Untitled

lohith

2023-04-24

```
rm(list = ls())

library('rugarch')
```

```
## Warning: package 'rugarch' was built under R version 4.2.3
```

```
## Loading required package: parallel
```

```
##
## Attaching package: 'rugarch'
```

```
## The following object is masked from 'package:stats':
##
##     sigma
```

```
library('RHmm')
```

```
## Loading required package: MASS
```

```
## Loading required package: nlme
```

```
library('reshape2')
```

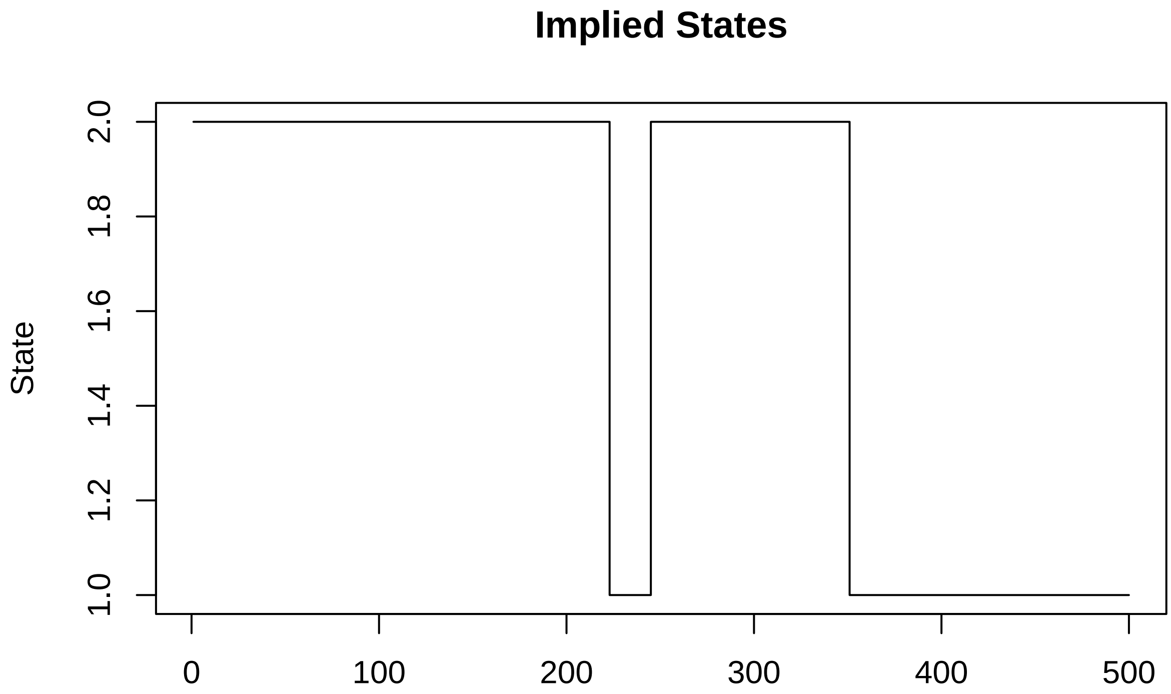
```
## Warning: package 'reshape2' was built under R version 4.2.3
```

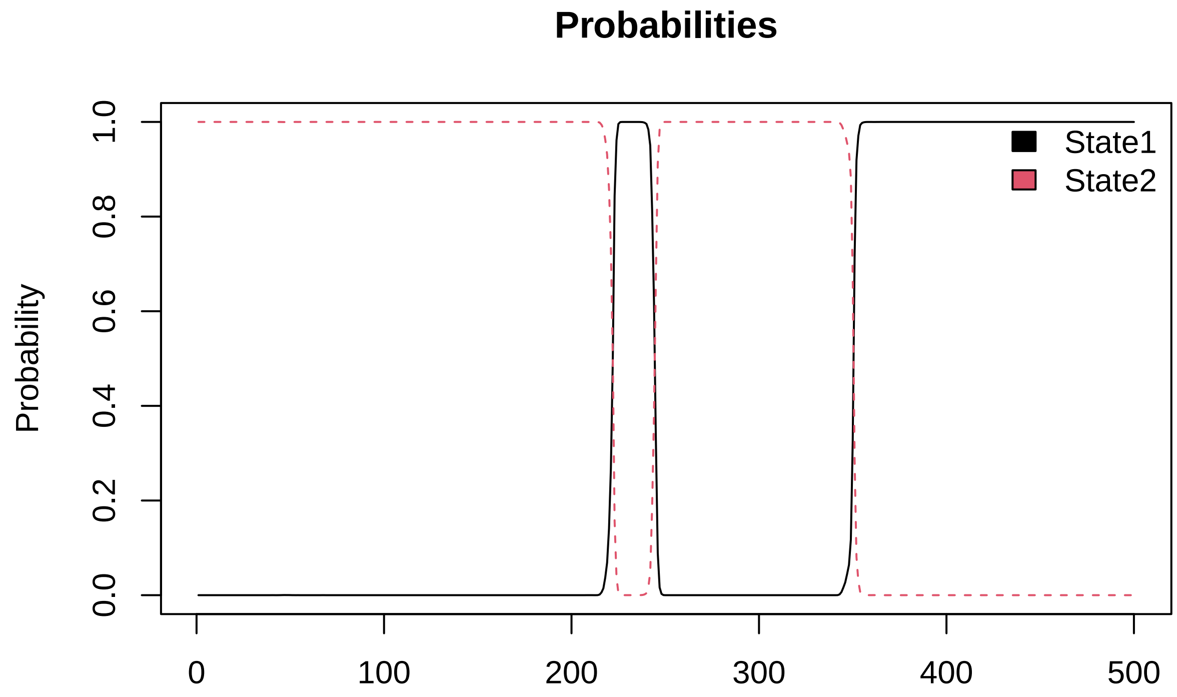
```
library('ggplot2')
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

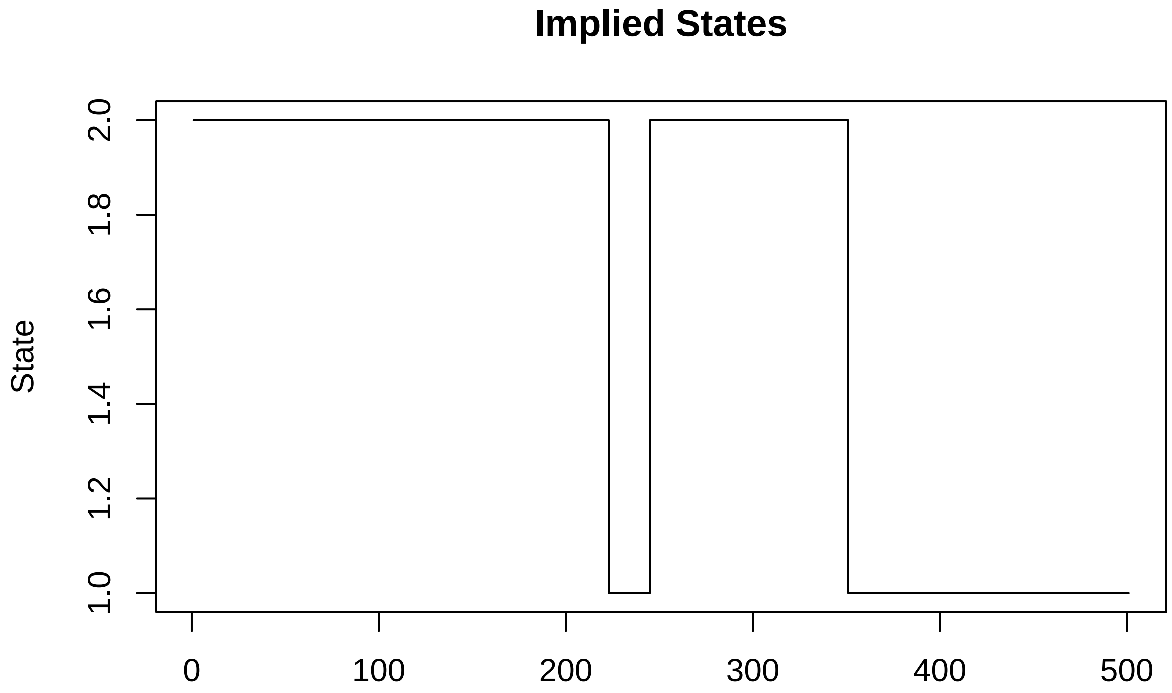
```
data = EuStockMarkets[1:500, 4]

for(x in 1 : 5)
{
y= data
Fit = HMMFit(data, nStates=2)
Path = viterbi(Fit, data)
fb = forwardBackward(Fit, data)
```

```r
plot(Path$states, type='s', main='Implied States', xlab='', ylab='State')

matplot(fb$Gamma, type='l', main='Probabilities', ylab='Probability')

legend(x='topright', c('State1','State2'),  fill=1:2, bty='n')

diff = Path$states[-1] - Path$states[-length(Path$states)]
state2 = which(Path$states %in% unique(Path$states)[1])
state1 = which(Path$states %in% unique(Path$states)[2])
vals = unique(diff)
state21 = length(which(diff %in% vals[2]))
state12 = length(which(diff %in% vals[3]))
state11 = (length(state1) - length(state12)) / length(state1)
state22 = (length(state2) - length(state21)) / length(state2)
state21 = state21 / length(state2)
state12 = state12 / length(state1)

PMat = matrix(c(state11,state12,state21,state22),ncol=2)

#Determining the most optimal p and q parameters of GARCH for each regime

p1 = 0
q1 = 0
p2 = 0
q2 = 0
for(p in 1 : 2)
{
  for(q in 1 : 2)
  {
    garch.spec.norm = ugarchspec(variance.model=list(garchOrder=c(p,q)),mean.model= list(armaOrder=c(0,0

    garch.fit.norm1 = ugarchfit(spec=garch.spec.norm,data=data[state2],solver.control=list(trace = 1))

    garch.fit.norm2 = ugarchfit(spec=garch.spec.norm,data=data[state1],solver.control=list(trace = 1))
   {

    if(!exists('max1'))
    {
      max1 = garch.fit.norm1@fit$LLH
      p1 = p
      q1 = q
    }
    else
    {
      if(garch.fit.norm1@fit$LLH > max1)
      {
        max1 = garch.fit.norm1@fit$LLH
        p1 = p
        q1 = q
      }
    }
  }
  {
    if(!exists('max2'))
    {
      max2 = garch.fit.norm2@fit$LLH
      p2 = p
      q2 = q
    }

    else
    {
      if(garch.fit.norm2@fit$LLH > max1)
      {
        max2 = garch.fit.norm2@fit$LLH
        p2 = p
        q2 = q
      }
    }
  }

  }
}

#Applying the most optimal p nd q parameters for GARCH models in both the regimes and building the GARC
# Building the GARCH model for regime 1 using the specification optimat p and q, i.e.,  p1 and q1 deriv
garch.spec.norm = ugarchspec(variance.model=list(garchOrder=c(p1,q1)),mean.model= list(armaOrder=c(0,0))
garch.fit.norm1 = ugarchfit(spec=garch.spec.norm,data=data[state2],solver.control=list(trace = 1))

# Building the GARCH model for regime 2 using the specification optimat p and q, i.e.,  p2 and q2 deriv
garch.spec.norm2 = ugarchspec(variance.model=list(garchOrder=c(p2,q2)),mean.model= list(armaOrder=c(0,0
garch.fit.norm2 = ugarchfit(spec=garch.spec.norm2,data=data[state1],solver.control=list(trace = 1))

statecurrent = Path$states[length(Path$states)]

coef1 = garch.fit.norm1@fit$coef
coef2 = garch.fit.norm2@fit$coef

m11 = c()
m12 =c()
m21 = c()
m22 =c()
for( p in 1 : p1)
{
  m11[p] =(garch.fit.norm1@fit$residuals[length(garch.fit.norm1@fit$residuals) - (p - 1)]) ^ 2
}
for( p in 1 : q1)
{
  m12[p] =(garch.fit.norm1@fit$var[length(garch.fit.norm1@fit$var) - (p - 1)])
}
for( p in 1 : p2)
{
  m21[p] =(garch.fit.norm2@fit$residuals[length(garch.fit.norm2@fit$residuals) - (p - 1)]) ^ 2
}
for( p in 1 : q2)
{
  m22[p] =(garch.fit.norm2@fit$var[length(garch.fit.norm2@fit$var) - (p - 1)])
}

m1 = c(1,1,m11,m12)
m2 = c(1,1,m21,m22)

futureval = ((PMat[statecurrent,1] * sum(coef1 * m1)) + (PMat[statecurrent,2] * sum(coef2 * m2))) ^ 0.5

data = c(data,futureval)

}
```



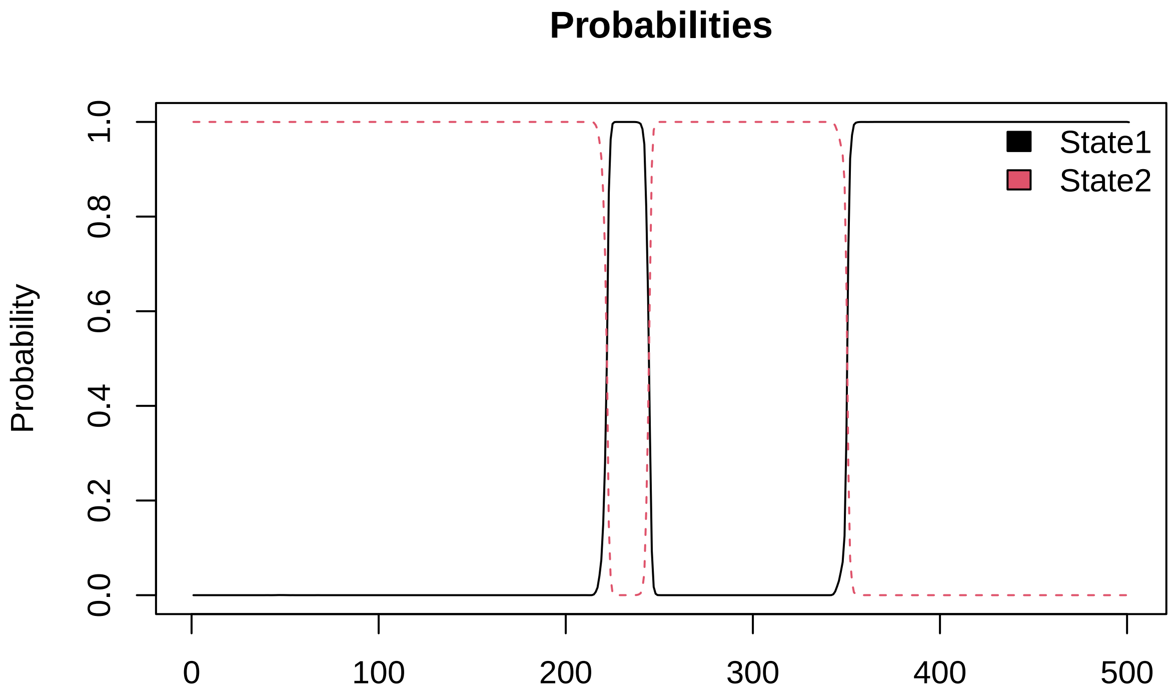


```
##
## Iter: 1 fn: 1820.7872     Pars:  2556.530267683  487.996329680    0.916299694    0.000007949
## Iter: 2 fn: 1820.7872     Pars:  2556.516201288  488.466433852    0.916129966    0.000006836
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 899.8773  Pars:  2832.343974504  230.925399138    0.888027778    0.000001986
## Iter: 2 fn: 899.8772  Pars:  2832.347242790  230.747019451    0.888022786    0.000001565
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 1820.7727     Pars:  2.557e+03 4.881e+02 9.174e-01 3.386e-08 1.210e-08
## Iter: 2 fn: 1820.7727     Pars:  2.557e+03 4.882e+02 9.174e-01 2.712e-08 8.245e-09
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 900.3089  Pars:  2.832e+03 2.309e+02 8.876e-01 1.954e-10 1.168e-09
## Iter: 2 fn: 900.3089  Pars:  2.832e+03 2.310e+02 8.876e-01 5.275e-11 1.018e-09
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 1820.7727     Pars:  2.557e+03 4.881e+02 9.174e-01 2.109e-08 2.350e-08
## Iter: 2 fn: 1820.7727     Pars:  2.557e+03 4.881e+02 9.174e-01 2.060e-08 2.298e-08
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 900.3089  Pars:  2.832e+03 2.309e+02 8.876e-01 5.597e-10 2.818e-09
## Iter: 2 fn: 900.3089  Pars:  2.832e+03 2.310e+02 8.876e-01 3.051e-10 2.524e-09
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 1820.7727     Pars:  2.557e+03 4.881e+02 9.172e-01 9.773e-09 1.343e-08 1.358e-09
## Iter: 2 fn: 1820.7727     Pars:  2.557e+03 4.880e+02 9.173e-01 9.305e-09 1.287e-08 1.009e-09
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 900.3089  Pars:  2.832e+03 2.309e+02 8.876e-01 3.487e-09 8.575e-09 1.048e-08
## Iter: 2 fn: 900.3089  Pars:  2.832e+03 2.309e+02 8.876e-01 3.216e-09 8.188e-09 1.006e-08
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 1820.7727     Pars:  2.557e+03 4.881e+02 9.174e-01 2.109e-08 2.350e-08
## Iter: 2 fn: 1820.7727     Pars:  2.557e+03 4.881e+02 9.174e-01 2.060e-08 2.298e-08
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 900.3089  Pars:  2.832e+03 2.309e+02 8.876e-01 3.487e-09 8.575e-09 1.048e-08
## Iter: 2 fn: 900.3089  Pars:  2.832e+03 2.309e+02 8.876e-01 3.216e-09 8.188e-09 1.006e-08
## solnp--> Completed in 2 iterations
```

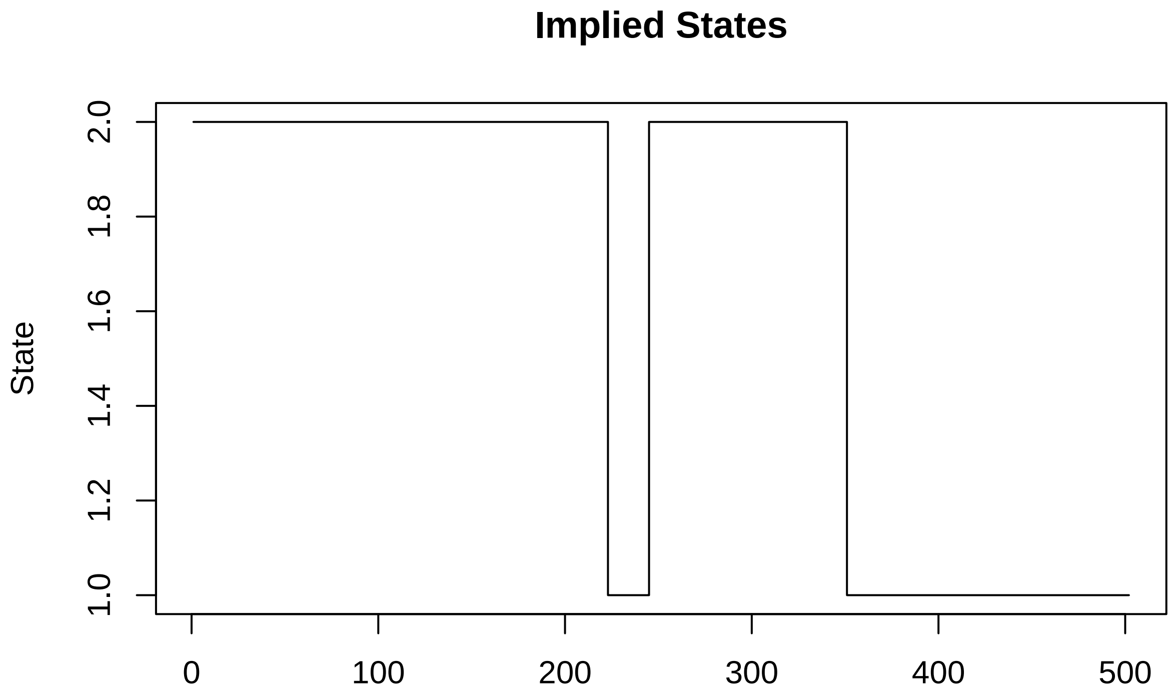


```
##
## Iter: 1 fn: 1820.7872     Pars:  2556.530267683  487.996329680    0.916299694    0.000007949
## Iter: 2 fn: 1820.7872     Pars:  2556.516201288  488.466433852    0.916129966    0.000006836
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 915.9010  Pars:  2824.50321125  296.87927757    0.99898647    0.00001118
## Iter: 2 fn: 915.8497  Pars:  2.825e+03 3.079e+02 9.501e-01 1.328e-12
## Iter: 3 fn: 915.8497  Pars:  2.825e+03 3.079e+02 9.501e-01 2.023e-13
## solnp--> Completed in 3 iterations
##
```

```
## Iter: 1 fn: 1820.7727     Pars:  2.557e+03 4.881e+02 9.174e-01 3.386e-08 1.210e-08
## Iter: 2 fn: 1820.7727     Pars:  2.557e+03 4.882e+02 9.174e-01 2.712e-08 8.245e-09
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 916.2390  Pars:  2.825e+03 3.075e+02 9.506e-01 1.329e-07 1.308e-09
## Iter: 2 fn: 916.2390  Pars:  2.825e+03 3.074e+02 9.508e-01 1.241e-07 4.181e-10
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 1820.7727     Pars:  2.557e+03 4.881e+02 9.174e-01 2.109e-08 2.350e-08
## Iter: 2 fn: 1820.7727     Pars:  2.557e+03 4.881e+02 9.174e-01 2.060e-08 2.298e-08
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 916.2390  Pars:  2.825e+03 3.071e+02 9.511e-01 1.531e-08 1.467e-08
## Iter: 2 fn: 916.2390  Pars:  2.825e+03 3.074e+02 9.508e-01 7.177e-09 7.551e-09
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 1820.7727     Pars:  2.557e+03 4.881e+02 9.172e-01 9.773e-09 1.343e-08 1.358e-09
## Iter: 2 fn: 1820.7727     Pars:  2.557e+03 4.880e+02 9.173e-01 9.305e-09 1.287e-08 1.009e-09
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 916.2390  Pars:  2.825e+03 3.073e+02 9.511e-01 1.116e-07 1.148e-07 1.718e-09
## Iter: 2 fn: 916.2390  Pars:  2.825e+03 3.072e+02 9.511e-01 1.107e-07 1.140e-07 1.601e-09
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 55968.1686    Pars:  2518.87409    8.88827
## solnp--> Solution not reliable....Problem Inverting Hessian.

## Warning in .sgarchfit(spec = spec, data = data, out.sample = out.sample, :
## ugarchfit-->warning: solver failer to converge.
```



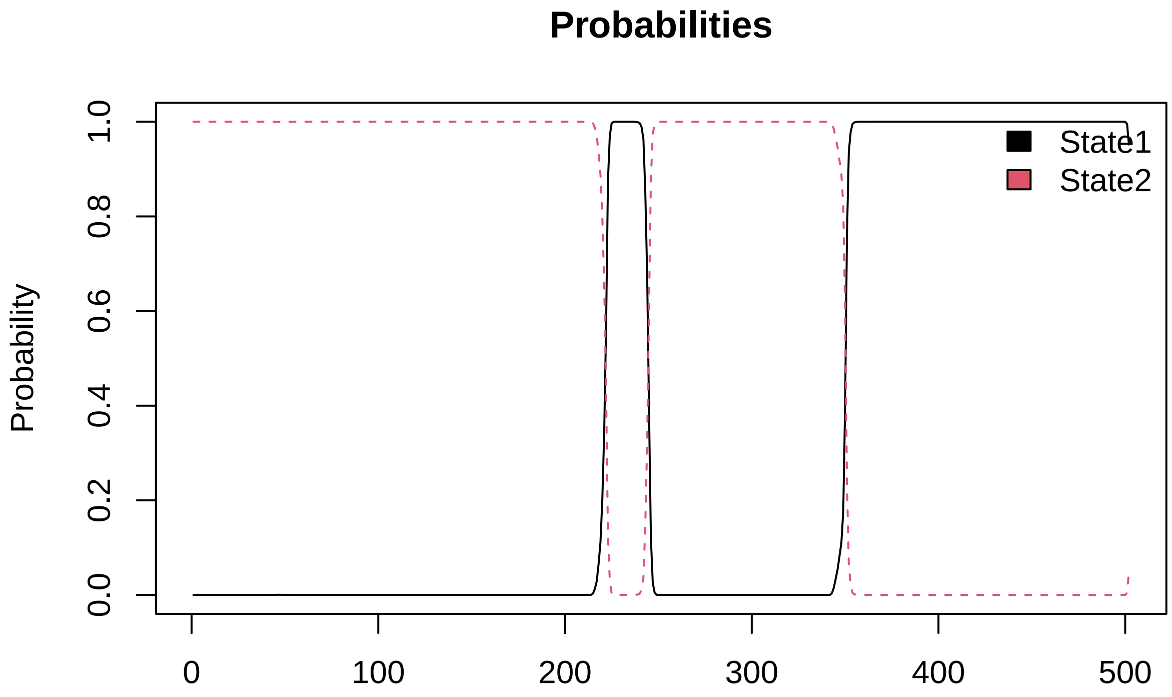
```
##
## Iter: 1 fn: 916.2390  Pars:  2.825e+03 3.073e+02 9.511e-01 1.116e-07 1.148e-07 1.718e-09
## Iter: 2 fn: 916.2390  Pars:  2.825e+03 3.072e+02 9.511e-01 1.107e-07 1.140e-07 1.601e-09
## solnp--> Completed in 2 iterations
```



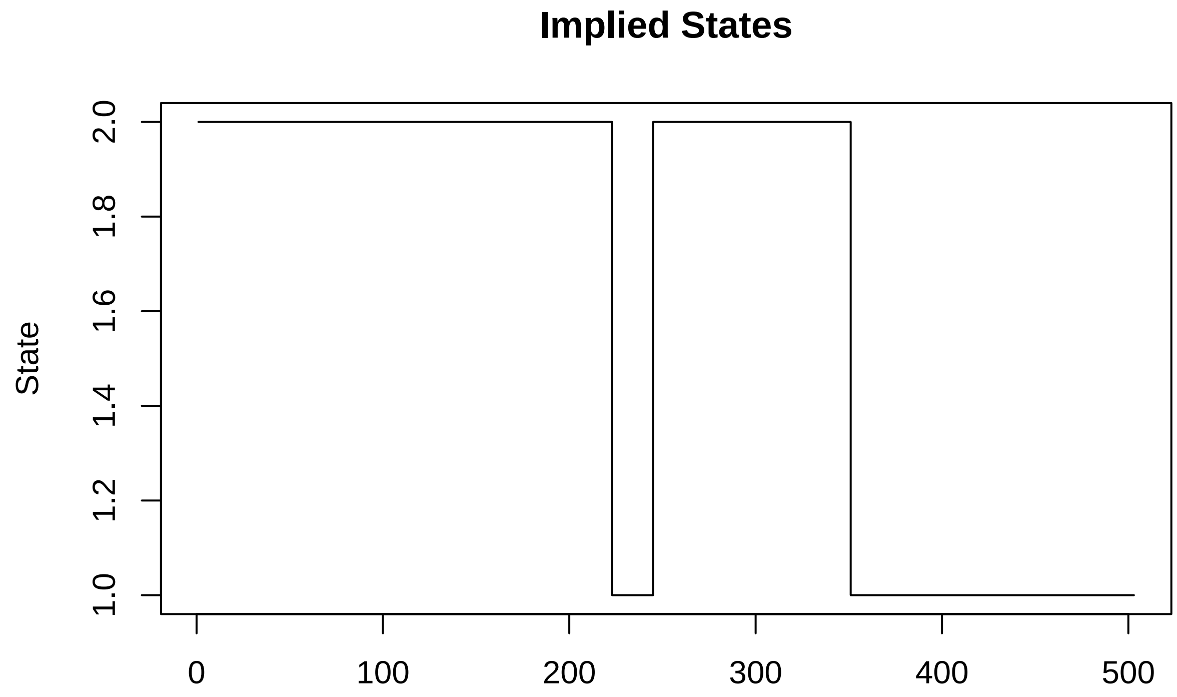
```
##
## Iter: 1 fn: 1820.7872     Pars:  2556.530267683  487.996329680    0.916299694    0.000007949
## Iter: 2 fn: 1820.7872     Pars:  2556.516201288  488.466433852    0.916129966    0.000006836
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 923.4276  Pars:  2.825e+03 3.016e+02 9.845e-01 4.607e-08
## Iter: 2 fn: 923.4276  Pars:  2.825e+03 3.016e+02 9.847e-01 2.385e-08
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 1820.7727     Pars:  2.557e+03 4.881e+02 9.174e-01 3.386e-08 1.210e-08
## Iter: 2 fn: 1820.7727     Pars:  2.557e+03 4.882e+02 9.174e-01 2.712e-08 8.245e-09
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 923.7838  Pars:  2.825e+03 3.018e+02 9.854e-01 7.153e-08 9.483e-10
## Iter: 2 fn: 923.7838  Pars:  2.825e+03 3.014e+02 9.857e-01 6.728e-08 1.329e-10
## Iter: 3 fn: 923.7838  Pars:  2.825e+03 3.014e+02 9.857e-01 6.726e-08 1.300e-10
## solnp--> Completed in 3 iterations
##
## Iter: 1 fn: 1820.7727     Pars:  2.557e+03 4.881e+02 9.174e-01 2.109e-08 2.350e-08
## Iter: 2 fn: 1820.7727     Pars:  2.557e+03 4.881e+02 9.174e-01 2.060e-08 2.298e-08
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 923.7838  Pars:  2824.6175380845  301.2666242737    0.9858036034    0.0000001411    0.000
## Iter: 2 fn: 923.7838  Pars:  2824.6183856248  301.3321496759    0.9857557684    0.0000001223    0.000
## solnp--> Completed in 2 iterations
##
```

```
## Iter: 1 fn: 1820.7727     Pars:  2.557e+03 4.881e+02 9.172e-01 9.773e-09 1.343e-08 1.358e-09
## Iter: 2 fn: 1820.7727     Pars:  2.557e+03 4.880e+02 9.173e-01 9.305e-09 1.287e-08 1.009e-09
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 923.7838  Pars:  2.825e+03 3.016e+02 9.856e-01 6.392e-08 4.937e-08 1.481e-10
## Iter: 2 fn: 923.7838  Pars:  2.825e+03 3.016e+02 9.856e-01 6.380e-08 4.930e-08 1.245e-10
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 55968.1686    Pars:  2518.87409    8.88827
## solnp--> Solution not reliable....Problem Inverting Hessian.
```

```
## Warning in .sgarchfit(spec = spec, data = data, out.sample = out.sample, :
## ugarchfit-->warning: solver failer to converge.
```



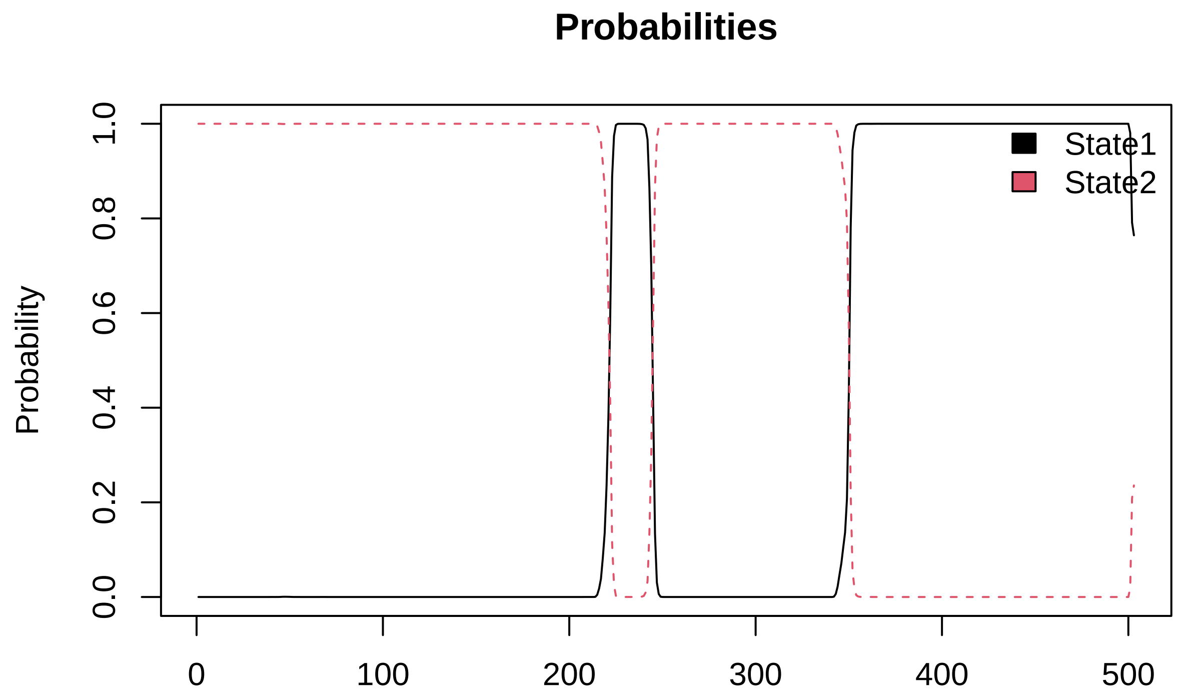
```
##
## Iter: 1 fn: 923.7838  Pars:  2.825e+03 3.016e+02 9.856e-01 6.392e-08 4.937e-08 1.481e-10
## Iter: 2 fn: 923.7838  Pars:  2.825e+03 3.016e+02 9.856e-01 6.380e-08 4.930e-08 1.245e-10
## solnp--> Completed in 2 iterations
```



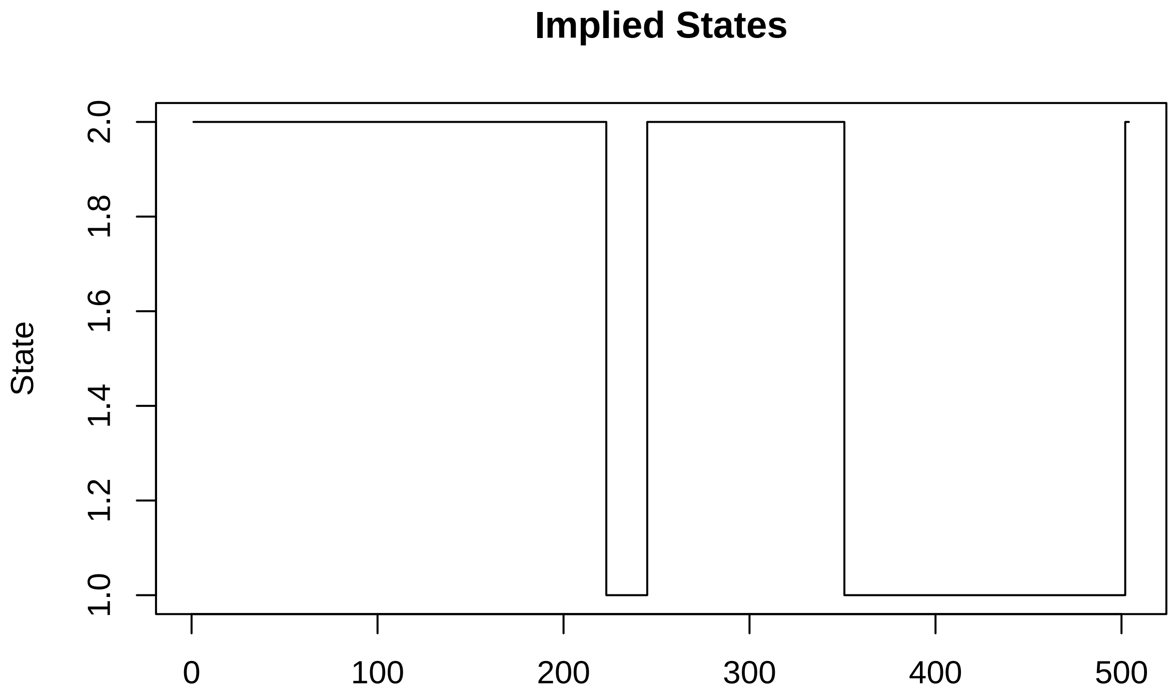
```
##
## Iter: 1 fn: 1820.7872     Pars:  2556.530267683  487.996329680    0.916299694    0.000007949
## Iter: 2 fn: 1820.7872     Pars:  2556.516201288  488.466433852    0.916129966    0.000006836
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 930.0806  Pars:  2.825e+03 3.018e+02 9.841e-01 9.856e-09
## Iter: 2 fn: 930.0806  Pars:  2.825e+03 3.017e+02 9.843e-01 6.165e-09
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 1820.7727     Pars:  2.557e+03 4.881e+02 9.174e-01 3.386e-08 1.210e-08
## Iter: 2 fn: 1820.7727     Pars:  2.557e+03 4.882e+02 9.174e-01 2.712e-08 8.245e-09
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 930.4099  Pars:  2.825e+03 3.011e+02 9.854e-01 7.364e-08 1.523e-10
## Iter: 2 fn: 930.4099  Pars:  2.825e+03 3.011e+02 9.854e-01 7.349e-08 1.287e-10
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 1820.7727     Pars:  2.557e+03 4.881e+02 9.174e-01 2.109e-08 2.350e-08
## Iter: 2 fn: 1820.7727     Pars:  2.557e+03 4.881e+02 9.174e-01 2.060e-08 2.298e-08
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 930.4099  Pars:  2.825e+03 3.003e+02 9.860e-01 6.680e-09 5.529e-09
## Iter: 2 fn: 930.4099  Pars:  2.825e+03 3.011e+02 9.855e-01 2.745e-09 2.123e-09
## Iter: 3 fn: 930.4099  Pars:  2.825e+03 3.011e+02 9.855e-01 2.624e-09 2.016e-09
## solnp--> Completed in 3 iterations
##
```

```
## Iter: 1 fn: 1820.7727     Pars:  2.557e+03 4.881e+02 9.172e-01 9.773e-09 1.343e-08 1.358e-09
## Iter: 2 fn: 1820.7727     Pars:  2.557e+03 4.880e+02 9.173e-01 9.305e-09 1.287e-08 1.009e-09
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 930.4099  Pars:  2.825e+03 3.011e+02 9.853e-01 4.089e-07 4.192e-07 5.997e-08
## Iter: 2 fn: 930.4099  Pars:  2.825e+03 3.012e+02 9.853e-01 3.781e-07 3.880e-07 5.471e-08
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 55968.1686    Pars:  2518.87409    8.88827
## solnp--> Solution not reliable....Problem Inverting Hessian.
```

```
## Warning in .sgarchfit(spec = spec, data = data, out.sample = out.sample, :
## ugarchfit-->warning: solver failer to converge.
```



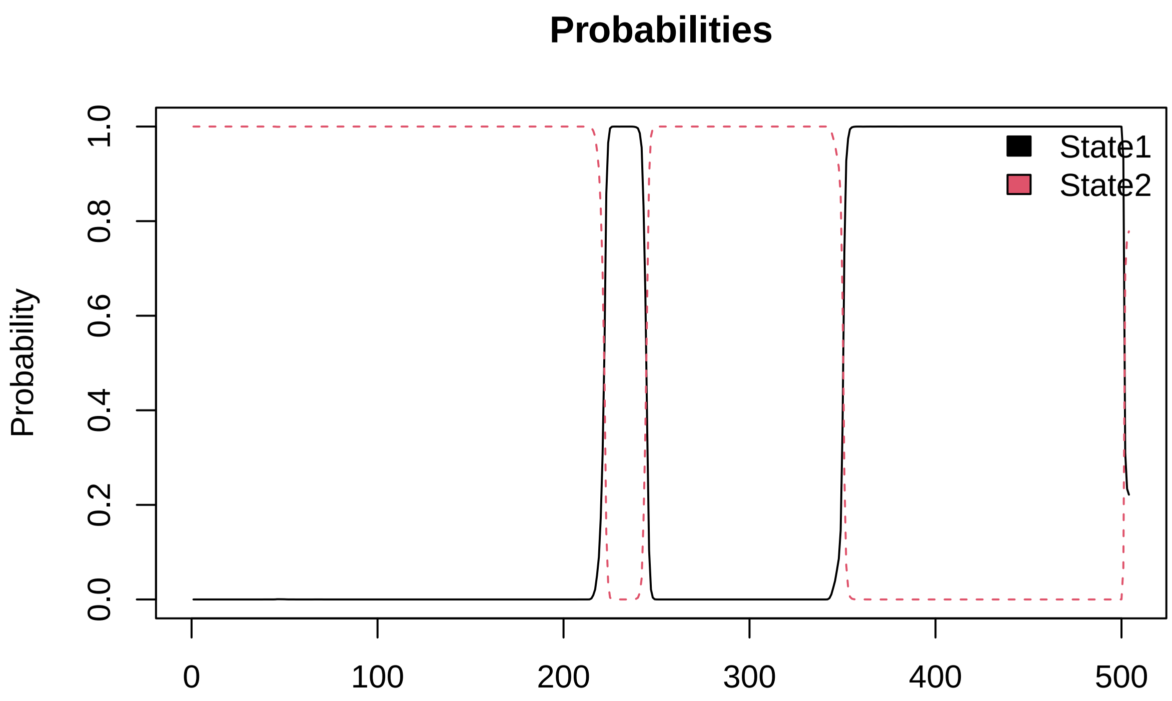
```
##
## Iter: 1 fn: 930.4099  Pars:  2.825e+03 3.011e+02 9.853e-01 4.089e-07 4.192e-07 5.997e-08
## Iter: 2 fn: 930.4099  Pars:  2.825e+03 3.012e+02 9.853e-01 3.781e-07 3.880e-07 5.471e-08
## solnp--> Completed in 2 iterations
```



```
##
## Iter: 1 fn: 1838.0095     Pars:  2557.258687088  485.184037762    0.916509523    0.000003679
## Iter: 2 fn: 1838.0095     Pars:  2557.258682356  485.184459974    0.916509398    0.000003679
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 915.9010  Pars:  2824.50321125  296.87927757    0.99898647    0.00001118
## Iter: 2 fn: 915.8497  Pars:  2.825e+03 3.079e+02 9.501e-01 1.328e-12
## Iter: 3 fn: 915.8497  Pars:  2.825e+03 3.079e+02 9.501e-01 2.023e-13
## solnp--> Completed in 3 iterations
##
## Iter: 1 fn: 1837.9946     Pars:  2557.2777543279  483.9545572346    0.9178512923    0.0000002395    0
## Iter: 2 fn: 1837.9946     Pars:  2557.2784439720  484.1691061610    0.9177982902    0.0000001910    0
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 916.2390  Pars:  2.825e+03 3.075e+02 9.506e-01 1.329e-07 1.308e-09
## Iter: 2 fn: 916.2390  Pars:  2.825e+03 3.074e+02 9.508e-01 1.241e-07 4.181e-10
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 1837.9946     Pars:  2557.2738247995  484.2574099207    0.9177532376    0.0000003512    0
## Iter: 2 fn: 1837.9946     Pars:  2557.2737426734  484.2607576968    0.9177524056    0.0000003500    0
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 916.2390  Pars:  2.825e+03 3.071e+02 9.511e-01 1.531e-08 1.467e-08
## Iter: 2 fn: 916.2390  Pars:  2.825e+03 3.074e+02 9.508e-01 7.177e-09 7.551e-09
## solnp--> Completed in 2 iterations
##
```

```
## Iter: 1 fn: 1837.9947     Pars:  2557.289793524  483.953820269    0.918111754    0.000001543    0.000
## Iter: 2 fn: 1837.9947     Pars:  2557.2869362986  483.9510299329    0.9178111740    0.0000009407    0
## Iter: 3 fn: 1837.9947     Pars:  2557.2875455572  483.9788198536    0.9179551247    0.0000008995    0
## solnp--> Completed in 3 iterations
##
## Iter: 1 fn: 916.2390  Pars:  2.825e+03 3.073e+02 9.511e-01 1.116e-07 1.148e-07 1.718e-09
## Iter: 2 fn: 916.2390  Pars:  2.825e+03 3.072e+02 9.511e-01 1.107e-07 1.140e-07 1.601e-09
## solnp--> Completed in 2 iterations
##
## Iter: 1 fn: 56803.2876    Pars:  2519.87337    8.91708
## solnp--> Solution not reliable....Problem Inverting Hessian.
```

```
## Warning in .sgarchfit(spec = spec, data = data, out.sample = out.sample, :
## ugarchfit-->warning: solver failer to converge.
```

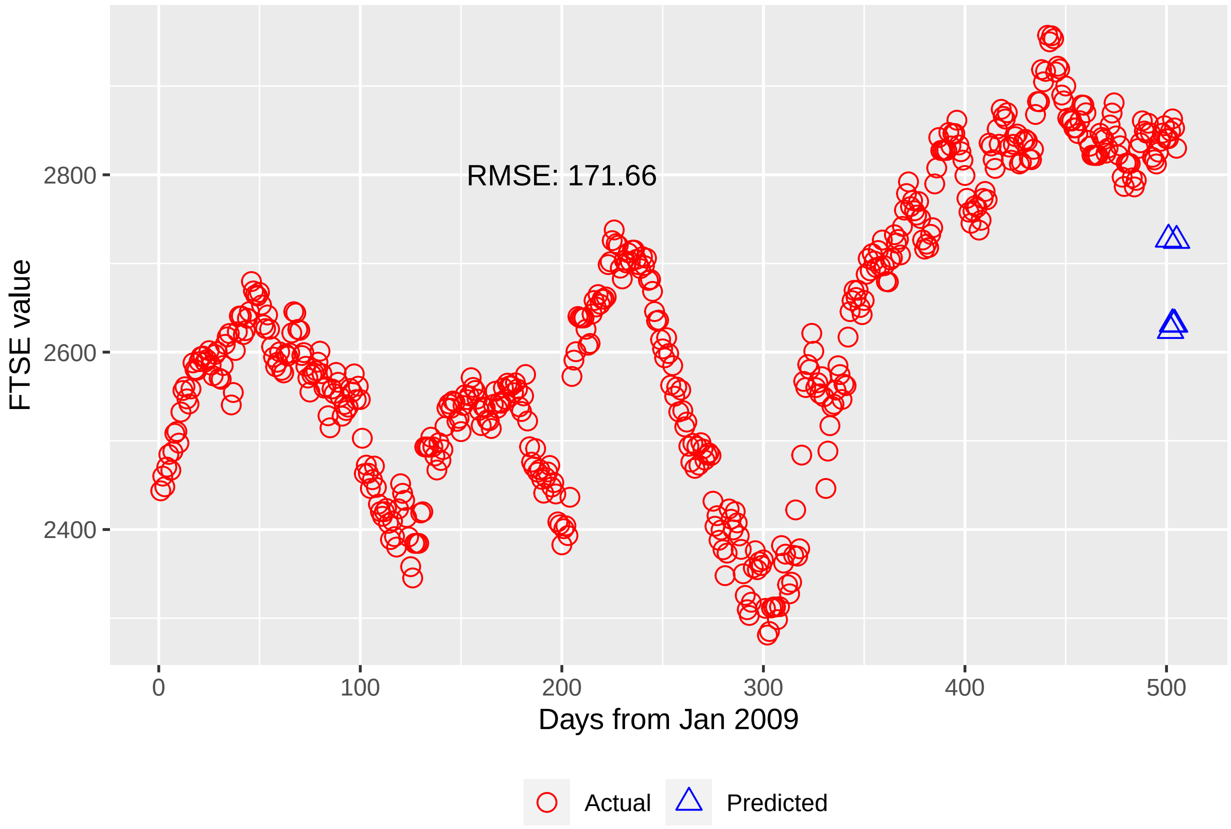


```
##
## Iter: 1 fn: 916.2390  Pars:  2.825e+03 3.073e+02 9.511e-01 1.116e-07 1.148e-07 1.718e-09
## Iter: 2 fn: 916.2390  Pars:  2.825e+03 3.072e+02 9.511e-01 1.107e-07 1.140e-07 1.601e-09
## solnp--> Completed in 2 iterations
```

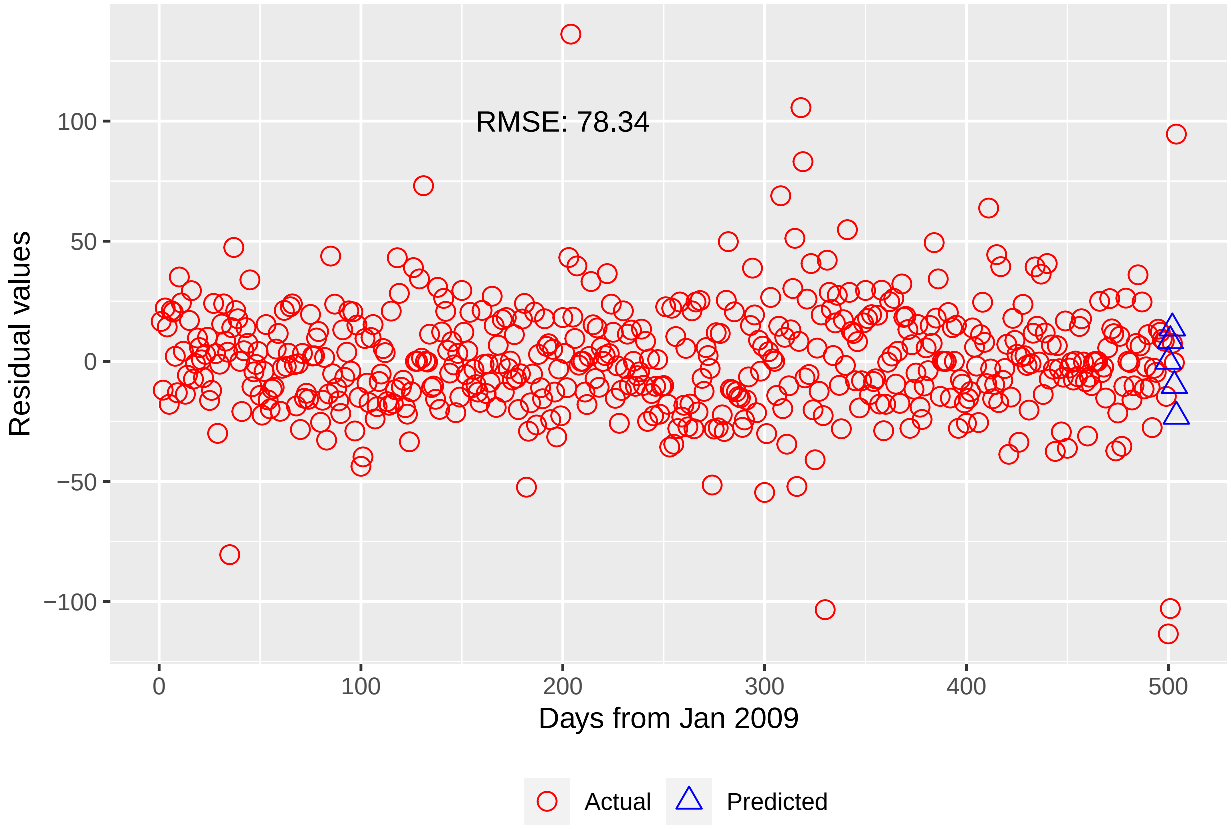
```
##########################################################################

RMSE = mean(((data[500:505] - EuStockMarkets[500:505,4])^2))^0.5
RMSE
```

```
## [1] 171.6614
```

```
xaxis = c(1:505,501:505)
yaxis = c(data,EuStockMarkets[501:505,4])
col = as.character(c(rep(1,500),rep(2,5),rep(1,5)))

df = data.frame(X = xaxis, Y = yaxis, Z = col)

plot = ggplot(df,aes(x = X, y = Y, col = Z,shape = Z)) + geom_point(size= 3)
plot = plot + scale_shape_manual(values = c("1" = 1, "2" = 2),labels = c("1" = "Actual","2" = "Predicte
plot = plot + scale_color_manual(values = c("1" = "red", "2" = "blue"),labels = c("1" = "Actual","2" =
plot = plot + annotate("text", label = paste("RMSE:",round(RMSE,2)), x = 200, y = 2800) + theme(legend.
plot = plot + xlab("Days from Jan 2009") + ylab("FTSE value")
plot
```



```
residueactual = EuStockMarkets[501:505] - EuStockMarkets[500:504]
residuefcst = data[501:505] - data[500:504]

RMSE = (mean((residueactual - residuefcst)^2)) ^ 0.5
xaxis = c(1:504,500:504)
yaxis = c((data[-1] - data[-length(data)]),(EuStockMarkets[501:505,4] - EuStockMarkets[500:504,4]))
col = as.character(c(rep(1,504),rep(2,5)))
df = data.frame(X = xaxis, Y = yaxis, Z = col)

plot = ggplot(df,aes(x = X, y = Y, col = Z,shape = Z)) + geom_point(size =3)
plot = plot + scale_shape_manual(values = c("1" = 1, "2" = 2),labels = c("1" = "Actual","2" = "Predicte
plot = plot + scale_color_manual(values = c("1" = "red", "2" = "blue"),labels = c("1" = "Actual","2" =
plot = plot + annotate("text", label = paste("RMSE:",round(RMSE,2)), x = 200, y = 100) + theme(legend.p
plot = plot + xlab("Days from Jan 2009") + ylab("Residual values")
plot
```


RMSE: 78.34
Residual values
Days from Jan 2009
○ Actual
△ Predicted