Project 3 : Markov switching models

Statistics for Smart Data

Hugo Brehier - Mattia Guerri

November 29^{th} 2019

Contents

Introduction

In this project, we model the return of two financial indicators. We are interested in considering the process as a special case of hidden Markov models, Markov switching models.

At first, in order to understand the provided code, we analyse simulated data. The observed signal is defined by the financial returns time series. We assume that the returns at each time interval are drawn by either one of two Gaussian distributions and that the returns switch between the various distributions. The hidden state space is thus represented by the regimes with different mean and variance. We take into account normal distribution with positive mean and small variance, modelling a market bull phase, and distribution with mean very close to zero and large variance, to simulate a market in a bear stage.

After the simulation, we describe with more details the modelling strategy.

Subsequently, we focus our attention on real financial data, in particular, on two stock market indexes:

- Standard and Poor's 500, an index measuring the performance of the shares of 500 large companies listed on the USA stock exchanges
- IWM, linked to the performance of 2000 America companies with smaller market cap.

We will use the following R packages:

- depmixS4 to estimate the models
- quantmod to download 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

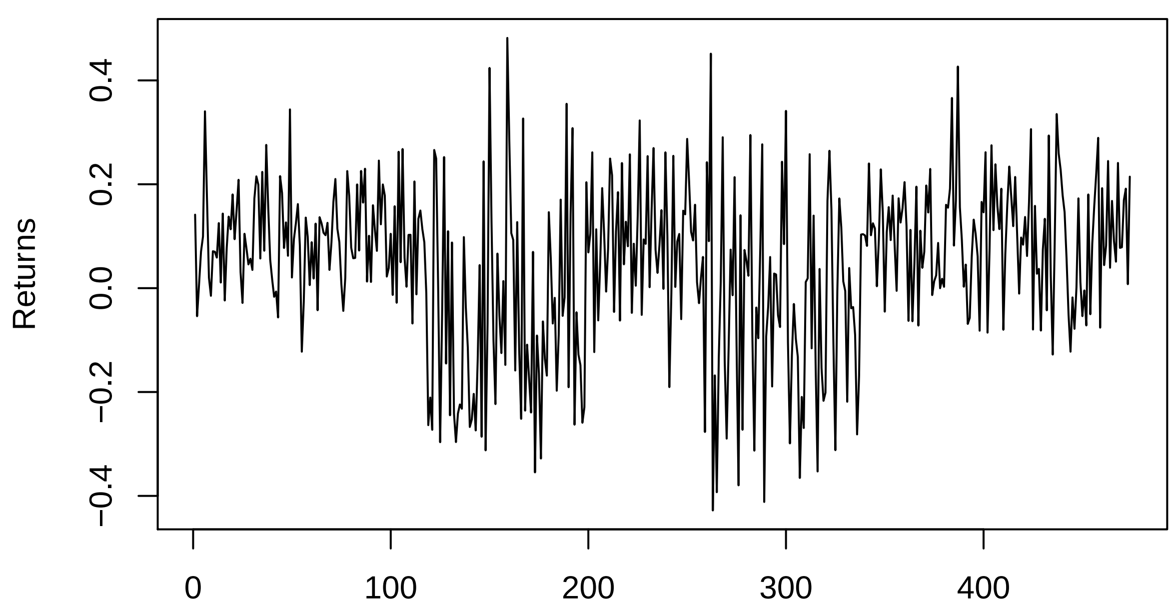
```
marketbull1 <- rnorm( days[1], bullmean, bullvar )
marketbear2 <- rnorm( days[2], bearmean, bearvar )
marketbull3 <- rnorm( days[3], bullmean, bullvar )
marketbear4 <- rnorm( days[4], bearmean, bearvar )
marketbull5 <- rnorm( days[5], bullmean, bullvar )
```

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")
```



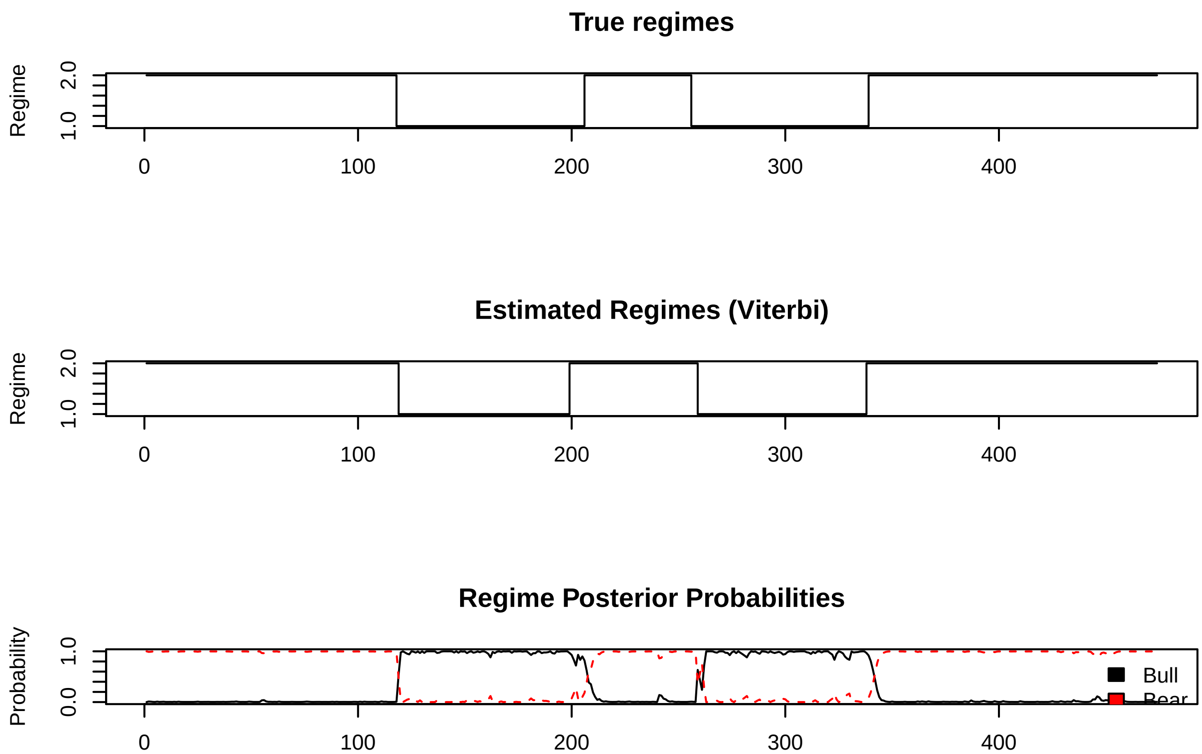
Create and fit the Hidden Markov Model

```
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')
```



The regimes are well retrieved by the model. Only at around $t = 210$ can we observe a difficult period to estimate correctly.

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 by an observation random process $(Y_t)_t$ and a hidden state process $(S_t)_t$, such that [2] [1] :

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

$\forall k \in \mathbb{S}$, 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 density of states at time t is defined by the vector $\boldsymbol{\pi_k} = (\pi_k)_k = (\mathbb{P}(S_t = k))_k$.

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

We estimate MSM to daily returns on opening values of our financial indexes. To choose the optimal number of regimes, we compare AIC of models with $n = \{2, \ldots, 15\}$ regimes.

S&P Index

Obtain S&P500 data from 2004 onwards with quantmod.

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

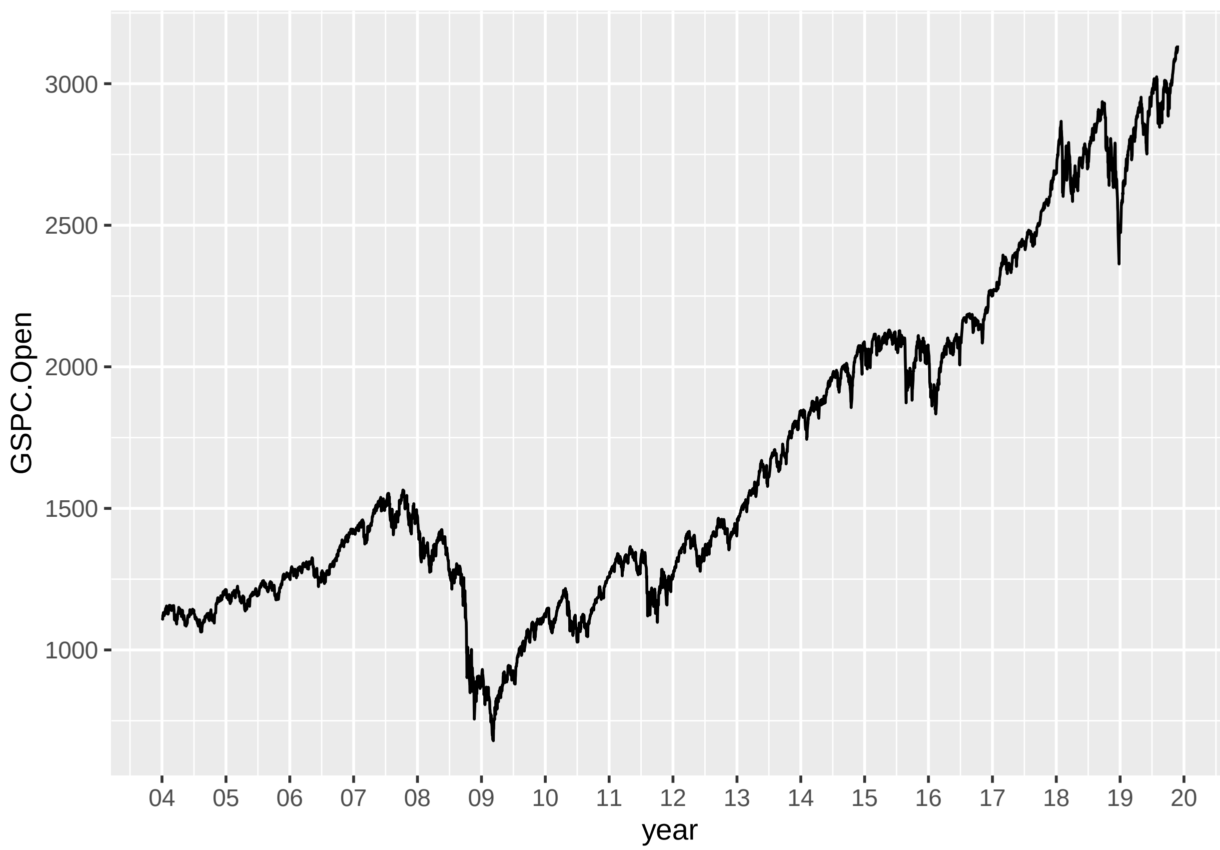
```
## [1] "^GSPC"
```

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

There are 4004 opening values of the index.

Opening values of S&P500

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

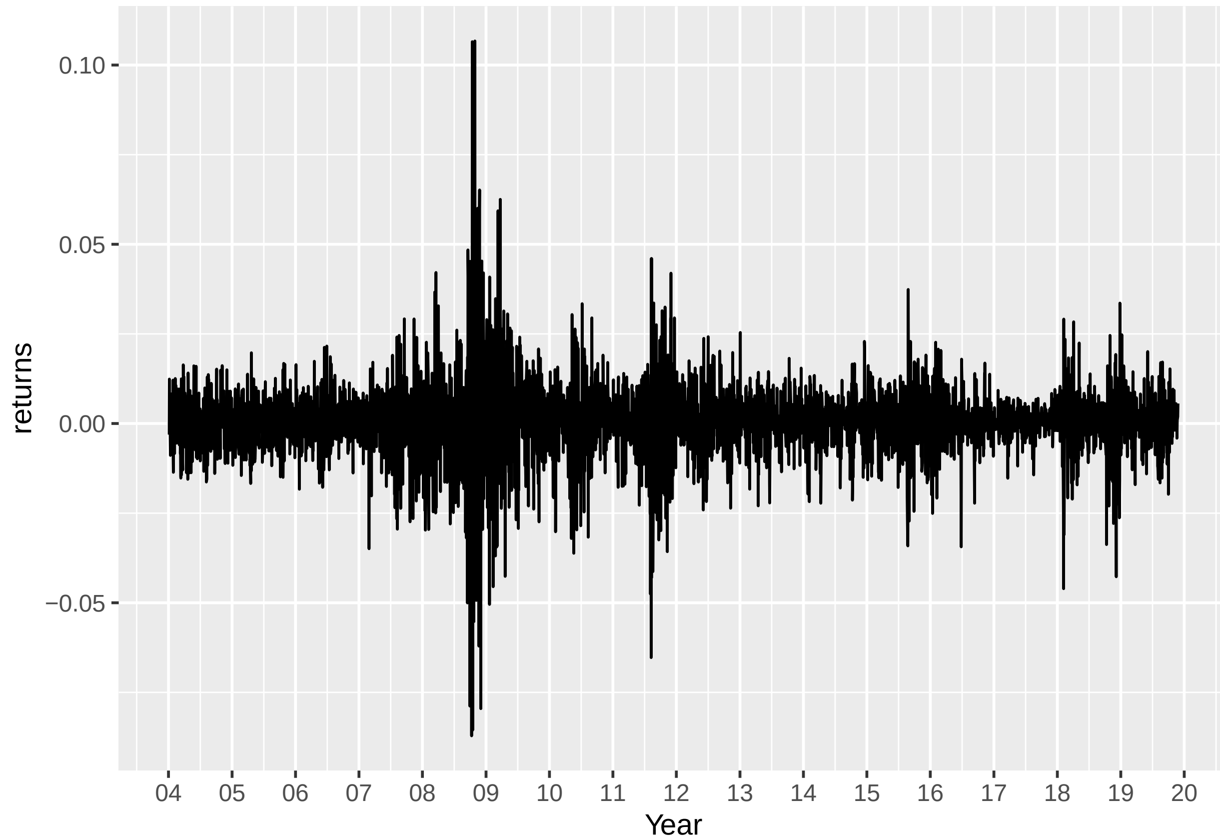


Returns of S&P500

```
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])
```

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

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




```r
k=15
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 55 with logLik: 13383.94
## converged at iteration 99 with logLik: 13568.32
## converged at iteration 342 with logLik: 13602.58
```

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

```
## [1] 10
```

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

```r
summary(hmmfit)
```

```
## Initial state probabilties model
##  pr1  pr2  pr3  pr4  pr5  pr6  pr7  pr8  pr9 pr10
##    1    0    0    0    0    0    0    0    0    0
##
## Transition matrix
##          toS1  toS2  toS3  toS4  toS5  toS6  toS7  toS8  toS9 toS10
## fromS1  0.456 0.000 0.033 0.114 0.000 0.164 0.000 0.000 0.232 0.000
## fromS2  0.000 0.966 0.000 0.000 0.000 0.000 0.000 0.006 0.000 0.028
## fromS3  0.000 0.057 0.089 0.000 0.000 0.000 0.000 0.003 0.000 0.851
## fromS4  0.808 0.000 0.000 0.005 0.118 0.069 0.000 0.000 0.000 0.000
## fromS5  0.005 0.000 0.000 0.000 0.622 0.058 0.315 0.000 0.000 0.000
## fromS6  0.000 0.000 0.000 0.000 0.711 0.073 0.098 0.000 0.118 0.000
## fromS7  0.048 0.000 0.000 0.000 0.272 0.373 0.147 0.000 0.159 0.000
## fromS8  0.000 0.036 0.000 0.000 0.000 0.000 0.000 0.964 0.000 0.000
## fromS9  0.709 0.000 0.215 0.000 0.000 0.015 0.000 0.000 0.060 0.000
## fromS10 0.000 0.000 0.267 0.158 0.000 0.000 0.000 0.000 0.000 0.575
##
## Response parameters
## Resp 1 : gaussian
##      Re1.(Intercept) Re1.sd
## St1            0.001  0.004
## St2            0.000  0.017
## St3           -0.013  0.008
## St4            0.013  0.004
## St5            0.001  0.003
## St6            0.007  0.003
## St7           -0.004  0.004
## St8           -0.003  0.039
## St9           -0.007  0.004
## St10           0.003  0.008
```

```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 = date_breaks("years"),
  labels = date_format('%y')) + xlab('Year') + scale_color_brewer(palette="Set3")
```

The lowest AIC is obtained at 10 regimes with AIC -2.7251731×10^4. Looking at the response parameters from the summary, the most volatil bear state appears almost only during the subprime crisis. It shows the market was in an exceptional state.

IWM

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" )
```

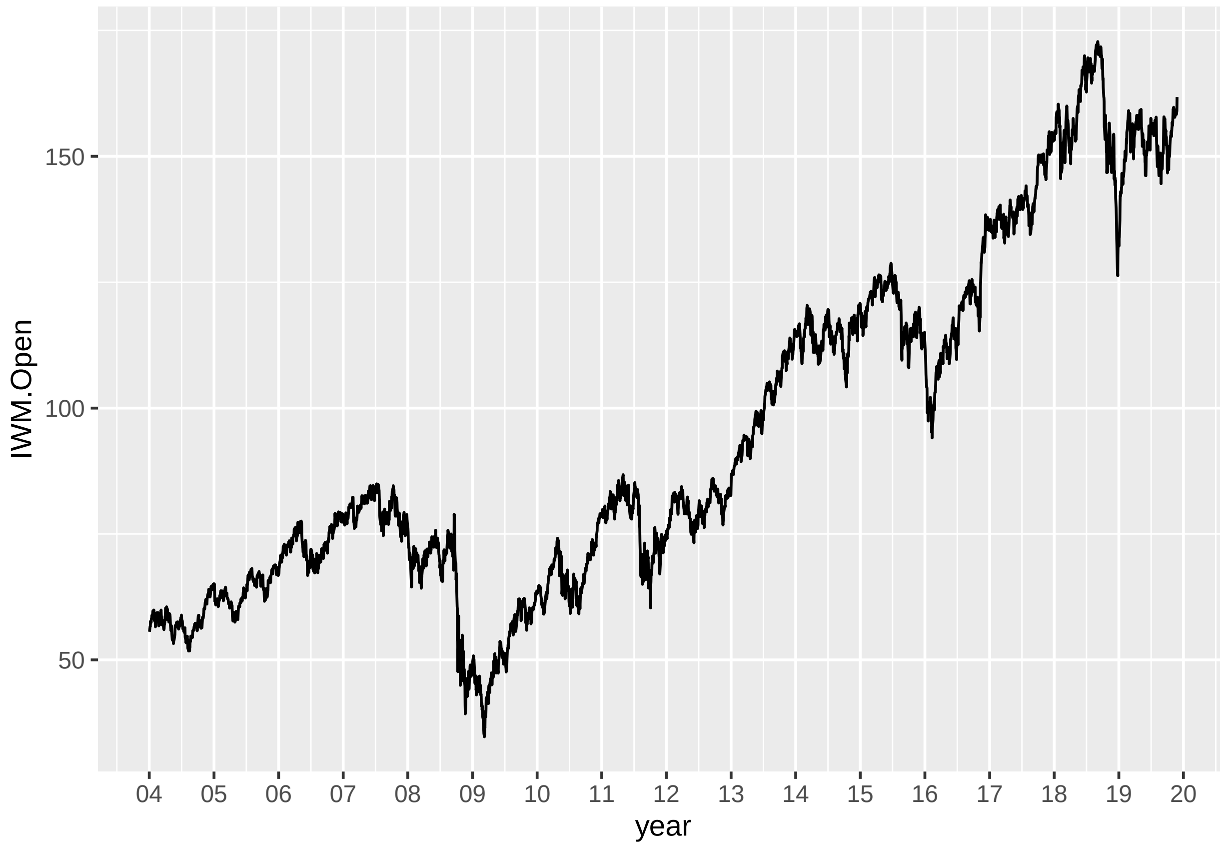
```
## [1] "IWM"
```

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

There are 4004 opening values of the index.

Opening values of IWM

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

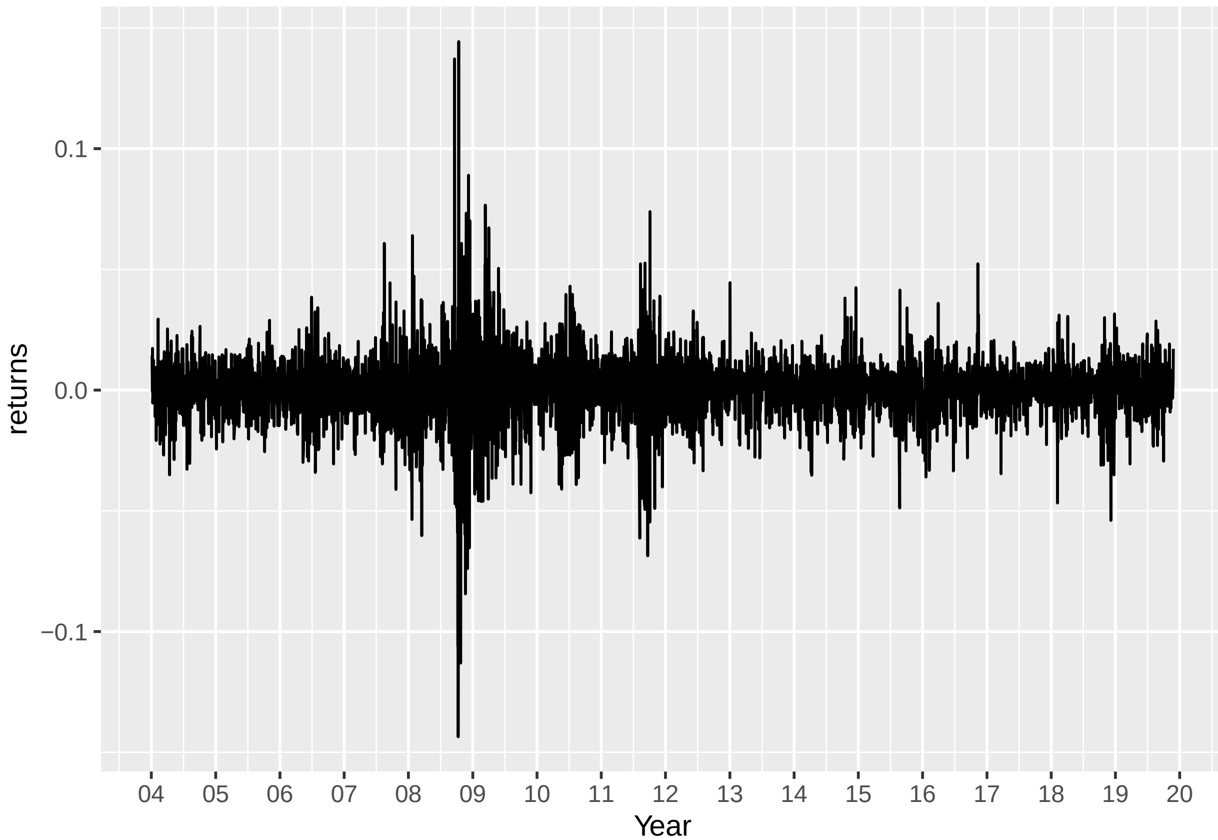


Returns of IWM

```
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("years"),
  labels = date_format('%y')) + xlab('Year')
```




```r
k=15
AICs2 = 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)
  AICs2[i-1] = AIC(hmmfit)
}
```

```
## converged at iteration 41 with logLik: 11961.62
## converged at iteration 129 with logLik: 12085
```

```r
bestk2 = which.min(AICs2) +1
bestk2
```

```
## [1] 6
```

```r
hmm <- depmix(returns ~ 1, family = gaussian(), nstates = bestk2, data=data.frame(iwm.returns))
hmmfit <- fit(hmm, verbose = FALSE)
```

```r
summary(hmmfit)
```

```
## Initial state probabilties model
## pr1 pr2 pr3 pr4 pr5 pr6
##   0   0   1   0   0   0
##
## Transition matrix
##         toS1  toS2  toS3  toS4  toS5  toS6
## fromS1 0.336 0.000 0.000 0.000 0.050 0.614
## fromS2 0.067 0.607 0.327 0.000 0.000 0.000
```

```
## fromS3 0.123 0.841 0.036 0.000 0.000 0.000
## fromS4 0.000 0.000 0.000 0.974 0.026 0.000
## fromS5 0.000 0.000 0.000 0.005 0.967 0.028
## fromS6 0.346 0.153 0.046 0.000 0.000 0.455
##
## Response parameters
## Resp 1 : gaussian
##     Re1.(Intercept) Re1.sd
## St1          -0.009  0.007
## St2           0.002  0.005
## St3           0.001  0.011
## St4          -0.002  0.040
## St5           0.000  0.017
## St6           0.007  0.007
```

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

```
ggplot(dat, aes(x=date, y=ret,color=factor(state),group = 1)) +
    geom_line() + scale_x_date(breaks = date_breaks("years"),
  labels = date_format('%y')) + xlab('Year') + scale_color_brewer(palette="Set3")
```

The lowest AIC is obtained at 6 regimes with AIC -2.4181211×10^4. Like with the S&P index, the most volatil bear state appears during the crisis, and in the end of 2011 shortly.
However, the optimal number of regimes has diminished from 10 to 6; the IWN is less heterogeneous in nature than the S&P500.

Conclusion

We have looked at an extension of Hidden Markov Models, called Markov Switching Models. We have first done a simulation study. The model recovered well the periods of bear and bull markets.
We have achieved Markov Switching Model estimation on financial data. It retrieved a special regime for the crisis characterized by high volatility.

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.