Project 3 : Markov switching models

Statistics for Smart Data

Mattia Guerri - Hugo Brehier

November 29th 2019

Contents

Introduction

In this project, we study Markov Switching Models. First, a simulation study is done. Next, we define the model and the estimation algorithm. Finally, we apply it to real financial data.

Simulated Data

Install the depmixS4 and quantmod libraries

```
library('depmixS4')
library('quantmod')
library(ggplot2)
library(scales)
set.seed(1)
```

Create the parameters for the bull and bear market returns distributions

```
Nklower <- 50
Nkupper <- 150
bullmean <- 0.1
bullvar <- 0.1
bearmean <- -0.05
bearvar <- 0.2
```

Create the list of durations (in days) for each regime

```
days = replicate(5, sample(Nklower :Nkupper, 1))
```

Create the various bull and bear markets returns

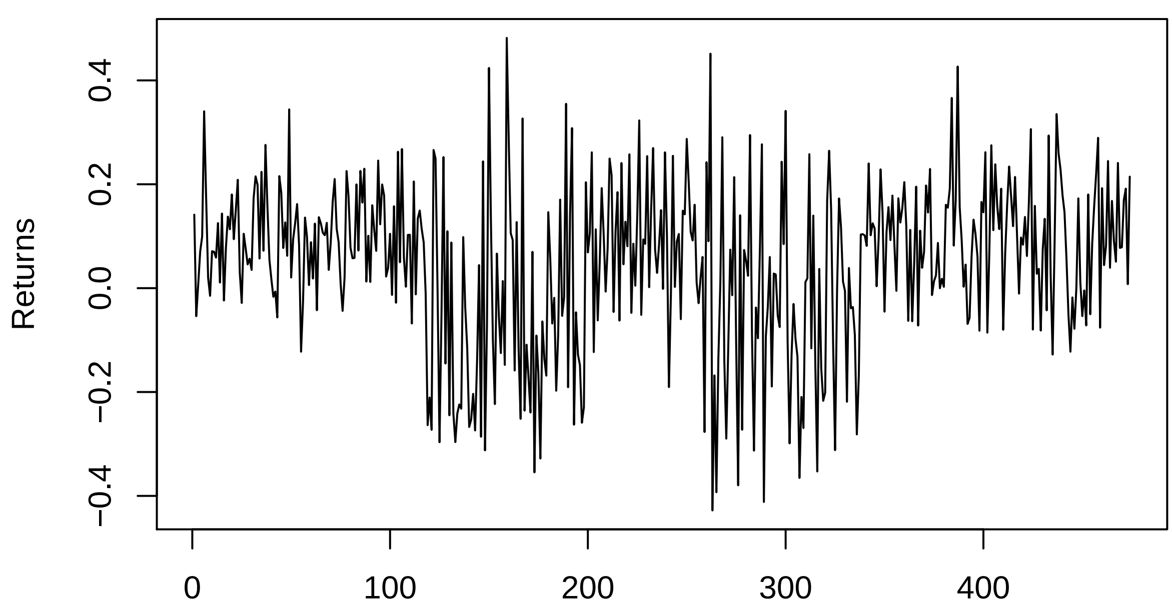
```
marketbull1 <- rnorm( days[1], bullmean, bullvar )
marketbear2 <- rnorm( days[2], bearmean, bearvar )
marketbull3 <- rnorm( days[3], bullmean, bullvar )
marketbear4 <- rnorm( days[4], bearmean, bearvar )
marketbull5 <- rnorm( days[5], bullmean, bullvar )
```

Create the list of true regime states and full returns list

```
trueregimes <-c( rep(2,days[1]), rep(1,days[2]), rep(2,days[3]), rep(1,days[4]), rep(2,days[5]))
returns <-c( marketbull1, marketbear2, marketbull3, marketbear4, marketbull5)
```

Plotting the returns shows the clear changes in mean and variance between the regime switches.

```
plot(returns, type="l", xlab='', ylab="Returns")
```



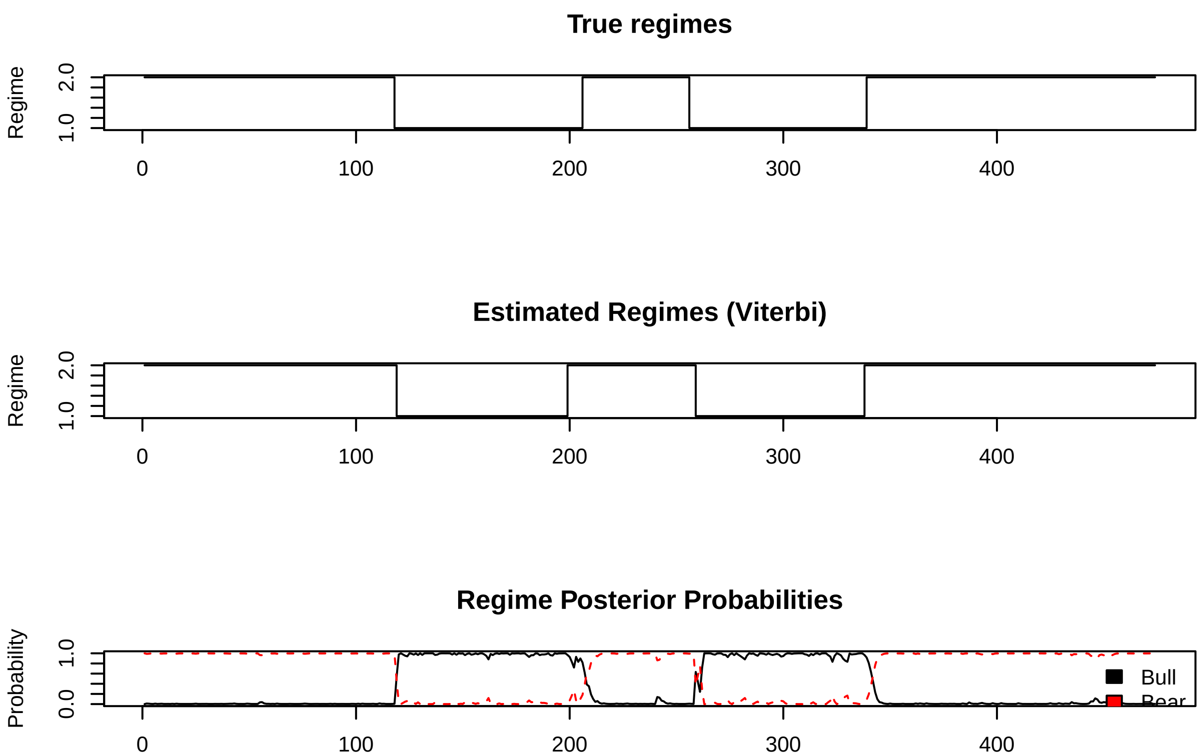
Create and fit the Hidden Markov Model

```
hmm = depmix(returns ~ 1, family = gaussian(), nstates = 2, data=data.frame(returns))
hmmfit = fit(hmm, verbose = FALSE)
```

```
## converged at iteration 24 with logLik: 289.6389
```

Output both the true regimes and the posterior probabilities of the regimes

```
postprobs <- posterior(hmmfit)
layout(1 :3)
plot(trueregimes,type = 's',main = 'True regimes',xlab='', ylab='Regime')
plot(postprobs$state, type='s', main='Estimated Regimes (Viterbi)', xlab='', ylab='Regime')
matplot(postprobs[, -1], type='l', main='Regime Posterior Probabilities',ylab='Probability')
legend(x='topright', c('Bull','Bear'), fill=1:2, bty='n')
```



Markov Switching Models

The model used in the simulation study is called a Markov Switching Model [3] with $k = 2$ regimes, bear and bull. Markov Switching Model are defined as [2] [1] :

$$Y_t = \mu_{S_t} + \sigma_{S_t}\epsilon_t \text{ where } \epsilon_t \sim \mathcal{N}(0,1) \text{ and } S_t \in \mathbb{S} = \{1,\dots,m\}$$

For each state $k \in \mathbb{S}$, we have:

$$\theta_k = (\mu_k, \sigma_k)$$

The hidden state process S_t is markovian :

$$\forall i,j \quad \mathbb{P}(S_t = j|S_{t-1} = i, S_{1:t-2}, (\epsilon_t)_t) = \mathbb{P}(S_t = j|S_{t-1} = i) = p_{i,j}$$

We have $P = (p_{i,j})_{i,j}$ the $m \times m$ transition matrix. It is stochastic:

$$\forall i,j \quad p_{i,j} \in [0,1], \quad \forall i \quad \sum_j p_{i,j} = 1$$

Then the distribution of Y_t conditional to S_t is:

$$(Y_t|S_t = k) \sim \mathcal{D}(\theta_k) = \mathcal{N}(\mu_k, \sigma_k^2)$$

The initial probability distribution of states is defined by the vector $\boldsymbol{\pi_1} = (\pi_{1k})_k = (\mathbb{P}(S_1 = k))_k$.

The non-conditional density of Y_t is:

$$\sum_{n=1}^{k} \pi_{tk} p(Y_t|\mu_k, \sigma_k)$$

In our case the hidden state space is discrete (regimes) , while the observations are continuous (returns). We have to optimize the model parameters, composed of the regimes parameters $\theta_k = (\mu_k, \sigma_k)$,the transition matrix P and $\boldsymbol{\pi_t}$. Based on observations $y_{1:n}$,random or prior initial parameters, we use the EM algorithm for the discrete state space. (see course chap 2 part 4.1) After this, the estimation of the most likely hidden states sequence to have occured based on the observations and estimated parameters is done by the Viterbi algorithm.(see course chap 2 part 3)

Application to real financial data

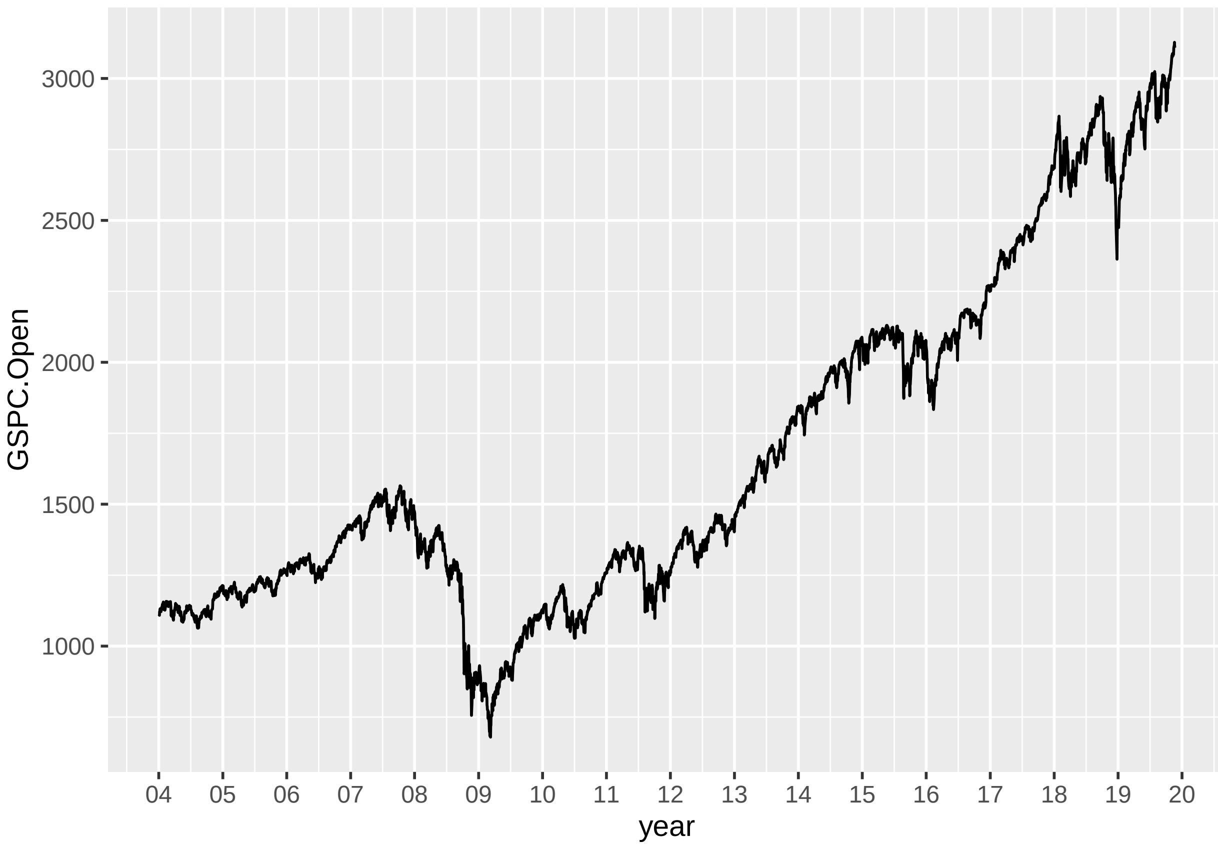
S&P Index

Obtain S&P500 data from 2004 onwards

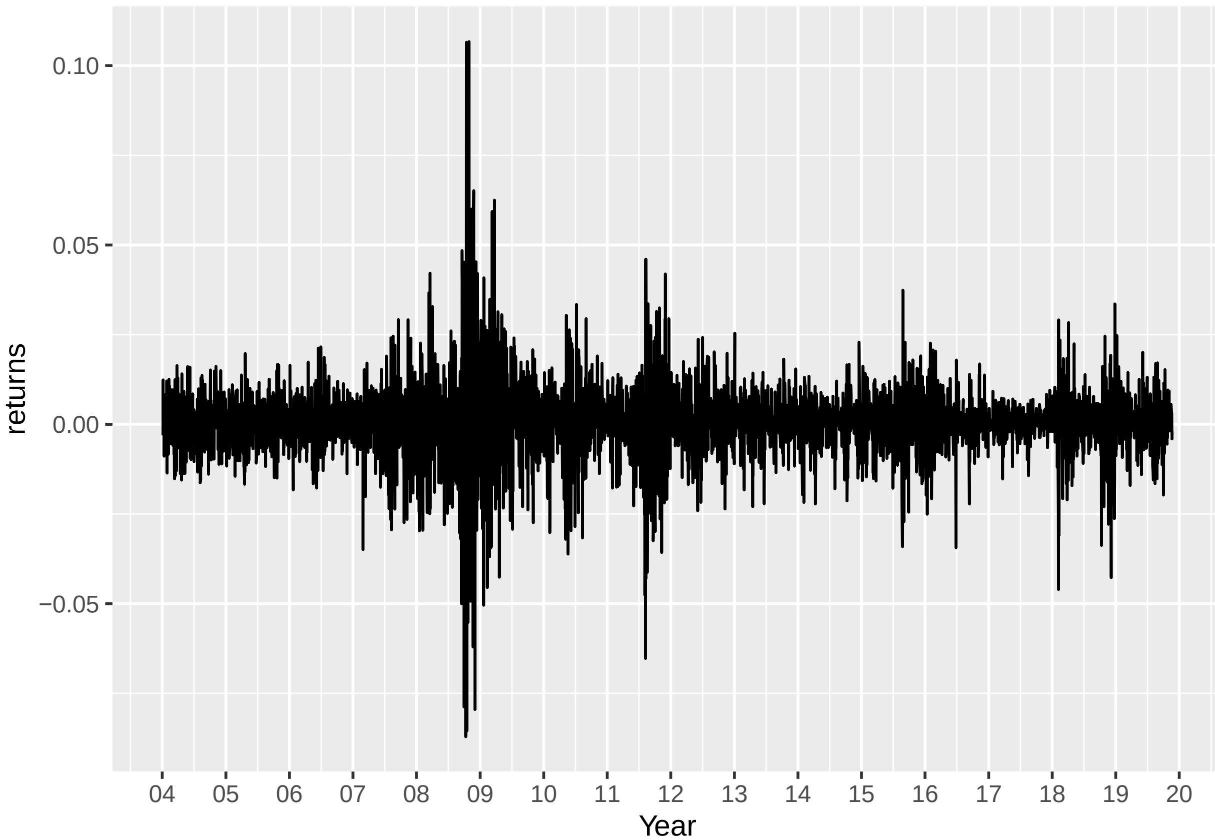
```
getSymbols( '^GSPC',src="yahoo", from="2004-01-01")
```

```
## [1] "^GSPC"
```

```
values = as.data.frame(GSPC)
n= nrow(values)
values$date = as.Date(rownames(values))

ggplot(values, aes(x=date, y=GSPC.Open,group = 1)) + geom_line() + scale_x_date(breaks = date_breaks("ye
  labels = date_format('%y')) + xlab('year')
```




```
gspc.returns = as.data.frame((values[2:n, 1] - values[1:(n-1), 1]) /  values[1:(n-1), 1])
gspc.returns$date = as.Date(rownames(values)[2:n])

colnames(gspc.returns) = c('returns','date')

ggplot(gspc.returns, aes(x=date, y=returns,group = 1)) + geom_line() + scale_x_date(breaks = date_break
  labels = date_format('%y')) + xlab('Year')
```




```
k=10
AICs = numeric(length(2:k))

for (i in 2:k) {
  hmm <- depmix(returns ~ 1, family = gaussian(), nstates = i, data=data.frame(gspc.returns))
  hmmfit <- fit(hmm, verbose = FALSE)
  AICs[i-1] = AIC(hmmfit)
}
```

```
## converged at iteration 54 with logLik: 13371.82
## converged at iteration 90 with logLik: 13555.77
## converged at iteration 453 with logLik: 13589.83
## converged at iteration 341 with logLik: 13609.15
```

```
bestk = which.min(AICs) + 1
bestk
```

```
## [1] 9
```

```
hmm <- depmix(returns ~ 1, family = gaussian(), nstates = bestk, data=data.frame(gspc.returns))
hmmfit <- fit(hmm, verbose = FALSE)
```

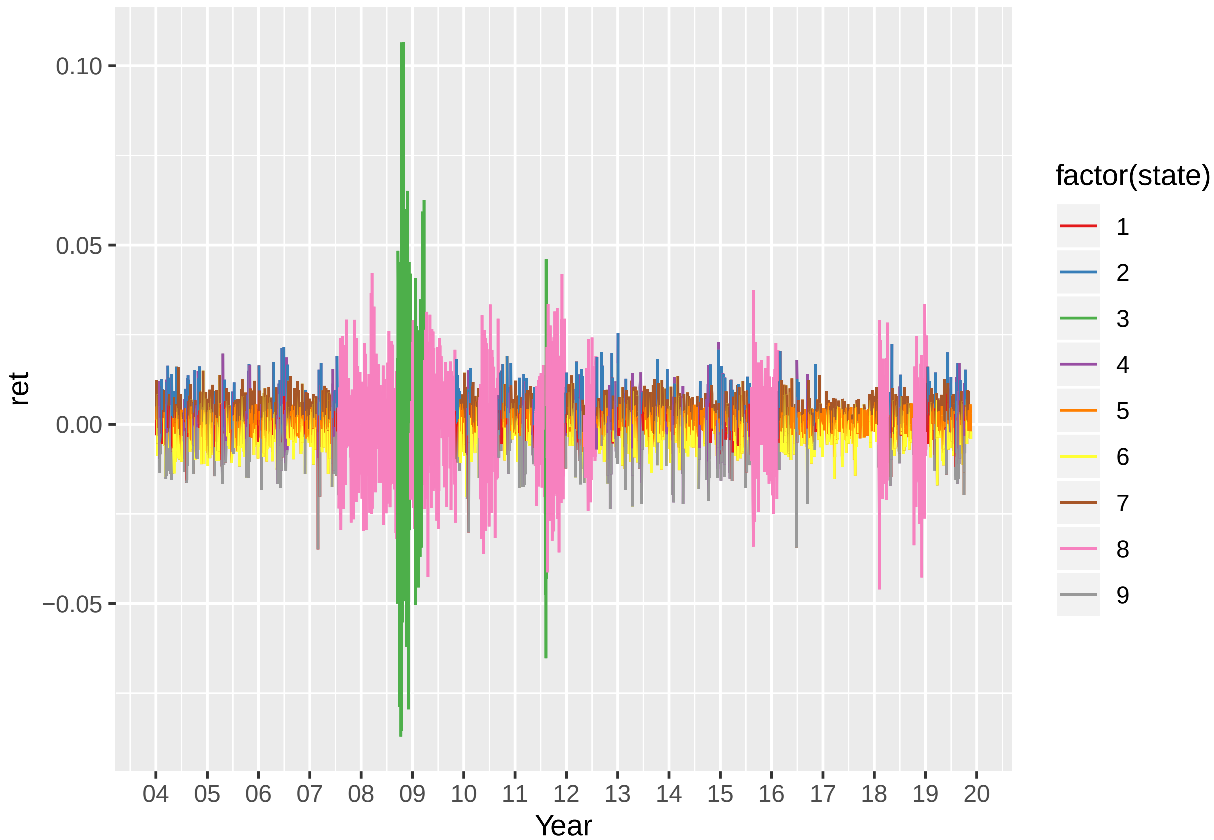
```
summary(hmmfit)
```

```
## Initial state probabilties model
## pr1 pr2 pr3 pr4 pr5 pr6 pr7 pr8 pr9
##   1   0   0   0   0   0   0   0   0
##
## Transition matrix
##         toS1  toS2  toS3  toS4  toS5  toS6  toS7  toS8  toS9
```

```
## fromS1 0.442 0.130 0.000 0.000 0.000 0.000 0.214 0.000 0.214
## fromS2 0.760 0.029 0.000 0.000 0.114 0.000 0.096 0.000 0.001
## fromS3 0.000 0.000 0.969 0.000 0.000 0.000 0.000 0.031 0.000
## fromS4 0.000 0.271 0.000 0.386 0.000 0.000 0.000 0.000 0.343
## fromS5 0.000 0.003 0.000 0.000 0.571 0.313 0.113 0.000 0.000
## fromS6 0.099 0.000 0.000 0.000 0.165 0.242 0.419 0.000 0.075
## fromS7 0.000 0.000 0.000 0.000 0.582 0.218 0.183 0.000 0.016
## fromS8 0.000 0.000 0.004 0.023 0.000 0.000 0.000 0.972 0.000
## fromS9 0.000 0.000 0.004 0.734 0.000 0.000 0.000 0.038 0.224
##
## Response parameters
## Resp 1 : gaussian
##     Re1.(Intercept) Re1.sd
## St1           0.000  0.004
## St2           0.012  0.005
## St3          -0.003  0.036
## St4           0.003  0.007
## St5           0.001  0.002
## St6          -0.004  0.004
## St7           0.006  0.003
## St8           0.000  0.015
## St9          -0.010  0.007
```

```r
dat = posterior(hmmfit)
dat$date = gspc.returns$date
dat$ret = gspc.returns$returns
```

```r
ggplot(dat, aes(x=date, y=ret,color=factor(state),group = 1)) + geom_line() + scale_x_date(breaks = dat
  labels = date_format('%y')) + xlab('Year') + scale_color_brewer(palette="Set1")
```



The most volatil bear state appears almost only during the subprime crisis. It shows the market was in an exceptional state.

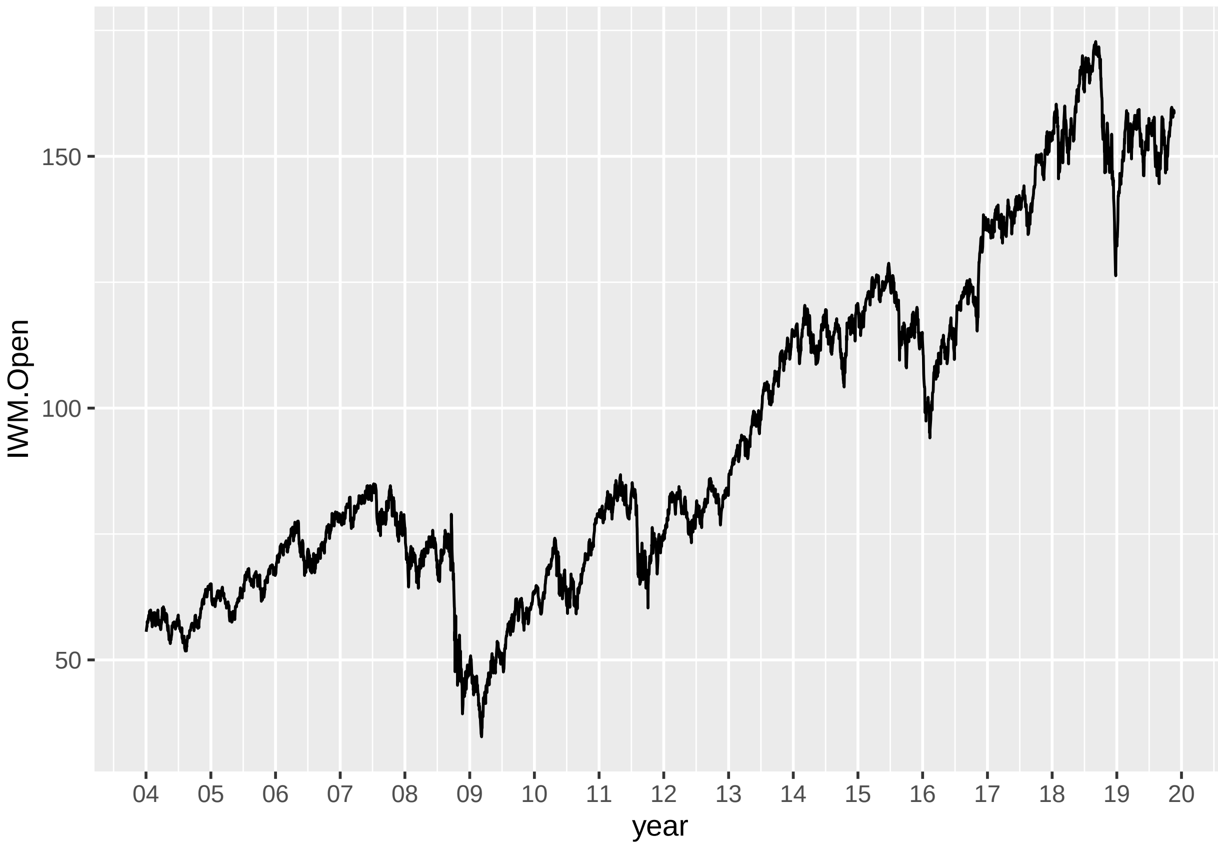
IWM

From etf.com: IWM tracks a market-cap-weighted index of US small-cap stocks. The index selects stocks ranked 1,001-3,000 by market cap.

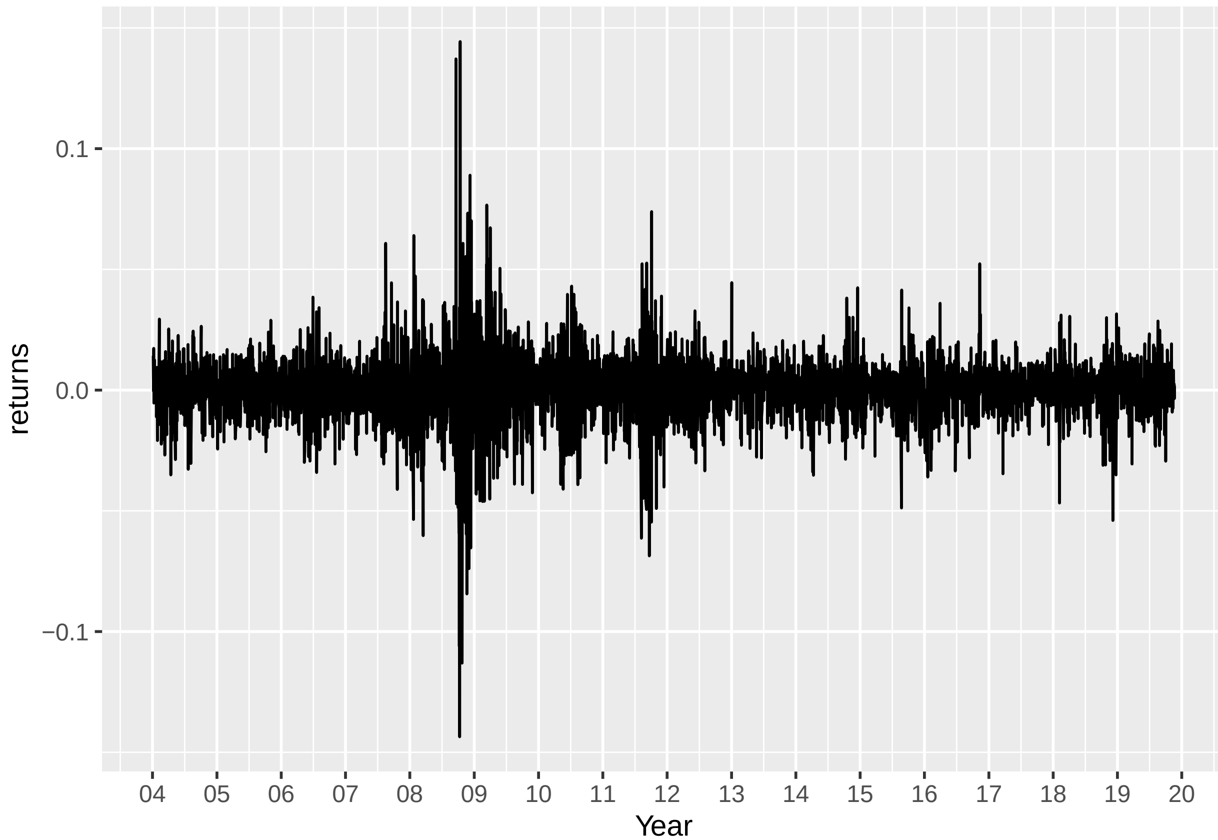
```
getSymbols( 'IWM',src="yahoo", from="2004-01-01" )
```

```
## [1] "IWM"
```

```
values = as.data.frame(IWM)
n= nrow(values)
values$date = as.Date(rownames(values))

ggplot(values, aes(x=date, y=IWM.Open,group = 1)) + geom_line() + scale_x_date(breaks = date_breaks("yea
  labels = date_format('%y')) + xlab('year')
```




```
iwm.returns = as.data.frame((values[2:n, 1] - values[1:(n-1), 1]) /  values[1:(n-1), 1])
iwm.returns$date = as.Date(rownames(values)[2:n])

colnames(iwm.returns) = c('returns','date')

ggplot(iwm.returns, aes(x=date, y=returns,group = 1)) + geom_line() + scale_x_date(breaks = date_breaks
  labels = date_format('%y')) + xlab('Year')
```




```
k=10
AICs = numeric(length(2:k))

for (i in 2:k) {
  hmm <- depmix(returns ~ 1, family = gaussian(), nstates = i, data=data.frame(iwm.returns))
  hmmfit <- fit(hmm, verbose = FALSE)
  AICs[i-1] = AIC(hmmfit)
}
```

```
## converged at iteration 42 with logLik: 11952.08
## converged at iteration 144 with logLik: 12075.47
## converged at iteration 452 with logLik: 12104.5
```

```
bestk = which.min(AICs) +1
bestk
```

```
## [1] 7
```

```
hmm <- depmix(returns ~ 1, family = gaussian(), nstates = bestk, data=data.frame(iwm.returns))
hmmfit <- fit(hmm, verbose = FALSE)
```

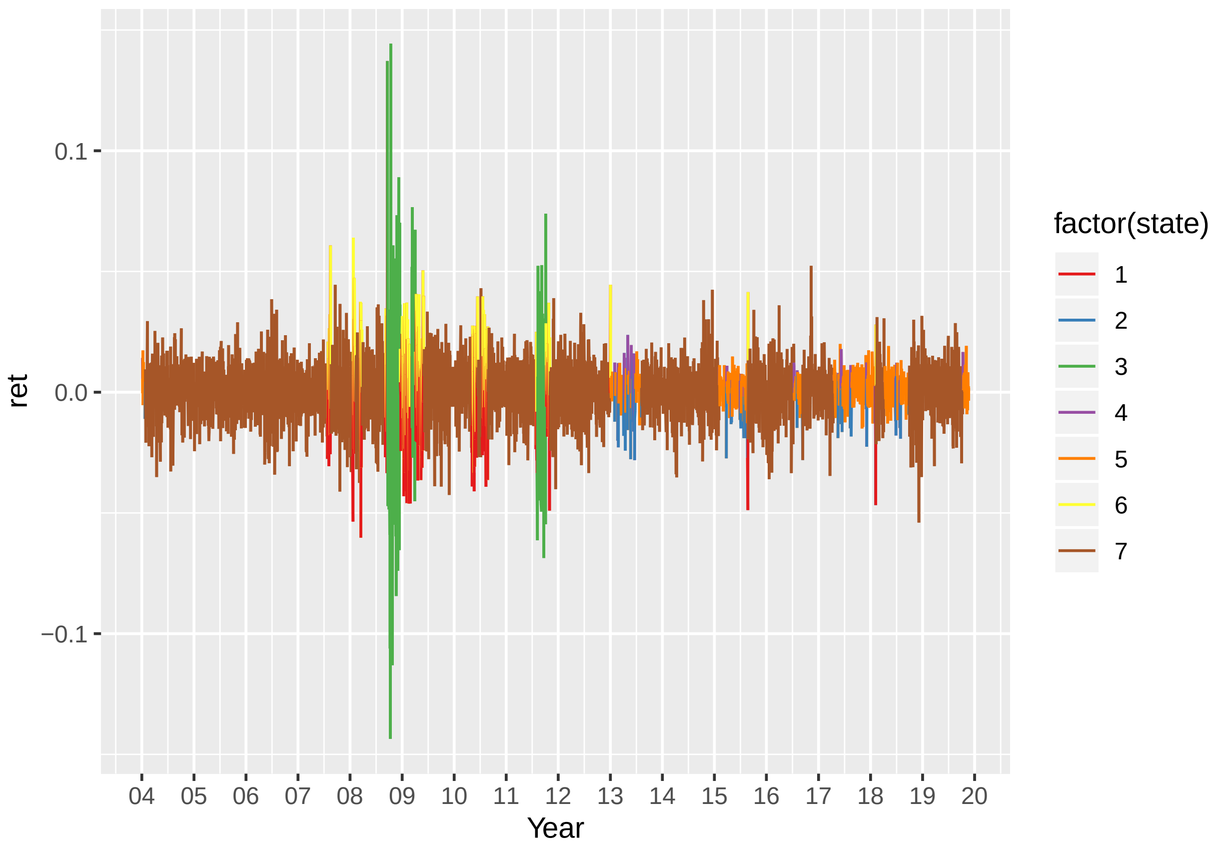
```
summary(hmmfit)
```

```
## Initial state probabilties model
## pr1 pr2 pr3 pr4 pr5 pr6 pr7
##   0   0   0   1   0   0   0
##
## Transition matrix
##         toS1  toS2  toS3  toS4  toS5  toS6  toS7
## fromS1 0.538 0.000 0.020 0.000 0.000 0.443 0.000
```

```
## fromS2 0.014 0.316 0.000 0.645 0.000 0.000 0.024
## fromS3 0.000 0.000 0.964 0.000 0.000 0.036 0.000
## fromS4 0.000 0.266 0.000 0.516 0.219 0.000 0.000
## fromS5 0.000 0.144 0.000 0.000 0.856 0.000 0.000
## fromS6 0.544 0.000 0.000 0.000 0.034 0.312 0.110
## fromS7 0.014 0.000 0.000 0.013 0.000 0.000 0.973
##
## Response parameters
## Resp 1 : gaussian
##     Re1.(Intercept) Re1.sd
## St1          -0.014  0.016
## St2          -0.009  0.008
## St3          -0.001  0.045
## St4           0.006  0.007
## St5           0.002  0.006
## St6           0.019  0.014
## St7           0.000  0.013
```

```r
dat = posterior(hmmfit)
dat$date = iwm.returns$date
dat$ret = iwm.returns$returns
```

```r
ggplot(dat, aes(x=date, y=ret,color=factor(state),group = 1)) + geom_line() + scale_x_date(breaks = dat
  labels = date_format('%y')) + xlab('Year') + scale_color_brewer(palette="Set1")
```



Like with the S&P index, the most volatil bear state appears during the crisis.

References

[1] Franck Arnaud. Markov switching models, 2012.

[2] Gilles de Truchis and Elena Dumitrescu. Économétrie non-linéaire, 2016.

[3] James D. Hamilton. Regime-switching models. 2005.