

시계열 기말 프로젝트

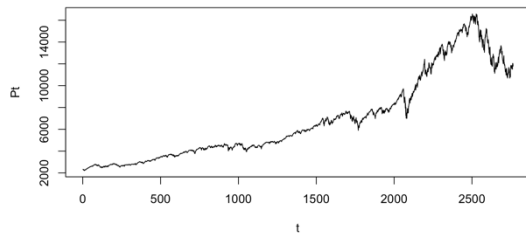
212STG36

백나림

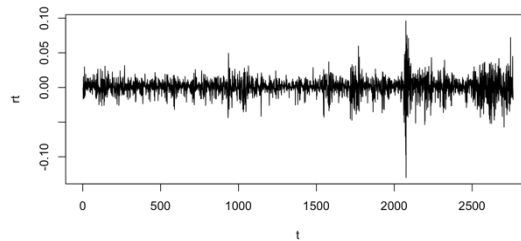
DATA: NASDAQ 100 가격 & 변동성 지수 2011.12.5~2022.12.5

A1. 기초통계분석

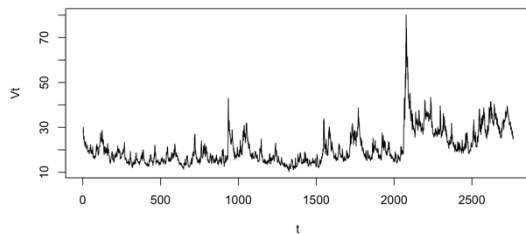
#1 일별가격,로그수익율,변동성지수를 구해서 시도표 그리기



Pt 시도표



rt 시도표



Vt 시도표

Pt, Rt에서 변동이 클 때 변동성지수, 즉 Vt값이 크게 나타난다.

#2. Pt, Rt, Vt의 기본통계량 표를 작성하라 -대표값, 산포, 비대칭성, 첨도, 정규성 등등 이들 통계량값을 통해 자료의 특징을 간단하게 기술하라.

	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis
Pt	6895.58	3879.48	5725.38	6445.45	3137.83	2215.27	16573.34	14358.07	0.87	-0.43
Rt	0	0.01	0	0	0.01	-0.13	0.1	0.23	-0.56	8.4
Vt	20.87	7.55	18.41	19.79	5.26	10.31	80.08	69.77	1.8	5.7

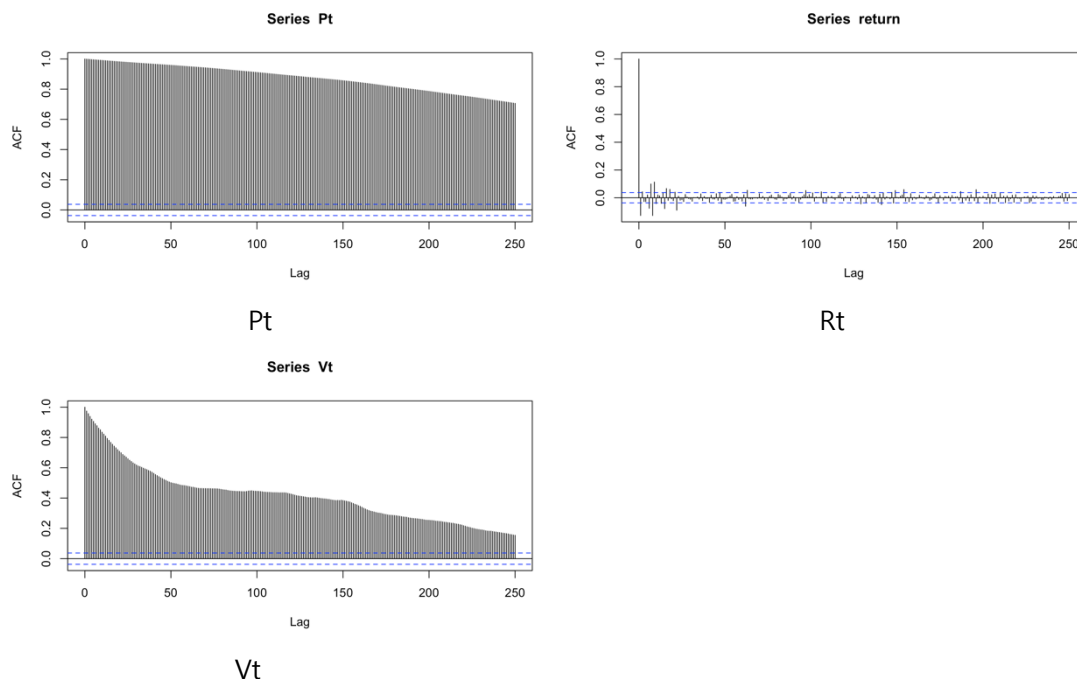
Pt는 평균 6895.58, 중앙값은 5725.38으로 평균과 중앙값이 차이가 많이 나고, 표준편차 3900정도로 크기 때문에 데이터들이 넓게 퍼져 있음을 알 수 있다. skew는 0.87, kurtosis는 -0.43으로 Rt, Vt에 비하면 그나마 정규분포랑 가장 가까운 분포를 가진다.

Rt 수익율은 평균, 중앙값이 거의 0으로 비슷하고, 표준편차는 0.01로 매우 작다. 즉 이는 거의 0에 가까운 값들로 구성되어 있음을 보여준다. 그리고 skew는 -0.5, kurtosis는 8.4로 뾰족한 분포를 가지고 있다.

Vt는 평균 20.87, 중앙값 18.41, 표준편차 7.55이고 skew는 1.8, kurtosis는 5.7로 Pt, Rt에 비해 옆으로 치우치고 약간 뾰족한 분포를 가지고 있다. 또한 Skew가 양수인 큰 값으로 보아 오른쪽으로 긴 꼬리를 가지는 것을 알 수 있다.

Normal분포이면, skewness는 0, kurtosis는 2이다. 하지만 Pt, Vt가 skew가 0에서 크게 벗어나있고, Pt, Rt, Vt kurtosis가 2에서 크게 벗어나있기 때문에 Pt, return, Vt 모두 정규성을 만족하지 못한다.

#3 Pt, return, Vt의 표본자기상관함수를 lag 을 250 까지 그리기, 각 계열의 자기상관의 특징을 간단하게 설명하라.



Pt는 자기상관이 크고 ACF가 느리게 감소하므로 비정상성을 의미한다. 변동성지수보다 더 완만하게 감소한다.

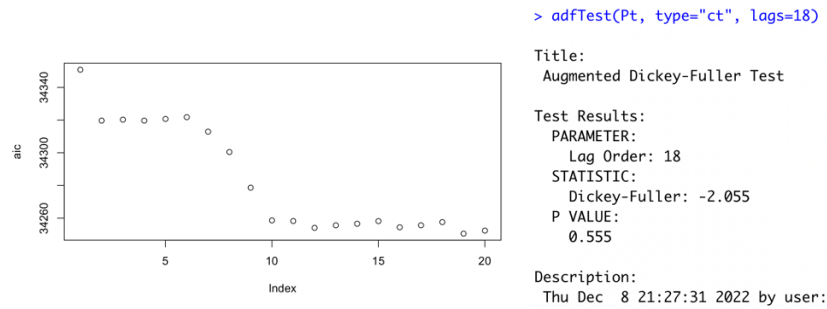
Rt은 acf가 거의 0에 가까우며 자기상관이 매우 적다. 즉 선형 기억성이 거의 없음을 알 수 있다. 즉 이는 이전 수익률과, 오늘 수익률은 거의 관계가 없다는 사실을 의미한다. 이에 비해 Pt, Vt는 자기상관이 매우 강하다. 특히 Pt는 굉장히 느리게 acf가 감소함을 보이기 때문에 장기기억성을 가진다는 사실을 알 수 있다.

Vt는 Rt에 비해 자기상관이 크고 Pt에 비해 자기상관이 작다. ACF가 Rt에 비해 느리게 감소하므로 비정상성이 있을수도 있음을 의미한다.

A2. 모형추정, 검진 및 예측

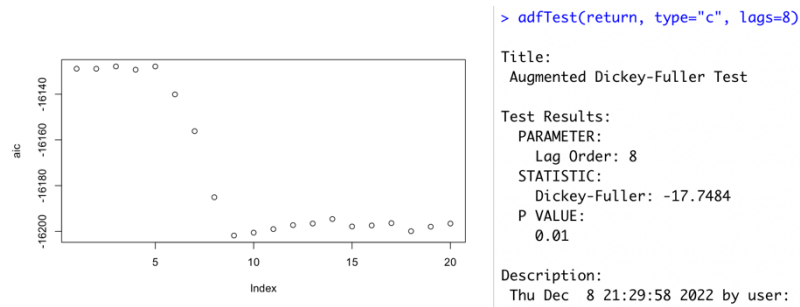
#1. Pt, return, Vt각에 대해 정상성 여부를 ADF 검정

Pt



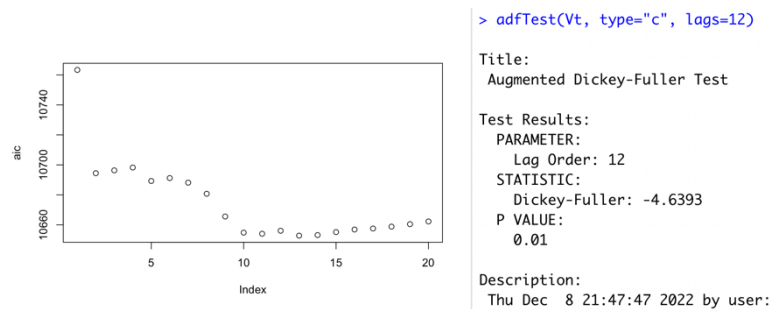
- ➔ Adf 검정 결과 p-value > 0.05, H_0 do not rejected at 5%
- ➔ Pt는 확률적 추세이다. 즉 단위근 계열로 차분이 필요하다.

Rt



- ➔ Adf 검정 결과 p-value < 0.05, H_0 rejected at 5%
- ➔ Rt는 확정적 추세이다. 즉 차분이 필요없다

Vt



- ➔ Adf 검정 결과 p-value < 0.05, H_0 rejected at 5%
- ➔ Vt는 확정적 추세이다. 즉 차분이 필요없다

#2 Pt, return, Vt각에 대해 ARIMA 모델을 BIC 기준으로 identify하고 모형의 추정식을 써라. 추정치의 표준오차도 명시하라.

Pt : Arima(9,1,0)

$$P_t = -0.086 P_{t-1} + 0.007 P_{t-2} - 0.028 P_{t-3} - 0.015 P_{t-4} + 0.009 P_{t-5} - 0.051 P_{t-6} + 0.061 P_{t-7} - 0.085 P_{t-8} + 0.089 P_{t-9}$$

Series: Pt

ARIMA(9,1,0)

Coefficients:

	ar1	ar2	ar3	ar4	ar5	ar6	ar7
	-0.0859	0.0072	-0.0284	-0.015	0.0085	-0.0508	0.0614
s.e.	0.0190	0.0190	0.0189	0.019	0.0190	0.0190	0.0190
	ar8	ar9					
	-0.0846	0.0892					
s.e.	0.0190	0.0190					

sigma^2 = 14026: log likelihood = -17107.66

AIC=34235.32 AICc=34235.4 BIC=34294.56

표준오차 sqrt(14026) = 118.4314

Rt: Arima(9,0,0)

$$R_t = -0.1(R_{t-1}) + 0.011(R_{t-2}) - 0.018(R_{t-3}) + 0.004(R_{t-4}) - 0.06(R_{t-5}) + 0.067(R_{t-6}) - 0.097(R_{t-7}) + 0.082(R_{t-8}) + 0.067(R_{t-9}) + a_t$$

Series: return

ARIMA(9,0,0) with non-zero mean

Coefficients:

	ar1	ar2	ar3	ar4	ar5	ar6	ar7
	-0.100	0.0108	-0.0175	-0.0328	0.0044	-0.0576	0.0669
s.e.	0.019	0.0190	0.0189	0.0189	0.0189	0.0189	0.0190
	ar8	ar9	mean				
	-0.0967	0.0824	6e-04				
s.e.	0.0190	0.0190	2e-04				

sigma^2 = 0.0001656: log likelihood = 8111.91

AIC=-16201.81 AICc=-16201.72 BIC=-16136.65

표준오차 : 0.01286857

Vt : Arima(2,0,0)

$$\text{추정식 : } V_t - 20.87 = 0.82(V_{t-1} - 20.87) + 0.16(V_{t-2} - 20.87) + a_t$$

> arima.fit

Series: Vt

ARIMA(2,0,0) with non-zero mean

Coefficients:

	ar1	ar2	mean
	0.8193	0.1591	20.8677
s.e.	0.0188	0.0188	1.4508

sigma^2 = 2.797: log likelihood = -5343.22

AIC=10694.44 AICc=10694.45 BIC=10718.14

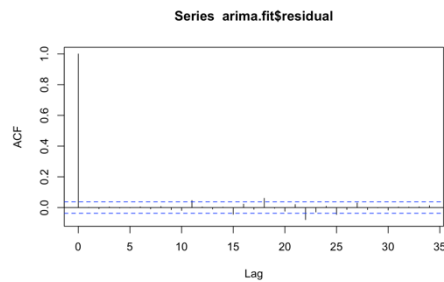
표준오차 : 0.01286857

#3. 위에서 추정한 모형의 타당성을 검진하라

#Pt - arima(9,1,0)

> Portmanteau

	L	Q	p
[1,]	6	0.33204	0.999326279
[2,]	12	8.10216	0.777096834
[3,]	18	25.86966	0.102775718
[4,]	24	48.49032	0.002193795

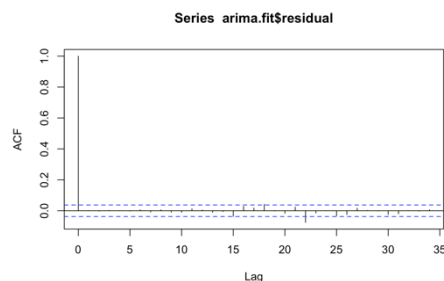


➔ Lag 24일때를 제외하고 모두 기각되지 않는다. 또한 잔차의 acf가 거의 유의하지 않으므로 ARIMA(9,1,0) 모형은 적절하다고 판단할 수 있다.

#return - arima(9,0,0)

> Portmanteau

	L	Q	p
[1,]	6	0.1391644	0.9999467
[2,]	12	1.4291726	0.9998994
[3,]	18	13.5667975	0.7568630
[4,]	24	32.4236176	0.1168310

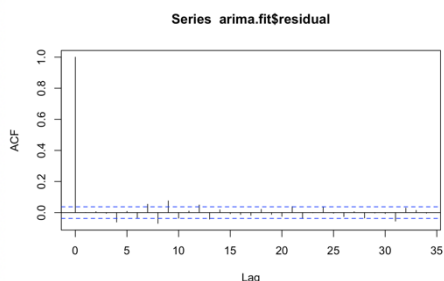


➔ 포트맨토우 검정 결과 모두 기각되지 않는다. 또한 잔차의 acf가 모두 유의하지 않으므로 ARIMA(9,0,0) 모형은 적절하다고 판단할 수 있다.

#Vt - arima(2,0,0)

> Portmanteau

	L	Q	p
[1,]	6	13.62054	3.417446e-02
[2,]	12	61.55045	1.175761e-08
[3,]	18	70.18824	4.199636e-08
[4,]	24	84.80726	1.024664e-08

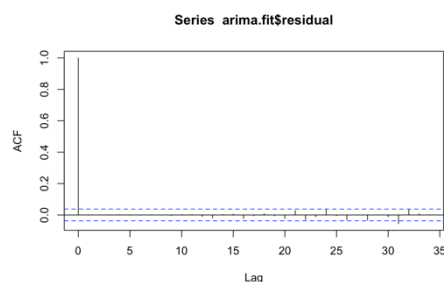


➔ 모두 기각되지 않는다. 또한 잔차의 acf도 몇몇 지점에서 유의하다. 그래서 p를 bic기준이 아닌, aic기준 선택된 13을 이용해서 포트맨토우 검정을 다시 실시해본다.

Vt-arima(13,0,0)

> Portmanteau

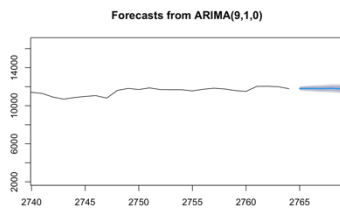
	L	Q	p
[1,]	6	0.01833437	0.9999999
[2,]	12	0.35617623	1.0000000
[3,]	18	3.30803232	0.9999417
[4,]	24	13.98406454	0.9470095



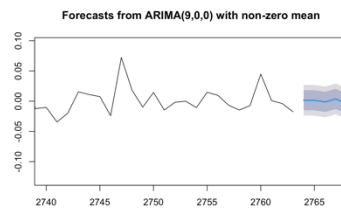
➔ 포트맨토우 검정결과 모두 기각되지 않는다, 또한 잔차의 acf도 모두 유의하지 않으므로 추정된 Vt모형은 ARIMA(13,0,0)이 적절해 보인다.

#4. P_t , $return$, V_t 각에 각각에 대해 데이터의 마지막 시점 T 에서 일주일간 미래 $T+1, T+2, \dots, T+5$ 에서의 미래값을 예측하고 95% 예측구간을 그려라. 이그림은 데이터 시점 $T-22, T-21, \dots, T, T+1, \dots, T+5$ 만 포함되게 한다

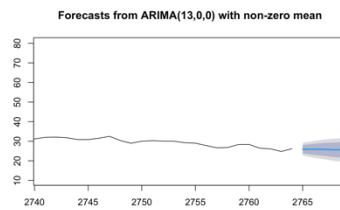
P_t - arima(9,1,0)



$return$ - arima(9,0,0)



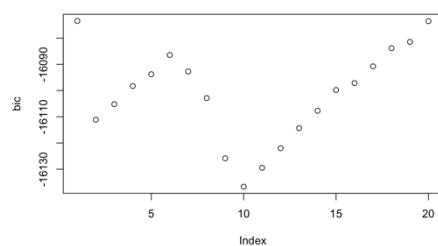
V_t - arima(13,0,0)



A3. out-of-sample 예측력 비교

#1. $return$ (R_t)에 대해 전체 데이터중 앞부분 85% 를 이용하여 AIC order AR 모형과 BIC order AR 모형을 추정하라

BIC 기준 ARIMA(9,0,0)



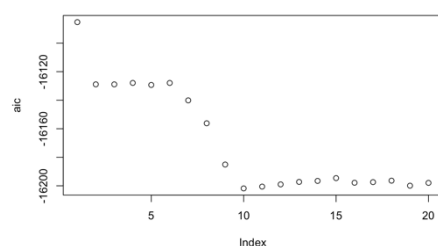
Series: train
ARIMA(9,0,0) with non-zero mean

Coefficients:

	ar1	ar2	ar3	ar4	ar5	ar6	ar7
	-0.100	0.0108	-0.0175	-0.0328	0.0044	-0.0576	0.0669
s.e.	0.019	0.0190	0.0189	0.0189	0.0189	0.0189	0.0190
	ar8	ar9	mean				
	-0.0967	0.0824	6e-04				
s.e.	0.0190	0.0190	2e-04				

sigma^2 = 0.0001656: log likelihood = 8111.91
AIC=-16201.81 AICc=-16201.72 BIC=-16136.65

AIC 기준 ARIMA(9,0,0)



Series: train
ARIMA(9,0,0) with non-zero mean

Coefficients:

	ar1	ar2	ar3	ar4	ar5	ar6	ar7
	-0.100	0.0108	-0.0175	-0.0328	0.0044	-0.0576	0.0669
s.e.	0.019	0.0190	0.0189	0.0189	0.0189	0.0189	0.0190
	ar8	ar9	mean				
	-0.0967	0.0824	6e-04				
s.e.	0.0190	0.0190	2e-04				

sigma^2 = 0.0001656: log likelihood = 8111.91
AIC=-16201.81 AICc=-16201.72 BIC=-16136.65

➔ R_t 에 대해 AIC과 BIC 를 기준으로 AR모형 추정 결과, 둘 다 AR(9)로 추정되었다.

#2. 각 모형의 최근 15% 데이터의 1-step, 2-step, 3-step, 4-step, 5-step 예측치의 RMSE, MAE, MAPE 비교표를 작성하라

위 1번에서 AIC BIC기준이 동일하게 AR(9)가 나왔다. 그렇기 때문에 BIC AIC plot을 그려본 결과 AR(9)다음으로 AIC, BIC값이 낮은 AR(10)을 비교모델로 선정하고, RMSE, MAE, MAPE기준으로 비교해보면 결과는 아래와 같다

AR(9)				AR(10)			
	RMSE	MAE	MAPE		RMSE	MAE	MAPE
1	0.01706309	0.01296452	1.297060	1	0.01705341	0.01293439	1.294063
2	0.01701223	0.01279180	1.283550	2	0.01700103	0.01275213	1.278468
3	0.01700721	0.01277853	1.282457	3	0.01700425	0.01277357	1.282139
4	0.01701002	0.01280303	1.282863	4	0.01700955	0.01277917	1.281234
5	0.01700965	0.01312281	1.319826	5	0.01700876	0.01310147	1.317579

#3. 비교 결과를 간단히 설명하라.

➔ . rmse, mae, mape 로 비교한 결과 두 모델간의 차이는 거의 나지 않지만 RMSE, MAE, MAPE 값은 AR(9)에 비해 AR(10)이 아주 약간 더 낮게 나타난다. 즉 이는 앞 85%데이터를 가지고 모형을 추측한 결과, AIC, BIC기준 AR(9)가 best로 나왔지만, 최근 15%데이터를 가지고 out of sample 예측력을 RMSE, MAE, MAPE 기준으로 비교한 결과, 1~5단계 예측에서 AR(9)보다 AR(10)이 약간이지만, 좀 더 나은 결과를 가져온다는 사실을 확인할 수 있었다.

A5. Rt에 대한 GARCH모형 추정 및 Pt, Rt의 5%VaR

#1. rt에 대해 AR(0)+GARCH(1,1) 과 AR(0)+GJR-GARCH(1,1) 모형을 추정하고 추정식을 써라. 정규 분포와 표준화 t-분포 사용하라. (총 4개의 모형을 추정)

#AR(0)+GARCH(1,1) + 정규분포

$$r = \mu + a_t$$

$$a_t = \sigma_t \varepsilon_t \quad \varepsilon_t \sim iid N(0,1)$$

$$\sigma_t^2 = 0.000005 \pm 0.144727 a_{t-1}^2 + 0.823771 \sigma_{t-1}^2$$

```

*-----*
*          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|)
omega  0.000005  0.000002  2.7016 0.006901
alpha1 0.144727  0.012957  11.1696 0.000000
beta1   0.823771  0.007394  111.4175 0.000000

```

#AR(0)+GARCH(1,1) + t분포

$$r = \mu + a_t$$

$$a_t = \sigma_t \varepsilon_t \quad \varepsilon_t \sim iid t(df = 5.64)$$

$$\sigma_t^2 = 0.000003 + 0.147735 a_{t-1}^2 + 0.841785 \sigma_{t-1}^2$$

```

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

Conditional Variance Dynamics
-----
GARCH Model      : sGARCH(1,1)
Mean Model       : ARFIMA(0,0,0)
Distribution      : std

Optimal Parameters
-----
      Estimate Std. Error t value Pr(>|t|)
omega  0.000003  0.000002  2.0254 0.042828
alpha1 0.147735  0.020861  7.0818 0.000000
beta1   0.841785  0.020534  40.9939 0.000000
shape   5.640680  0.635134  8.8811 0.000000

```


#AR(0)+GJR-GARCH(1,1) + 정규분포

$$r = \mu + a_t$$

$$a_t = \sigma_t \varepsilon_t \quad \varepsilon_t \sim iid N(0,1)$$

$$\sigma_t^2 = 0.000005 + 0.018093 a_{t-1}^2 + 0.184960 s_{t-1}^- a_{t-1}^2 + 0.85359 \sigma_{t-1}^2$$

$$s_t^- = 1, a_t < 0; s_t^- = 0, a_t \geq 0$$

```

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

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

Optimal Parameters
-----
      Estimate Std. Error t value Pr(>|t|)
omega    0.000005  0.000000  18.1691 0.000000
alpha1    0.018093  0.005416   3.3407 0.000836
beta1     0.853590  0.009423  90.5851 0.000000
gamma1    0.184960  0.017758  10.4155 0.000000

```

AR(0)+GJR-GARCH(1,1) + t분포

$$r = \mu + a_t$$

$$a_t = \sigma_t \varepsilon_t \quad \varepsilon_t \sim iid t(df = 5.93)$$

$$\sigma_t^2 = 0.000004 + 0.257918 s_{t-1}^- a_{t-1}^2 + 0.849232 \sigma_{t-1}^2$$

$$s_t^- = 1, a_t < 0; s_t^- = 0, a_t \geq 0$$

```

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

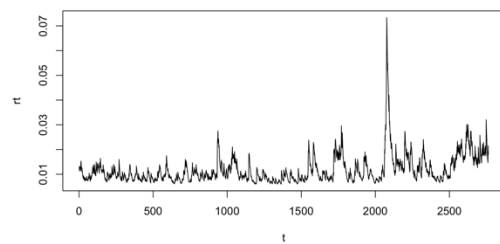
Conditional Variance Dynamics
-----
GARCH Model      : gjrGARCH(1,1)
Mean Model       : ARFIMA(0,0,0)
Distribution      : std

Optimal Parameters
-----
      Estimate Std. Error t value Pr(>|t|)
omega    0.000004  0.000001  8.176593 0.000000
alpha1    0.000000  0.001684  0.000076 0.99994
beta1     0.849232  0.011961  70.997936 0.000000
gamma1    0.257918  0.026422  9.761497 0.000000
shape    5.927278  0.643147  9.216055 0.000000

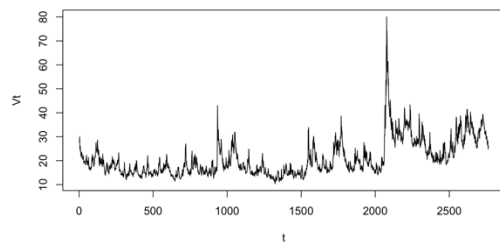
```

#2. AR(0)+GARCH(1,1) 모형으로부터 계산된 sigma hat 시도표를 그려보고 이를 rt, IVt의 시도표와 비교 설명하여라.

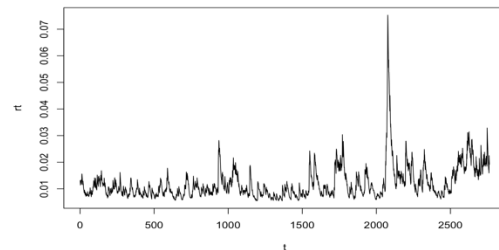
AR(0)+GARCH(1,1)+ norm



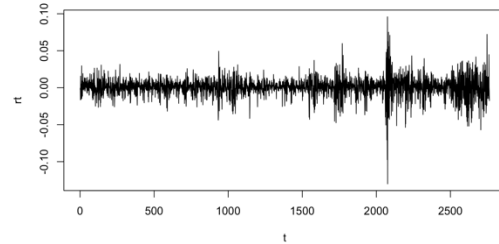
Vt



AR(0)+GARCH(1,1)+ t

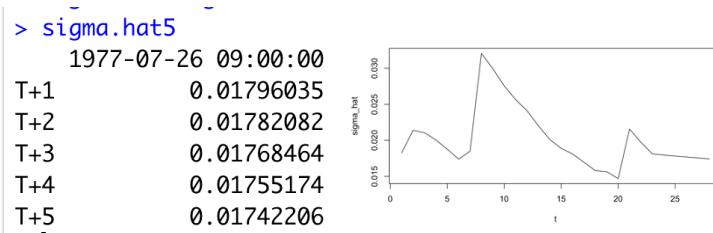


rt



➔ AR(0)+GARCH(1,1) 모형으로부터 계산된 sigma hat 시도표와, Volatility 시도표는 거의 비슷한 패턴을 보인다. 또한 rt 시도표의 +방향쪽의 그래프와도 형태가 비슷함을 알 수 있다.

#3. r_{t+1}, \dots, r_{t+5} 의 조건부 표준편차의 예측치를 구하시오. 또한 이를 시도표 그림으로 나타내라
이 그림은 데이터 시점 $t = T-22, T-21, \dots, T, T+1, \dots, T+5$ 에서의 σ 만 포함되게 한다



#4. 위의 각 4개의 모형을 이용하여 r_{t+1} 의 1%, 5% VaR 구하고 표로 작성하라. 이중 어느 모형의 VaR 더 좋은지 어떻게 평가할지를 설명하라 또 P_{t+1} 의 1% 5% VAR 구하고 표를 작성하라

R_{t+1} 의 VAR

분포	5%	1%
AR(0)+GARCH(1,1) normal	-0.029	-0.041
AR(0)+GARCH(1,1) t분포	-0.036	-0.059
AR(0)+GJR-GARCH(1,1) normal	-0.023	-0.033
AR(0)+GJR-GARCH(1,1) t분포	-0.027	-0.042

P_{t+1} 의 VAR

분포	5%	1%
AR(0)+GARCH(1,1) normal	11653.83	11512.05
AR(0)+GARCH(1,1) t분포	11568.44	11307.81
AR(0)+GJR-GARCH(1,1) normal	11720.85	11606.09
AR(0)+GJR-GARCH(1,1) t분포	11678.67	11494.08

➔ 최근 NASDAQ의 데이터의 P_t, R_t 을 보면, 변동성이 커지는 형태를 보인다. 또한 증가에 비해, 감소가 더 자주 일어나면서, 그 값은 적은 전형적인 비대칭성을 보인다. 그렇기 때문에 데이터의 비대칭성이 반영된 GJR-GARCH모형을 이용하는 경우의 VAR이 더 좋다고 평가할 수 있다.

#AR6 GBM

#1. 위A1-1의 데이터를 이용하여 μ , σ 를 추정(단위 명시)

$\mu.\text{hat} : 0.1684/\text{year}$ $\sigma.\text{hat} : 0.2081/\text{year}$

#2. $p_s = \ln P_s$ 라 할 때 p_s 가 만족하는 모형을 적어라. 모수값도 적어야함

$\ln P_s = p_s$ 는 generalized wiener process를 만족한다.

$p_s^* = \ln P_s^* \sim N(\ln P_0 + (\mu.\text{hat} - \sigma.\text{hat}^2 / 2) * s, \sigma.\text{hat}^2 * s)$

즉, $N(9.374736 + 0.1467989 * s, 0.04331835 * s)$

#3 마지막 시점 T (대응되는 s 값은 $T/250$)서 조건부로 2년후의 미래값 P_{s+2} 의 조건부 분포를 구하고 평균 분산을 구하시오.

$p_2 - p_0 \sim N(2 * (\mu - \sigma^2 / 2), 2 * \sigma^2)$

$\log(P_2) = p_2 \sim N(\log(P_0) + 2 * (\mu - \sigma^2 / 2), 2 * \sigma^2)$

P_2 는 log normal 분포로 평균은 16508.85 , 분산은 24665193 이다.

#4. 마지막 시점 T (대응되는 s 값은 $T/250$)서 조건부로 향후 2년간의 미래수익률 $r_{s+2} = \ln P_{s+2} - \ln P_s$ 의 조건부 분포를 구하고 평균 분산을 구하시오.

향후 2년간 미래수익률의 조건부 분포는 정규분포로, 분산은 $2 * \sigma^2$ 이고 평균은 $2 * (\mu - \sigma^2 / 2)$ 이다.

즉 $r_{(s+2)} = \log(P_{s+2}) - \log(P_s) \sim \text{Normal}(0.1467989, 0.02165917)$ 이다.

#5. 이자산의 마지막가격 P_t 를 쓰시오. 또 만기6개월,행사가 P_t 인 Europeancall option 의 공정한 현재가를 구하시오.

$P_t = 11786.8$ #자산의 마지막 가격

$h_1 = (\log(P_t/K)) + (r + \sigma^2 / 2) * T / (\sigma * \sqrt{T})$

$h_2 = (\log(P_t/K)) + (r - \sigma^2 / 2) * T / (\sigma * \sqrt{T})$

$C_t = P_t * \text{pnorm}(h_1) - K * \exp(-r * T) * \text{pnorm}(h_2) = 1245.071$

Europeancall option 의 공정한 현재가는 1245.07 이다.

#6. $Q_s = P_s^2$ 이 Geometric Brownian Motion을 따름을 보이고 이 모형의 mean, variance 의 값을 적으시오.

→ P_s 가 GBM을 따른다고 가정했으므로,

$dP_s = \mu P_s dt + \sigma P_s dW_t$ 이다.

$2 P_s dP_s = dP_s^2$ 이므로, 양변에 $2P_s$ 곱하면

$dP_s^2 = 2 \mu P_s^2 dt + 2 \sigma P_s^2 dW_t$ 이다. 이때 $Q_s = P_s^2$ 이므로

$dQ_s = 2 \mu Q_s dt + 2 \sigma Q_s dW_t$ 이다.

즉 Q_s 는 평균 2μ , 표준편차 2σ 인 GBM을 따른다.

Mean : 2μ , variance: $4 \sigma^2$

AR7 Simulation for GARCH and GJR-GARCH

#1. 위 A5-1에서 추정된 $AR(0)+GARCH(1,1)$ 과 $AR(0)+GJR-GARCH(1,1)$ 모수를 이용하여 추정된 모형을 참모형으로 간주하고 $r_t, t=1..n, n=1000$ simulate 하여라. 그 후 각 모형별로 skewness 를 계산하여라.

분포	$AR(0)+GARCH(1,1)$ +normal	$AR(0)+GARCH(1,1)$ +t(5.6)	$AR(0)+GJR-GARCH(1,1)$ + normal	$AR(0)+GJR-GARCH(1,1)$ + t(5.9)
skew	0.082	0.306	0.153	-0.286

#2. 위 1을 500 번 반복하여 skewness 의 평균을 구하여라.

분포	$AR(0)+GARCH(1,1)$ +normal	$AR(0)+GARCH(1,1)$ +t	$AR(0)+GJR-GARCH(1,1)$ + normal	$AR(0)+GJR-GARCH(1,1)$ + t
skew	-0.009	-0.197	-0.004	0.0254

#3. 위 2 의 결과에 대해 간략한 토론을 하여라.

: 전반적으로 skewness 평균값이 0에 가깝다. 주로 $AR+GARCH$ 모형이나, $AR+GJR-GARCH$ 모형이나 normal을 가정할 때가 좀 더 skewness가 0에 가까움을 확인할 수 있다. 위 4 모형중에서는 $AR+GARCH+t$ 분포 가정 시, skewness가 가장 0에서 벗어남을 확인할 수 있다.

<Appendix>

R-code

```
library(lubridate)
library(tidyverse)
library(psych)
library(forecast)
library(fUnitRoots)
library(forecast)
library(rugarch)
```

#load data

```
data=read.csv("/Users/baeknarim/2022_2/시계열분석/project/Nasdaq 100 Historical Data.csv")
vol=read.csv("/Users/baeknarim/2022_2/시계열분석/project/NASDAQ 100 Volatility Historical Data-3.csv")
```

```
data$Date = mdy(data$Date)
vol$Date = mdy(vol$Date)
result=inner_join(data, vol, by="Date")
head(result)
result2=result[,c(1,2,8)]
names(result2)=c("Date", "Pt", "Vt")
result2$Pt= as.numeric(gsub(",","",result2$Pt))
```

```
result2 = result2[order(result2$Date),]
attach(result2)
```

#A1, 기초통계분석

###1 일별가격,로그수익율,변동성지수를 구해서 시도표그리기

#Pt 시도표

```
plot(Pt, type="l",xlab="t", ylab="Pt")
```

#로그수익률 시도표

```
return =diff(log(result2$Pt))
```

```
plot(return, type="l",xlab="t", ylab="rt")
```

#변동성지수 시도표

```
plot(Vt, type="l", xlab="t", ylab="Vt")
```

###2. Pt, Rt, Vt의 기본통계량 표를 작성하라 -대표값, 산포, 비대칭성, 첨도, 정규성 등등 이들 통계량값을 통해 자료의 특징을 간단하게 기술하라

```
#Pt
```

```
describe(Pt)
```

```
#Rt
```

```
describe(return)
```

```
#Vt
```

```
describe(Vt)
```

###3. Pt, return, Vt의 표본자기상관함수를 lag 을 250 까지 그리기, 각 계열의 자기상관의 특징을 간단하게 설명하라.

```
acf(Pt, lag=250)
```

```
acf(return, lag=250)
```

```
acf(Vt, lag=250)
```

#A2. 모형추정, 검진 및 예측

###Pt, return, Vt각에 대해 정상성 여부를 ADF 검정

```
#Pt
```

```
aic=c()
```

```
for(p in 1:20){
```

```
  ar.fit=Arima(Pt,order=c(p,0,0))
```

```
  aic[p] = ar.fit$aic
```

```
}
```

```
plot(aic)
```

```
which.min(aic) #19
```

```
adfTest(Pt, type="ct", lags=18)
```

```
#p-value > 0.05 H0 do not rejected at 5%
```

```
#Pt는 확률적 추세이다.단위근 계열로 차분이 필요하다.
```

```
#return
```

```

aic=c()
for(p in 1:20){
  ar.fit=Arima(return,order=c(p,0,0))
  aic[p] = ar.fit$aic
}
plot(aic)
which.min(aic) #9

```

```

adfTest(return, type="c", lags=8)
#p-value < 0.05 H0 rejected at 5%
#Rt는 확정적 추세이다. 차분이 필요없다

```

```

#Vt
aic=c()
for(p in 1:20){
  ar.fit=Arima(Vt,order=c(p,0,0))
  aic[p] = ar.fit$aic
}
plot(aic)
which.min(aic) #13

```

```

adfTest(Vt, type="c", lags=12)
#p-value < 0.05 H0 rejected at 5%
#Vt는 확정적 추세이다. 차분이 필요없다

```

```

###Pt, return, Vt각에 대해 ARIMA 모델을 BIC 기준으로 identify 추정치의 표준오차도 명시
#Pt - arima(9,1,0)
bic=c()
for(p in 1:20){
  arima.fit = Arima(Pt, order = c(p-1,1,0)); bic[p] = arima.fit$bic
}
which.min(bic) #10

```

```

arima.fit = Arima(Pt, order = c(9,1,0))
arima.fit
#표준오차 :118.4314

```

```

#return - #arima(9,0,0)
bic=c()

```

```
for(p in 1:20){
  arima.fit = Arima(return, order = c(p-1,0,0)); bic[p] = arima.fit$bic}
which.min(bic) #10
```

```
arima.fit = Arima(return, order = c(9,0,0))
arima.fit
#표준오차 0.01286857
```

```
#Vt - #arima(2,0,0)
bic=c()
for(p in 1:20){
  arima.fit = Arima(Vt, order = c(p-1,0,0)); bic[p] = arima.fit$bic}
which.min(bic) #3
```

```
arima.fit = Arima(Vt, order = c(2,0,0))
arima.fit
#표준오차 1.672423
```

```
aic=c()
for(p in 1:20){
  arima.fit = Arima(Vt, order = c(p-1,0,0)); aic[p] = arima.fit$aic}
which.min(aic) #14
```

```
arima.fit = Arima(Vt, order = c(13,0,0))
arima.fit
#표준오차 1.656502
```

```
###위에서 추정한 모형의 타당성을 검진하라
#Pt - arima(9,1,0)
arima.fit = Arima(Pt, order = c(9,1,0))
L=c();Q=c();p=c()
for(i in 1:4){
  L[i]=6*i
  Q[i]=Box.test(arima.fit$residual, lag=L[i],type="Ljung-Box")$statistic
  df=L[i]
  p[i]=1-pchisq(Q[i],df)}
Portmanteau=cbind(L,Q,p)
Portmanteau
#6,12,18이 기각되지 않기 때문에 arima(9,1,0) 적절해 보인다
```



```
acf(arima.fit$residual)
```

```
#return - #arima(9,0,0)
```

```
arima.fit = Arima(return, order = c(9,0,0))
```

```
L=c();Q=c();p=c()
```

```
for(i in 1:4){
```

```
  L[i]=6*i
```

```
  Q[i]=Box.test(arima.fit$residual, lag=L[i],type="Ljung-Box")$statistic
```

```
  df=L[i]
```

```
  p[i]=1-pchisq(Q[i],df)}
```

```
Portmanteau=cbind(L,Q,p)
```

```
Portmanteau
```

```
#모두 기각되지 않기 때문에 arima(9,1,0) 적절해 보인다
```

```
acf(arima.fit$residual)
```

```
#Vt - #arima(2,0,0)
```

```
#bic 기준 2
```

```
arima.fit = Arima(Vt, order = c(2,0,0))
```

```
L=c();Q=c();p=c()
```

```
for(i in 1:4){
```

```
  L[i]=6*i
```

```
  Q[i]=Box.test(arima.fit$residual, lag=L[i],type="Ljung-Box")$statistic
```

```
  df=L[i]
```

```
  p[i]=1-pchisq(Q[i],df)}
```

```
Portmanteau=cbind(L,Q,p)
```

```
Portmanteau
```

```
acf(arima.fit$residual)
```

```
#모두 기각되지 않는다. 그래서 p를 bic기준이 아닌, aic기준 선택된 13을 이용한다
```

```
#13 - aic 기준
```

```
arima.fit = Arima(Vt, order = c(13,0,0))
```

```
L=c();Q=c();p=c()
```

```
for(i in 1:4){
```

```
  L[i]=6*i
```

```
  Q[i]=Box.test(arima.fit$residual, lag=L[i],type="Ljung-Box")$statistic
```

```
  df=L[i]
```

```
  p[i]=1-pchisq(Q[i],df)}
```

```
Portmanteau=cbind(L,Q,p)
```

```
Portmanteau
```

#모두 기각되지 않으므로 Vt모형은 ARIMA(13,0,0)이 적절해 보인다.

```
acf(arima.fit$residual)
```

###Pt, return, Vt각에 각각에 대해 데이터의 마지막 시점 T 에서 일주일간 미래 T+1, T+2, ..T+5에서의 미래값을 예측하고 95% 예측구간을 그리기 이그림은 데이터 시점 T-22, T-21, ..., T, T+1, ..., T+5 만 포함되게 한다

```
#Pt - arima(9,1,0)
```

```
arima.fit = Arima(Pt, order = c(9,1,0))
```

```
ar.hat = forecast(arima.fit, h=5)
```

```
plot(ar.hat, xlim=c(2741,2768))
```

```
#return - arima(9,0,0)
```

```
arima.fit = Arima(return, order = c(9,0,0))
```

```
ar.hat = forecast(arima.fit, h=5)
```

```
plot(ar.hat, xlim=c(2740,2767))
```

```
#Vt - arima(13,0,0)
```

```
arima.fit = Arima(Vt, order = c(13,0,0))
```

```
ar.hat = forecast(arima.fit, h=5)
```

```
plot(ar.hat, xlim=c(2741,2768))
```

#A3. out-of-sample 예측력 비교

#return 에 대해 AIC order AR 모형과 BIC order AR 모형의 예측력을 비교하라

###1. 전체데이터중 앞부분 85% 를 이용하여 AIC order AR 모형과 BIC order AR 모형을 추정하라

```
#return
```

```
n=length(return); m=round(0.15*n)
```

```
train=return[1:n]
```

```
#bic 기준 arima(9,0,0)
```

```
bic=c()
```

```
for(p in 1:20){
```

```
  arima.fit = Arima(train, order = c(p-1,0,0)); bic[p] = arima.fit$bic
```

```
which.min(bic) #10
```

```
plot(bic)
```

```
arima.fit = Arima(train, order = c(9,0,0))
```

```
arima.fit
```

```
#aic 기준 arima(9,0,0)
aic=c()
for(p in 1:20){
  arima.fit = Arima(train, order = c(p-1,0,0)); aic[p] = arima.fit$aic}
which.min(aic) #10
plot(aic)
```

```
arima.fit = Arima(train, order = c(9,0,0))
arima.fit
```

###2.각모형의 최근 15% 데이터의 1-step, 2-step, 3-step, 4-step, 5-step 예측치의 RMSE, MAE, MAPE 비교표를 작성하라

#위 1번에서 AIC BIC기준이 동일하게 AR(9)가 나왔다. 그렇기 때문에 BIC AIC plot을 그려본 결과 그다음으로 낮은 AR(10)을 비교모델로 선정하고, RMSE, MAE, MAPE기준으로 비교해보겠다.

```
n = length(return) #2763
m = round(0.15*n) #414
y = return
```

#1단계 예측

```
#AR(9)
e.aic_step1 = c()
for(k in 1:m){
  N = n-k
  ar1 = Arima(y[1:N], order = c(9,0,0)) #1~2349 데이터로 fitting
  y.hat_aic_step1 = forecast(ar1, h=1)$mean[1] #2349~ 부터 예측
  e.aic_step1[k] = y[(N+1)] - y.hat_aic_step1
}
```

#2단계 예측

```
e.aic_step2 = c()
for(k in m:1){
  N = n-k
  ar = Arima(y[1:N], order = c(9,0,0))
  y[2349] = forecast(ar, h=2)$mean[1]

  y.hat = forecast(ar1, h=2)$mean[2]

  e.aic_step2[1] = return[2349] - y[2349]
```

```

    e.aic_step2[m+2-k] = y[(N+2)] - y.hat
}

```

#3단계 예측

```

e.aic_step3 = c()
for(k in m:1){
  N = n-k
  ar = Arima(y[1:N], order = c(9,0,0))
  y[2349] = forecast(ar, h=3)$mean[1]
  y[2350] = forecast(ar, h=3)$mean[2]

  y.hat = forecast(ar1, h=3)$mean[3]

  e.aic_step3[1] = return[2349] - y[2349]
  e.aic_step3[2] = return[2350] - y[2350]
  e.aic_step3[m+3-k] = y[(N+3)] - y.hat
}

```

#4단계 예측

```

e.aic_step4 = c()
for(k in m:1){
  N = n-k
  ar = Arima(y[1:N], order = c(9,0,0))
  y[2349] = forecast(ar, h=4)$mean[1]
  y[2350] = forecast(ar, h=4)$mean[2]
  y[2351] = forecast(ar, h=4)$mean[3]

  y.hat = forecast(ar1, h=4)$mean[4]

  e.aic_step4[1] = return[2349] - y[2349]
  e.aic_step4[2] = return[2350] - y[2350]
  e.aic_step4[3] = return[2351] - y[2351]

  e.aic_step4[m+4-k] = y[(N+4)] - y.hat
}

```

#5단계 예측

```
e.aic_step5 = c()
```

```
for(k in m:1){
```

```
  N = n-k
```

```
  ar = Arima(y[1:N], order = c(9,0,0))
```

```
  y[2349] = forecast(ar, h=5)$mean[1]
```

```
  y[2350] = forecast(ar, h=5)$mean[2]
```

```
  y[2351] = forecast(ar, h=5)$mean[3]
```

```
  y[2352] = forecast(ar, h=5)$mean[4]
```

```
  y.hat = forecast(ar1, h=5)$mean[5]
```

```
  e.aic_step5[1] = return[2349] - y[2349]
```

```
  e.aic_step5[2] = return[2350] - y[2350]
```

```
  e.aic_step5[3] = return[2351] - y[2351]
```

```
  e.aic_step5[4] = return[2352] - y[2352]
```

```
  e.aic_step5[m+5-k] = y[(N+5)] - y.hat
```

```
}
```

#AIC order model

```
RMSE<-
```

```
c(sd(e.aic_step1),sd(e.aic_step2[1:m]),sd(e.aic_step3[1:m]),sd(e.aic_step4[1:m]),sd(e.aic_step5[1:m]))
```

```
MAE<-
```

```
c(mean(abs(e.aic_step1)),mean(abs(e.aic_step2[1:m])),mean(abs(e.aic_step3[1:m])),mean(abs(e.aic_step4[1:m])),mean(abs(e.aic_step5[1:m])))
```

```
y.test = return[2350:2763]
```

```
MAPE<-c(100*mean(abs(e.aic_step1/(y.test+1))),
```

```
  100*mean(abs(e.aic_step2[1:m]/(y.test+1))),
```

```
  100*mean(abs(e.aic_step3[1:m]/(y.test+1))),
```

```
  100*mean(abs(e.aic_step4[1:m]/(y.test+1))),
```

```
  100*mean(abs(e.aic_step5[1:m]/(y.test+1))))
```

```
)
```

```
result_aic<- data.frame(RMSE = RMSE, MAE = MAE, MAPE = MAPE)
```

```
result_aic
```

```

#AR(10)
#1단계 예측
e.aic_step1 = c()
for(k in 1:m){
  N = n-k
  ar1 = Arima(y[1:N], order = c(10,0,0)) #1~2349 데이터로 fitting
  y.hat_aic_step1 = forecast(ar1, h=1)$mean[1] #2349~ 부터 예측
  e.aic_step1[k] = y[(N+1)] - y.hat_aic_step1
}

```

```

#2단계 예측
e.aic_step2 = c()
for(k in m:1){
  N = n-k
  ar = Arima(y[1:N], order = c(10,0,0))
  y[2349] = forecast(ar, h=2)$mean[1]

  y.hat = forecast(ar1, h=2)$mean[2]

  e.aic_step2[1] = return[2349] - y[2349]
  e.aic_step2[m+2-k] = y[(N+2)] - y.hat
}

```

```

#3단계 예측
e.aic_step3 = c()
for(k in m:1){
  N = n-k
  ar = Arima(y[1:N], order = c(10,0,0))
  y[2349] = forecast(ar, h=3)$mean[1]
  y[2350] = forecast(ar, h=3)$mean[2]

  y.hat = forecast(ar1, h=3)$mean[3]

  e.aic_step3[1] = return[2349] - y[2349]
  e.aic_step3[2] = return[2350] - y[2350]
  e.aic_step3[m+3-k] = y[(N+3)] - y.hat
}

```

```
}
```

#4단계 예측

```
e.aic_step4 = c()
```

```
for(k in m:1){
```

```
  N = n-k
```

```
  ar = Arima(y[1:N], order = c(10,0,0))
```

```
  y[2349] = forecast(ar, h=4)$mean[1]
```

```
  y[2350] = forecast(ar, h=4)$mean[2]
```

```
  y[2351] = forecast(ar, h=4)$mean[3]
```

```
  y.hat = forecast(ar1, h=4)$mean[4]
```

```
  e.aic_step4[1] = return[2349] - y[2349]
```

```
  e.aic_step4[2] = return[2350] - y[2350]
```

```
  e.aic_step4[3] = return[2351] - y[2351]
```

```
  e.aic_step4[m+4-k] = y[(N+4)] - y.hat
```

```
}
```

#5단계 예측

```
e.aic_step5 = c()
```

```
for(k in m:1){
```

```
  N = n-k
```

```
  ar = Arima(y[1:N], order = c(10,0,0))
```

```
  y[2349] = forecast(ar, h=5)$mean[1]
```

```
  y[2350] = forecast(ar, h=5)$mean[2]
```

```
  y[2351] = forecast(ar, h=5)$mean[3]
```

```
  y[2352] = forecast(ar, h=5)$mean[4]
```

```
  y.hat = forecast(ar1, h=5)$mean[5]
```

```
  e.aic_step5[1] = return[2349] - y[2349]
```

```
  e.aic_step5[2] = return[2350] - y[2350]
```

```
  e.aic_step5[3] = return[2351] - y[2351]
```

```
  e.aic_step5[4] = return[2352] - y[2352]
```

```

e.aic_step5[m+5-k] = y[(N+5)] - y.hat
}

#AIC order model
RMSE<-
c(sd(e.aic_step1),sd(e.aic_step2[1:m]),sd(e.aic_step3[1:m]),sd(e.aic_step4[1:m]),sd(e.aic_step5[1:m]))

MAE<-
c(mean(abs(e.aic_step1)),mean(abs(e.aic_step2[1:m])),mean(abs(e.aic_step3[1:m])),mean(abs(e.aic_step4[1:m])),mean(abs(e.aic_step5[1:m])))

y.test = return[2350:2763]
MAPE<-c(100*mean(abs(e.aic_step1/(y.test+1))),
        100*mean(abs(e.aic_step2[1:m]/(y.test+1))),
        100*mean(abs(e.aic_step3[1:m]/(y.test+1))),
        100*mean(abs(e.aic_step4[1:m]/(y.test+1))),
        100*mean(abs(e.aic_step5[1:m]/(y.test+1))))
)

result_aic<- data.frame(RMSE = RMSE, MAE = MAE, MAPE = MAPE)
result_aic

```

###3. 비교 결과를 간단히 설명하라.

#A5. Rt에 대한 GARCH모형추정및 Pt, Rt의 5%VaR
 ### rt에 대해 AR(0)+GARCH(1,1) 과 AR(0)+GJR-GARCH(1,1) 모형을 추정하고 추정식을 써라.정규 분포와 표준화 t-분포 사용하라. 총 4개의 모형을 추정

```

r.t = return
#AR(0)+GARCH(1,1)
#정규분포
spec.garch = ugarchspec(variance.model=list(model="sGARCH",garchOrder=c(1,1)),
                        mean.model=list(armaOrder=c(0,0), include.mean=FALSE),
                        distribution.model="norm")

a.t = r.t - mean(r.t)
garch.fit = ugarchfit(data = a.t, spec=spec.garch)
garch.fit

```

```

#t분포
spec.garch.t = ugarchspec(variance.model=list(model="sGARCH",garchOrder=c(1,1)),

```



```
mean.model=list(armaOrder=c(0,0), include.mean=FALSE),  
distribution.model="std")
```

```
a.t = r.t - mean(r.t)
```

```
garch.fit.t = ugarchfit(data = a.t, spec=spec.garch.t)
```

```
garch.fit.t
```

```
#AR(0)+GJR-GARCH(1,1) 모형
```

```
#정규분포
```

```
gjr.garch = ugarchspec(variance.model=list(model="gjrGARCH",garchOrder=c(1,1)),  
mean.model=list(armaOrder=c(0,0), include.mean=FALSE),  
distribution.model="norm")
```

```
a.t=r.t-mean(r.t)
```

```
gjr.garch.fit=ugarchfit(data=a.t, spec=gjr.garch)
```

```
gjr.garch.fit
```

```
#t분포
```

```
gjr.garch.t = ugarchspec(variance.model=list(model="gjrGARCH",garchOrder=c(1,1)),  
mean.model=list(armaOrder=c(0,0), include.mean=FALSE),  
distribution.model="std")
```

```
a.t=r.t-mean(r.t)
```

```
gjr.garch.fit.t=ugarchfit(data=a.t, spec=gjr.garch.t)
```

```
gjr.garch.fit.t
```

AR(0)+GARCH(1,1) 모형으로부터 계산된 $\hat{\sigma}$ 시도표를 그려보고 이를 r_t , IV_t 의 시도표와 비교 설명하여라

```
r.t = return
```

```
v.t = Vt
```

```
#AR(0)+GARCH(1,1)
```

```
#정규분포
```

```
spec.garch = ugarchspec(variance.model=list(model="sGARCH",garchOrder=c(1,1)),  
mean.model=list(armaOrder=c(0,0), include.mean=FALSE),  
distribution.model="norm")
```

```
a.t = r.t - mean(r.t)
```

```
garch.fit = ugarchfit(data = a.t, spec=spec.garch)
```

```
#sigma hat 시도표
sigma.fit=garch.fit@fit$sigma
plot(sigma.fit, type="l",xlab="t", ylab="rt")
```

```
#t분포
spec.garch.t = ugarchspec(variance.model=list(model="sGARCH",garchOrder=c(1,1)),
                           mean.model=list(armaOrder=c(0,0), include.mean=FALSE),
                           distribution.model="std")
```

```
a.t = r.t - mean(r.t)
garch.fit.t = ugarchfit(data = a.t, spec=spec.garch.t)
garch.fit.t
#sigma hat 시도표
sigma.fit=garch.fit.t@fit$sigma
plot(sigma.fit, type="l",xlab="t", ylab="rt")
```

```
#rt
plot(r.t, type="l",xlab="t", ylab="rt")
```

```
#IVt 시도표 (Vt)
#변동성지수 시도표
plot(v.t, type="l",xlab="t", ylab="Vt")
```

```
### rt+1, .. rt+5의 조건부 표준편차의 예측치를 구하시오 또한 이를 시도표 그림으로 나타내라
이그림은 데이터 시점 t = T-22, T-21, ..., T, T+1,..., T+5 에서의 sigma만 포함되게 한다
sigma.hat5=ugarchforecast(garch.fit, n.ahead=5)@forecast$sigmaFor
sigma.hat5
```

```
#plot
plot(c(sigma.fit[2741:2763] , sigma.hat5), type="l",xlab="t", ylab="sigma_hat")
```

```
### 위1의 각4개의 모형을 이용하여 rt+1의 1%, 5% VaR 구하고 표로 작성하라. 이중 어느 모형
의 VaR 더 좋은지 어떻게 평가할지를 설명하라 또 Pt+1의 1% 5% VAR 구하고 표를 작성하라
#내일 log return의 1% 5% VAR
#AR(0)+GARCH(1,1)
#정규분포
ar.fit = Arima(r.t, order = c(0,0,0))
```

```
spec.garch = ugarchspec(variance.model=list(model="sGARCH",garchOrder=c(1,1)),
```

```

mean.model=list(armaOrder=c(0,0), include.mean=FALSE),
distribution.model="norm")

a.t = r.t - mean(r.t)
garch.fit = ugarchfit(data = a.t, spec=spec.garch)
sigma.hat=ugarchforecast(garch.fit, n.ahead=1)@forecast$sigmaFor
mu= ar.fit$coef[1]

# 5% value at risk of return
garch_nor_rt_5=mu + qnorm(0.05) * sigma.hat
# 1% value at risk of return
garch_nor_rt_1=mu + qnorm(0.01) * sigma.hat

# Pt 5% value at risk of return
log.Pt_1 = mu + qnorm(0.05) * sigma.hat + log(Pt[n]) #7.688656
garch_nor_pt_5=exp(log.Pt_1)*exp(sigma.hat^2/2)

# Pt 1% value at risk of return
log.Pt_1 = mu + qnorm(0.01) * sigma.hat + log(Pt[n]) #7.688656
garch_nor_pt_1=exp(log.Pt_1)*exp(sigma.hat^2/2)

#AR(0)+GARCH(1,1)
#t분포
ar.fit = Arima(r.t, order = c(0,0,0))

spec.garch.t = ugarchspec(variance.model=list(model="sGARCH",garchOrder=c(1,1)),
                           mean.model=list(armaOrder=c(0,0), include.mean=FALSE),
                           distribution.model="std")

a.t = r.t - mean(r.t)
garch.fit.t = ugarchfit(data = a.t, spec=spec.garch.t)
sigma.hat=ugarchforecast(garch.fit.t, n.ahead=1)@forecast$sigmaFor
mu= ar.fit$coef[1]

# 5% value at risk of return
garch_t_rt_5=mu + qt(0.05,df=6) * sigma.hat

# 1% value at risk of return
garch_t_rt_1=mu + qt(0.01,df=6) * sigma.hat

```

```
# Pt 5% value at risk of return
log.Pt_1 = mu + qt(0.05,df=6) * sigma.hat + log(Pt[n]) #7.688656
garch_t_pt_5=exp(log.Pt_1)*exp(sigma.hat^2/2)
```

```
# Pt 1% value at risk of return
log.Pt_1 = mu + qt(0.01,df=6) * sigma.hat + log(Pt[n]) #7.688656
garch_t_pt_1=exp(log.Pt_1)*exp(sigma.hat^2/2)
```

```
#AR(0)+GJR-GARCH(1,1) 모형
#정규분포
ar.fit = Arima(r.t, order = c(0,0,0))
gjr.garch = ugarchspec(variance.model=list(model="gjrGARCH",garchOrder=c(1,1)),
                        mean.model=list(armaOrder=c(0,0), include.mean=FALSE),
                        distribution.model="norm")
```

```
a.t=r.t-mean(r.t)
gjr.garch.fit=ugarchfit(data=a.t, spec=gjr.garch)
```

```
sigma.hat=ugarchforecast(gjr.garch.fit, n.ahead=1)@forecast$sigmaFor
mu= ar.fit$coef[1]
```

```
# 5% value at risk of return
GJR_nor_rt_5=mu + qnorm(0.05) * sigma.hat
```

```
# 1% value at risk of return
GJR_nor_rt_1=mu + qnorm(0.01) * sigma.hat
```

```
# Pt 5% value at risk of return
log.Pt_1 = mu + qnorm(0.05) * sigma.hat + log(Pt[n]) #7.688656
GJR_nor_pt_5=exp(log.Pt_1)*exp(sigma.hat^2/2)
```

```
# Pt 1% value at risk of return
log.Pt_1 = mu + qnorm(0.01) * sigma.hat + log(Pt[n]) #7.688656
GJR_nor_pt_1=exp(log.Pt_1)*exp(sigma.hat^2/2)
```

```
#AR(0)+GJR-GARCH(1,1) 모형
#t분포
```

```

ar.fit = Arima(r.t, order = c(0,0,0))
gjr.garch.t = ugarchspec(variance.model=list(model="gjrGARCH",garchOrder=c(1,1)),
                        mean.model=list(armaOrder=c(0,0), include.mean=FALSE),
                        distribution.model="std")

```

```

a.t=r.t-mean(r.t)
gjr.garch.fit.t=ugarchfit(data=a.t, spec=gjr.garch.t)
sigma.hat=ugarchforecast(gjr.garch.t, n.ahead=1)@forecast$sigmaFor
mu= ar.fit$coef[1]

```

```

# 5% value at risk of return
GJR_t_rt_5=mu + qt(0.05,df=7) * sigma.hat

```

```

# 1% value at risk of return
GJR_t_rt_1=mu + qt(0.01,df=7) * sigma.hat

```

```

# Pt 5% value at risk of return
log.Pt_1 = mu + qt(0.05,df=7) * sigma.hat + log(Pt[n]) #7.688656
GJR_t_pt_5=exp(log.Pt_1)*exp(sigma.hat^2/2)

```

```

# Pt 1% value at risk of return
log.Pt_1 = mu + qt(0.01,df=7) * sigma.hat + log(Pt[n]) #7.688656
GJR_t_pt_1=exp(log.Pt_1)*exp(sigma.hat^2/2)

```

```

#result dataframe
var_rt=data.frame(var5=c(garch_nor_rt_5,garch_t_rt_5,GJR_nor_rt_5,GJR_t_rt_5),
                  var1=c(garch_nor_rt_1,garch_t_rt_1,GJR_nor_rt_1,GJR_t_rt_1))

```

#현재가 11994.26

```

var_pt=data.frame(var5=c(garch_nor_pt_5,garch_t_pt_5,GJR_nor_pt_5,GJR_t_pt_5),
                  var1=c(garch_nor_pt_1,garch_t_pt_1,GJR_nor_pt_1,GJR_t_pt_1))

```

```

var_rt;var_pt

```

#AR6 GBM

###위A1-1의데이터를이용하여 mu, sigma를추정(단위 명시)

#Pt: 11년치 data

```
r_bar<-mean(return) #일별 평균 로그수익률
```

```
s_r<-sd(return) #일별 로그수익률 표준편차
```

```
sigma.hat=sd(return) * sqrt(250) # 연 표준편차
```

```
mu.hat=mean(return) * 250 + sigma^2/2 # 연 평균
```

```
mu.hat;sigma.hat # per year
```

```
###ps = lnPs라 할 때 ps가 만족하는 모형을 적어라. 모수값도 적어야함
```

```
mu.hat-sigma.hat^2/2
```

```
sigma.hat
```

```
P0=tail(Pt,1)
```

```
log(P0)
```

```
###마지막 시점 T (대응되는 s 값은 T/250)서 조건부로 2년후의 미래값  $P_{s+2}$  의 조건부 분포를  
구하고 평균 분산을 구하시오.
```

```
#lnP3 ~N(9.374736+ 0.1467989*2, 0.04331835*2)
```

```
log(P0)
```

```
mu.hat2 = 9.374736+ 0.1467989*2
```

```
var.hat2 =0.04331835*2
```

```
exp(mu.hat2+var.hat2/2)
```

```
exp(2*mu.hat2+var.hat2)*(exp(var.hat2)-1)
```

```
###마지막 시점 T (대응되는 s 값은 T/250)서조건부로 향후 2년간의 미래수익률  $r_{s+2} = \ln P_{s+2}$   
-  $\ln P_s$ 의 조건부 분포를 구하고 평균 분산을 구하시오.
```

```
#r~ N(mu.hat - sig.hat^2/2, sig.hat^2/2)
```

```
mu.hat - sigma.hat^2/2
```

```
sigma.hat^2/2
```

```
###이자산의 마지막가격  $P_t$  를쓰시오.또만기6개월,행사가  $P_t$ 인 Europeancall option 의 공정한  
현재가를 구하시오.
```

```
P0=tail(Pt,1) #현재가
```

```
K=11786.8 #행사가
```

```
T=6/12
```

```
r=mu.hat
```

```
vol=sigma.hat
```

```
h1=(log(P0/K))+(r+vol^2/2)*T/(vol*sqrt(T))
```

```

h2=(log(P0/K))+(r-vol^2/2)*T/(vol*sqrt(T))
Ct =P0*pnorm(h1) - K*exp(-r*T)*pnorm(h2)
Ct

```

#AR7 Simulation for GARCH and GJR-GARCH

###위 A5-1에서 추정된 AR(0)+GARCH(1,1) 과 AR(0)+GJR-GARCH(1,1) 모수를 이용하여 추정된 모형을 참모형으로 간주하고 r_t, t=1..n,n=1000 simulate 하여라. 그 후 각 모형별로 skewness 를 계산하여라.

```

#AR(0)+GARCH(1,1)
#garch.fit
#omega=0.000005, alpha=0.144727, beta1=0.823771

sigma=c();epsilon=c();a=c();
n = 1000;set.seed(1); epsilon=rnorm(n);
alpha_0=0.000005; alpha_1=0.144727; beta_1=0.823771
sigma[1]=sqrt(alpha_0/(1-alpha_1-beta_1)) #uncond. variance
a[1] = sigma[1]*epsilon[1]
for (t in 2:n){
  sigma[t]= sqrt(alpha_0 + alpha_1*(a[t-1])^2+beta_1*(sigma[t-1])^2); a[t]=sigma[t]*epsilon[t]}

skew(a)

```

```

#AR(0)+GARCH(1,1) + t분포 추정
#omega=0.000003, alpha=0.147735 , beta1=0.841785, shape(df)=5.640680
sigma=c();epsilon=c();a=c();
alpha_0=0.000003; alpha_1=0.147735; beta_1=0.841785; shape=5.640680
n = 1000;set.seed(1); epsilon=rt(n,df=shape);
sigma[1]=sqrt(alpha_0/(1-alpha_1-beta_1)) #uncond. variance
a[1] = sigma[1]*epsilon[1]
for (t in 2:n){
  sigma[t]= sqrt(alpha_0 + alpha_1*(a[t-1])^2+beta_1*(sigma[t-1])^2); a[t]=sigma[t]*epsilon[t]}

skew(a)

```

#AR(0)+GJR-GARCH(1,1) 모형 + normal 분포

```
#gjr.garch.fit
#omega=0.000005 , alpha=0.018093, beta1=0.853590 , gamma1= 0.184960
sigma=c();epsilon=c();a=c();
alpha_0=0.000005; alpha_1=0.018093; beta_1=0.853590;gamma1=0.184960
n = 1000;set.seed(1); epsilon=rnorm(n);
sigma[1]=sqrt(alpha_0/(1-alpha_1-gamma1/2-beta_1)) #uncond. variance
a[1] = sigma[1]*epsilon[1]
for (t in 2:n){
  sigma[t]= sqrt(alpha_0 + alpha_1*(a[t-1])^2 + gamma1*ifelse(a[t-1]<0,1,0)*a[t-
1]^2+beta_1*(sigma[t-1])^2)
  a[t]=sigma[t]*epsilon[t]}

skew(a)
```

```
#AR(0)+GJR-GARCH(1,1) + t분포 추정
#gjr.garch.fit.t
#omega=0.000004 , alpha=0.000000 , beta1=0.849232 , gamma1=0.257918, shape=5.927278
sigma=c();epsilon=c();a=c();
alpha_0=0.000004; alpha_1=0.000000; beta_1=0.849232;gamma1=0.257918;shape=5.927278

n = 1000;set.seed(1); epsilon=rt(n,df=shape);
sigma[1]=sqrt(alpha_0/(1-alpha_1-gamma1/2-beta_1)) #uncond. variance
a[1] = sigma[1]*epsilon[1]
for (t in 2:n){
  sigma[t]= sqrt(alpha_0 + alpha_1*(a[t-1])^2 + gamma1*ifelse(a[t-1]<0,1,0)*a[t-
1]^2+beta_1*(sigma[t-1])^2)
  a[t]=sigma[t]*epsilon[t]}

skew(a)
```

###2. 위 1을 500 번 반복하여 skewness 의 평균을 구하여라.

```
#AR(0)+GARCH(1,1)
#garch.fit
n = 1000;m=500;set.seed(1);
sigma=matrix(nrow=n,ncol=m);epsilon=c();a=matrix(nrow=n,ncol=m);
alpha_0=0.000005; alpha_1=0.144727; beta_1=0.823771
for(i in 1:m){
  epsilon=rnorm(n)
```



```

sigma[1,i]=sqrt(alpha_0/(1-alpha_1-beta_1)) #uncond. variance
a[1,i] = sigma[1,i]*epsilon[1]
for (t in 2:n){
  sigma[t,i]= sqrt(alpha_0 + alpha_1*(a[t-1,i])^2+beta_1*(sigma[t-1,i])^2)
  a[t,i]=sigma[t,i]*epsilon[t]}
}
mean(apply(a,2,skew)) #-0.007062047

```

#AR(0)+GARCH(1,1) + t분포 추정

```

#garch.fit.t
n = 1000;m=500;set.seed(1);
sigma=matrix(nrow=n,ncol=m);epsilon=c();a=matrix(nrow=n,ncol=m);
alpha_0=0.000003; alpha_1=0.147735; beta_1=0.841785; shape=5.640680
for(i in 1:m){
  epsilon=rt(n,df=shape)
  sigma[1,i]=sqrt(alpha_0/(1-alpha_1-beta_1)) #uncond. variance
  a[1,i] = sigma[1,i]*epsilon[1]
  for (t in 2:n){
    sigma[t,i]= sqrt(alpha_0 + alpha_1*(a[t-1,i])^2+beta_1*(sigma[t-1,i])^2)
    a[t,i]=sigma[t,i]*epsilon[t]}
}
mean(apply(a,2,skew))

```

#AR(0)+GJR-GARCH(1,1) 모형 + normal 분포

```

#gjr.garch.fit
n = 1000;m=500;set.seed(1);
sigma=matrix(nrow=n,ncol=m);epsilon=c();a=matrix(nrow=n,ncol=m);
alpha_0=0.000005; alpha_1=0.018093; beta_1=0.853590;gamma1=0.184960
for(i in 1:m){
  epsilon=rnorm(n)
  sigma[1,i]=sqrt(alpha_0/(1-alpha_1-gamma1/2-beta_1)) #uncond. variance
  a[1,i] = sigma[1,i]*epsilon[1]
  for (t in 2:n){
    sigma[t,i]= sqrt(alpha_0 + alpha_1*(a[t-1,i])^2+beta_1*(sigma[t-1,i])^2)
    a[t,i]=sigma[t,i]*epsilon[t]}
}
mean(apply(a,2,skew))

```

```

#AR(0)+GJR-GARCH(1,1) + t분포 추정
#gjr.garch.fit.t
#omega=0.000004 , alpha=0.000000 , beta1=0.849232 , gamma1=0.257918, shape=5.927278
n = 1000;m=500;set.seed(1);
sigma=matrix(nrow=n,ncol=m);epsilon=c();a=matrix(nrow=n,ncol=m);
alpha_0=0.000004; alpha_1=0.000000; beta_1=0.849232;gamma1=0.257918;shape=5.927278
for(i in 1:m){
  epsilon=rt(n,df=shape)
  sigma[1,i]=sqrt(alpha_0/(1-alpha_1-gamma1/2-beta_1)) #uncond. variance
  a[1,i] = sigma[1,i]*epsilon[1]
  for (t in 2:n){
    sigma[t,i]= sqrt(alpha_0 + alpha_1*(a[t-1,i])^2+beta_1*(sigma[t-1,i])^2)
    a[t,i]=sigma[t,i]*epsilon[t]
  }
}
mean(apply(a,2,skew))

```