Volatility modeling and forecasting

Sebastiano Manzan

ECO 4051 | Spring 2018
Introduction

- The Figure below shows the *returns* and *absolute returns* starting in January 1980 at the daily frequency for:
  - S&P 500 Index returns (left)
  - The JPY/USD exchange rate (right)
The ACF of the returns and absolute returns up to lag 50 (left: S&P; right: JPY/USD)
The *CBOE Volatility Index* (VIX) is a measure of market uncertainty that is constructed from implied volatilities of S&P 500 Index options. It is widely used as a proxy for market volatility.

```r
VIX <- getSymbols("~VIX", from="1990-01-01", auto.assign=FALSE) %>% Cl
p0 <- autoplot(VIX, ts.colour="tomato1") + theme_bw() + labs(x=NULL, y=NULL, title = "VIX")
```

How can we construct a measure of volatility (similar to VIX) for any asset we are interested in?
- The AR(1) assumes that $R_{t+1} = \beta_0 + \beta_1 R_{t} + \epsilon_{t+1}$, where:
  - $E_t(R_{t+1}) = \beta_0 + \beta_1 R_{t}$
  - $Var_t(R_{t+1}) = \sigma^2$

- The $E_t(\cdot)$ and $Var_t(\cdot)$ represent the expected return of $R_{t+1}$ conditional on the information available at time $t$

- The evidence in the previous graph indicates that this might not be a realistic assumption because returns are heteroskedastic (i.e., the variance and the standard deviation change over time)

- In other words, we need $Var(\epsilon_t)$ to be $\sigma^2_t$ instead of $\sigma^2$

- Why should we care about modeling the time variation of the standard deviation $\sigma^2_t$? What is the usefulness of forecasting volatility instead of or in addition to forecasting returns? Numerous applications in finance:
  - pricing derivatives
  - measuring risk (Value-at-Risk)
  - portfolio allocation
A volatility model

- We assume that returns follow the model

\[ R_{t+1} = \mu_{t+1} + \eta_{t+1} \]

that decomposes the return in two components:

- **Expected return** \( \mu_{t+1} \): the predictable component of the return; it can be assumed equal to 0, equal to a constant \( \mu \), or following an AR(1) process \( \mu_{t+1} = \phi_0 + \phi_1 \times R_t \)

- **Shock** \( \eta_{t+1} \): the unpredictable component of the return; we can assume that it is equal to \( \eta_{t+1} = \sigma_{t+1} \epsilon_{t+1} \) where:
  - **Volatility**: \( \sigma_{t+1} \) is the standard deviation of the error at time \( t \) which we assume varies over time; for example, \( \sigma_{t+1} = \omega + \alpha R^2_t \)
  - **Standardized shock**: \( \epsilon_{t+1} \) represents an unpredictable error term or a shock with mean zero and variance 1 (additionally, we can assume that it is normally distributed)
Both $\mu_{t+1}$ and $\sigma_{t+1}$ can be assumed to be functions of past values.

At the market closing in day $t$ I can then calculate the forecast of the return the following day, $\mu_{t+1}$, and the expected volatility the following day, $\sigma_{t+1}$.

How to model the volatility process $\sigma_t$? We consider three approaches:

- **Moving Average (MA)**
- **Exponential Moving Average (EMA)**
- **Generalized Auto-Regressive Conditional Heteroskedasticity (GARCH) model**

At high frequencies (daily or intra-daily) there is typically very little predictability in the mean but significant predictability in the conditional variance; it is thus reasonable to assume that $\mu_{t+1} = 0$ (of course, we can statistically test this hypothesis . . . )
Moving Average (MA)

- The **MA model** consists of averaging the square returns on a window of the latest $M$ days (where $M$ denotes the window size).

- The estimate of the variance in day $t$ is $\sigma_t^2$ and it is given by

$$
\sigma_{t+1}^2 = \frac{1}{M} \left( R_t^2 + R_{t-1}^2 + \cdots + R_{t-M+1}^2 \right) = \frac{1}{M} \sum_{j=1}^{M} R_{t-j+1}^2
$$

- Dependence on $M$:
  - $M = 1$: $\sigma_{t+1}^2 = R_t^2$
  - $M = t$: $\sigma_{t+1}^2 = \sigma^2$

- Using the squared returns in the summation relies on the assumption that $\mu_{t+1} = 0$ so that $R_{t+1} = \eta_{t+1}$

- If $\mu_{t+1} \neq 0$ then we apply MA on $\eta_{t+1} = R_{t+1} - \mu_{t+1}$ that is

$$
\sigma_{t+1}^2 = \left( \sum_{j=1}^{M} \eta_{t-j+1}^2 \right) / M
$$
The MA estimate of volatility can be interpreted as a weighted average of the square returns

\[ \sigma_{t+1}^2 = \sum_{j=1}^{t} w_j R_j^2 \]

where the weight \( w_j \) of day \( j \) takes the following form:

- \( w_j = 1/M \) if \((t - M + 1) \leq j \leq t\)
  
  (ex: for \( M = 25 \) \( w_j = 1/25 = 0.04 \) or 4%)

- \( w_j = 0 \) if \( j < (t - M + 1) \)

The MA approach can be implemented in R using the \texttt{rollmean()} function in package \texttt{zoo} which takes as arguments:

- the window size \( M \)
- if the estimate should be aligned to the left, center, or right of the window
GSPC <- getSymbols("^GSPC", from="1980-01-01", auto.assign=FALSE)
sp500daily <- 100 * ClCl(GSPC) %>% na.omit
names(sp500daily) <- "RET"
sigma25 <- zoo::rollmean(sp500daily^2, 25, align="right")
names(sigma25) <- "MA25"

ggplot(merge(sigma25, sp500daily)) + geom_line(aes(time(sp500daily), abs(RET)), color="gray80") + theme_bw() + geom_line(aes(index(sp500daily), MA25^0.5), color="orangered3") + labs(x=NULL, y=NULL)
The effect of increasing the window size $M$ is to provide a smoother (slowly changing) estimate of volatility $\sigma_{t+1}$ since each daily return receives a smaller weight.

Below a comparison of $M = 25$ and 100:

```r
sigma100 <- zoo::rollmean(sp500daily^2, 100, align="right")
names(sigma100) <- "MA100"
ggplot(merge(sigma25, sigma100)) + 
  geom_line(aes(time(sigma25), MA25^0.5), color="orangered3", size=0.3, linetype="dashed") + 
  theme_bw() + 
  geom_line(aes(index(sigma25), MA100^0.5), color="springgreen4", size=0.6) + 
  labs(x=NULL, y=NULL) + ylim(c(0, 6))
```
Pros/Cons of the MA approach

▶ The MA approach is very simple to calculate and implement which is convenient when dealing with a large number of assets

▶ A drawback of the approach is that it is sensitive to large returns: when a large return enters/exit the estimation window \( M \) the estimate \( \sigma_{t+1}^2 \) has a upward/downward jump

▶ How to choose \( M \)? Practitioners use \( M = 25 \) (one trading month) as an ad hoc value rather than being chosen optimally
An alternative approach is the EMA approach which solves the problem of the sensitivity to large returns.

EMA is calculated as follows:

\[
\sigma_{t+1}^2 = \lambda \cdot R_t^2 + \lambda(1 - \lambda) \cdot R_{t-1}^2 + \lambda(1 - \lambda)^2 \cdot R_{t-2}^2 + \ldots \\
= \lambda \sum_{j=0}^{\infty} (1 - \lambda)^j R_{t-j}^2
\]

where \( \lambda \) is a smoothing parameter (equivalent of \( M \) for MA).
Based on the formula for $\sigma_{t+1}^2$, its lagged value $\sigma_t^2$ is given by

$$\sigma_t^2 = \lambda \cdot R_{t-1}^2 + \lambda(1 - \lambda) \cdot R_{t-2}^2 + \ldots$$

This expression for $\sigma_{t-1}^2$ can be used to rewrite $\sigma_t^2$ as:

$$\sigma_{t+1}^2 = \lambda \cdot R_t^2 + \lambda(1 - \lambda) \cdot R_{t-1}^2 + \lambda(1 - \lambda)^2 \cdot R_{t-2}^2 + \ldots$$

$$= \lambda \cdot R_t^2 + (1 - \lambda) \cdot \left( \lambda R_{t-1}^2 + \lambda(1 - \lambda) R_{t-2}^2 + \ldots \right)$$

$$= \lambda \cdot R_t^2 + (1 - \lambda) \sigma_t^2$$

which shows that the current volatility estimate is given by a weighted average of the previous day estimate ($\sigma_t^2$) and the current day square return ($R_t^2$)

The weight function $w_j$ for EMA is $w_j = \lambda \cdot (1 - \lambda)^j$; compared to the MA weight function:

- It weighs all observations (not only the $M$ most recent)
- The weighs are (smoothly) declining the farther in the past
Can you match the color to the method?

- MA(25), MA(100), EMA(0.06), and EMA(0.04)
- darkolivegreen4, royalblue1, tomato4, and royalblue4
Practitioners have found that a value of $\lambda$ of 0.06 for EMA works well for a large set of assets.

The package TTR (part of quantmod) provides functions to calculate moving averages, both simple and of the exponential type as in code below:

```r
library(TTR)
# SMA() function for Simple Moving Average; n = number of days
ma25 <- SMA(sp500daily^2, n=25)
names(ma25) <- "MA25"
# EMA() function for Exponential Moving Average; ratio = lambda
ema06 <- EMA(sp500daily^2, ratio=0.06)
names(ema06) <- "EMA06"
autoplot(merge(ma25, ema06)^0.5, ncol=2) + theme_bw()
```
Comparison of the MA and EMA volatility estimators between 2008 and 2010 (gray lines are the absolute returns)

```r
temp <- merge(sp500daily, ma25, ema06) %>% window(.,
start="2008-01-01", end="2009-12-31")
ggplot(temp) + geom_point(aes(time(temp), abs(RET)), color="gray45", size=0.4) +
  geom_line(aes(time(temp), MA25^0.5), color="indianred1", size = 0.8) +
  geom_line(aes(time(temp), EMA06^0.5), color="seagreen4", size = 0.8) +
  theme_bw() + labs(x=NULL, y=NULL)
```
In day $t$ we estimate volatility based on the current and past returns using MA or EMA.

We then assume that variance next period $E_t(\sigma_{t+1}^2) = \sigma_t^2$ and, more generally, $E_t(\sigma_{t+k}^2) = \sigma_t^2$.

This assumption implies that the variance follows a random walk model (without drift) and that volatility is non-stationary.

How do you feel assuming that volatility is not mean-reverting??
The MA and EMA are simple tools that work in many situations of practical relevance, in particular forecasting volatility at short horizons.

However, they require to introduce some assumptions that might be questionable:

- ad hoc choice of the parameters ($M$ and $\lambda$) rather than being estimated
- volatility is non-stationary (random walk)

An alternative approach is represented by the ARCH model.
Assume for now that $\mu_{t+1} = 0$ so that $R_{t+1} = \eta_{t+1}$

The ARCH model assumes that the conditional variance $\sigma^2_{t+1}$ is a function of the current squared return, that is,

$$\sigma^2_{t+1} = \omega + \alpha * R^2_t$$

where $\omega$ and $\alpha$ are parameters to be estimated.

The model above is called ARCH(1) since only one lag is included, but it can be generalized to include $p$ lagged returns.

A more general specification is the Generalized ARCH (GARCH) model which is characterized by the following Equation for the conditional variance:

$$\sigma^2_{t+1} = \omega + \alpha * R^2_t + \beta * \sigma^2_t$$

where, the previous variance forecast $\sigma^2_t$ is included in addition to the current return $R^2_t$.

We typically refer to the previous model as ARCH(1) and GARCH(1,1) since we included one lag of $R^2_t$ and $\sigma^2_t$; this can be generalized to more lags.
We make a distinction between:

- $\sigma^2$: the *unconditional* variance (i.e. the long-run variance)
- $\sigma^2_{t+1}$: the *conditional* variance (i.e. based on information available at time $t$)

They measure the volatility that you expect by holding the asset for a long-period (e.g., 20 years) vs holding the asset for a short period (e.g., one day)

The unconditional mean of the conditional variance, $E(\sigma^2_{t+1}) = \sigma^2$, for the GARCH(1,1) is given by:

$$\sigma^2 = \frac{\omega}{1 - (\alpha + \beta)}$$

The unconditional variance of the returns is finite if $\alpha + \beta < 1$.

On the other hand, if $\alpha + \beta = 1$ the long-run variance is not finite and variance is non-stationary (as for EMA)
What does it mean that variance (or volatility) is non-stationary? Do we expect the variance to be mean reverting?

If volatility is mean-reverting it means that:

- $\sigma^2_{t+1}$ oscillates (more/less persistently) around its unconditional mean $\sigma^2$
- Periods of high volatility (higher than $\sigma^2$) will be followed by periods of low volatility (lower than $\sigma^2$)

This is consistent with the discussion in the first slide in which we observe returns alternating between periods of high and low volatility (volatility clustering)
On the other hand, non-stationary volatility means that we expect variance (or volatility) to stay at the current level forever.

This implication seems at odds with the observation that volatility alternates between periods of high and low volatility.

Although the forecasts assuming $\alpha + \beta = 1$ might be as accurate as those from models assuming $\alpha + \beta < 1$ in the short-run, they might differ significantly in the long-run.

When do we need accurate long-run forecast of volatility? Pricing long-dated options, investment allocation, etc.
GARCH vs EMA

If we set $\omega = 0$, $\alpha = \lambda$ and $\beta = 1 - \lambda$ in the GARCH conditional variance equation we obtain

$$\sigma_{t+1}^2 = \lambda R_t^2 + (1 - \lambda)\sigma_t^2$$

which is the conditional variance for EMA.

So EMA is a restricted version of a GARCH model in which $\alpha$ and $\beta$ are restricted to be equal to $\lambda$ and $1 - \lambda$ so that ...

... $\alpha + \beta = \lambda + 1 - \lambda = 1$ means that EMA volatility is non-stationary.

Empirically, this hypothesis can be tested by assuming the null hypothesis $\alpha + \beta = 1$ versus the one sided hypothesis that $\alpha + \beta < 1$. 
The expected variance $k$ days from the current period $t$ relative to the long-run variance $\sigma^2$ is given by

$$E_t(\sigma^2_{t+k} - \sigma^2) = (\alpha + \beta)^k \cdot (\sigma^2_t - \sigma^2)$$

and shows that:

- If $\alpha + \beta$ is less than 1 the forecast of the conditional variance will converge to the unconditional variance as $k$ increases
- For $\alpha + \beta = 1$ we expect $E(\sigma^2_{t+k} - \sigma^2) = (\sigma^2_t - \sigma^2)$ whatever $k$ is (non-stationarity)
- EMA assumes that $\alpha + \beta = 1$ and thus volatility is never expected to revert back to its long run mean $\sigma^2$
GJR-GARCH

Another GARCH specification that has become popular was proposed by Glosten, Jagganathan and Runkle (hence, **GJR-GARCH**) with the conditional variance given by

\[
\sigma_{t+1}^2 = \omega + \alpha_1 R_t^2 + \gamma_1 R_t^2 \times I(R_t \leq 0) + \beta \sigma_t^2
\]

where the effect of the unexpected return depends on its sign:

- if positive its effect on the conditional variance is \(\alpha_1\)
- if negative the effect is \(\alpha_1 + \gamma_1\)

Testing the hypothesis that \(\gamma_1 = 0\) thus provides a test of the asymmetric effect of current and past squared returns on volatility.
Estimation of GARCH models

- Assume that returns are modeled as \( R_{t+1} = \mu_{t+1}(\theta_{\mu}) + \sigma_{t+1}(\theta_{\sigma})\epsilon_{t+1} \)

- Example:
  - \( \theta_{\mu} = (\beta_0, \beta_1) \) (parameters of the conditional mean)
  - \( \mu_{t}(\theta_{\mu}) = \beta_0 + \beta_1 R_t \)
  - \( \theta_{\sigma} = (\omega, \alpha, \beta) \) (parameters of the conditional variance)
  - \( \sigma_{t}^2(\theta_{\sigma}) = \omega + \alpha \ast R_t^2 + \beta \sigma_t^2 \)

- OLS cannot be used because minimizing the Residuals Sum of Squares

\[
\sum_{t=1}^{T} (R_{t+1} - \mu_{t}(\theta_{\mu}))^2
\]

does not involve the \( \theta_{\sigma} \) parameters, but only \( \theta_{\mu} \)
We need an alternative approach to estimate the model that minimizes a measure of “error” that is also a function of the parameters of the conditional variance ($\theta_\sigma$)

This alternative approach is called Maximum Likelihood (ML) and consists of finding the parameter values that maximize the likelihood (or probability) of the sample observations.

ML requires that we make an assumption about the distribution of the $\epsilon_t$ and we typically assume that they follow a normal distribution (however, we can also assume different distributions such as a $t$ distribution)
Assume that $R_{t+1}$ follows a $N(\mu, \sigma^2)$ and we have a sample of $T$ observations ($t = 1, \cdots, T$)

Based on a sample of $T$ returns, how do we estimate $\mu$ and $\sigma^2$?

The Probability Density Function (PDF) for the normal distribution is given by

$$f(R_{t+1}|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( -\frac{1}{2} \left( \frac{R_{t+1} - \mu}{\sigma} \right)^2 \right)$$

```r
mu = 0
sigma = 1
ggfortify::ggdistribution(dnorm, seq(-6, 6, 0.1), mean = mu, sd = sigma) + theme_bw()
```
ML estimation of $\mu$ and $\sigma^2$ consist of the following steps:

1. The likelihood function for observation $t$ is given by

   $$L(R_{t+1}|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( -\frac{1}{2} \left( \frac{R_{t+1} - \mu}{\sigma} \right)^2 \right)$$

2. Take the logarithm of the likelihood:

   $$\log L(R_{t+1}|\mu, \sigma^2) = l(R_{t+1}|\mu, \sigma^2) = -\frac{1}{2} \log(2\pi\sigma^2) - \frac{1}{2} \left( \frac{R_{t+1} - \mu}{\sigma} \right)^2$$

3. Sum the log-likelihood of each observation for $t = 1, \cdots, T$ and obtain

   $$l(R|\mu, \sigma^2) = \sum_{t=1}^{T} -\frac{1}{2} \log(2\pi\sigma^2) - \frac{1}{2} \left( \frac{R_{t+1} - \mu}{\sigma} \right)^2 =$$

   $$-\frac{1}{2} \left[ T \log(2\pi\sigma^2) + \sum_{t=1}^{T} \left( \frac{R_{t+1} - \mu}{\sigma} \right)^2 \right]$$

4. Find the values of $\mu$ and $\sigma^2$ that maximize $l(R|\mu, \sigma^2)$ numerically
Maximizing $l(R|\mu, \sigma^2)$ means finding the values of the parameters that are more likely to have produced the sample.

In most cases the likelihood does not provide an analytical solution of the estimator for $\mu$ and $\sigma^2$ and this is the reason for using numerical methods instead.

Assume that $\sigma = 1$: maximizing the likelihood function is equivalent to minimize RSS since

$$
max_{\mu} l(R|\mu, \sigma^2) = -\frac{1}{2} \left[ T \log(2\pi) + \sum_{t=1}^{T} (R_{t+1} - \mu)^2 \right]
$$

$$
min_{\mu} RSS(\mu) = \sum_{t=1}^{T} (R_{t+1} - \mu)^2
$$

The next slide illustrates how ML works when assuming that $\mu = 0$ so that we need to maximize only over one parameter, $\sigma$.
In the next slide we plot the log-likelihood as a function of $\sigma$; we follow these steps:

1. Generate a time series from $N(0, \sigma^2)$ of length $T$
2. Create a grid of values for $\sigma^2$
3. Calculate the log-likelihood function $l(R|\sigma^2)$ for each value of $\sigma^2$ in the grid
4. Find the value of $\sigma^2$ that maximizes the log-likelihood function

In the simulation we assume the true value of $\sigma = 0.94$ and $T = 100$
```r
set.seed(1234)
mysd = 0.94 ## true value and we simulate from this
T = 100
# 1) simulate a random sample from the normal distribution
R = rnorm(T, mean=0, sd=mysd)
# 2) create a grid of value for calculating likelihood
sigma = seq(0.4,5,by=0.001)
# 3) calculate the loglik function at each point in the grid
loglik <- matrix(NA, length(sigma), 1)
for (i in 1:length(sigma)) loglik[i] = -0.5 * (T * log(2*pi*sigma[i]^2) + sum((R / sigma[i])^2))
# plot of the loglikelihood function and the true value of the standard deviation
qplot(sigma, loglik, geom="line") + geom_vline(xintercept=mysd, color="steelblue3", linetype="dashed") + theme_classic()
# value that maximizes the loglik function
sigma[which.max(loglik)]

[1] 0.951
```
The grid search is one way to find the maximum of a function
A more automatic way to find the maximum (or minimum) of a function is to use numerical optimization algorithms
There are several ways and functions to do this in R; in the example below I use the function `optim()` that requires three inputs:
- the function to maximize (i.e., log-likelihood function)
- starting values
- data

```r
set.seed(1234)
mysd = 0.94  ## true value and we simulate from this
T = 100      ## sample size
R = rnorm(T, mean=0, sd=mysd)  # simulate a random sample

# define the likelihood function
mylik <- function(data, sigma)
{
  # NB: I drop the minus sign because the function "optim" is designed to minimize a function
  loglik = 0.5 * (T * log(2*pi*sigma^2) + sum((data / sigma)^2))
  return(loglik)
}

# Use "optim" function to find the minimum of mylik with starting value sigma0
sigma0 <- 2
myopt  <- optim(sigma0, mylik, data=R)
myopt$par
```

```
[1] 0.95098
```
The discussion above simplified the application of ML to the case of a constant mean and variance, but the principle applies also when the mean and variance are functions of parameters $(\mu_{t+1}(\theta_{\mu})$ and $\mu_{t+1}(\theta_{\sigma}))$ instead of being the parameters.

The family of GARCH models is large but they can all be expressed as $R_{t+1} = \mu_{t+1} + \eta_{t+1}$ with $\eta_{t+1} = \sigma_{t+1}\epsilon_{t+1}$.

The only difference between specifications is the conditional variance $\sigma^2_{t+1}$; for example:

- GARCH(1,1): $\sigma^2_{t+1} = \omega + \alpha\eta^2_t + \beta\sigma^2_t$
- GJR-GARCH(1,1): $\sigma^2_{t+1} = \omega + \alpha_1 \eta^2_t + \gamma_1 \eta^2_t * I(\eta_t \leq 0) + \beta * \sigma^2_t$
- ...
The log-likelihood of any GARCH model can be expressed in terms of the parameters $\hat{\theta}_\mu$ and $\hat{\theta}_\sigma$ and it is given by

$$l(R|\theta_\mu, \theta_\sigma) = -\frac{1}{2} \sum_{t=1}^{T} \left[ \ln(2\pi) + \ln \sigma_t^2(\theta_\sigma) + \left( \frac{R_{t+1} - \mu_t(\theta_\mu)}{\sigma_t(\theta_\sigma)} \right)^2 \right]$$

since the first term $\ln(2\pi)$ does not depend on any parameter it can be dropped from the function; the estimates $\hat{\theta}_\mu$ and $\hat{\theta}_\sigma$ are then the result of the maximization of

$$l(R|\theta_\mu, \theta_\sigma) = -\frac{1}{2} \sum_{t=1}^{T} \left[ \ln \sigma_t^2(\theta_\sigma) + \left( \frac{R_{t+1} - \mu_t(\theta_\mu)}{\sigma_t(\theta_\sigma)} \right)^2 \right]$$
GARCH in R

- The package fGarch provides functionalities to estimate a wide array of GARCH models
- The function garchFit estimates by ML a model specified by the user:
  - \( \text{arma}(p,q) \) for the conditional mean (typically \( \text{arma}(0,0) \) or \( \text{arma}(1,0) \))
  - \( \text{garch}(r,s) \) for the conditional variance (typically \( \text{garch}(1,1) \))
- The code below estimates a GARCH(1,1) model:

```r
require(fGarch)
fit <- garchFit(~garch(1,1), data=sp500daily, trace=FALSE)
round(fit@fit$matcoef, 3)
```

|        | Estimate | Std. Error | t value | Pr(>|t|) |
|--------|----------|------------|---------|----------|
| mu     | 0.060    | 0.008      | 7.431   | 0        |
| omega  | 0.014    | 0.002      | 7.184   | 0        |
| alpha1 | 0.084    | 0.006      | 14.389  | 0        |
| beta1  | 0.905    | 0.007      | 135.880 | 0        |

- \( \alpha + \beta = 0.988 \) is very close to one and indicates that the EMA assumptions is likely to hold
- However, while practitioners use \( \lambda \) equal to 0.06 the estimation on our sample suggests a higher value of 0.084
We can add an AR(1) term to the conditional mean, while the conditional variance remains of the GARCH(1,1) type:

```r
fit <- garchFit(~ arma(1,0) + garch(1,1), data=sp500daily, trace=FALSE)
round(fit@fit$matcoef, 3)
```

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|----------|
| mu       | 0.060      | 0.008   | 7.395    | 0.000    |
| ar1      | 0.003      | 0.011   | 0.244    | 0.807    |
| omega    | 0.014      | 0.002   | 7.184    | 0.000    |
| alpha1   | 0.084      | 0.006   | 14.389   | 0.000    |
| beta1    | 0.905      | 0.007   | 135.842  | 0.000    |

- the AR(1) term is not significant
- The estimates of $\alpha$ and $\beta$ are the same as in the case of just the intercept
Plotting $\sigma_{t+1}$

- Once we have the coefficient estimates we can calculate the estimate of the conditional variance as $\sigma_{t+1}^2 = \hat{\omega} + \hat{\alpha} R_t^2 + \hat{\beta} \sigma_t^2$
- The function calculates $\sigma_t$ as output of the estimation step

```r
sigma <- fit@sigma.t  # class is numeric
qplot(time(sp500daily), sigma, geom="line", xlab=NULL, ylab="") + theme_bw() + labs(title="Conditional standard deviation")
```

![Conditional standard deviation](image)
Comparison of MA, EMA, and GARCH volatilities

- We considered three methods for modeling and forecasting volatility: MA, EMA, and GARCH
- How different are the volatility forecasts obtained from these three methods?
- Correlations: $\text{cor}(\text{MA}, \text{GARCH}) = 0.967$ and $\text{cor}(\text{EMA}, \text{GARCH}) = 0.981$
- Scatter plots:

```r
p1 <- ggplot() +
  geom_point(aes(ma25^0.5, sigma), color="gray70", size=0.3) +
  geom_abline(intercept=0, slope=1, color="steelblue3", linetype="dashed", size=0.8) +
  theme_bw() + labs(x="MA(25)",y="AR(1)-GARCH(1,1)")

p2 <- ggplot() +
  geom_point(aes(ema06^0.5, sigma), color="gray70", size=0.3) +
  geom_abline(intercept=0, slope=1, color="steelblue3", linetype="dashed", size=0.8) +
  theme_bw() + labs(x="EMA(0.06)",y="AR(1)-GARCH(1,1)")

ggridExtra::grid.arrange(p1,p2, ncol=2)
```
Standardized and Unstandardized residuals

- The residuals of the GARCH model are $\eta_{t+1} = \sigma_{t+1} \epsilon_{t+1}$
- According to our assumptions they should be:
  - independent over time (i.e., no auto-correlation)
  - normally distributed (i.e., we made this assumption)
- Two types of residuals:
  1. unstandardized residuals: $\eta_{t+1}$
  2. standardized residuals: $\epsilon_{t+1} = \eta_{t+1} / \sigma_{t+1}$

```r
res <- fit$residuals  # class is numeric
p1 <- qplot(time(sp500daily), res, geom="line", main="UNstandardized Residuals") + labs(x=NULL, y=NULL) + theme_bw() + theme(plot.title = element_text(size = 8))
p2 <- qplot(time(sp500daily), res/sigma, geom="line", main="Standardized Residuals") + labs(x=NULL, y=NULL) + theme_bw() + theme(plot.title = element_text(size = 8))
gridExtra::grid.arrange(p1, p2, ncol=2)
```
Are the standardized residuals approximately normal?

Figure below shows the histogram of $\epsilon_{t+1}$ and the standard normal distribution

The std. residuals seem to be more peaked at the center and with longer tails (not very visible at this scale)

But: skewness = -0.47189 and excess kurtosis = 3.32609

Conclusion:

1. normal distribution for $\epsilon_{t+1}$ is inaccurate
2. the GARCH(1,1) conditional variance equation is inaccurate
We started off with the ACF of the squared returns $R_{t+1}^2$ showing significant and persistent autocorrelation.

We now filter the residuals with the conditional standard deviation, $\eta_{t+1}/\sigma_{t+1}$ and we find all this autocorrelation has disappeared.

This indicates that filtering the returns with the GARCH(1,1) conditional standard deviation takes care of the dependence in the squared returns.

```r
p1 <- ggacf(res/sigma, lag=20)
p2 <- ggacf((res/sigma)^2, lag=20)
ggplot::grid.arrange(p1, p2, ncol=2)
```
Another package that estimates many GARCH-type models is **rugarch**

Estimation of a GARCH model proceeds in two steps:

1. **ugarchspec()**: specify the model in terms of the conditional mean and variance
2. **ugarchfit()**: estimate the model specified in the previous step

```r
require(rugarch)
spec = ugarchspec(variance.model=list(model="sGARCH",garchOrder=c(1,1)),
                   mean.model=list(armaOrder=c(1,0)))
fitgarch = ugarchfit(spec = spec, data = sp500daily)
round(coef(fitgarch), 3)
```

<table>
<thead>
<tr>
<th>mu</th>
<th>ar1</th>
<th>omega</th>
<th>alpha1</th>
<th>beta1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.060</td>
<td>0.003</td>
<td>0.014</td>
<td>0.084</td>
<td>0.905</td>
</tr>
</tbody>
</table>
The GJR-GARCH model can be easily estimated by simply setting the model option to `gjrGARCH` in the model specification function `ugarchspec()`.

```r
spec = ugarchspec(variance.model=list(model="gjrGARCH", garchOrder=c(1,1)),
                   mean.model=list(armaOrder=c(1,0)))
fitgjr = ugarchfit(spec = spec, data = sp500daily)
knitr::kable(data.frame(Estimate=coef(fitgjr), SE=fitgjr@fit$tval))
```

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>mu</td>
<td>0.03723</td>
<td>4.52342</td>
</tr>
<tr>
<td>ar1</td>
<td>0.00753</td>
<td>0.68824</td>
</tr>
<tr>
<td>omega</td>
<td>0.01875</td>
<td>8.48399</td>
</tr>
<tr>
<td>alpha1</td>
<td>0.01946</td>
<td>4.46062</td>
</tr>
<tr>
<td>beta1</td>
<td>0.90219</td>
<td>132.58377</td>
</tr>
<tr>
<td>gamma1</td>
<td>0.12181</td>
<td>12.04780</td>
</tr>
</tbody>
</table>

▶ Is the asymmetry coefficient significant?
▶ What do we learn about the S&P 500 volatility dynamics?
Comparison of GARCH(1,1) and GJR-GARCH(1,1) volatility estimates

```r
p1 <- autoplot(merge(GARCH = sigma(fitgarch), GJR = sigma(fitgjr)), scales="fixed") + theme_bw() + theme(strip.text = element_text(size=5), text = element_text(size=5))
p2 <- ggplot(data=merge(GARCH = sigma(fitgarch), GJR = sigma(fitgjr))) + geom_point(aes(x = GARCH, y=GJR), color="gray70", size=0.3) + geom_abline(intercept=0, slope=1, color="steelblue3", linetype="dashed", size=0.8) + theme_bw() + theme(text = element_text(size=5))
ggridExtra::grid.arrange(p1, p2, ncol=2)
```
GJR on FX returns?

- Is the asymmetric effect of returns on volatility also relevant for exchange rate volatility?
- Estimation of GJR-GARCH(1,1) on JPY/USD daily returns:

```r
fitgjr.fx = ugarchfit(spec = spec, data = DEXJPUS)
data.frame(Estimate=coef(fitgjr.fx), SE=fitgjr.fx@fit$tval)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>mu</td>
<td>-0.0042226</td>
<td>-0.65700</td>
</tr>
<tr>
<td>ar1</td>
<td>0.0040482</td>
<td>0.36879</td>
</tr>
<tr>
<td>omega</td>
<td>0.0078450</td>
<td>2.55350</td>
</tr>
<tr>
<td>alpha1</td>
<td>0.0393570</td>
<td>4.08780</td>
</tr>
<tr>
<td>beta1</td>
<td>0.9390349</td>
<td>56.07169</td>
</tr>
<tr>
<td>gamma1</td>
<td>0.0077128</td>
<td>1.39328</td>
</tr>
</tbody>
</table>

- Is the asymmetry coefficient significant?
- What do we learn about the FX volatility dynamics?
How do we compare volatility models?

- The Akaike Information Criterion (AIC) discussed for AR models can also be used to select GARCH models.
- The model with the smallest criterion value is selected.

\[
\begin{array}{ll}
\text{GARCH} & \text{GJR} \\
\text{Akaike} & 2.6571 \quad 2.6339 \\
\text{Bayes} & 2.6608 \quad 2.6383 \\
\text{Shibata} & 2.6571 \quad 2.6339 \\
\text{Hannan-Quinn} & 2.6583 \quad 2.6354 \\
\end{array}
\]

- Which model shall we select: GARCH or GJR-GARCH?

- For FX returns:

\[
\begin{array}{ll}
\text{GARCH} & \text{GJR} \\
\text{Akaike} & 1.9383 \quad 1.9382 \\
\text{Bayes} & 1.9421 \quad 1.9429 \\
\text{Shibata} & 1.9383 \quad 1.9382 \\
\text{Hannan-Quinn} & 1.9396 \quad 1.9398 \\
\end{array}
\]

- Which model shall we select: GARCH or GJR-GARCH?
Forecasts from GARCH models

- The estimated $\sigma_t$ at the end of the sample (2018-04-11) is 1.423 for GARCH and 1.478 for GJR-GARCH
- We then forecast volatility for the following 250 days as shown below:

```r
nforecast = 250
garchforecast <- ugarchforecast(fitgarch, n.ahead = nforecast)
gjrchọrdouserc -< ugarchforecast(fitgjr, n.ahead = nforecast)
temp <- data.frame(Date = end(sp500daily) + 1:nforecast,
                   GJR = as.numeric(sigma(gjrchọrdouserc)),
                   GARCH = as.numeric(sigma(garchforecast)))
ggplot(temp) + geom_line(aes(Date, GARCH), color="tomato3") +
                     geom_line(aes(Date, GJR), color="steelblue2") +
                     geom_hline(yintercept = sd(sp500daily), color="seagreen4", linetype="dashed") +
                     labs(x=NULL, y=NULL, title="Volatility forecasts",
                          subtitle=paste("Forecast date: ", end(sp500daily))) + theme_bw()
```

Volatility forecasts
Forecast date: 2018-04-11
The GJR-GARCH implies an asymmetric response to shocks ($\epsilon_{t-1}$) relative to the GARCH model which is evident when plotting the **news impact curve** (the response of conditional variance $\sigma^2_t$ to a shock $\epsilon_{t-1}$).

```r
newsgarch <- newsimpact(fitgarch)
newsgjr  <- newsimpact(fitgjr)
p1 <- qplot(newsgarch$zx, newsgarch$zy, geom="line", main="GARCH") + labs(x="shock", y="volatility") + geom_vline(xintercept = 0, linetype="dashed") + theme_bw() + theme(text = element_text(size=5))
p2 <- qplot(newsgjr$zx, newsgjr$zy, geom="line", main="GJR") + labs(x="shock", y="volatility") + geom_vline(xintercept = 0, linetype="dashed") + theme_bw() + theme(text = element_text(size=5))
gridExtra::grid.arrange(p1, p2, ncol=2)
```