Stochastic Processes Project

Daniel Snyder and Yibei Chen

5/4/2017

Methods

Here we use a Hidden Markov model to make predictions of stock prices given historical data. Under this model, the unobserved process is a Markov chain whose states represent states of the stock market (Bull market and Bear market). Let X_i denote the unobserved Markov Process with discrete state space. Let Y_i denote the observed, Markov dependent process. Let Y_i be the log of price ratio from open to close of a given stock symbol. Assume a normal distribution of Y_i given X_i. Symbolically, $Y_i|X_i \sim N(\mu, \sigma^2)$. The observed process, Y_i depends on the unknown market state, X_i.

Model Parameters

Firstly, the initial state of the Markov chain has some inital distribution: $X_0 \sim \pi_0$. This is a distribution of probability to all possible market states at time zero. The number of states of the Markov chain is a parameter that may affect the predictive accuracy of the model.

The Markov chain has some time-homogeneous transition matrix, A, which dictates the probability of transition from one unknown market state to another. The $(i,j)^{th}$ entry is defined to be:

$$A(i,j) = \mathbb{P}[X_{t+1} = j | X_t = i] B(i,j) = \mathbb{P}[Y_t = j | X_t = i] \tag{1}$$

Finally, the model has parameters dictating the distribution of the daily price change Y_i conditional upon the unknown market state X_i. Since the Y_i are assumed to be Gaussian the parameterization will be mean and standard deviation conditional upon market state.[**?**]

$$\mathbb{P}(Y_t = y_t | X_t = x_t, \theta) \sim N(\mu_X, \sigma^2_{x_t}) \tag{2}$$

where

$$\theta := \{\pi_i, A(i,j), \mu_i, \sigma_i\} \tag{3}$$

Now the problem reduces to find the maxima of the posterior likelihood.[**?**]

$$\max_{\theta} \mathbb{P}\{Y_0 = y_0, \cdots, Y_N = y_N\} \tag{4}$$

$$
\begin{aligned}
L_c(\theta) =& \mathbb{P}(Y_0 = y_0, \cdots, Y_N = y_N, X_0 = y_0, \cdots, X_N = y_N|\theta) \\
=& \mathbb{P}(X_0 = x_0|\theta)\mathbb{P}(Y_0 = y_0|X_0 = x_0; \theta), \cdots, \\
& \mathbb{P}(X_N = x_N|X_{N-1} = x_{N-1}\theta)\mathbb{P}(Y_N = y_N|X_N = x_N; \theta) \\
=& \mathbb{P}(X_0 = x_0|\theta)\mathbb{P}(Y_0 = y_0|X_0 = x_0; \theta) \\
& \prod_{t=1}^{N} \mathbb{P}(X_t = x_t|X_{t-1} = x_{t-1}\theta)\mathbb{P}(Y_t = y_t|X_t = x_t; \theta) \\
\log L_c(\theta) =& \log \pi_{x_0} + \sum_{t+1}^{N} \log p_{x_t x_{t-1}} + \sum (-\frac{1}{2} \log 2\pi\sigma_{x_t}^2 - \frac{(y_t - \mu_{x_t})^2}{2\sigma_{x_t}^2}) \qquad (5) \\
E[\log L_c(\theta)|Y_1, \cdots, Y_N; \tilde{\theta}] =& \sum_i \mathbb{P}(X_{-0} = i|Y_0, \cdots, Y_N; \tilde{\theta}) \log \pi_i \\
& + \sum_{i,j} \sum_{t=1}^{N} \mathbb{P}(X_t = j, X_{t-1=i}|Y_0, \cdots, Y_N; \tilde{\theta}) \log A(i,j) \\
& + \sum_{i,j} \sum_{t=0}^{N} \mathbb{P}(X_t = i|Y_1, \cdots, Y_N; \tilde{\theta})(-\frac{1}{2} \log 2\pi\sigma_i^2 - \frac{(y_t - \mu_i)^2}{2\sigma_i^2})
\end{aligned}
$$

Expectation Maximimization

To estimate the maximally likely parameters, we will use the Baum Welch algorithm which is a form of Expectation Maximization [1].

Baum Welch Algorithm

Let

$$
\begin{aligned}
\alpha_t(j) &= \mathbb{P}(Y_0 = y_0, \cdots, Y_t = y_t, X_t = j) \\
\beta_t(j) &= \mathbb{P}(Y_{t+1} = y_{t+1}, \cdots, Y_N = y_N|X_t = j)
\end{aligned} \qquad (6)
$$

Their value can be obtained recursively.

$$
\begin{aligned}
\alpha_{t+1}(j) =& \mathbb{P}(X_{t+1} = j, Y_0 = y + 0, \cdots, Y_{t+1} = y_{t+1}) \\
=& \sum_i \mathbb{P}(X_{t+1} = j, X_t = i, Y_0 = y + 0, \cdots, Y_{t+1} = l) \\
=& \sum_i \mathbb{P}(Y_{t+1} = l)|X_{t+1} = j, X_t = i, Y_0 = y + 0, \cdots, Y_t = y_t) \\
& \mathbb{P}(X_{t+1} = j|X_t = i, Y_0 = y + 0, \cdots, Y_t = y_t)\mathbb{P}(Y_0 = y_0, \cdots, Y_t = y_t, X_t = j) \\
=& \sum_i B(j,l)A(i,j)\alpha_t(i) \qquad (7) \\
\beta_t(j) =& \mathbb{P}(Y_{t+1} = l, \cdots, Y_N = y_N|X_t = j) \\
=& \sum_k \mathbb{P}(X_{t+1} = k, Y_{t+1} = l, \cdots, Y_N =_N |X_t = j) \\
=& \sum_k \mathbb{P}(Y_{t+2} = y_{t+2}, \cdots, Y_N = y_N|X_{t+1} = k) \\
& \mathbb{P}(Y_{t+1} = l|X_{t+1} = k)\mathbb{P}(X_{t+1} = k|X_t = j) \\
=& \beta_{t+1}(k)B(k,l)A(j,k)
\end{aligned}
$$

where,

$$\begin{aligned} A(i,j) &= \mathbb{P}[X_{t+1} = j | X_t = i] \\ B(i,j) &= \mathbb{P}[Y_t = j | X_t = i] \\ \pi_0(i) &= \mathbb{P}[X_0 = i] \end{aligned} \tag{8}$$

Then,

$$\alpha_t(j)\beta_t(j) = \mathbb{P}(X_t = j, Y_0 = y_0, \ldots, Y_N = y_N) \tag{9}$$

So, [**?**]

$$d_t(j) = \mathbb{P}(X_t = j | Y_0 = y_0, \cdots, Y_N = y_N) = \frac{\alpha_t(j)\beta_t(j)}{\sum_j \alpha_t(j)\beta_t(j)} \tag{10}$$

$$\begin{aligned} \alpha_t(i)A(i,y_{t+1})B(j,y_{t+1})\beta_{t+1}(j) =& \mathbb{P}(X_t = i, Y_0 = y_0, \ldots, Y_N = y_N) \\ & \mathbb{P}(X_{t+1} = j | X_t = i)\mathbb{P}(Y_{t+1} = y_{t+1} | X_{t+1} = j) \\ & \mathbb{P}(Y_{t+2} = y_{t+2}, \cdots, Y_N = y_N | X_{t+1} = j) \\ =& \mathbb{P}(X_{t+1} = j, X_t = i, Y_0 = y_0, \ldots, Y_N = y_N) \end{aligned} \tag{11}$$

Therefore,[**?**]

$$\begin{aligned} e_t(i,j) =& \mathbb{P}(X_{t+1} = j, X_t = i | Y_0 = y_0, \ldots, Y_N = y_N) \\ =& \frac{\alpha_t(i)A(i,y_{t+1})B(j,y_{t+1})\beta_{t+1}(j)}{\sum_{i,j} \alpha_t(i)A(i,y_{t+1})B(j,y_{t+1})\beta_{t+1}(j)} \end{aligned} \tag{12}$$

Now we can find the maxima by the steps below.[**?**]

$$\begin{aligned} & \max_\theta E[\log L_c(\theta) | Y_1, \cdots, Y_N; \tilde{\theta}] \\ =& \max_\theta \sum_i d_0(i) \log \pi_i + \sum_{i,j,k} \sum_{t=1}^{N} e_{t-1}(i,j,k) \log Ai, j^k + \sum_i \sum_{t=0}^{N} (d_t(i) - \frac{(y_t - \mu_i)^2}{2\sigma_i^2}) \\ & s.t. \sum_i \pi_i = 1, \sum_j A(i,j) = 1 \end{aligned} \tag{13}$$

We can get,[**?**]

$$\pi_i^* = d_0(i), A(i,j)^* = \frac{\sum_{t=0}^{N-1} e_t(i,j,k)}{\sum_j \sum_{t=0}^{N-1} e_t(i,j,k)}, \mu_i^* = \frac{\sum_{t=0}^{N} d_t(i) y_t}{\sum_{t=0}^{N} d_t(i)}, \sigma_i^* = \frac{\sum_{t=0}^{N} d_t(i)(y_t - \mu_i^*)^2}{\sum_{t=0}^{N} d_t(i)} \tag{14}$$

The algorithm is implemented using the HiddenMarkov package for R [2].

Prediction of Next Day's Return

Here we will predict the next day's stock return (log price ratio) using the Hidden Markov Model. The EM procedure produced estimates of which state the hidden process is in at the all times considered. We will predict the next day's return as the exepecation of the random variable Y_{t+1}. Using the assumed Markov property, we may calculate the marginal distribution of tomorrow's hidden state.

$$\pi_{t+1} = \pi_t \Pi \tag{15}$$

Given this distrbution of tomorrow's hidden state, we may calculate the expectation of tomorrow's daily return as the inner product of the hidden state distribution and the vector of Markov dependent means.

$$\mathbb{E}[Y_{t+1}] = \sum_{i \in S_X} \mathbb{E}[Y_{t+1} | X_{t+1} = i] \cdot \mathbb{P}[X_{t+1} = i] \tag{16}$$

Results

The software package chosen to perform the task of predicting stock returns is the R package *HiddenMarkov*. While some texts refer to R package *depmixS4*, this package was chosen because it is newer and because of its syntactic niceness. The package has capbilites for Markov modulated generalized linear models although these capbilities were not tested during the course of this project.

Code

The following are pre processing functions used to import stock market historical quote data. This data was retrieved from the historical quote tool available at NASDAQ.com [3]. The pre processing function here calculates the log price ratio for each day and sorts by date ascending.

```
library(HiddenMarkov)
library(expm)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'expm'
```

```
## The following object is masked from 'package:Matrix':
##
##     expm
```

```
pre.process <- function(f.name){
  # Read a csv file of stock price data
  #   having columns: "open", "close", "date"
  # param f.name: string denoting file name to load
  x <- read.csv(f.name)
  x <- x[order(as.POSIXct(x$date)),] # Sort by date
  x$logdiff <- log(x$close) - log(x$open)
  return(x)
}
```

The Baum Welch algorithm, available in *HiddenMarkov*, uses EM algorithm to estimate most likely parameters for the HMM given the data.

```
bwelch <- function(hmm){
  # Perform Baum Welch algorithm
  #   for MLE of model parameters
  # param hmm: dthmm {HiddenMarkov} object
  hmm <- BaumWelch(hmm, control=bwcontrol(prt=F))
  return(hmm)
}
```

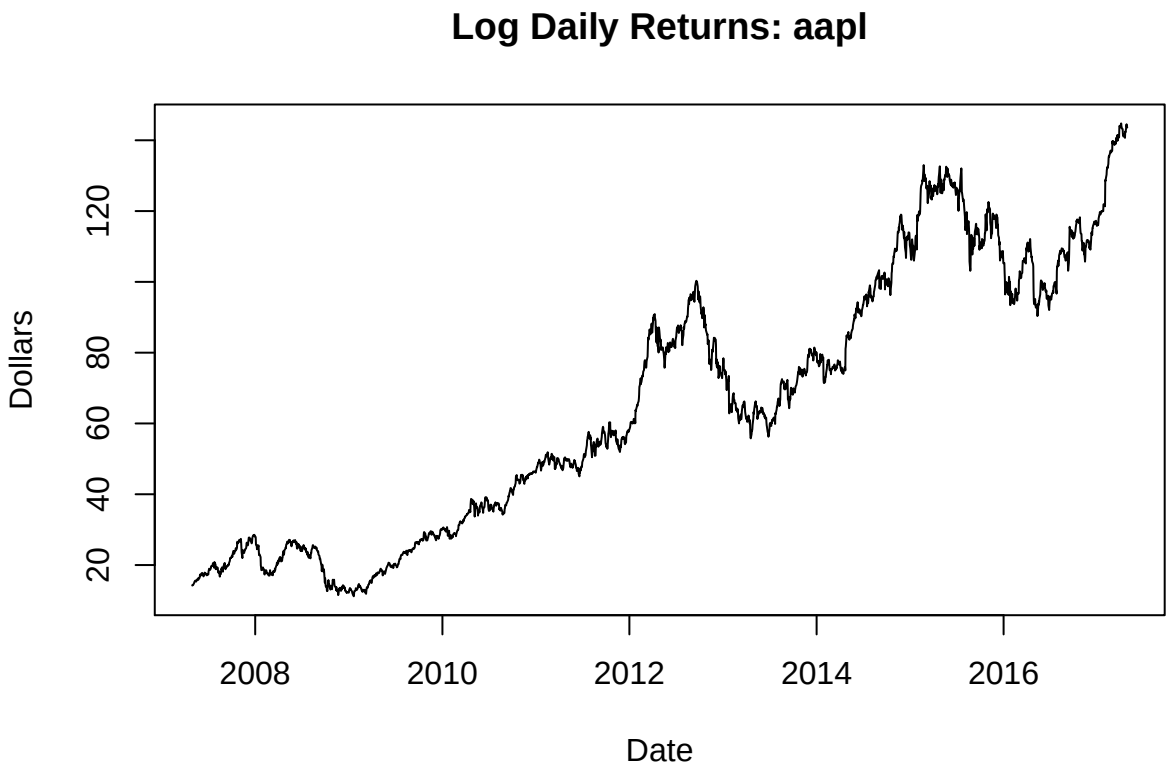
To instantiate the dthmm (discrete time hidden markov model) class, provide a time series of data and starting points for the model parameters.

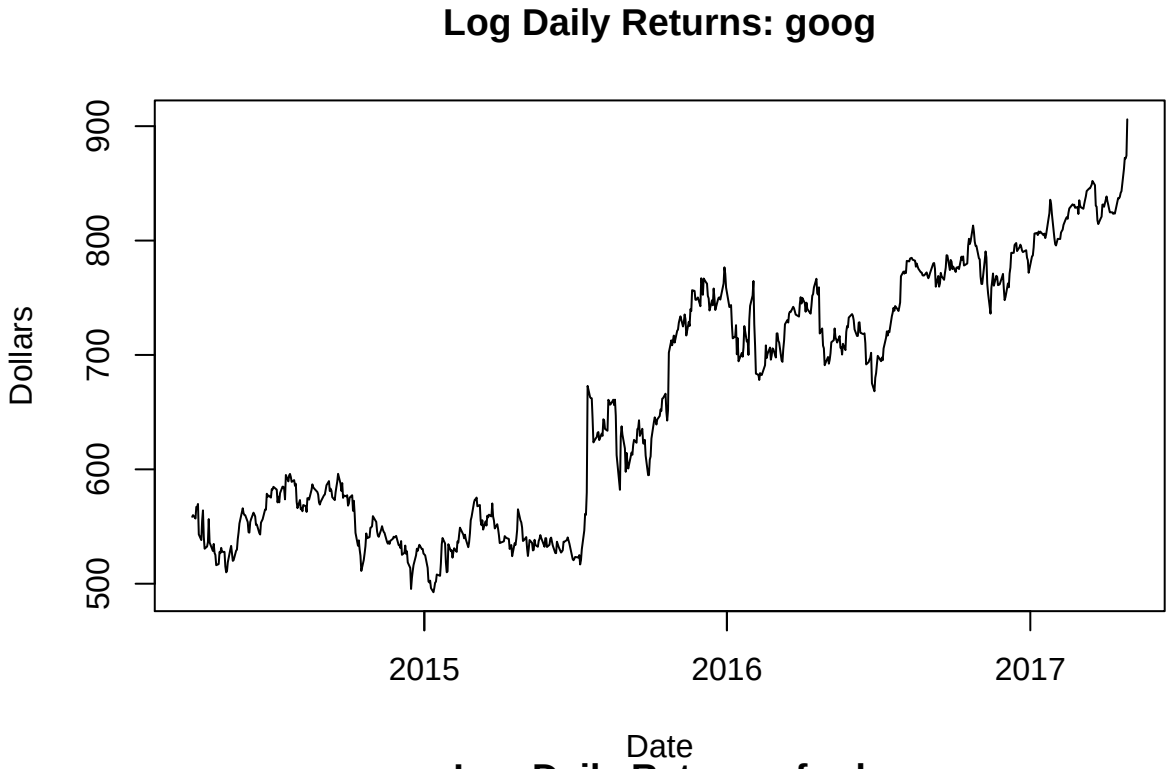
```
get.hmm <- function(x, Tr=nrow(x), s=2){
  # Instantiate HMM object
  # param x: data frame with column 'logdiff'
  # param Tr: Truncation point
  #   (index of last observation included in training set)
  # params s: number of hidden Markov States
  Tr <- min(Tr, nrow(x))
  v <- x[1:Tr, "logdiff"]
  k <- 30.
  p <- 1 / k; q <- 1 - p
  A <- matrix(q, s, s)  # Initial guess at Markov Transition Matrix
  diag(A) <- p
  xi <- matrix(1 / s, 1, s)
  m1 <- mean(v[v < 0]); m2 <- mean(v[v >= 0]);
  s1 <- sd(v[v < 0]); s2 <- sd(v[v >= 0]);
  means <- c(m1,m2)
  stds <- c(s1,s2)
  cond.dist <- list(mean = means,
                    sd = stds)
  hmm <- dthmm(x = v, Pi = A, delta = xi,
               distn = 'norm', pm = cond.dist)
  hmm <- bwelch(hmm)
  return(hmm)
}

update.hmm <- function(hmm, xnew){
  # Update the Hidden Markov model with new observation.
  # Re-solve for parameters using Baum-Welch.
  # param hmm: dthmm {HiddenMarkov} object
  # param xnew: float. New observation to append
  hmm$x <- c(hmm$x, xnew)
  hmm <- bwelch(hmm)
  return(hmm)
}


forecast.hmm <- function(hmm, horizon=1){
  # Forecast tomorrow's return (log dollars)
  # param hmm: dthmm{HiddenMarkov} object
  # param horizon: int. how many steps ahead to forecast
  A <- hmm$Pi  # MLE of Markov Chain Transition Matrix
```

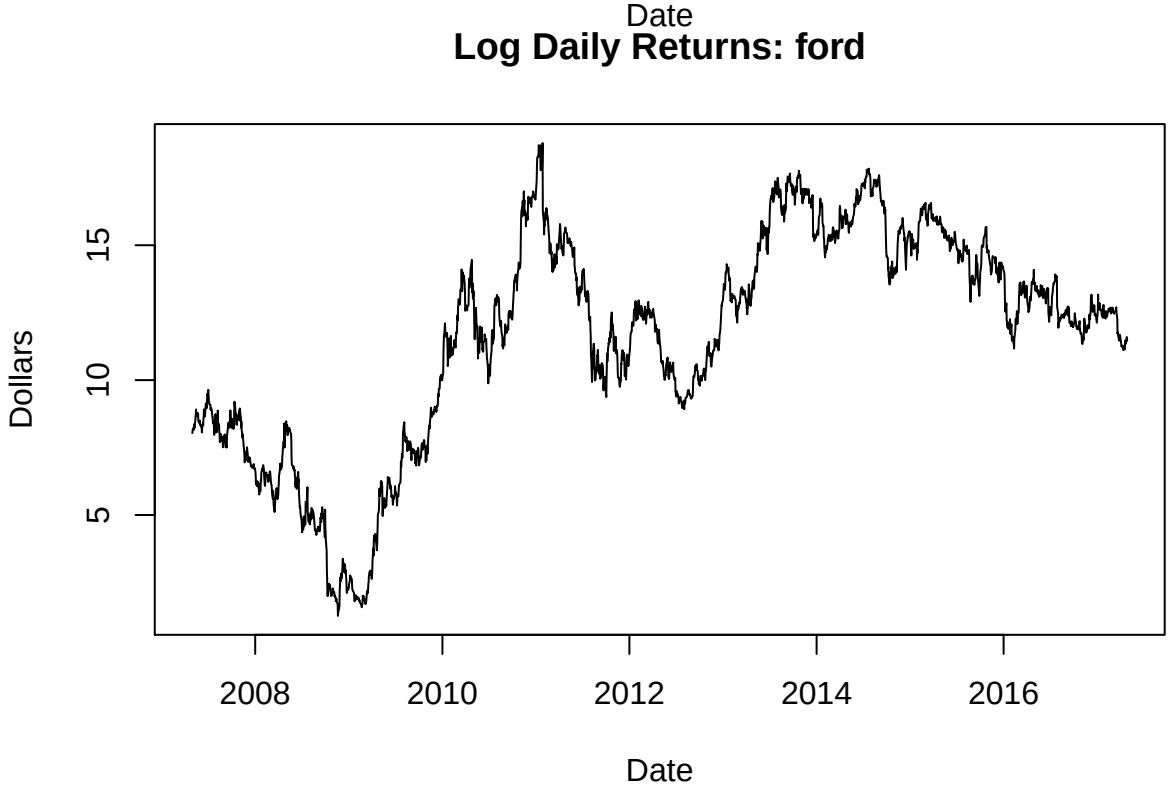
```
  Eyx <- hmm$pm$mean  # Expectation of Y given X
  t <- length(hmm$x)  # Series length
  px.T0 <- hmm$u[t,]  # Estimated distribution of Xt
  Ah <- A %^% horizon    # Exponentiate MC transition matrix
  px.Th <- px.T0 %*% Ah  # Estimated distribution of X(t+h)
  Ey.Th <- px.Th %*% Eyx  # Expectation of Y(t+h)
  return(Ey.Th)
}


plot.conditional.dist <- function(hmm){
  # Plot the conditional distributions
  #   of Y given hidden state X
  # param hmm: dthmm {HiddenMarkov} object
  m1 <- hmm$pm$mean[1]
  m2 <- hmm$pm$mean[2]
  s1 <- hmm$pm$sd[1]
  s2 <- hmm$pm$sd[2]
  xlim <- c(min(m1 - 3  * s1,
                m2 - 3  * s2),
            max(m1 + 3  * s1,
                m2 + 3  * m2))
  x <- seq(xlim[1],xlim[2],.001)
  y1 <- dnorm(x, m1, s1)
  y2 <- dnorm(x, m2, s2)
  ymax <- max(max(y1), max(y2))
  plot(x, y1, type="l", col="red",
       ylim=c(0,ymax), main="Conditional Distribution: P[Y|X]",
       ylab="Probability Density", xlab="y")
  abline(v = m1, col='red')
  lines(x, y2, col="blue")
  abline(v = m2, col="blue")
  legend(xlim[1], ymax / 2, c('X=1', 'X=2'), lty=c(1,1),
         lwd=c(2,2), col=c('red','blue'))
}


stock.timeseries.plot <- function(x){
  # Plot the stock's value over time as
  #   well as its daily log returns
  # param x: list of data frames with columns "logdiff" and "close"
  # param symb: stock ticker symbol
  sym <- names(x)
  for(name in sym){
    plot(close ~ as.POSIXct(date), x[[name]], type="l",
         xlab="Date", ylab="Dollars",
         main=paste("Log Daily Returns:", name))
  }
}
```

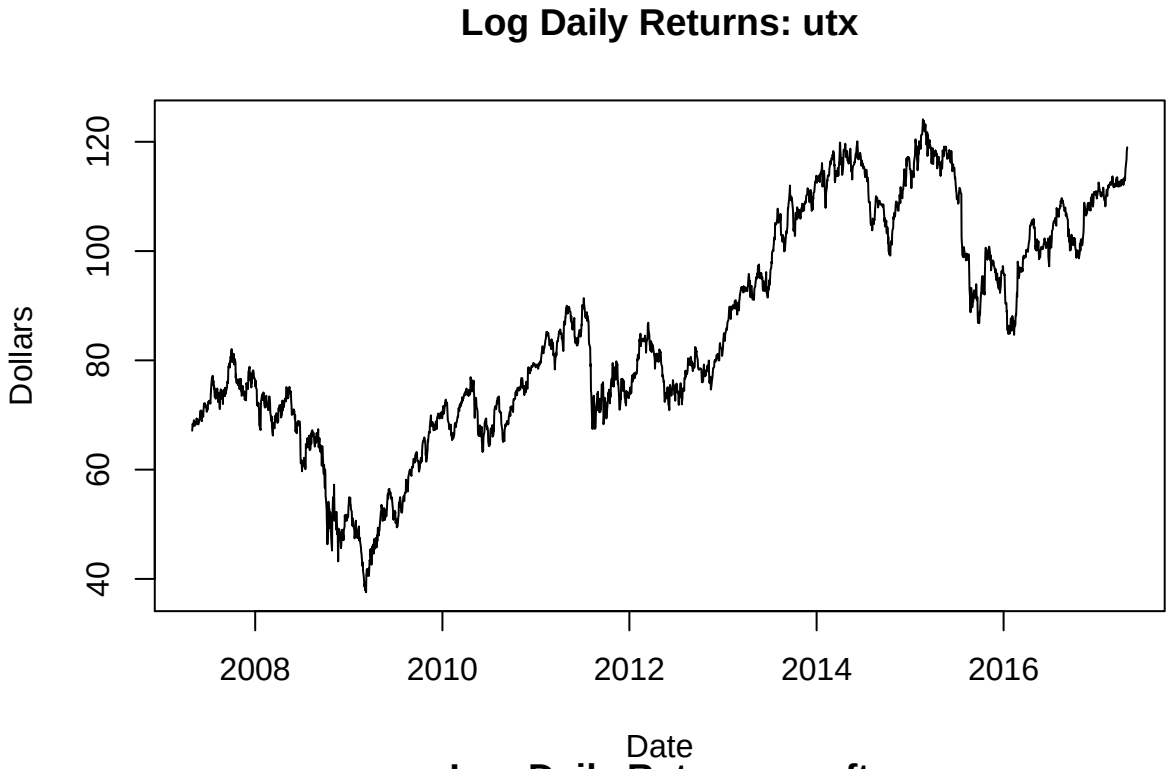
To demonstrate the concept, here we will instantiate a HMM object and train it on a time series of data. Then we will show the estimates for the conditional distribution of Y_i given state of X_i. On the density plots, note how the two states different in both mean and variance. One state is associated with high volatililty (high variance in log returns) and one state shows lower volatility (lower variance in log returns).

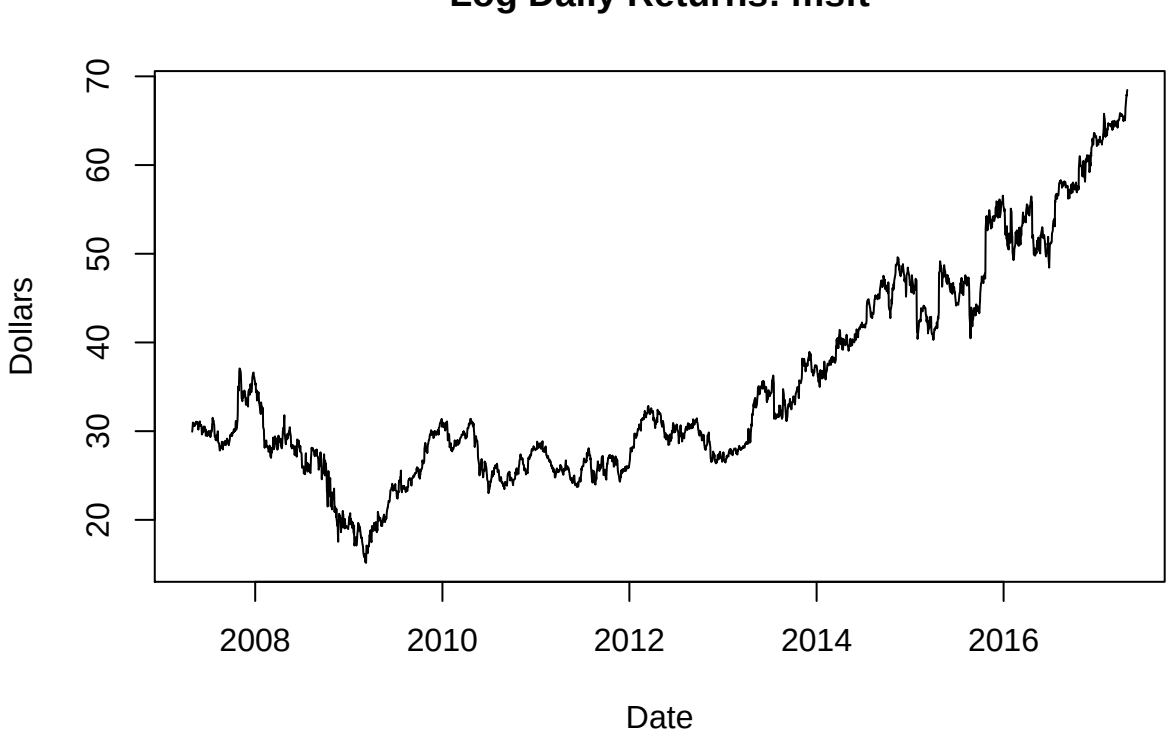
```
# Read stock data (date, open, close, volume)
aapl <- pre.process("./stock_data/aapl_decade.csv")
goog <- pre.process("./stock_data/goog_decade.csv")
ford <- pre.process("./stock_data/ford_decade.csv")
utx <- pre.process("./stock_data/utx_decade.csv")
msft <- pre.process("./stock_data/msft_decade.csv")
sbux <- pre.process("./stock_data/sbux_decade.csv")

# Examine the data as a plot of price and of log returns
stocklist <- list(aapl=aapl, goog=goog, ford=ford,
                  utx=utx, msft=msft, sbux=sbux)
stock.timeseries.plot(stocklist)
```

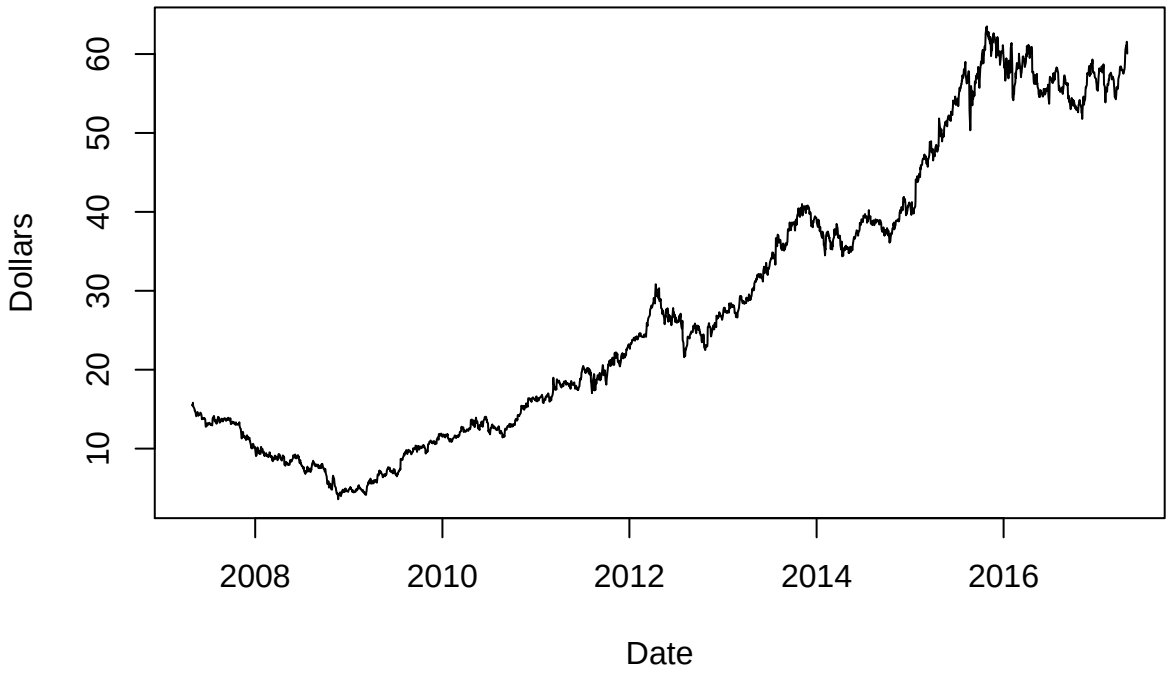



Log Daily Returns: goog
Dollars
500
600
700
800
900
2015
2016
2017
Date


Log Daily Returns: ford
Dollars
5
10
15
2008
2010
2012
2014
2016
Date


Log Daily Returns: utx
Dollars
40
60
80
100
120
2008
2010
2012
2014
2016
Date


Log Daily Returns: msft
Dollars
20
30
40
50
60
70
2008
2010
2012
2014
2016
Date



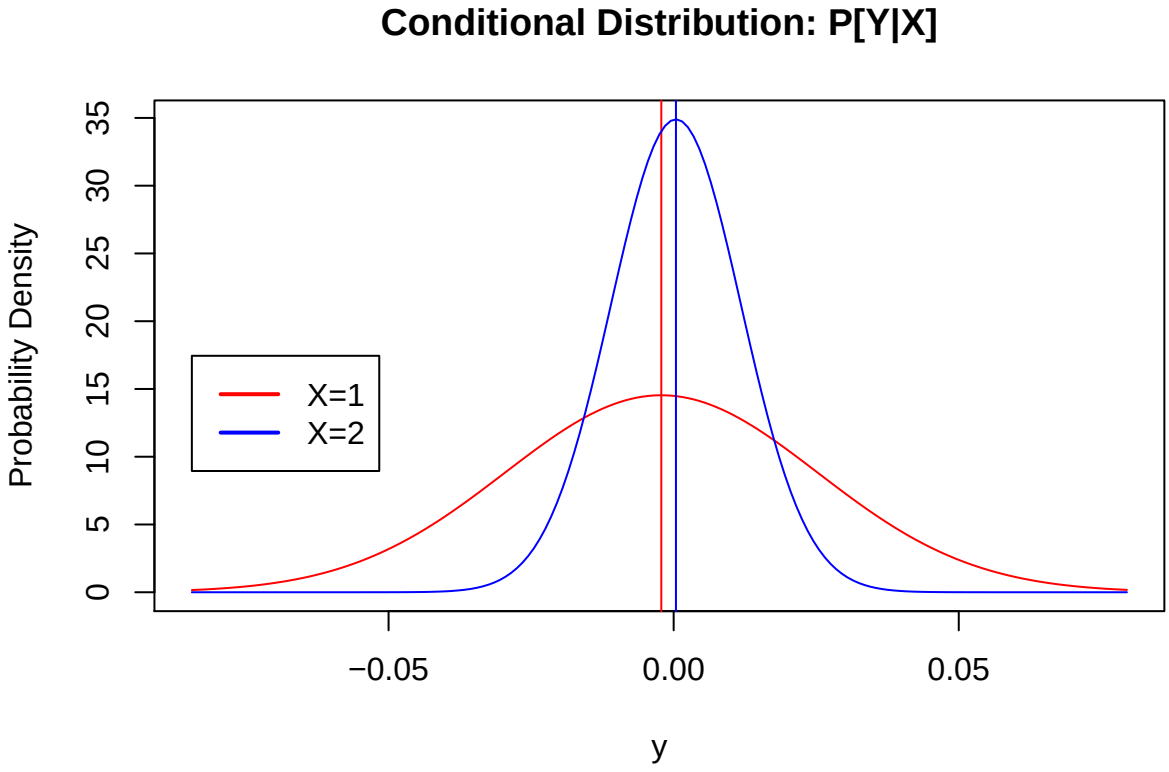

```
# Fit the HMM
hmm <- get.hmm(aapl)

# List the estimated conditional means
hmm$pm$mean
```

```
## [1] -0.0021680570  0.0004014585
```

```
# List the estimated conditional standard deviations
hmm$pm$sd
```

```
## [1] 0.02744515 0.01143210
```

```
# Plot conditional distribution of Y given X (EM estimated parameters)
# The model posits that these are (most likely) the distributions
# that Y (log returns) come from given the underlying state of the market.
plot.conditional.dist(hmm)
```



References

[1] Zucchini, Walter. Iain L. MacDonald, Ronald Langrock. *Hidden Markov Models for Time Series* An Introduction Using R. CRC Press. Boca Raton, FL. 2016

[2] Harte, David. *HiddenMarkov: Hidden Markov Models*. R package version 1.8-7. Statistics Research Associates. Wellington. ftp://ftp.gns.cri.nz/pub/davidh/sslib/r-repo/. 2016

[3] NASDAQ historical stock quote tool.http://www.nasdaq.com/quotes/historical_quotes.aspx