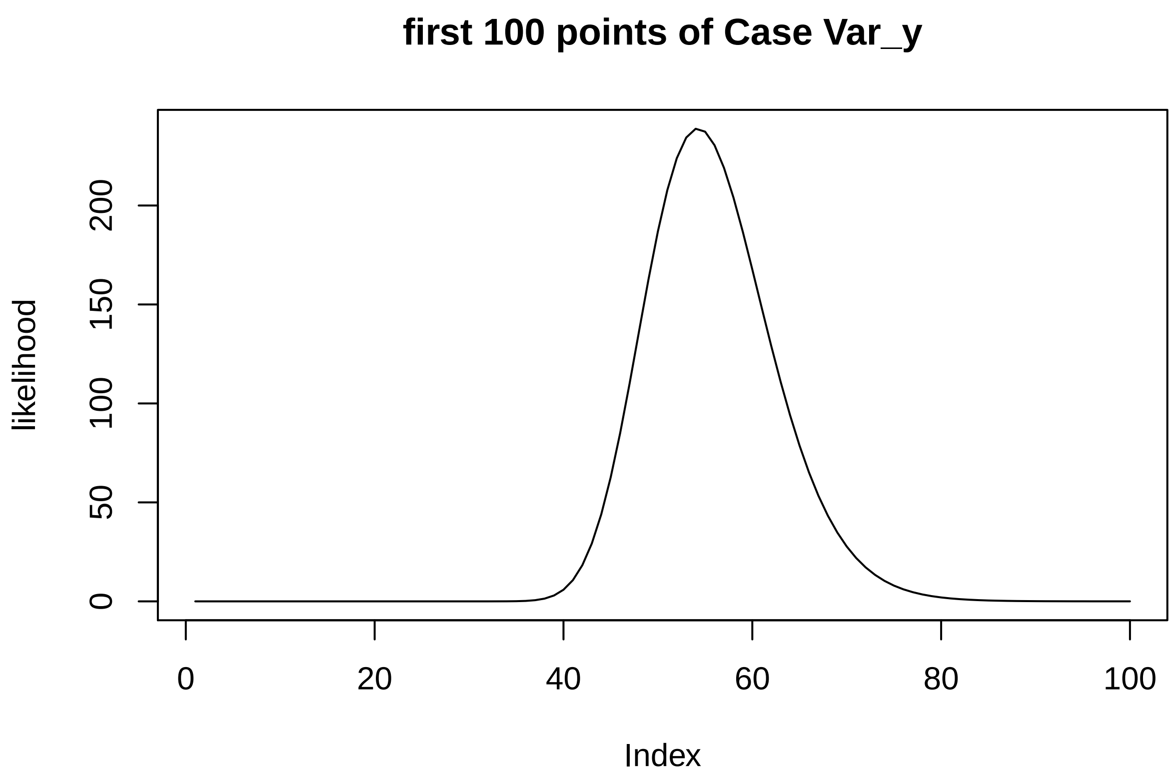


I take a look at the first few points specially since there is a change at the beginning:

```
plot(output1_log[5:500], type = 'l',
     main = '[5:500] points of Case Var_y', ylab = 'loglikelihood')
```




```
plot(output1_like[1:100], type = 'l',
     main = 'first 100 points of Case Var_y', ylab = 'likelihood')
```



MAP

Because our paramters are variances, that is to say they are greater than 0, so the priors on them should be some distribution that is defined on $(0, \infty)$.

Prior

```
# if we let prior is exponential function
prior <- function(x) exp(-x)
```

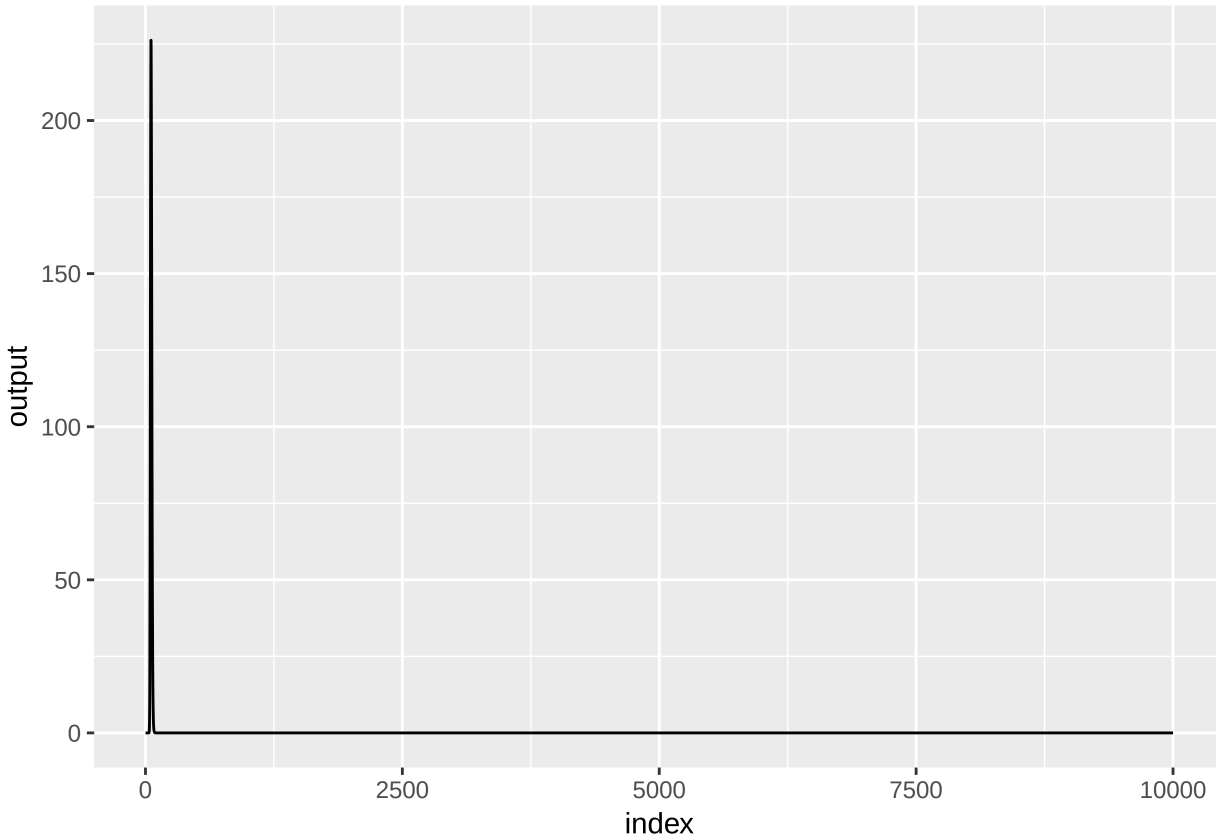
Posterior/MAP

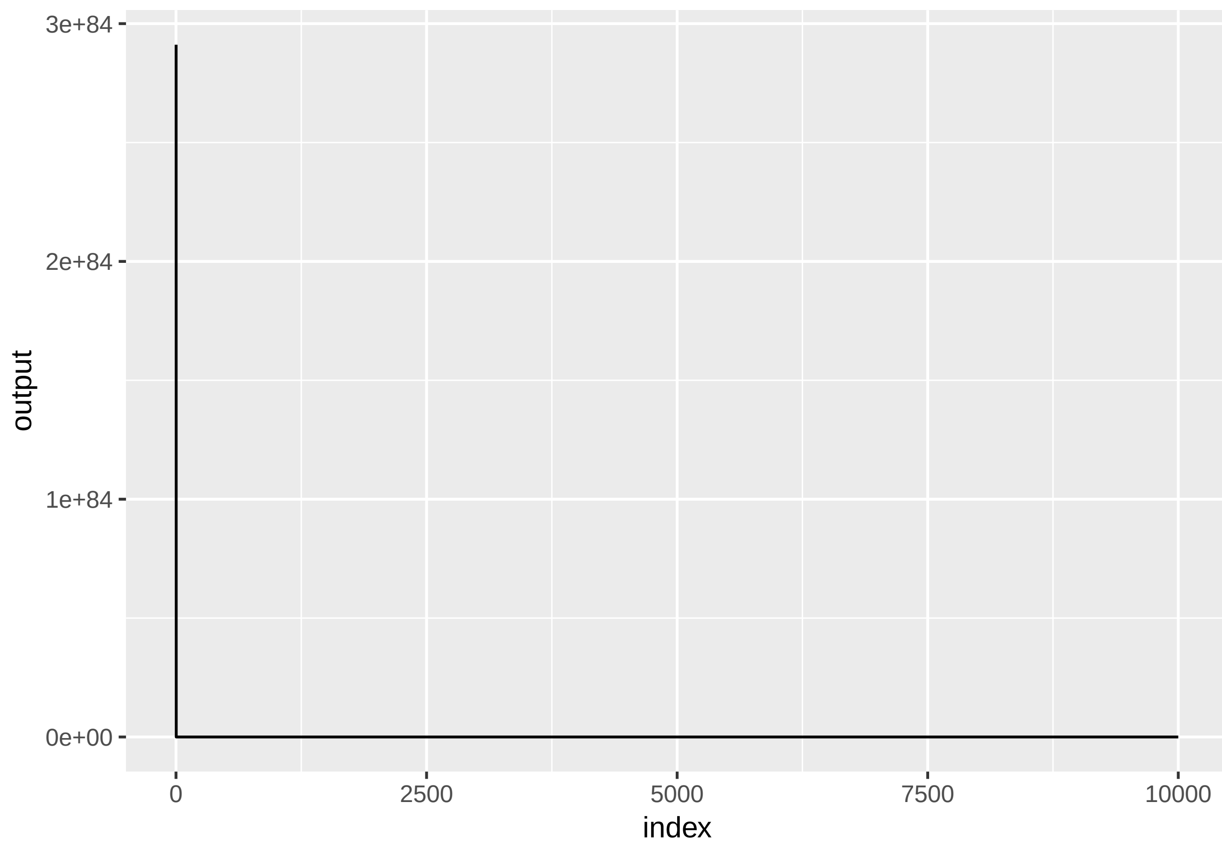
It is similar to the likelihood function but we times the prior:

```
posterior_single_gs <- function(data, type) {

  if(type == 1){
    map1 <- c()
   for (i in 1:10000) {
     ssm <- SSModel(data ~ SSMtrend(1, Q=list(1)) +
                      SSMseasonal(12, sea.type = 'dummy', Q = 1 ), H = i*0.001)
     mod <- KFS(ssm)
     Trend <- coef(mod, states = 'trend')
     Season <- rowSums(coef(mod, states = 'seasonal'))
     theta <- c(i*0.001,1,1)
     map <- likelihood(theta=theta,Trend=Trend,Season=Season,Obs = data) *
```

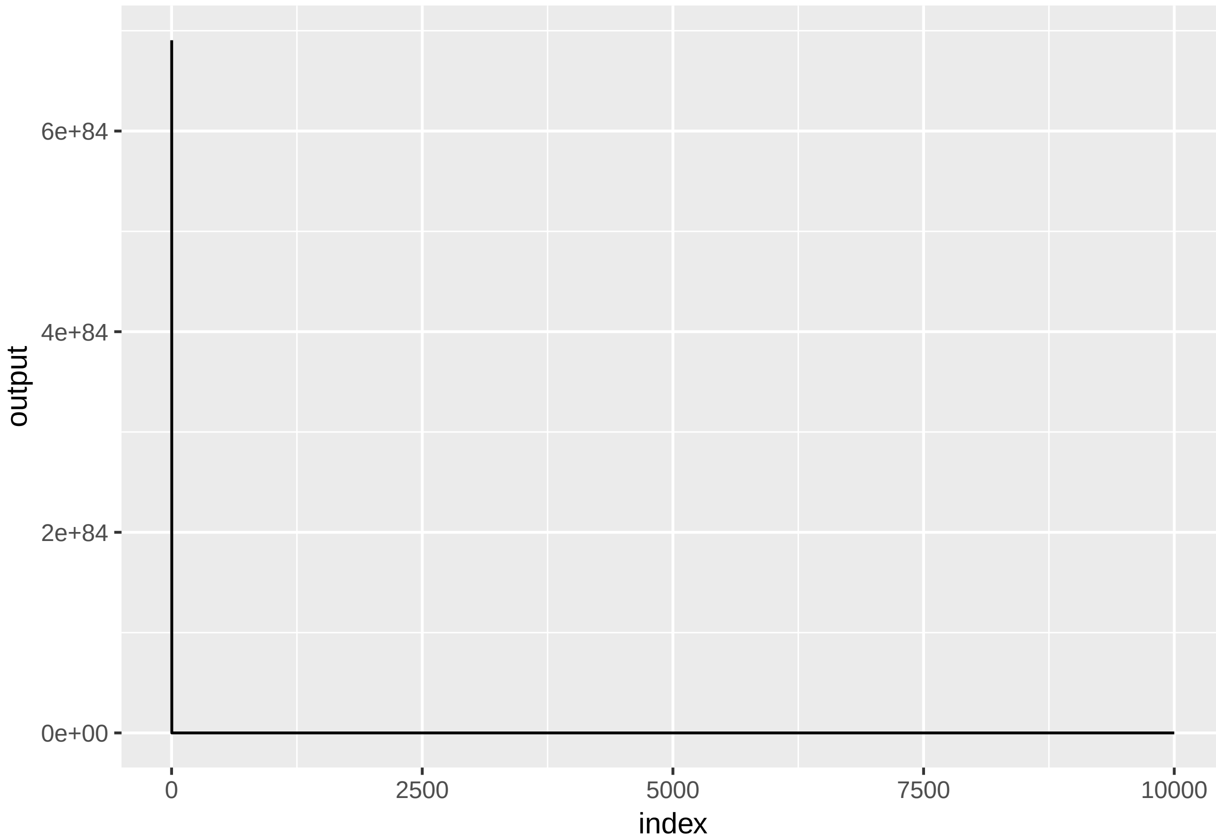
```
      prior(theta[type])
    map1 <- c(map1, map)
  }
   return(map1)
  }

  if(type == 2){
   map2 <- c()
  for (i in 1:10000) {
    ssm <- SSModel(data ~ SSMtrend(1, Q=list(0.01*i)) +
                     SSMseasonal(12, sea.type = 'dummy', Q = 1 ), H = 1)
    mod <- KFS(ssm)
    Trend <- coef(mod, states = 'trend')
    Season <- rowSums(coef(mod, states = 'seasonal'))
    theta <- c(1,i*0.01,1)
    map <- likelihood(theta=theta,Trend=Trend,Season=Season,Obs = data) *
      prior(theta[type])
    map2 <- c(map2, map)
  }
   return(map2)
  }

  if(type == 3){
   map3 <- c()
  for (i in 1:10000) {
    ssm <- SSModel(data ~ SSMtrend(1, Q=list(1)) +
                     SSMseasonal(12, sea.type = 'dummy', Q = 0.01*i ), H = 1)
    mod <- KFS(ssm)
    Trend <- coef(mod, states = 'trend')
    Season <- rowSums(coef(mod, states = 'seasonal'))
    theta <- c(1,1,0.01*i)
    map <- likelihood(theta=theta,Trend=Trend,Season=Season,Obs = data) *
      prior(theta[type])
    map3 <- c(map3, map)
  }
   return(map3)
  }
}
```

Then these are the posterior curves:(still on AirPassengers)


200
150
100
50
0
output
0
2500
5000
7500
10000
index

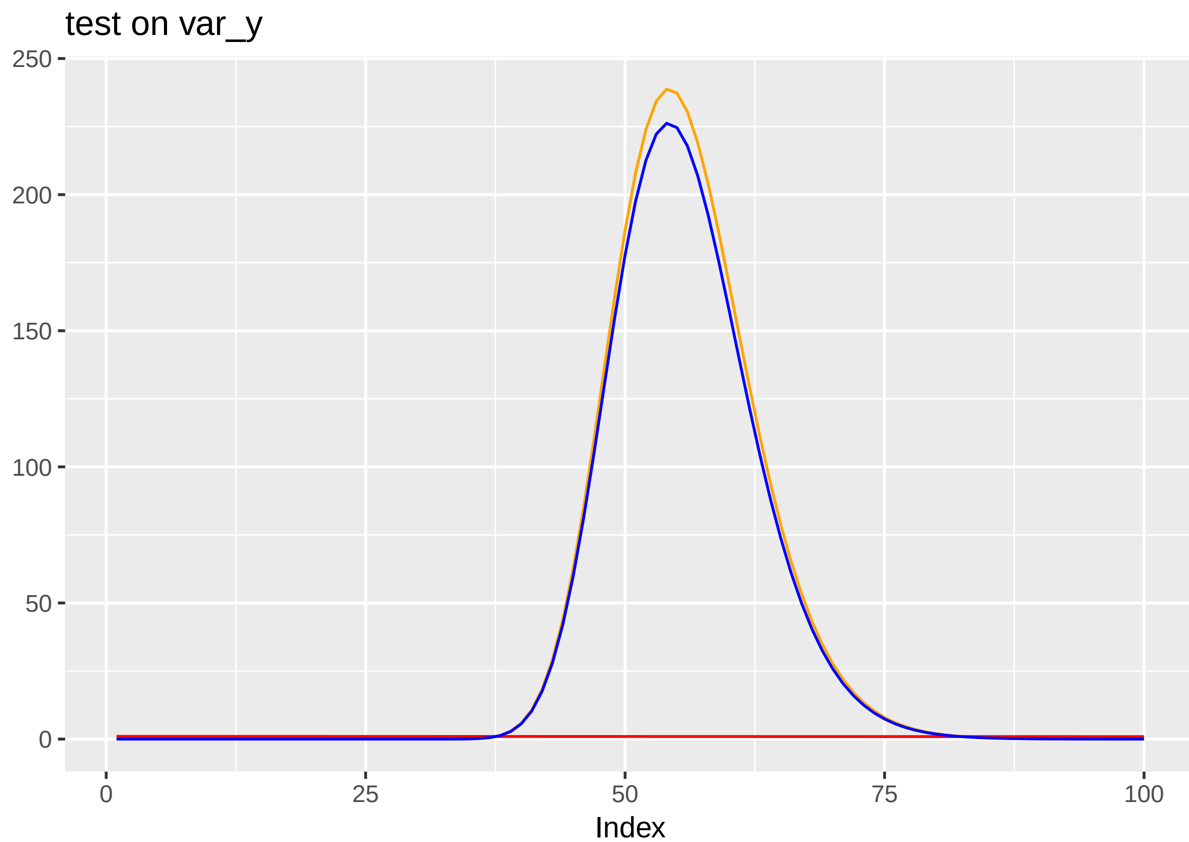


3e+84
2e+84
1e+84
0e+00
output
0
2500
5000
7500
10000
index



Let't take a look at the first 100 points of likelihood, prior and posterior curves at the same time(Fixed σ_T^2, σ_S^2 and $\sigma_y^2 = 0.001, 0.002, \ldots, 10$):

Orange: likelihood Red: prior Blue: posterior



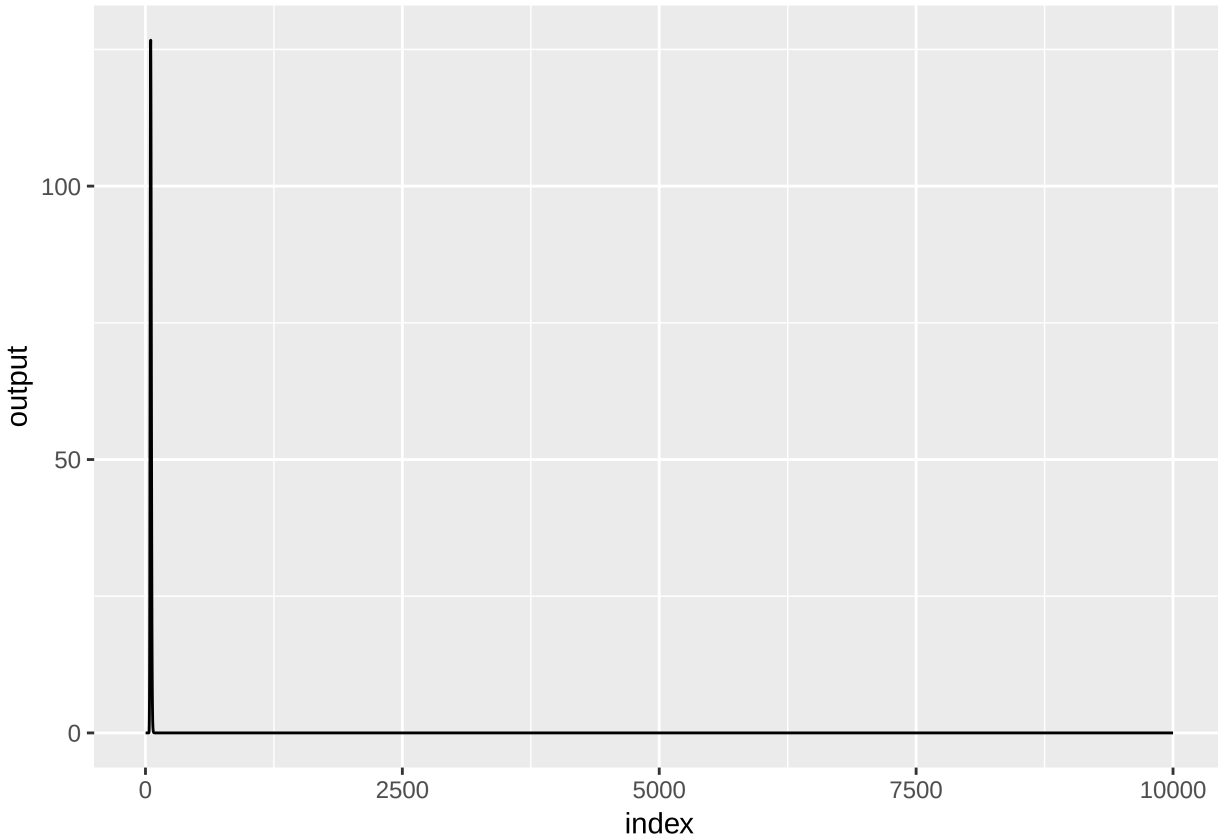
If we change the prior to a generic exponential distribution

```
prior <- function(x, lambda) lambda * exp(-lambda * x)
```

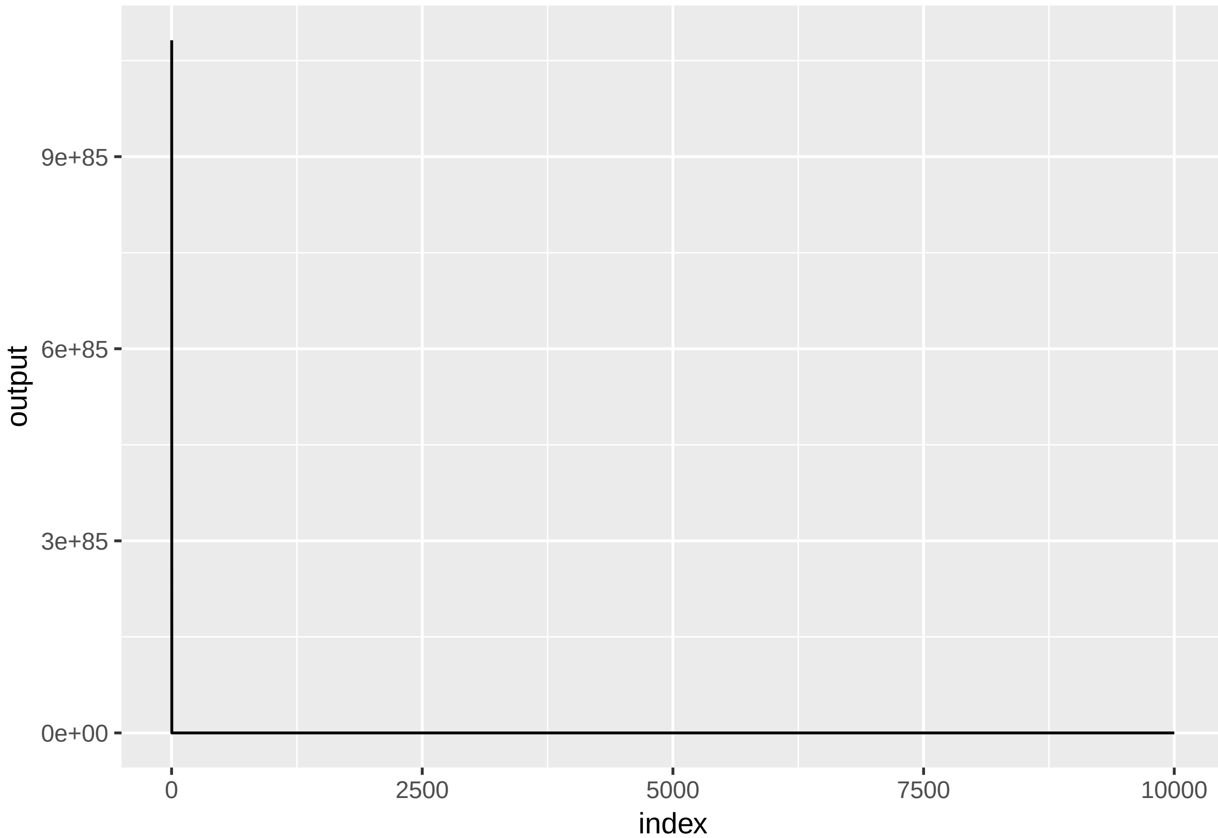
Update the posterior

We let lambda = 100:

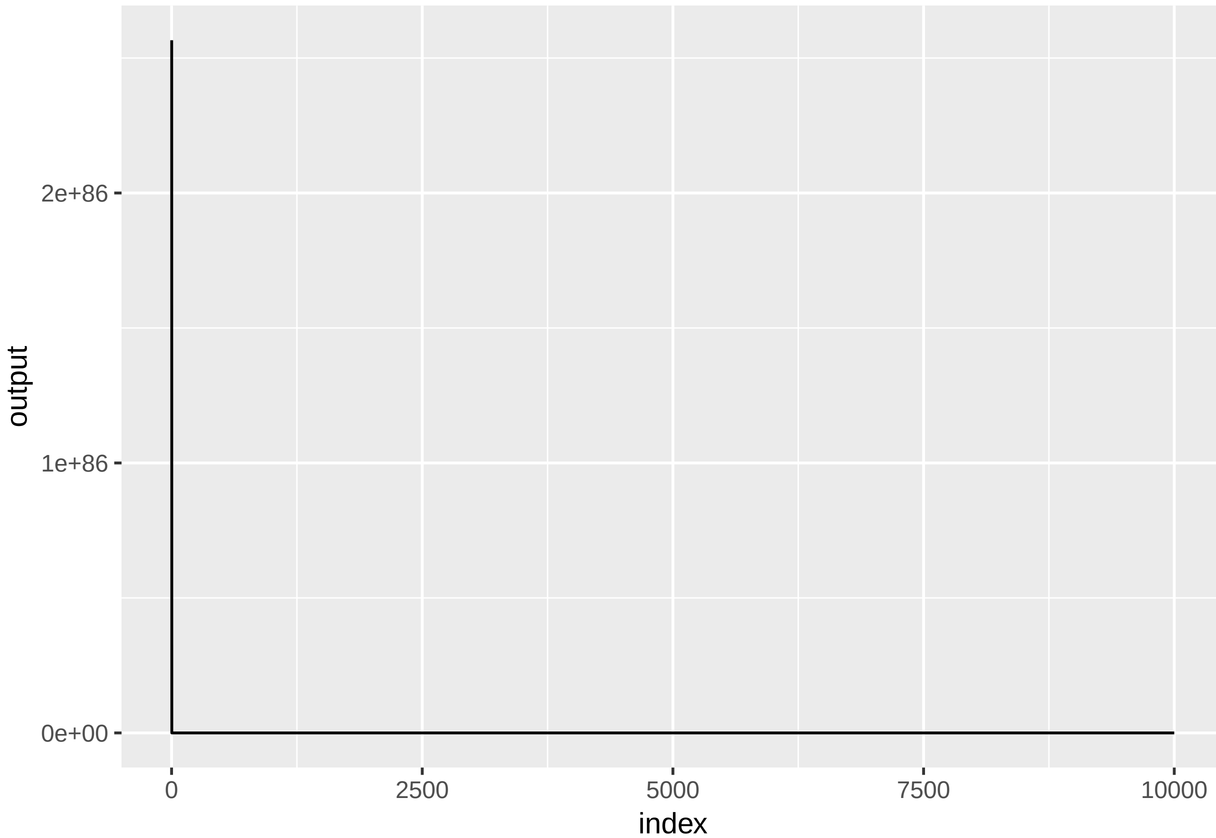
```
lambda <- 100

output1_post <- posterior_single_gs(data=log(AirPassengers), type = 1, lambda)

output2_post <- posterior_single_gs(data=log(AirPassengers), type = 2, lambda)

output3_post <- posterior_single_gs(data=log(AirPassengers), type = 3, lambda)


ggplot(data.frame(output=output1_post, index = c(1:length(output1_post))),
       aes(x=index, y=output)) +
  geom_line()
```



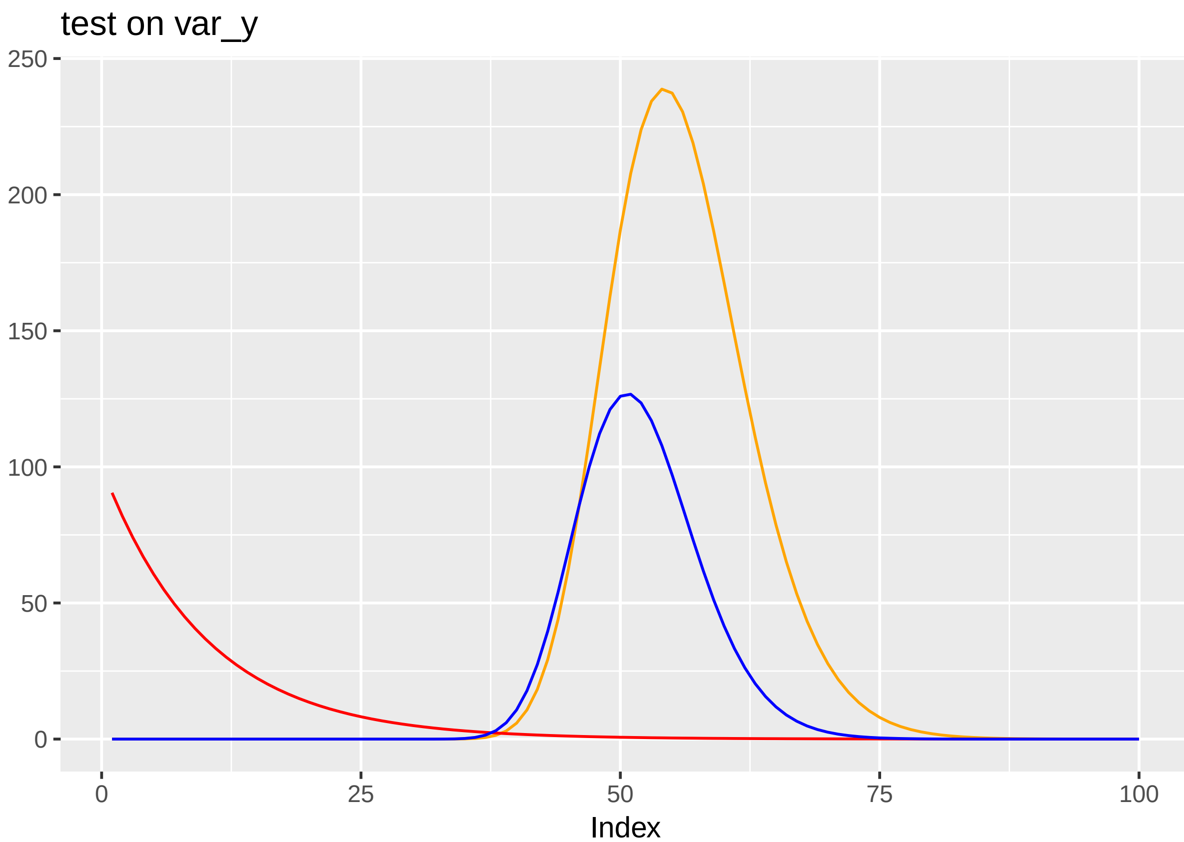

```r
ggplot(data.frame(output=output2_post, index = c(1:length(output2_post))),
       aes(x=index, y=output)) +
  geom_line()
```




```
ggplot(data.frame(output=output3_post, index = c(1:length(output3_post))),
       aes(x=index, y=output)) +
  geom_line()
```




```r
ggplot(data.frame(output1_like=output1_like[1:100],
                  output1_prior=prior(seq(0.001,10,0.001),lambda)[1:100],
                  output1_post=output1_post[1:100],
                  index=c(1:100))) +
  geom_line(aes(y=output1_like, x=index),color='orange') +
  geom_line(aes(y=output1_prior, x=index), color='red') +
  geom_line(aes(y=output1_post, x=index),color='blue') +
  labs(title = 'test on var_y', x='Index', y='')
```



TBD