제13강: 변동성 모형

금융 통계 및 시계열 분석

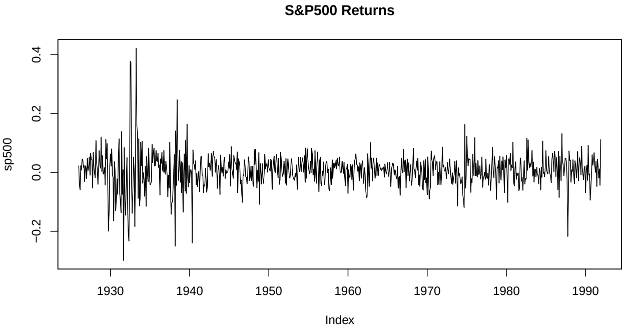
TRADE INFORMATIX

2014년 2월 21일

목차

S&P500 수익률

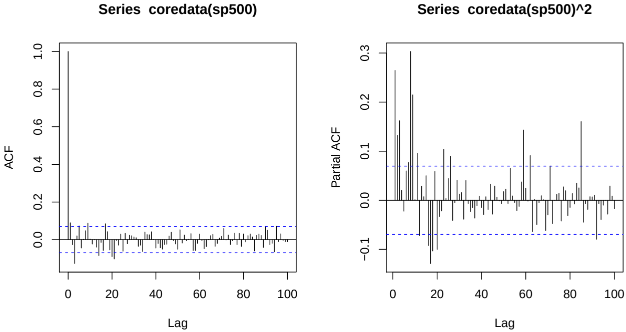
```
> library("FinTS")
> data(sp500)
> plot(sp500, type='l', main="S&P500 Returns")
```



수익률 시계열의 상관관계 특성

- ❑ 수익률 자체는 auto-correlation이 없다.
- ❑ 수익률의 크기는 강한 auto-correlation이 있다.

```
> layout(matrix(1:2, 1, 2, byrow=TRUE))
> acf(coredata(sp500), lag.max=100)
> pacf(coredata(sp500)**2, lag.max=100)
```



수익률 시계열의 변동성 특성

1. 변동성 클러스터링 (clustering)
   - ❑ 일단 변동성이 높아지면 일정기간동안 변동성이 높게 유지된다.
2. 변동성 변화는 정상 (stationary) 특성을 가진다
   - ❑ 변동성의 변동성 (volatility of volatility) 는 일정하게 유지된다.
3. 레버리지 효과 (leverage effect)
   - ❑ 변동성이 증가할 때는 빠르게 증가하고 감소할 때는 천천히 감소한다.

ARCH 모형

- Conditional Heteroskedasticity 특성
  - 변동성이 이전 시간까지의 변동성 정보에 의존한다.

$$
\begin{aligned}
E[r_t] &= \mu \\
\sigma_t^2 &= \mathrm{Var}(r_t) = E[(r_t - \mu)^2 | F_{t-1}]
\end{aligned}
$$

- AutoRegressive Conditional Heteroskedasticity (ARCH)
  - Engle 제안 (1982)
  - 변동성이 이전 시간까지의 변동성에 의존하는 AR모형과 유사
  - 변동성은 이전 시간까지의 실현 수익률 크기 a_t^2 에 의존

$$
\begin{aligned}
a_t &= r_t - \mu = \sigma_t e_t \\
\sigma_t^2 &= \alpha_0 + \alpha_1 a_{t-1}^2 + \ldots + \alpha_m a_{m-1}^2
\end{aligned}
$$

  - e_t : 평균 0, 분산 1인 I.I.D 확률변수. 보통 정규분포 가정
  - α_i : 양수 (positive) 인 계수

ARCH 효과 테스트 1 : Ljung-Box 테스트

- Ljung-Box 테스트
  - $\{a_t^2\}$ 시계열의 auto-correlation이 m차까지 0인지 검정

$$Q(m) = n\,(n+2)\sum_{k=1}^{h}\frac{\hat{\rho}_k^2}{n-k}$$

- `Box.test(x, lag, type)`
  - `x` : 시계열 자료
  - `lag` : 검정 차수
  - `type` : 검정 방법 `"Box-Pierce"`, `"Ljung-Box"`

```
> Box.test(sp500**2, lag=20, type="Ljung-Box")

Box-Ljung test

data:  sp500^2
X-squared = 508.0743, df = 20, p-value < 2.2e-16
```

ARCH 효과 테스트 2 : Lagrange Multiplier 테스트

- Lagrange Multiplier 테스트
  - a_t^2 를 $a_{t-1}^2, \ldots, a_{t-m}^2$ 으로 회귀분석한 계수가 0인지 검정

$$a_t^2 \quad = \quad \alpha_0 + \alpha_1 a_{t-1}^2 + \ldots + \alpha_m a_{m-1}^2 + e_t$$

- `ArchTest(x, lags, demean` (FinTS 패키지)
  - `x` : 시계열 자료
  - `lag` : 검정 차수
  - `demean` : `TRUE`이면 샘플평균 제거후 분석

```
> ArchTest(sp500)

ARCH LM-test; Null hypothesis: no ARCH effects

data:  sp500
Chi-squared = 193.7156, df = 12, p-value < 2.2e-16
```

ARCH(1) 모형의 특성

❑ ARCH(1)

$$\begin{aligned} a_t &= \sigma_t e_t \\ \sigma_t^2 &= \alpha_0 + \alpha_1 a_{t-1}^2 \end{aligned}$$

❑ 무조건부 평균(unconditional mean)은 0

$$E[a_t] = E[E[a_t|F_{t-1}]] = E[\sigma E[e_t]] = 0$$

❑ 무조건부 분산(unconditional variance)은 $\dfrac{\alpha_0}{1-\alpha_1}$

$$\begin{aligned} \text{Var}[a_t] &= E[a_t^2] = E[E[a_t^2|F_{t-1}]] = E[\alpha_0 + \alpha_1 a_{t-1}^2] \\ &= \alpha_0 + \alpha_1 E[a_{t-1}^2] = \alpha_0 + \alpha_1 \text{Var}[a_{t-1}] \end{aligned}$$

❑ Excess Kurtosis

$$\frac{E[a_t^4]}{[E[a_t^2]]^2} = \frac{3(1+\alpha_1)}{(1-\alpha_1)(1-3\alpha_1^2)} > 3$$

❑ 추가적 제한조건

$$1 - 3\alpha_1^2 > 0$$

ARCH 모형의 장단점

- 장점
  - 단순한 수학적 모형
  - 정상 신호 모형
  - 변동성 clustering 표현 가능
  - fai-tail 표현 가능
- 단점
  - 레버리지 효과 표현 불가능
  - 계수 $\{\alpha_i\}$에 제한이 많음
  - 변동성 쇼크에 대한 반응속도가 느리기때문에 변동성 과대평가

ARCH 모형 사용 방법

- 차수 결정
  - ARCH(p) 모형의 차수 결정
  - a_t^2 의 PACF 이용
- 계수 추정
  - MLE(Maximum Likelihood Estimation) 사용
  - 보통 e_t 가 Normal 분포인 경우 가정
- 모형 검증
  - 올바른 모형인 경우 표준잔차 a_t/σ_t 는 iid
  - Ljung-Box test 사용
- 예측
  - 모형 수식에 의한 회귀적 예측 가능

GARCH 모형

- Generalized AutoRegressive Conditional Heteroskedasticity
- Bollerslev 제안 (1986)
- 변동성이 이전 시간까지의 변동성 및 실현변동성 모두에 의존하는 AR모형
- ARCH 모형보다 적은 차수로 수익률 시계열 모형 가능
- GARCH(m, s) 모형

$$
\begin{aligned}
a_t &= \sigma_t e_t \\
\sigma_t^2 &= \alpha_0 + \sum_{i=1}^{m} \alpha_i a_{t-i}^2 + \sum_{j=1}^{s} \beta_j \sigma_{t-j}^2
\end{aligned}
$$

  - $\alpha_0, \alpha_i, \beta_i$: 양수 (positive) 인 계수
  - 계수 조건

$$
\sum_{i}^{\max(m,s)} (\alpha_i + \beta_i) < 1
$$

- GARCH(m, 0) = ARCH(m)

GARCH(1,1) 모형 특징

❑ GARCH(1, 1) 모형

$$\sigma_t^2 \quad = \quad \alpha_0 + \alpha_1 a_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

$$\alpha_0 > 0,\ 0 < \alpha_i > 1,\ 0 < \beta_i < 1,\ \alpha_i + \beta_i < 1$$

❑ Excess Kurtosis

$$\frac{E[a_t^4]}{[E[a_t^2]]^2} = \frac{3(1-(\alpha_1+\beta_1)^2)}{1-(\alpha_1+\beta_1)^2 - 2\alpha^2} > 3$$

❑ 추가적 제한조건

$$1 - (\alpha_1 + \beta_1)^2 - 2\alpha^2 > 0$$

GARCH(1,1) 모형 예측

❑ 1-스텝 예측

$$\begin{aligned} \sigma_{t+1}^2 &= \alpha_0 + \alpha_1 a_t^2 + \beta_1 \sigma_t^2 \\ E[\sigma_{t+1}^2 | F_t] &= \alpha_0 + \alpha_1 a_t^2 + \beta_1 \sigma_t^2 \end{aligned}$$

❑ 2-스텝 예측

$$\begin{aligned} \sigma_{t+2}^2 &= \alpha_0 + \alpha_1 a_{t+1}^2 + \beta_1 \sigma_{t+1}^2 \\ &= \alpha_0 + (\alpha_1 + \beta_1)\sigma_{t+1}^2 + \alpha_1 \sigma_t^2 (e_{t+1}^2 - 1) \\ E[\sigma_{t+2}^2 | F_t] &= \alpha_0 + (\alpha_1 + \beta_1)\sigma_{t+1}^2 \end{aligned}$$

❑ l-스텝 예측

$$\begin{aligned} E[\sigma_{t+l}^2 | F_t] &= \alpha_0 + (\alpha_1 + \beta_1)\sigma_{t+l-1}^2 \\ &= \frac{\alpha_0(1 - (\alpha_1 + \beta_1)^{l-1})}{1 - (\alpha_1 + \beta_1)} + (\alpha_1 + \beta_1)^{l-1} \sigma_{t+1} \\ E[\sigma_{t+l}^2 | F_t] &\to \frac{\alpha_0}{1 - (\alpha_1 + \beta_1)} \quad \text{as l} \to \infty \end{aligned}$$

ARCH/GARCH 명령

- fGarch 패키지
- `garchSpec`
  - 시뮬레이션을 위한 GARCH 모형 정의
- `garchSim`
  - GARCH 모형 시뮬레이션
- `garchFit`
  - GARCH 모형 파라미터 추정
- GARCH(m, s) 모형 정의에서 차수 s=0 이면 ARCH 모형

ARCH/GARCH 모형 정의 명령

- `garchSpec(model, cond.dist, rseed)`
- 평균 프로세스와 변동성 프로세스 동시 정의

$$
\begin{aligned}
x_t &= \mu + \sum_{i=1}^{p} a_i x_{t-i} + \sum_{i=0}^{q} b_i a_{t-i} \\
a_t &= \sigma_t e_t \\
\sigma_t^2 &= \alpha_0 + \sum_{i=1}^{m} \alpha_i a_{t-i}^2 + \sum_{j=1}^{s} \beta_j \sigma_{t-j}^2
\end{aligned}
$$

- ▶ `model=list(omega, alpha, beta, mu, ar, ma)`
  - `mu, ar, ma` : 평균 프로세스 계수
  - `omega=0.8, alpha=0.1, beta=0.8` : 변동성 프로세스 계수 ($\omega = \alpha_0$)
- ▶ `cond.dist` : 표준잔차 이노베이션 확률변수 정의
  - `norm` : Normal 분포
  - `ged` : Generalized Error 분포
  - `std` : Student-t 분포
- ▶ `rseed` : 랜덤 시드

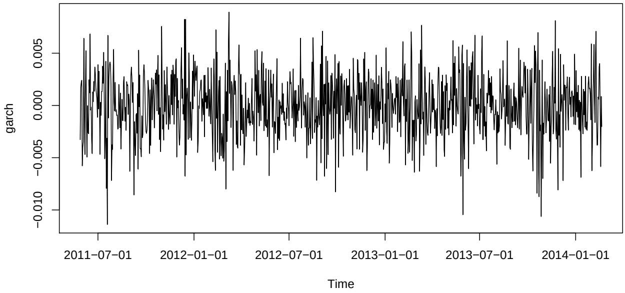
ARCH/GARCH 모형 정의의 예

- ARCH(2)
  - `garchSpec(model = list(alpha = c(0.2, 0.4), beta = 0))`
- AR(1)-ARCH(2)
  - `garchSpec(model = list(ar = 0.5, alpha = c(0.3, 0.4), beta = 0))`
- GARCH(1,1)
  - `garchSpec(model = list(alpha = 0.2, beta = 0.7))`
- ARMA(1,2)-GARCH(1,1)
  - `garchSpec(model = list(ar = 0.5, ma = c(0.3, -0.3)))`

ARCH/GARCH 모형 시뮬레이션 명령

- `garchSim(spec, n)`
  - `spec` : `garchSpec` 명령으로 정의된 모형
  - `n` : 시뮬레이션 수

```
> library("fGarch")
> spec <- garchSpec(model = list())
> x <- garchSim(spec, n=1000)
> plot(x)
```



ARCH/GARCH 모형 추정 명령

- `garchFit(formula, data, cond.dist, trace)`
- 입력 변수
  - `formula` : ARCH/GARCH 모형 구조를 정의하는 formula
  - `cond.dist` : 표준잔차 이노베이션 확률변수 정의
  - `trace` : `FALSE` 이면 최적화 과정을 표시하지 않음
- 출력 변수 : S4 class
  - `data` : 원본 자료
  - `fit` : 추정된 파라메터
  - `residuals` : 잔차
  - `fitted` : 추정된 파라미터로 fitted 된 시계열

ARCH/GARCH 모형 추정 예 1

```
> x <- as.vector(garchSim(garchSpec(rseed=1985), n=200)[,1])
> m <- garchFit(~garch(1,1), data=x, trace=FALSE)
> summary(m)

Title:
 GARCH Modelling

Call:
 garchFit(formula = ~garch(1, 1), data = x, trace = FALSE)

Mean and Variance Equation:
 data ~ garch(1, 1)
<environment: 0x1932e5c0>
 [data = x]

Conditional Distribution:
 norm

Coefficient(s):
        mu       omega      alpha1       beta1
3.5418e-05  1.0819e-06  8.8855e-02  8.1200e-01

Std. Errors:
 based on Hessian

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
mu     3.542e-05   2.183e-04    0.162    0.871
omega  1.082e-06   1.051e-06    1.030    0.303
alpha1 8.885e-02   5.450e-02    1.630    0.103
beta1  8.120e-01   1.242e-01    6.538 6.25e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

ARCH/GARCH 모형 추정 예 1 (계속)

```
Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  1.114092  0.5728988
 Shapiro-Wilk Test  R    W      0.9932315 0.4910797
 Ljung-Box Test     R    Q(10)  7.30396   0.6964714
 Ljung-Box Test     R    Q(15)  8.712828  0.8920477
 Ljung-Box Test     R    Q(20)  9.766983  0.972203
 Ljung-Box Test     R^2  Q(10)  11.88456  0.2928571
 Ljung-Box Test     R^2  Q(15)  14.93927  0.4558004
 Ljung-Box Test     R^2  Q(20)  20.08938  0.4523512
 LM Arch Test       R    TR^2   11.57234  0.480607

Information Criterion Statistics:
      AIC       BIC       SIC      HQIC
-8.579494 -8.513527 -8.580273 -8.552798
```

ARCH/GARCH 모형 추정 예 2-1

```
> (m1 <- garchFit(~arma(3,0)+garch(1,1), data=sp500, trace=FALSE))

Title:
 GARCH Modelling

Call:
 garchFit(formula = ~arma(3, 0) + garch(1, 1), data = sp500, trace = FALSE)

Mean and Variance Equation:
 data ~ arma(3, 0) + garch(1, 1)
<environment: 0x19415a28>
 [data = sp500]

Conditional Distribution:
 norm

Coefficient(s):
         mu          ar1          ar2          ar3
 7.7078e-03   3.1969e-02  -3.0262e-02  -1.0650e-02
      omega       alpha1        beta1
 7.9746e-05   1.2424e-01   8.5302e-01

Std. Errors:
 based on Hessian

Error Analysis:
         Estimate  Std. Error  t value Pr(>|t|)
mu      7.708e-03   1.607e-03    4.798 1.61e-06 ***
ar1     3.197e-02   3.837e-02    0.833  0.40471
ar2    -3.026e-02   3.841e-02   -0.788  0.43075
ar3    -1.065e-02   3.756e-02   -0.284  0.77675
omega   7.975e-05   2.810e-05    2.838  0.00454 **
alpha1  1.242e-01   2.247e-02    5.529 3.22e-08 ***
beta1   8.530e-01   2.183e-02   39.076  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log Likelihood:
 1272.179    normalized:  1.606287

Description:
 Fri Feb 21 09:41:28 2014 by user:
```

ARCH/GARCH 모형 추정 예 2-2

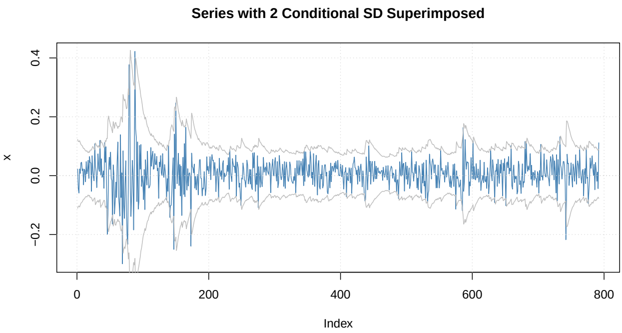
```
> (m2 <- garchFit(~garch(1,1), data=sp500, trace=FALSE))

Title:
 GARCH Modelling

Call:
 garchFit(formula = ~garch(1, 1), data = sp500, trace = FALSE)

Mean and Variance Equation:
 data ~ garch(1, 1)
<environment: 0x19662f08>
 [data = sp500]

Conditional Distribution:
 norm

Coefficient(s):
        mu       omega      alpha1       beta1
7.4497e-03  8.0615e-05  1.2198e-01  8.5436e-01

Std. Errors:
 based on Hessian

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
mu     7.450e-03   1.538e-03    4.845 1.27e-06 ***
omega  8.061e-05   2.833e-05    2.845  0.00444 **
alpha1 1.220e-01   2.202e-02    5.540 3.02e-08 ***
beta1  8.544e-01   2.175e-02   39.276  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log Likelihood:
 1269.455    normalized:  1.602848

Description:
 Fri Feb 21 09:41:28 2014 by user:
```

ARCH/GARCH 모형 추정 결과 Plot

- fGARCH S4 클래스 명령
- `plot(x, which)`
- `which` 인수 : 분석 결과 선택
  1. Time SeriesPlot
  2. Conditional Standard Deviation Plot
  3. Series Plot with 2 Conditional SD Superimposed
  4. Autocorrelation function Plot of Observations
  5. Autocorrelation function Plot of Squared Observations
  6. Cross Correlation Plot
  7. Residuals Plot
  8. Conditional Standard Deviations Plot
  9. Standardized Residuals Plot
  10. ACF Plot of Standardized Residuals
  11. ACF Plot of Squared Standardized Residuals
  12. Cross Correlation Plot between r^2 and r
  13. Quantile-Quantile Plot of Standardized Residuals

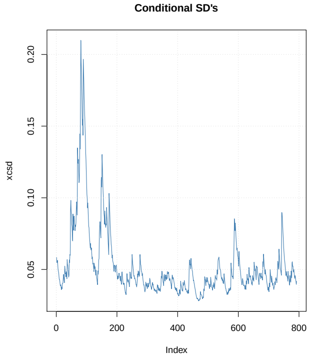
ARCH/GARCH 모형 결과 Plot 예 1-1

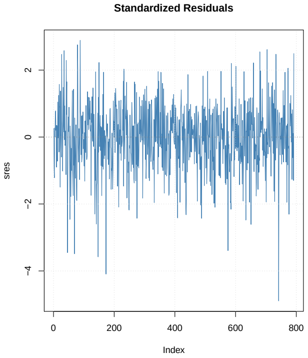
```
> plot(m2, which=3)
```



ARCH/GARCH 모형 결과 Plot 예 1-2

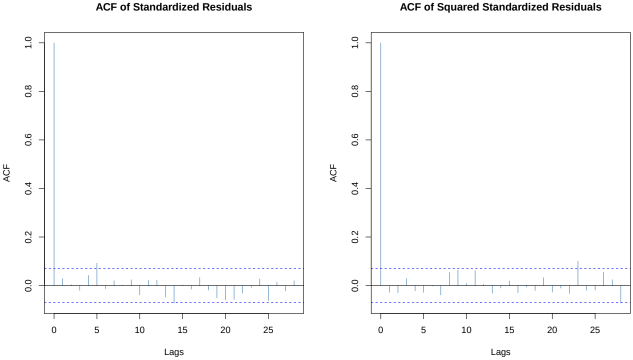
```
> layout(matrix(c(1,2), 1, 2, byrow=TRUE))
> plot(m2, which=8)
> plot(m2, which=9)
```





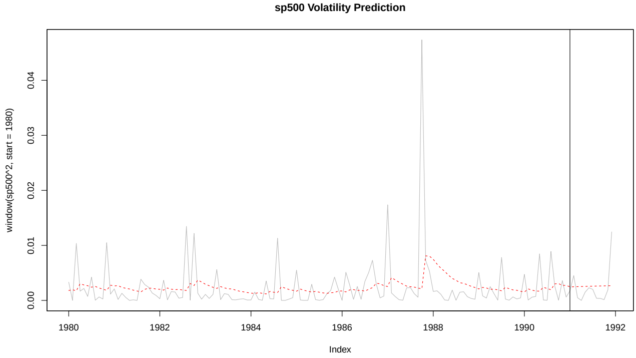
ARCH/GARCH 모형 결과 Plot 예 1-3

```
> layout(matrix(c(1,2), 1, 2, byrow=TRUE))
> plot(m2, which=10)
> plot(m2, which=11)
```



ARCH/GARCH 모형 예측 예

```
> N <- length(sp500)
> sp500.2 <- sp500[-(N-12+1:N)]
> m3 <- garchFit(~garch(1,1), data=sp500.2, trace=FALSE)
> p <- predict(m3, 12)
> sp500.p <- ts(c(m3@sigma.t**2, (p$standardDeviation)**2), start=1926, frequency=12)
> plot(window(sp500**2, start=1980), type="l", col="gray", main="sp500 Volatility Prediction")
> lines(window(sp500.p, start=1980), lty=2, col="red"); abline(v=1991)
```



GARCH 변형 모형

- IGARCH : Integrated GARCH
  - GARCH ARMA 계수가 unit root를 가지는 모형
- GARCH-M : GARCH in the mean
  - 시계열 자체가 변동성에 의존
- EGARCH : Exponential GARCH
  - 레버리지 효과 부가
- TGARCH : Threshold GARCH
  - 레버리지 효과 부가

rugarch 패키지: GARCH 변형 모형

- ❑ rugarch 패키지
- ❑ R Univariate GARCH Model
- ❑ ARFIMAX 평균 프로세스 지원
- ❑ 다음과 같은 GARCH-변형 변동성 프로세스 지원
  - sGARCH
  - eGARCH
  - gjrGARCH
  - apARCH
  - iGARCH
  - csGARCH
  - TGARCH
  - AVGARCH
  - NGARCH
  - NAGARCH
  - APARCH

rugarch 모형 추정 명령

- `ugarchspec(variance.model, mean.model, distribution.model)`
  - `variance.model=list(model, garchOrder, submodel)` : 변동성 프로세스 모형
  - `model` : `"sGARCH", "fGARCH", "eGARCH", "gjrGARCH", "apARCH", "iGARCH", "csGARCH"`
  - `garchOrder=c(1,1)` : 변동성 프로세스 차수
  - `submodel` : `model="fGARCH"`인 경우 `"GARCH", "TGARCH", "AVGARCH", "NGARCH", "NAGARCH", "APARCH","GJRGARCH" and "ALLGARCH"`
  - `mean.model=list(armaOrder, include.mean, garchm)` : `FALSE` 이면 최적화 과정을 표시하지 않음
  - `armaOrder=c(0,0)` : 평균 ARMA 프로세스 차수
  - `include.mean` : constant mean 추가 여부
  - `garchm` : GARCH-in-the-mean 추가 여부
- `ugarchfit(spec, data)`
  - `spec` : `ugarchspec`로 정의된 모형
  - `data` : 시계열 자료

IGARCH : Integrated GARCH

- GARCH ARMA 계수가 unit root를 가지는 모형
- 변동성 이노베이션의 영향이 영구적
- IGARCH(1,1)

$$\sigma_t^2 \quad = \quad \alpha_0 + \beta_1 \sigma_{t-1}^2 + (1 - \beta_1) a_{t-1}^2$$

IGARCH : Integrated GARCH

```
> library(rugarch)
> spec.igarch <- ugarchspec(variance.model=list(model="iGARCH", garchOrder=c(1,1)),
+   mean.model=list(armaOrder=c(0, 0), include.mean=TRUE, arfima=FALSE))
> mod.fit.igarch <- ugarchfit(spec=spec.igarch, data=as.numeric(sp500))
> show(mod.fit.igarch)

*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model : iGARCH(1,1)
Mean Model  : ARFIMA(0,0,0)
Distribution : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.007417    0.001525   4.8621 0.000001
omega   0.000051    0.000018   2.9238 0.003458
alpha1  0.142951    0.021443   6.6667 0.000000
beta1   0.857049          NA       NA       NA

Robust Standard Errors:
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.007417    0.001587   4.6726 0.000003
omega   0.000051    0.000019   2.6913 0.007118
alpha1  0.142951    0.024978   5.7230 0.000000
beta1   0.857049          NA       NA       NA

LogLikelihood : 1268.238

Information Criteria
------------------------------------

Akaike       -3.1950
Bayes        -3.1773
Shibata      -3.1951
Hannan-Quinn -3.1882

Q-Statistics on Standardized Residuals
------------------------------------
              statistic p-value
```

GARCH-M : GARCH in the mean

- 시계열 자체가 변동성에 의존
- 수익률 시계열은 리스크 프리미엄을 가짐

$$
\begin{aligned}
x_t &= \mu + c\sigma_t^2 + a_t \\
a_t &= \sigma_t e_t \\
\sigma_t^2 &= \alpha_0 + \sum_{i=1}^{m} \alpha_i a_{t-i}^2 + \sum_{j=1}^{s} \beta_j \sigma_{t-j}^2
\end{aligned}
$$

- 시계열 자체가 변동성에 의존

GARCH-M : GARCH in the mean

```
> spec.garchm <- ugarchspec(variance.model=list(model="sGARCH", garchOrder=c(1,1)),
+   mean.model=list(armaOrder=c(0, 0), include.mean=TRUE, garchm=TRUE))
> mod.fit.garchm <- ugarchfit(spec=spec.garchm, data=as.numeric(sp500))
> show(mod.fit.garchm)

*---------------------------------*
*          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       0.00745    0.001538   4.8450 0.000001
omega    0.00008    0.000028   2.8287 0.004674
alpha1   0.12226    0.022102   5.5315 0.000000
beta1    0.85435    0.021811  39.1708 0.000000

Robust Standard Errors:
        Estimate  Std. Error  t value Pr(>|t|)
mu       0.00745    0.001717   4.3382 0.000014
omega    0.00008    0.000034   2.3921 0.016754
alpha1   0.12226    0.028162   4.3412 0.000014
beta1    0.85435    0.026420  32.3374 0.000000

LogLikelihood : 1269.455

Information Criteria
------------------------------------

Akaike       -3.1956
Bayes        -3.1720
Shibata      -3.1956
Hannan-Quinn -3.1865

Q-Statistics on Standardized Residuals
------------------------------------
              statistic p-value
Lag[1]            0.627  0.4285
```

EGARCH : Exponential GARCH

- ❑ 분산(변동성)의 log값 모형
- ❑ 레버리지 효과 부가
- ❑ 변동성 이노베이션 값이 양수일 때와 음수일 때 비중 변화

$$\log(\sigma_t^2) \quad = \quad \alpha_0 + \frac{\sum_{j=1}^{s} \beta_j B^{j-1}}{\sum_{i=1}^{m} \alpha_i B^i} g(e_{t-1})$$

- ❑ 비중함수

$$g(e_t) = \begin{cases} (\theta + \gamma)e_t - \gamma E[|e_t|] & \text{if } e_t \geq 0 \\ (\theta - \gamma)e_t - \gamma E[|e_t|] & \text{if } e_t < 0 \end{cases}$$

EGARCH : Exponential GARCH

```
> spec.egarch <- ugarchspec(variance.model=list(model="eGARCH", garchOrder=c(1,1)),
+      mean.model=list(armaOrder=c(0, 0), include.mean=TRUE, arfima=FALSE))
> mod.fit.egarch <- ugarchfit(spec=spec.egarch, data=as.numeric(sp500))
> show(mod.fit.egarch)

*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model : eGARCH(1,1)
Mean Model : ARFIMA(0,0,0)
Distribution : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.006865    0.001554   4.4173 0.000010
omega  -0.153512    0.057027  -2.6919 0.007104
alpha1 -0.058325    0.022518  -2.5902 0.009592
beta1   0.973450    0.009362 103.9758 0.000000
gamma1  0.226981    0.033965   6.6828 0.000000

Robust Standard Errors:
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.006865    0.001702   4.0328 0.000055
omega  -0.153512    0.075748  -2.0266 0.042703
alpha1 -0.058325    0.030653  -1.9028 0.057070
beta1   0.973450    0.011912  81.7234 0.000000
gamma1  0.226981    0.044380   5.1145 0.000000

LogLikelihood : 1271.916

Information Criteria
------------------------------------

Akaike       -3.1993
Bayes        -3.1698
Shibata      -3.1994
Hannan-Quinn -3.1879

Q-Statistics on Standardized Residuals
```

TGARCH : Threshold GARCH

- 레버리지 효과 부가
- 실현 수익률 값이 양수일 때와 음수일 때 비중 변화

$$
\begin{aligned}
a_t &= \sigma_t e_t \\
\sigma_t^2 &= \alpha_0 + \sum_{i=1}^{m} (\alpha_i + \gamma_i N_{t-i}) a_{t-i}^2 + \sum_{j=1}^{s} \beta_j \sigma_{t-j}^2
\end{aligned}
$$

$$
N_{t-i} = \begin{cases} 1 & \text{if } a_{t-i} < 0 \\ 0 & \text{if } a_{t-i} \geq 0 \end{cases}
$$

TGARCH : Threshold GARCH

```
> spec.tgarch <- ugarchspec(variance.model=list(model="fGARCH", submodel="TGARCH", garchOrder=c(1,1)),
+      mean.model=list(armaOrder=c(0, 0), include.mean=TRUE, arfima=FALSE))
> mod.fit.tgarch <- ugarchfit(spec=spec.tgarch, data=as.numeric(sp500))
> show(mod.fit.tgarch)

*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model : fGARCH(1,1)
fGARCH Sub-Model : TGARCH
Mean Model : ARFIMA(0,0,0)
Distribution : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.006789    0.001568   4.3291 0.000015
omega   0.001773    0.000617   2.8726 0.004071
alpha1  0.122429    0.018760   6.5260 0.000000
beta1   0.871693    0.019076  45.6968 0.000000
eta11   0.336830    0.120995   2.7838 0.005372

Robust Standard Errors:
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.006789    0.001712   3.9646 0.000074
omega   0.001773    0.000764   2.3225 0.020206
alpha1  0.122429    0.023733   5.1586 0.000000
beta1   0.871693    0.024652  35.3600 0.000000
eta11   0.336830    0.166260   2.0259 0.042772

LogLikelihood : 1274.897

Information Criteria
------------------------------------

Akaike       -3.2068
Bayes        -3.1773
Shibata      -3.2069
Hannan-Quinn -3.1955
```

Stochastic Volatility 모형

❑ 변동성 프로세스의 이노베이션이 독립적 확률변수

$$\begin{aligned} a_t &= \sigma_t e_t \\ 1 - \sum_{i=1}^{m} \alpha_i B^i \log(\sigma_t^2) &= \alpha_0 + v_t \end{aligned}$$

❑ 모델 자유도가 증가하는 대신 파라미터 추정이 어려움

❑ Kalman Filter / MCMC (Monte Carlo Markov Chain) 사용

Extreme Value 모형

- daily 종가뿐이 아니라 시고저종(OHLC) 값을 모두 사용하여 변동성 추정
- Garman-Klass 변동성 추정

$$
\begin{aligned}
\sigma^2_{0,t} &= (C_t - C_{t-1})^2 \\
\sigma^2_{1,t} &= \frac{(O_t - C_{t-1})^2}{2f} + \frac{(C_t - O_t)^2}{2(1-f)} \\
\sigma^2_{2,t} &= \frac{(H_t - L_t)^2}{4\log 2} \\
\sigma^2_{3,t} &= 0.17\frac{(O_t - C_{t-1})^2}{f} + 0.83\frac{(H_t - L_t)^2}{(1-f)4\log 2} \\
\sigma^2_{5,t} &= 0.5(H_t - L_t)^2 - (2\log 2 - 1)(C_t - O_t)^2 \\
\sigma^2_{6,t} &= 0.12\frac{(O_t - C_{t-1})^2}{f} + 0.88\frac{\sigma^2_{5,t}}{1-f}
\end{aligned}
$$

Extreme Value 모형 (계속)

❑ Yang-Zhang 변동성 추정

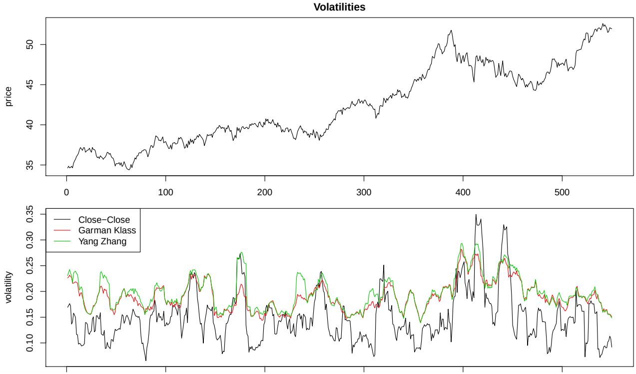
$$
\begin{aligned}
o_t &= \log O_t - \log C_{t-1} \\
c_t &= \log C_t - \log O_t \\
u_t &= \log H_t - \log O_t \\
d_t &= \log L_t - \log O_t \\
\sigma_{yz}^2 &= \sigma_o^2 + k\sigma_c^2 + (1-k)\sigma_{rs}^2 \\
\sigma_o^2 &= \frac{1}{n-1}\sum_{t=1}^{n}(o_t - \bar{o}_t)^2 \\
\sigma_c^2 &= \frac{1}{n-1}\sum_{t=1}^{n}(c_t - \bar{c}_t)^2 \\
\sigma_{rs}^2 &= \frac{1}{n-1}\sum_{t=1}^{n}(u_t(u_t - c_t) + d_t(d_t - c_t)) \\
k &= \frac{0.34}{1.34 + (n+1)/(n-1)}
\end{aligned}
$$

R에서 Extreme Value 모형 사용

- `TTR` 패키지 사용
- `volatility(OHLC, n=10, calc="close", N=260, ...)`
  - `OHLC` : Open-High-Low-Close dataframe
  - `n` : 변동성 계산용 윈도우 크기
  - `calc` : 변동성 계산 방법
    - `"close"` : Close-to-Close
    - `"garman.klass"` : Garman Klass
    - `"rogers.satchell"` : Rogers Satchell
    - `"parkinson"` : Parkinson
    - `"gk.yz"` : Garman Klass - Yang Zhang
    - `"yang.zhang"` : Yang Zhang
  - `N` : Normalization을 위한 1년의 일수

R에서 Extreme Value 모형 사용예

```
> require("quantmod")
> data(ttrc); ohlc <- ttrc[,c("Open","High","Low","Close")]
> vClose <- volatility(ohlc, calc="close")
> vGK <- volatility(ohlc, calc="garman")
> vGKTZ <- volatility(ohlc, calc="gk.yz")
> vYZ <- volatility(ohlc, calc="yang.zhang")
> old.par <- par(); par(mfrow=c(2,1), mar=c(1,4,2,2))
> plot(ohlc$Close[-(1:5000)], type='l', col=1, xlab="", ylab="price",main="Volatilities")
> plot(vClose[-(1:5000)], type='l', col=1, xlab="", ylab="volatility")
> lines(vGK[-(1:5000)], col=2)
> lines(vYZ[-(1:5000)], col=3)
> legend("topleft", c("Close-Close", "Garman Klass", "Yang Zhang"), lwd=1, col=1:3)
> par(old.par)
```



Realized Volatility 모형

- 변동성 변화보다 high frequency인 자료를 사용하여 샘플 변동성 측정
- 아래의 가격 프로세스에 대해

$$Y_t = \mu_t dt + \sigma_t dW_t, \tag{1}$$

- Δ 간격으로 n 개의 시세 샘플링으로 구한 실현변동성 (realized volatility)

$$RV_t = \sum_{i=1}^{n} (Y_{t,i} - Y_{t,i-i})^2 = \sum_{i=1}^{n} r_{t,i}^2 \tag{2}$$

- $RV_t - \sigma_t^2$ 는 n 이 증가하면 0으로 수렴
  - 일간 변동성 : 보통 10분 - 30분 샘플링 시세 사용
  - 월간 변동성 : 일일 시세 사용
- frequency가 높아지면 market microstructure noise로 인한 변동성 과대 평가
- frequency가 낮아지면 변동성 측정 오차 증가

Realized Volatility 모형의 어려움

- correlated-return : RV 추정 필터 필요
  - 예를 들어 수익률이 MA(1) 모형인 경우

$$RV_t = \frac{n}{n-1}\sum_{i=1}^{n}(r_{t,i} - \bar{r}_t)^2 + 2\sum_{i=1}^{n-1}(r_{t,i} - \bar{r}_t)(r_{t,i+1} - \bar{r}_t) \qquad (3)$$

- market microstructure noise (MMN)
  - frequency가 높아지면 MMN로 인한 변동성 과대 평가
  - frequency가 낮아지면 변동성 측정 오차 증가