

Modélisation GARCH de la volatilité

Chapitre 18 : Mise en œuvre sur R



Jaouad Madkour

Faculté d'Économie et de Gestion de Tanger · 27 août 2026

Avant de commencer

Étape 1 — Décrire et détecter

Étape 2 — Estimer

Étape 3 — Valider

Étape 4 — Prévoir et mesurer le risque

Synthèse

Avant de commencer

Le fichier rentabilites.csv

2 500 rentabilités quotidiennes **simulées**, livrées avec le cours. Le mécanisme générateur est un GJR-GARCH à innovations de Student : aucune mémoire en niveau, forte persistance de la volatilité, effet de levier.

Pourquoi simulées

Parce qu'on connaît la réponse. On peut vérifier que la démarche retrouve la persistance, l'asymétrie et l'épaisseur des queues qu'on y a mises — et rien d'autre. Sur données réelles, cette vérification est impossible.

```
install.packages(c("rugarch", "FinTS", "moments", "tseries"))  
library(rugarch)    # ugarchspec, ugarchfit, ugarchforecast  
library(FinTS)      # ArchTest  
library(moments)     # skewness, kurtosis
```

Étape 1 — Décrire et détecter

```
don <- read.csv("rentabilites.csv")
r <- don$rentabilite
T <- length(r)                                # 2500

c(moyenne = mean(r), ecart_type = sd(r),
  asymetrie = skewness(r), kurtosis = kurtosis(r))
```

moyenne	ecart_type	asymetrie	kurtosis
0.0200	1.1060	-0.2900	6.0500

Trois faits stylisés, en une ligne

Kurtosis de 6,05 contre 3 : queues épaisses. Asymétrie négative : les baisses sont plus marquées. Écart-type de 1,1 % par jour, soit 17,5 % annualisé.

L'équation de moyenne, d'abord

```
Box.test(r, lag = 20, type = "Ljung-Box")
```

X-squared = 17.033, df = 20, p-value = 0.6516

Conclusion

Non rejeté : aucune autocorrélation. L'équation de moyenne se réduit à une constante, exactement comme le cours ARMA l'annonçait.

On peut passer à la variance — et **seulement** parce que cette étape a été faite.


```
Box.test(r^2, lag = 20, type = "Ljung-Box")  
ArchTest(r, lags = 12)
```

X-squared = 968.13, df = 20, p-value < 2.2e-16

ARCH LM-test

Chi-squared = 274.53, df = 12, p-value < 2.2e-16

Le contraste, en deux nombres

$Q_{LB}(20)$ vaut 17,0 sur r_t et 968,1 sur r_t^2 . Non corrélées, mais très loin d'être indépendantes.

Étape 2 — Estimer

Le GARCH(1,1) gaussien

```
spec <- ugarchspec(  
  variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),  
  mean.model      = list(armaOrder = c(0, 0), include.mean = TRUE),  
  distribution.model = "norm")  
g_norm <- ugarchfit(spec, data = r)
```

	Estimate	Std. Error	t value	Pr(> t)
mu	0.057500	0.019800	2.904	0.003683
omega	0.020500	0.005100	4.020	0.000058
alpha1	0.100000	0.014200	7.042	0.000000
beta1	0.887300	0.015900	55.805	0.000000

LogLikelihood : -3543.2 Bayes (BIC) : 7117.8

Trois lectures immédiates

Persistance $\hat{\pi} = 0,100 + 0,887 = 0,987$. Demi-vie $\ln 0,5 / \ln 0,987 = 53$ jours.

Variance de long terme $0,0205 / (1 - 0,987) = 1,58$, soit une volatilité de 1,26 % par jour.

Tous les coefficients sont massivement significatifs – c'est la règle sur ce type de données.

Passer à la loi de Student

```
spec_t <- ugarchspec(  
  variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),  
  mean.model      = list(armaOrder = c(0, 0), include.mean = TRUE),  
  distribution.model = "std")          # std = Student  
g_t <- ugarchfit(spec_t, data = r)
```

	Estimate	Std. Error	t value
mu	0.044400	0.018900	2.349
omega	0.019500	0.006300	3.095
alpha1	0.099500	0.017100	5.819
beta1	0.889900	0.018200	48.896
shape	6.460000	0.812000	7.956

LogLikelihood : -3491.0

Bayes (BIC) : 7021.1

Le gain

Le BIC passe de 7 117,8 à 7 021,1 : **97 points** pour un paramètre. Le seuil usuel d'un écart significatif

```
signbias(g_t)
```

	t-value	prob	sig
Sign Bias	2.8140	0.0049	***
Negative Sign Bias	3.1620	0.0016	***
Joint Effect	18.4300	0.0004	***

Verdict

Rejet net : il reste de l'asymétrie que le GARCH symétrique ne capte pas. Il faut un GJR.

```
spec_gjr <- ugarchspec(  
  variance.model = list(model = "gjrGARCH", garchOrder = c(1, 1)),  
  mean.model      = list(armaOrder = c(0, 0), include.mean = TRUE),  
  distribution.model = "std")  
g_gjr <- ugarchfit(spec_gjr, data = r)
```

	Estimate	Std. Error	t value	Pr(> t)
mu	0.030700	0.018600	1.651	0.098800
omega	0.023200	0.006600	3.515	0.000440
alpha1	0.043400	0.015300	2.837	0.004556
beta1	0.886200	0.018400	48.163	0.000000
gamma1	0.115700	0.028100	4.117	0.000038
shape	6.620000	0.851000	7.779	0.000000

LogLikelihood : -3476.8 Bayes (BIC) : 7000.5

La lecture du coefficient d'asymétrie

$\hat{\gamma} = 0,116$, significatif à 4,1 écarts-types. Un choc négatif pèse $0,043 + 0,116 = 0,159$ contre 0,043 pour un choc positif : **près de quatre fois plus**.

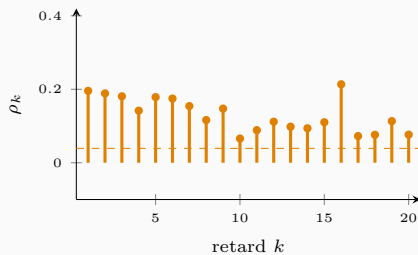
Persistence : $0,043 + 0,116/2 + 0,886 = 0,987$.

Étape 3 — Valider

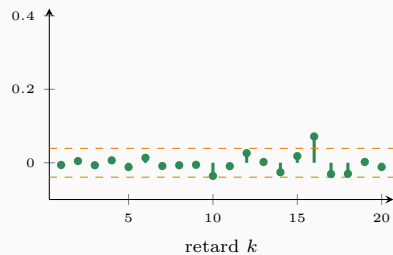
```
z <- residuals(g_gjr, standardize = TRUE)
Box.test(z, lag = 20, type = "Ljung-Box")
Box.test(z^2, lag = 20, type = "Ljung-Box")
```

```
z      : X-squared =  9.72, df = 20, p-value = 0.9738
z^2    : X-squared = 27.10, df = 20, p-value = 0.1320
```

avant : ACF de r_t^2



après : ACF de z_t^2

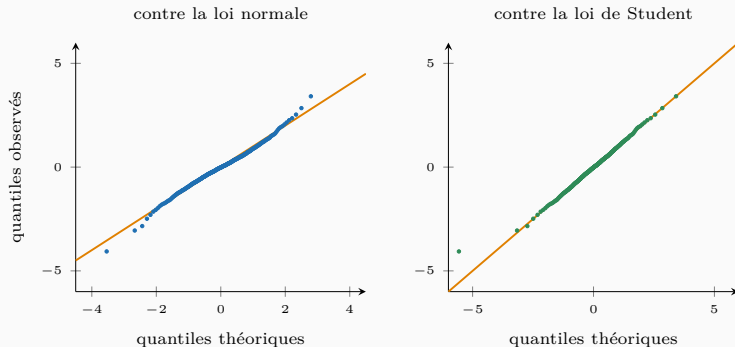


Le résultat central du cours

$Q_{LB}(20)$ sur les carrés passe de **968** à **27**. Le modèle a absorbé la totalité de la dépendance de la variance.

Le diagramme quantile-quantile

```
plot(g_gjr, which = 9) # QQ-plot des résidus standardisés
```



Lecture

Contre la normale, les points décrochent aux deux extrémités. Contre la Student estimée, ils suivent la droite jusque dans les queues : la loi choisie convient.

```
sapply(list(g_norm, g_t, g_gjr), infocriteria)[2, ] # ligne 2 = BIC
```

GARCH-norm	GARCH-t	GJR-t
7117.8	7021.1	7000.5

Étape 4 — Prévoir et mesurer le risque

```
p <- ugarchforecast(g_gjr, n.ahead = 60)
head(sigma(p), 3)
tail(sigma(p), 1)
```

```
[1] 0.7412 0.7534 0.7651
```

```
[60] 1.1893
```

Le retour à la moyenne, en trois nombres

La volatilité prévue part de 0,74 % — le marché est calme en fin d'échantillon — et remonte lentement vers 1,19 % à soixante jours.

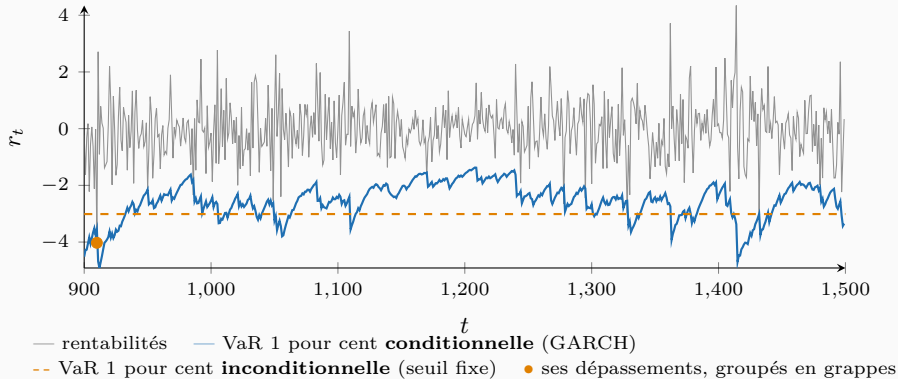
C'est exactement la courbe du chapitre 16 : convergence par le bas, à la vitesse $\pi = 0,987$.


```
q1  <- quantile(z, 0.01)           # quantile empirique des résidus
var_c <- coef(g_gjr)["mu"] + sigma(g_gjr) * q1
var_u <- quantile(r, 0.01)         # seuil fixe, pour comparaison

c(depassemements_conditionnelle = sum(r < var_c),
  depassemements_inconditionnelle = sum(r < var_u))
```

depassemements_conditionnelle	depassemements_inconditionnelle
25	25

La VaR conditionnelle par simulation filtrée (suite)



```
VaRTest(alpha = 0.01, actual = r, VaR = var_c)$uc.LRstat # Kupiec  
VaRTest(alpha = 0.01, actual = r, VaR = var_c)$cc.LRstat # Christoffersen
```

Le résultat qui conclut le cours

Mesure	Dépassements	LR_{uc}	LR_{ind}	Verdict
conditionnelle	25	0,00	1,38	acceptée
inconditionnelle	25	0,00	16,22	rejetée

Même comptage, verdict opposé

Le seuil fixe a exactement le bon nombre de dépassements — et les groupe en grappes, dans les périodes de crise. C'est ce que le test d'indépendance détecte, et c'est ce que le GARCH corrige.

Synthèse

L'essentiel du chapitre 18

- `Box.test` sur r puis sur r^2 : 17,0 contre 968,1. Non corrélées, non indépendantes.
- `ugarchspec` + `ugarchfit` : trois modèles emboîtés, à comparer par le BIC.
- Loi de Student : —97 points de BIC. Asymétrie GJR : —21 de plus.
- `signbias` teste l'effet de levier; ici rejet net, $\hat{\gamma} = 0,116$.
- Validation : $Q_{LB}(20)$ sur z^2 passe de 968 à 27.
- `ugarchforecast` donne le retour à la moyenne; `VaRTest` fait le backtesting.

Vers le chapitre 19

La même série, la même démarche, les mêmes chiffres — en Python. Non par redondance, mais pour vérifier que la méthode ne doit rien à l'outil.