Econométrie de la finance

Chapitre 2: Les modèles GARCH

Mohamed Essaied Hamrita

Octobre 2021

Introduction

- Comme nous l'avons déjà vu dans le chapitre précédent, que dans la plus part des cas, les séries financières remettent en cause la propriété d'homoscédasticité.

Introduction

- Comme nous l'avons déjà vu dans le chapitre précédent, que dans la plus part des cas, les séries financières remettent en cause la propriété d'homoscédasticité.
- L'approche ARCH/GARCH est proposée pour prendre en compte des variances conditionnelles dépendantes du temps.

Le modèle ARCH

Définition: Soit le processus $X_t = \{X_1, X_2, \ldots, X_T\}$ et $\mathcal{I}_{t-1} = \{X_1, X_2, \ldots, X_{t-1}\}$ l'information disponible à l'instant $t-1$. X_t est dit un processus **ARCH** d'ordre p, noté $X_t \sim ARCH(p)$, s'il vérifie la relation suivante:

$$\begin{cases} X_t = \varepsilon_t, \;\; \varepsilon \sim N(0, \sigma_t) & \text{(Mean conditional equation)} \\ \varepsilon_t = Z_t \sigma_t, \;\; Z_t \overset{iid}{\sim} N(0,1) & \\ \sigma_t^2 = a_0 + a_1 \varepsilon_{t-1}^2 + a_2 \varepsilon_{t-2}^2 + \ldots + a_p \varepsilon_{t-p}^2 & \text{(Variance conditional equat} \end{cases}$$

avec $a_0 > 0$, $a_1, \ldots, a_p \geq 0$ et $a_1 + a_2 + \ldots + a_p < 1$.

σ_t^2 est la **variance conditionnelle** du processus X_t, $\sigma_t^2 = \mathbb{V}(X_t | \mathcal{I}_{t-1})$.

Ce processus est proposé par (Engle 1982).

Exemple: Soit $X_t \sim ARCH(1)$. La figure suivante est une réalisation (simulation) du modèle $ARCH(1)$ définit par: $X_t = \varepsilon_t = \sigma_t Z_t$ et $\sigma_t^2 = 0.4 + 0.7\varepsilon_{t-1}^2$.

```
library(fGarch); set.seed(12345)
arch1=garchSim(garchSpec(model=list(omega=0.4,alpha=0.7, be
par(mfrow=c(1,2))
plot(arch1$garch, type="l", col=2, main="Simulation ARCH(1)
plot(arch1$sigma^2, type="l", col=2, main="Variance conditi
```

Propriétés statistiques

Soit $X_t \sim ARCH(p)$. On a pour tout t et $h \geq 1$,

- $\mathbb{E}\left(X_t | \mathcal{I}_{t-1}\right) = 0$ et $\mathbb{E}\left(X_t\right) = 0$.

Propriétés statistiques

Soit $X_t \sim ARCH(p)$. On a pour tout t et $h \geq 1$,

- $\mathbb{E}\left(X_t | \mathcal{I}_{t-1}\right) = 0$ et $\mathbb{E}\left(X_t\right) = 0$.
- $\mathbb{V}\left(X_t | \mathcal{I}_{t-1}\right) = \sigma_t^2$, $\forall t$ et $\mathbb{V}\left(X_t\right) = \dfrac{a_0}{1 - \sum_{i=1}^{p} a_i}$.

Propriétés statistiques

Soit $X_t \sim ARCH(p)$. On a pour tout t et $h \geq 1$,

- $\mathbb{E}\left(X_t | \mathcal{I}_{t-1}\right) = 0$ et $\mathbb{E}\left(X_t\right) = 0$.
- $\mathbb{V}\left(X_t | \mathcal{I}_{t-1}\right) = \sigma_t^2,\ \forall t$ et $\mathbb{V}\left(X_t\right) = \dfrac{a_0}{1 - \displaystyle\sum_{i=1}^{p} a_i}$.
- $\mathbb{C}\rtimes\succapprox\left(X_t X_{t+h} | \mathcal{I}_{t-1}\right) = 0$ pour $h \geq 1$ et $\mathbb{C}\rtimes\succapprox\left(X_t X_{t+h}\right) = 0$.

Propriétés statistiques

Soit $X_t \sim ARCH(p)$. On a pour tout t et $h \geq 1$,

- $\mathbb{E}\left(X_t | \mathcal{I}_{t-1}\right) = 0$ et $\mathbb{E}\left(X_t\right) = 0$.
- $\mathbb{V}\left(X_t | \mathcal{I}_{t-1}\right) = \sigma_t^2,\ \forall t$ et $\mathbb{V}\left(X_t\right) = \dfrac{a_0}{1 - \displaystyle\sum_{i=1}^{p} a_i}$.
- $\mathbb{C}\rtimes\succapprox\left(X_t X_{t+h} | \mathcal{I}_{t-1}\right) = 0$ pour $h \geq 1$ et $\mathbb{C}\rtimes\succapprox\left(X_t X_{t+h}\right) = 0$.
- **Rappel**

Propriétés statistiques

Soit $X_t \sim ARCH(p)$. On a pour tout t et $h \geq 1$,

- $\mathbb{E}\left(X_t|\mathcal{I}_{t-1}\right) = 0$ et $\mathbb{E}\left(X_t\right) = 0$.
- $\mathbb{V}\left(X_t|\mathcal{I}_{t-1}\right) = \sigma_t^2,\ \forall t$ et $\mathbb{V}\left(X_t\right) = \dfrac{a_0}{1 - \displaystyle\sum_{i=1}^{p} a_i}$.
- $\mathbb{C}\rtimes\gtrapprox\left(X_t X_{t+h}|\mathcal{I}_{t-1}\right) = 0$ pour $h \geq 1$ et $\mathbb{C}\rtimes\gtrapprox\left(X_t X_{t+h}\right) = 0$.
- **Rappel**
- $\mathbb{E}(X) = \mathbb{E}(\mathbb{E}(X|Y))$.

Propriétés statistiques

Soit $X_t \sim ARCH(p)$. On a pour tout t et $h \geq 1$,

- $\mathbb{E}\left(X_t | \mathcal{I}_{t-1}\right) = 0$ et $\mathbb{E}\left(X_t\right) = 0$.
- $\mathbb{V}\left(X_t | \mathcal{I}_{t-1}\right) = \sigma_t^2,\ \forall t$ et $\mathbb{V}\left(X_t\right) = \dfrac{a_0}{1 - \displaystyle\sum_{i=1}^{p} a_i}$.
- $\mathbb{C}\rtimes\gtrapprox\left(X_t X_{t+h} | \mathcal{I}_{t-1}\right) = 0$ pour $h \geq 1$ et $\mathbb{C}\rtimes\gtrapprox\left(X_t X_{t+h}\right) = 0$.
- **Rappel**
- $\mathbb{E}(X) = \mathbb{E}\big(\mathbb{E}(X|Y)\big)$.
- Soit $A_1 \subseteq A_2$, $\mathbb{E}(X|A_1) = \mathbb{E}\big(\mathbb{E}(X|A_2)|A_1\big)$.

La construction du modèle ARCH

- La construction du modèle ARCH passe par 4 étapes:

La construction du modèle ARCH

- La construction du modèle ARCH passe par 4 étapes:
  1. Détermination de l'ordre p.

La construction du modèle ARCH

- La construction du modèle ARCH passe par 4 étapes:
    1. Détermination de l'ordre p.
    2. Estimation du modèle $ARCH(p)$.

La construction du modèle ARCH

- La construction du modèle ARCH passe par 4 étapes:
  1. Détermination de l'ordre p.
  2. Estimation du modèle $ARCH(p)$.
  3. Diagnostic du modèle estimé.

La construction du modèle ARCH

- La construction du modèle ARCH passe par 4 étapes:
  1. Détermination de l'ordre p.
  2. Estimation du modèle $ARCH(p)$.
  3. Diagnostic du modèle estimé.
  4. Prévision.

Détermination de l'ordre p:

- Similairement aux modèles ARMA, l'ordre p peut être déterminé en examinant la fonction d'auto-corrélation partielle de ε_t^2.

Détermination de l'ordre p:

- Similairement aux modèles ARMA, l'ordre p peut être déterminé en examinant la fonction d'auto-corrélation partielle de ε_t^2.
- On peut aussi faire recours aux critères de sélection: AIC, SIC et AICc.

Détermination de l'ordre p:

- Similairement aux modèles ARMA, l'ordre p peut être déterminé en examinant la fonction d'auto-corrélation partielle de ε_t^2.
- On peut aussi faire recours aux critères de sélection: AIC, SIC et AICc.
- $AIC = -2logL + 2k$, (Akaike information criteria) $k =$ nombre de paramètres dans le modèle estimé.

Détermination de l'ordre p:

- Similairement aux modèles ARMA, l'ordre p peut être déterminé en examinant la fonction d'auto-corrélation partielle de ε_t^2.
- On peut aussi faire recours aux critères de sélection: AIC, SIC et AICc.
- $AIC = -2logL + 2k$, (Akaike information criteria) $k =$ nombre de paramètres dans le modèle estimé.
- $BIC = -2logL + \log(T)k$. (Bayesian information criteria)

Détermination de l'ordre p:

- Similairement aux modèles ARMA, l'ordre p peut être déterminé en examinant la fonction d'auto-corrélation partielle de ε_t^2.
- On peut aussi faire recours aux critères de sélection: AIC, SIC et AICc.
- $AIC = -2logL + 2k$, (Akaike information criteria) $k =$ nombre de paramètres dans le modèle estimé.
- $BIC = -2logL + \log(T)k$. (Bayesian information criteria)
- Pour un échantillon de petite taille, on utilise le critère $AICc$ qui est défini par

$$AICc = AIC + \frac{2k(k+1)}{T-k-1}$$

Reprenons notre exemple du chapitre précédent, les rendements du bitcoin.

```
library(quantmod); library(zoo)
btc=getSymbols("BTC-USD", src="yahoo", from="2014-09-17",to
closedAdj=zoo(na.omit(btc[,6])); rt=diff(log(closedAdj))
pacf(rt^2,na.action = na.pass, col=4, xlab="")
```

D'après le graphique de la fonction d'auto-corrélation partielle de r_t^2, on remarque bien que le modèle ARCH d'ordre 1 est approprié aux rendements du BTC. On remarque aussi, que les pacf d'ordres 4 et 7 sont aussi significativement différents de zéro. Déterminons les valeurs des critères d'information pour les modèles ARCH(1) et ARCH(4). Sous R, il existe plusieurs packages permettant l'estimation des modèles GARCH tels que `tseries`, `fGarch`, `rugarch`. Ici, nous décrivons l'utilisation des packages `fGarch` et `rugarch`. Pour l'estimation du modèle ARCH, nous devons spécifier le modèle à estimer en donnant les ordres des différents modèles (mean and variance equations). La fonction à utiliser est `ugarchspec` (`rugarch`) et prend comme arguments principaux `variance.model`, `mean.model` et `distribution.model`. Les deux premiers arguments sont des listes et le toisième est une chaîne de caractère qui peut être `"norm"`, `"std"`, `"sstd "ged"` ou `sged`.

```r
library(rugarch)   # charger le package
spec1=ugarchspec(variance.model = list(garchOrder=c(1,0)),
                mean.model = list(armaOrder=c(0,0)))
spec4=ugarchspec(variance.model = list(garchOrder=c(4,0)),
                mean.model = list(armaOrder=c(0,0)))
fit1=ugarchfit(spec1,rt)  # Estimation
fit4=ugarchfit(spec4,rt)
t(infocriteria(fit1))  # critères d'information (normalisés
```

```
    Akaike     Bayes   Shibata Hannan-Quinn
 0.5254326 0.5322282 0.5254299    0.5278955
```

```r
t(infocriteria(fit4))
```

```
   Akaike     Bayes   Shibata Hannan-Quinn
 -3.75665 -3.743059 -3.756661    -3.751724
```

Estimation

- Sous l'hypothèse de normalité des erreurs, la fonction de vraisemblance d'un modèle $ARCH(p)$ est:

$$L(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_T|\mathbf{a}) = \prod_{i=p+1}^{T} \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left(-\frac{\varepsilon_t^2}{2\sigma_t^2}\right) \times f(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_T|\mathbf{a}$$

où $\mathbf{a} = (a_0, a_1, \ldots, a_p)$ et $f(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_T|\mathbf{a})$ la densité conjointe conditionnelle des erreurs.

Estimation

- Sous l'hypothèse de normalité des erreurs, la fonction de vraisemblance d'un modèle $ARCH(p)$ est:

$$L(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_T|\mathbf{a}) = \prod_{i=p+1}^{T} \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left(-\frac{\varepsilon_t^2}{2\sigma_t^2}\right) \times f(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_T|\mathbf{a}$$

  où $\mathbf{a} = (a_0, a_1, \dots, a_p)$ et $f(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_T|\mathbf{a})$ la densité conjointe conditionnelle des erreurs.

- Maximiser la fonction de vraisemblance conditionnelle est équivalent à maximiser son logarithme. Le logarithme de la vraisemblance conditionnelle est:

$$\ell(\varepsilon_{p+1}, \varepsilon_{p+2,\dots,\varepsilon_{T|\mathbf{a},a_{1,a_{2,\dots,a_{p)=\sum_{i=p+1}^{T}\left[-\frac{1}{2}\log(2\pi)-\frac{1}{2}\log(\sigma_t^2)-\frac{1}{2}\frac{\varepsilon_t^2}{\sigma_t^2}\right]}}}}}$$

  où $\sigma_t^2 = a_0 + a_1\varepsilon_{t-1}^2 + a_2\varepsilon_{t-2}^2 + \dots + a_p\varepsilon_{t-p}^2$ qui peut être calculé récursivement.

Estimation

- Sous l'hypothèse de normalité des erreurs, la fonction de vraisemblance d'un modèle $ARCH(p)$ est:

$$L(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_T|\mathbf{a}) = \prod_{i=p+1}^{T} \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left(-\frac{\varepsilon_t^2}{2\sigma_t^2}\right) \times f(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_T|\mathbf{a}$$

  où $\mathbf{a} = (a_0, a_1, \ldots, a_p)$ et $f(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_T|\mathbf{a})$ la densité conjointe conditionnelle des erreurs.

- Maximiser la fonction de vraisemblance conditionnelle est équivalent à maximiser son logarithme. Le logarithme de la vraisemblance conditionnelle est:

$$\ell(\varepsilon_{p+1}, \varepsilon_{p+2,\ldots,\varepsilon_{T|\mathbf{a},a_{1,a_{2,\ldots,a}}}}{}_{p)=\sum_{i=p+1}^{T}\left[-\frac{1}{2}\log(2\pi)-\frac{1}{2}\log(\sigma_t^2)-\frac{1}{2}\frac{\varepsilon_t^2}{\sigma_t^2}\right]}$$

  où $\sigma_t^2 = a_0 + a_1\varepsilon_{t-1}^2 + a_2\varepsilon_{t-2}^2 + \ldots + a_p\varepsilon_{t-p}^2$ qui peut être calculé récursivement.

- **Remarque:** On peut aussi utiliser d'autres distributions autre que la loi normale telles que la loi de student ou la loi GED

Exemple: Soit $X_t \sim ARCH(1)$. Donner les estimateurs de μ, a_0, et a_1 par la méthode de MV.

- Le logarithme de la vraisemblance conditionnelle est donnée par:

$$\ell(\varepsilon_2, \varepsilon_3, \ldots, \varepsilon_T | \mathbf{a}, a_1, a_2) = \sum_{i=2}^{T} \left[-\frac{1}{2} \log(2\pi) - \frac{1}{2} \log(a_0 + a_1 X_{t-1}^2) - \right.$$

Les paramètres μ, a_0 et a_1 se déduisent en résolvant le système suivant

$$\begin{cases} \frac{\partial \ell}{\partial \mu} = 0 \\ \frac{\partial \ell}{\partial a_0} = 0 \\ \frac{\partial \ell}{\partial a_1} = 0 \end{cases}$$

```r
fit1@fit$matcoef
```

```
          Estimate   Std. Error      t value    Pr(>|t|)
mu     1.774978e-01 5.624388e-05 3155.859739 0.000000000
omega  1.470274e-06 4.864592e-07    3.022399 0.002507798
alpha1 5.665866e-01 1.517078e-04 3734.722111 0.000000000
```

```r
show(fit1)
```

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*
Conditional Variance Dynamics
-----------------------------------
GARCH Model : sGARCH(1,0)
Mean Model  : ARFIMA(0,0,0)
Distribution    : norm
Optimal Parameters
------------------------------------
        Estimate  Std. Error   t value Pr(>|t|)
mu      0.177498    0.000056 3155.8597 0.000000
```

- Le modèle estimé est alors:

$$\begin{cases} X_t = 0.1775 + \sigma_t \varepsilon_t \\ \sigma_t^2 = 1.4702 \times 10^{-6} + 0.56658\, X_{t-1}^2 \end{cases}$$

- Estimation avec des erreurs de loi de student:

```
hist(rt,col=4, prob=T, breaks = 50) # histogramme or rt
lines(density(rt), col=2,lwd=3) # density estimation (kerne
curve(dnorm(x, mean(rt),sd(rt)), lwd=3, col="darkorchid3",
legend("topleft",bty="n",lwd=3, col=c(2,"darkorchid3"),
       legend=c("Kernel","Normal"))
```

- Estimation avec des erreurs de loi de student:
- Tout d'abord, examinons la distribution des erreurs et la comparons par la densité normale.

```
hist(rt,col=4, prob=T, breaks = 50) # histogramme or rt
lines(density(rt), col=2,lwd=3) # density estimation (kerne
curve(dnorm(x, mean(rt),sd(rt)), lwd=3, col="darkorchid3",
legend("topleft",bty="n",lwd=3, col=c(2,"darkorchid3"),
       legend=c("Kernel","Normal"))
```

```
specSt=ugarchspec(list(garchOrder=c(1,0)), list(armaOrder=c
fitst=ugarchfit(specSt,rt)
show(fitst)
```

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*
Conditional Variance Dynamics
-----------------------------------
GARCH Model : sGARCH(1,0)
Mean Model  : ARFIMA(0,0,0)
Distribution    : std
Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.002287    0.000474   4.8272 0.000001
omega   0.002465    0.000820   3.0058 0.002649
alpha1  0.999000    0.380806   2.6234 0.008706
shape   2.291507    0.123369  18.5745 0.000000
Robust Standard Errors:
```

Diagnostique du modèle estimé

- Pour un modèle ARCH bien approprié, les erreurs standards $\widetilde{\varepsilon}_t = \frac{\varepsilon_t}{\sigma_t}$ doivent être *iid*. La statistique de Ljung-Box peut être appliqué sur les erreurs standards $(\widetilde{\varepsilon}_t)$ pour examiner l'équation de la moyenne conditionnelle et sur leurs carrés $(\widetilde{\varepsilon}_t^2)$ pour examiner l'équation de la variance conditionnelle.

Diagnostique du modèle estimé

- Pour un modèle ARCH bien approprié, les erreurs standards $\widetilde{\varepsilon}_t = \frac{\varepsilon_t}{\sigma_t}$ doivent être *iid*. La statistique de Ljung-Box peut être appliqué sur les erreurs standards $(\widetilde{\varepsilon}_t)$ pour examiner l'équation de la moyenne conditionnelle et sur leurs carrés $(\widetilde{\varepsilon}_t^2)$ pour examiner l'équation de la variance conditionnelle.
- Lest tests de stochasticité (randomness tests) peuvent être aussi appliqué tels que le test BDS, run test, etc...

Diagnostique du modèle estimé

- Pour un modèle ARCH bien approprié, les erreurs standards $\widetilde{\varepsilon}_t = \frac{\varepsilon_t}{\sigma_t}$ doivent être *iid*. La statistique de Ljung-Box peut être appliqué sur les erreurs standards ($\widetilde{\varepsilon}_t$) pour examiner l'équation de la moyenne conditionnelle et sur leurs carrés ($\widetilde{\varepsilon}_t^2$) pour examiner l'équation de la variance conditionnelle.
- Lest tests de stochasticité (randomness tests) peuvent être aussi appliqué tels que le test BDS, run test, etc...
- Le package `rugarch` utilise plutôt les statistiques de Ljung-Box pondérée et LM-ARCH pondérée proposé par (Fisher and Gallagher 2012)

Diagnostique du modèle estimé

- Pour un modèle ARCH bien approprié, les erreurs standards $\widetilde{\varepsilon}_t = \frac{\varepsilon_t}{\sigma_t}$ doivent être *iid*. La statistique de Ljung-Box peut être appliqué sur les erreurs standards $(\widetilde{\varepsilon}_t)$ pour examiner l'équation de la moyenne conditionnelle et sur leurs carrés $(\widetilde{\varepsilon}_t^2)$ pour examiner l'équation de la variance conditionnelle.
- Lest tests de stochasticité (randomness tests) peuvent être aussi appliqué tels que le test BDS, run test, etc. . .
- Le package `rugarch` utilise plutôt les statistiques de Ljung-Box pondérée et LM-ARCH pondérée proposé par (Fisher and Gallagher 2012)
- Le package `fGarch` utilise les statistiques de Ljung-Box et LM-ARCH standards sur les erreurs standards et leurs carrés.

```
library(fGarch)
fitst2=garchFit(~garch(1,0),rt, cond.dist = "std", trace =
```

```
Warning: Using formula(x) is deprecated when x is a charact
  Consider formula(paste(x, collapse = " ")) instead.
```

```
fitst2@fit$matcoef   # fGarch
```

```
          Estimate   Std. Error   t value     Pr(>|t|)
mu     0.002290206 0.0004739852  4.831808 1.352984e-06
omega  0.002509288 0.0008199825  3.060172 2.212096e-03
alpha1 0.999999990 0.3677348660  2.719350 6.541026e-03
shape  2.286381480 0.1179821323 19.379049 0.000000e+00
```

```
fitst@fit$matcoef    # rugarch
```

```
          Estimate   Std. Error   t value     Pr(>|t|)
mu     0.002287490 0.0004738717  4.827234 1.384422e-06
omega  0.002465219 0.0008201576  3.005787 2.648943e-03
alpha1 0.998999883 0.3808058193  2.623384 8.706109e-03
shape  2.291507174 0.1233687554 18.574453 0.000000e+00
```

```
summary(fitst2)
```

```
Title:
 GARCH Modelling
Call:
 garchFit(formula = ~garch(1, 0), data = rt, cond.dist = "s
    trace = F)
Mean and Variance Equation:
 data ~ garch(1, 0)
<environment: 0x0000000034aef4d8>
 [data = rt]
Conditional Distribution:
 std
Coefficient(s):
       mu      omega     alpha1      shape
0.0022902  0.0025093  1.0000000  2.2863815
Std. Errors:
 based on Hessian
Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
```

Prévision

- Les prévisions du model ARCH est obtenues récursivement comme celles du modèle AR.

Prévision

- Les prévisions du model ARCH est obtenues récursivement comme celles du modèle AR.
- La prévision à une étape est $\sigma_h^2(1) = a_0 + a_1\varepsilon_h^2 + \ldots + a_p\varepsilon_{h+1-p}^2$.

Prévision

- Les prévisions du model ARCH est obtenues récursivement comme celles du modèle AR.
- La prévision à une étape est $\sigma_h^2(1) = a_0 + a_1\varepsilon_h^2 + \ldots + a_p\varepsilon_{h+1-p}^2$.
- La prévision à deux étapes est $\sigma_h^2(2) = a_0 + a_1\sigma_h^2(1) + a_2\varepsilon_h^2 + \ldots + a_p\varepsilon_{h+2-p}^2$.

Prévision

- Les prévisions du model ARCH est obtenues récursivement comme celles du modèle AR.
- La prévision à une étape est $\sigma_h^2(1) = a_0 + a_1\varepsilon_h^2 + \ldots + a_p\varepsilon_{h+1-p}^2$.
- La prévision à deux étapes est $\sigma_h^2(2) = a_0 + a_1\sigma_h^2(1) + a_2\varepsilon_h^2 + \ldots + a_p\varepsilon_{h+2-p}^2$.
- La prévision à k étapes est $\sigma_h^2(k) = a_0 + \displaystyle\sum_{j=1}^{p} a_j\sigma_h^2(k-j)$ où $\sigma_h^2(k-j) = \varepsilon_{h+k-1}^2$ si $k - j \leq 0$.

```r
predict(fitst2,3)                # fGarch
```

```
  meanForecast  meanError standardDeviation
1  0.002290206 0.05567025        0.05567025
2  0.002290206 0.07488968        0.07488968
3  0.002290206 0.09009857        0.09009857
```

```r
ugarchforecast(fitst, n.ahead=3)     # rugarch
```

```
*------------------------------------*
*       GARCH Model Forecast         *
*------------------------------------*
Model: sGARCH
Horizon: 3
Roll Steps: 0
Out of Sample: 0
0-roll forecast [T0=2021-10-20]:
      Series   Sigma
T+1 0.002287 0.05527
T+2 0.002287 0.07428
T+3 0.002287 0.08931
```

- Calcul des prévisions à la main:

- Calcul des prévisions à la main:
- On a $\widehat{\varepsilon}_T = 5.9002 \times 10^{-4}$, déterminons $\widehat{\sigma}^2_T(k)$, $k = 1, 2, 3$.

- Calcul des prévisions à la main:
- On a $\widehat{\varepsilon}_T = 5.9002 \times 10^{-4}$, déterminons $\widehat{\sigma}^2_T(k)$, $k = 1, 2, 3$.
- $\widehat{\sigma}^2_T(1) = a_0 + a_1 \varepsilon^2_h = 2.467 \times 10^{-3} + 1 \times 5.9002 \times 10^{-4} = 0.00305702$.

- Calcul des prévisions à la main:
- On a $\widehat{\varepsilon}_T = 5.9002 \times 10^{-4}$, déterminons $\widehat{\sigma}^2_T(k)$, $k = 1, 2, 3$.
- $\widehat{\sigma}^2_T(1) = a_0 + a_1 \varepsilon^2_h = 2.467 \times 10^{-3} + 1 \times 5.9002 \times 10^{-4} = 0.00305702$.
- $\widehat{\sigma}^2_T(2) = a_0 + a_1 \widehat{\sigma}^2_T(1) = 2.467 \times 10^{-3} + 1 \times 3.057 \times 10^{-3} = 0.005524$.

- Calcul des prévisions à la main:
- On a $\widehat{\varepsilon}_T = 5.9002 \times 10^{-4}$, déterminons $\widehat{\sigma}^2_T(k)$, $k = 1, 2, 3$.
- $\widehat{\sigma}^2_T(1) = a_0 + a_1 \varepsilon^2_h = 2.467 \times 10^{-3} + 1 \times 5.9002 \times 10^{-4} = 0.00305702$.
- $\widehat{\sigma}^2_T(2) = a_0 + a_1 \widehat{\sigma}^2_T(1) = 2.467 \times 10^{-3} + 1 \times 3.057 \times 10^{-3} = 0.005524$.
- $\widehat{\sigma}^2_T(3) = a_0 + a_1 \widehat{\sigma}^2_T(2) = 2.467 \times 10^{-3} + 1 \times 5.524 \times 10^{-3} = 0.007991$.

Le modèle GARCH

- Le modèle GARCH (*Generalized ARCH*) est proposé par (Bollerslev 1986).

Le modèle GARCH

- Le modèle GARCH (*Generalized ARCH*) est proposé par (Bollerslev 1986).
- $X_t \sim GARCH(p,q)$ si $X_t = \varepsilon_t = \sigma_t Z_t$ et $\sigma_t^2 = a_0 + \sum_{i=1}^{p} a_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2$ où $Z_t \overset{iid}{\sim} N(0,1)$, $a0 > 0$, $a_i \geq 0$, $\beta_j \geq 0$ et $\sum_{i=1}^{\max(p,q)} (a_i + \beta_j) < 1$.

Le modèle GARCH

- Le modèle GARCH (*Generalized ARCH*) est proposé par (Bollerslev 1986).
- $X_t \sim GARCH(p,q)$ si $X_t = \varepsilon_t = \sigma_t Z_t$ et $\sigma_t^2 = a_0 + \sum_{i=1}^{p} a_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2$ où $Z_t \overset{iid}{\sim} N(0,1)$, $a0 > 0$, $a_i \geq 0$, $\beta_j \geq 0$ et $\sum_{i=1}^{\max(p,q)} (a_i + \beta_j) < 1$.
- La dernière contrainte implique que la variance inconditionnelle de X_t est finie et que sa variance conditionnelle varie en fonction du temps.

Le modèle GARCH

- Le modèle GARCH (*Generalized ARCH*) est proposé par (Bollerslev 1986).
- $X_t \sim GARCH(p,q)$ si $X_t = \varepsilon_t = \sigma_t Z_t$ et
$\sigma_t^2 = a_0 + \sum_{i=1}^{p} a_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2$ où $Z_t \overset{iid}{\sim} N(0,1)$, $a0 > 0$,
$a_i \geq 0$, $\beta_j \geq 0$ et $\sum_{i=1}^{\max(p,q)} (a_i + \beta_j) < 1$.
- La dernière contrainte implique que la variance inconditionnelle de X_t est finie et que sa variance conditionnelle varie en fonction du temps.
- La prévision du modèle GARCH se fait similairement à un modèle ARMA. Soit $X_t = \sigma_t Z_t$ et $\sigma_t^2 = a_0 + a_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2$. On a alors:

Le modèle GARCH

- Le modèle GARCH (*Generalized ARCH*) est proposé par (Bollerslev 1986).

- $X_t \sim GARCH(p,q)$ si $X_t = \varepsilon_t = \sigma_t Z_t$ et
$$\sigma_t^2 = a_0 + \sum_{i=1}^{p} a_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 \text{ où } Z_t \overset{iid}{\sim} N(0,1),\ a0 > 0,$$
$$a_i \geq 0,\ \beta_j \geq 0 \text{ et } \sum_{i=1}^{\max(p,q)} (a_i + \beta_j) < 1.$$

- La dernière contrainte implique que la variance inconditionnelle de X_t est finie et que sa variance conditionnelle varie en fonction du temps.

- La prévision du modèle GARCH se fait similairement à un modèle ARMA. Soit $X_t = \sigma_t Z_t$ et $\sigma_t^2 = a_0 + a_1\varepsilon_{t-1}^2 + \beta_1\sigma_{t-1}^2$. On a alors:

- $\sigma_h(1) = a_0 + a_1\varepsilon_h^2 + \beta_1\sigma_h^2$,

Le modèle GARCH

- Le modèle GARCH (*Generalized ARCH*) est proposé par (Bollerslev 1986).
- $X_t \sim GARCH(p,q)$ si $X_t = \varepsilon_t = \sigma_t Z_t$ et $\sigma_t^2 = a_0 + \sum_{i=1}^{p} a_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2$ où $Z_t \overset{iid}{\sim} N(0,1)$, $a0 > 0$, $a_i \geq 0$, $\beta_j \geq 0$ et $\sum_{i=1}^{\max(p,q)} (a_i + \beta_j) < 1$.
- La dernière contrainte implique que la variance inconditionnelle de X_t est finie et que sa variance conditionnelle varie en fonction du temps.
- La prévision du modèle GARCH se fait similairement à un modèle ARMA. Soit $X_t = \sigma_t Z_t$ et $\sigma_t^2 = a_0 + a_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2$. On a alors:
- $\sigma_h(1) = a_0 + a_1 \varepsilon_h^2 + \beta_1 \sigma_h^2$,
- $\sigma_h(2) = a_0 + (a_1 + \beta_1)\sigma_h^2(1) + a_1 \sigma_h^2(1)(Z_{h+1}^2 - 1)$ et puisque

Le modèle GARCH

- Le modèle GARCH (*Generalized ARCH*) est proposé par (Bollerslev 1986).
- $X_t \sim GARCH(p,q)$ si $X_t = \varepsilon_t = \sigma_t Z_t$ et
$$\sigma_t^2 = a_0 + \sum_{i=1}^{p} a_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 \text{ où } Z_t \overset{iid}{\sim} N(0,1),\ a0 > 0,$$
$$a_i \geq 0,\ \beta_j \geq 0 \text{ et } \sum_{i=1}^{\max(p,q)} (a_i + \beta_j) < 1.$$
- La dernière contrainte implique que la variance inconditionnelle de X_t est finie et que sa variance conditionnelle varie en fonction du temps.
- La prévision du modèle GARCH se fait similairement à un modèle ARMA. Soit $X_t = \sigma_t Z_t$ et $\sigma_t^2 = a_0 + a_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2$. On a alors:
- $\sigma_h(1) = a_0 + a_1 \varepsilon_h^2 + \beta_1 \sigma_h^2$,
- $\sigma_h(2) = a_0 + (a_1 + \beta_1)\sigma_h^2(1) + a_1 \sigma_h^2(1)(Z_{h+1}^2 - 1)$ et puisque

Remarques:

- En générale, les modèles GARCH s'appliquent aux erreurs suite à un modèle linéaire (régression linéaire, ARMA).

Remarques:

- En générale, les modèles GARCH s'appliquent aux erreurs suite à un modèle linéaire (régression linéaire, ARMA).
- Comme dans le modèle ARCH, les erreurs Z_t dans le modèle GARCH peuvent être de loi Student ou GED.

Remarques:

- En générale, les modèles GARCH s'appliquent aux erreurs suite à un modèle linéaire (régression linéaire, ARMA).
- Comme dans le modèle ARCH, les erreurs Z_t dans le modèle GARCH peuvent être de loi Student ou GED.
- La détermination de l'ordre q du modèle GARCH se base sur la fonction d'auto-corrélation de X_t^2.

Remarques:

- En générale, les modèles GARCH s'appliquent aux erreurs suite à un modèle linéaire (régression linéaire, ARMA).
- Comme dans le modèle ARCH, les erreurs Z_t dans le modèle GARCH peuvent être de loi Student ou GED.
- La détermination de l'ordre q du modèle GARCH se base sur la fonction d'auto-corrélation de X_t^2.
- Estimons les rendements du BTC à l'aide d'un modèle GARCH(1,1).

```
garch11=garchFit(~garch(1,1),rt, trace = F)   #fGarch
garch11@fit$matcoef
```

```
            Estimate   Std. Error   t value     Pr(>|t|)
mu      2.182371e-03 6.334182e-04  3.445387 5.702419e-04
omega   6.937465e-05 1.063513e-05  6.523160 6.884138e-11
alpha1  1.339155e-01 1.523577e-02  8.789549 0.000000e+00
beta1   8.365670e-01 1.548134e-02 54.037112 0.000000e+00
```

```
spec11=ugarchspec(variance.model = list(garchOrder=c(1,1))
                  mean.model = list(armaOrder=c(0,0)))
Garch11=ugarchfit(spec11,rt)
Garch11@fit$matcoef
```

```
            Estimate   Std. Error   t value     Pr(>|t|)
mu      2.182108e-03 6.334382e-04  3.444864 5.713467e-04
omega   6.933686e-05 1.066024e-05  6.504249 7.808243e-11
alpha1  1.337894e-01 1.523686e-02  8.780639 0.000000e+00
beta1   8.366535e-01 1.551168e-02 53.937005 0.000000e+00
```

```
Title:
 GARCH Modelling
Call:
 garchFit(formula = ~garch(1, 1), data = rt, trace = F)
Mean and Variance Equation:
 data ~ garch(1, 1)
<environment: 0x0000000034cf1f20>
 [data = rt]
Conditional Distribution:
 norm
Coefficient(s):
        mu       omega      alpha1       beta1
2.1824e-03  6.9375e-05  1.3392e-01  8.3657e-01
Std. Errors:
 based on Hessian
Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
mu     2.182e-03   6.334e-04    3.445  0.00057 ***
omega  6.937e-05   1.064e-05    6.523 6.88e-11 ***
alpha1 1.339e-01   1.524e-02    8.790  < 2e-16 ***
```

```
show(Garch11)
```

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*
Conditional Variance Dynamics
-----------------------------------
GARCH Model : sGARCH(1,1)
Mean Model  : ARFIMA(0,0,0)
Distribution    : norm
Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.002182    0.000633   3.4449 0.000571
omega   0.000069    0.000011   6.5042 0.000000
alpha1  0.133789    0.015237   8.7806 0.000000
beta1   0.836654    0.015512  53.9370 0.000000
Robust Standard Errors:
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.002182    0.000729   2.9930 0.002762
```

```r
ugarchforecast(Garch11,n.ahead = 5)
```

```
*------------------------------------*
*       GARCH Model Forecast         *
*------------------------------------*
Model: sGARCH
Horizon: 5
Roll Steps: 0
Out of Sample: 0
0-roll forecast [T0=2021-10-20]:
      Series   Sigma
T+1 0.002182 0.03436
T+2 0.002182 0.03486
T+3 0.002182 0.03534
T+4 0.002182 0.03579
T+5 0.002182 0.03623
```

Références

Bollerslev, Tim. 1986. "Generalized Autoregressive Conditional Heteroskedasticity." *Journal of Econometrics* 31 (3): 307–27. https://doi.org/https://doi.org/10.1016/0304-4076(86)90063-1.

Engle, Robert F. 1982. "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation." *Econometrica* 50 (4): 987–1007.

Fisher, Thomas J., and Colin M. Gallagher. 2012. "New Weighted Portmanteau Statistics for Time Series Goodness of Fit Testing." *Journal of the American Statistical Association* 107 (498): 777–87. https://doi.org/10.1080/01621459.2012.688465.