

Timeseires__HW6

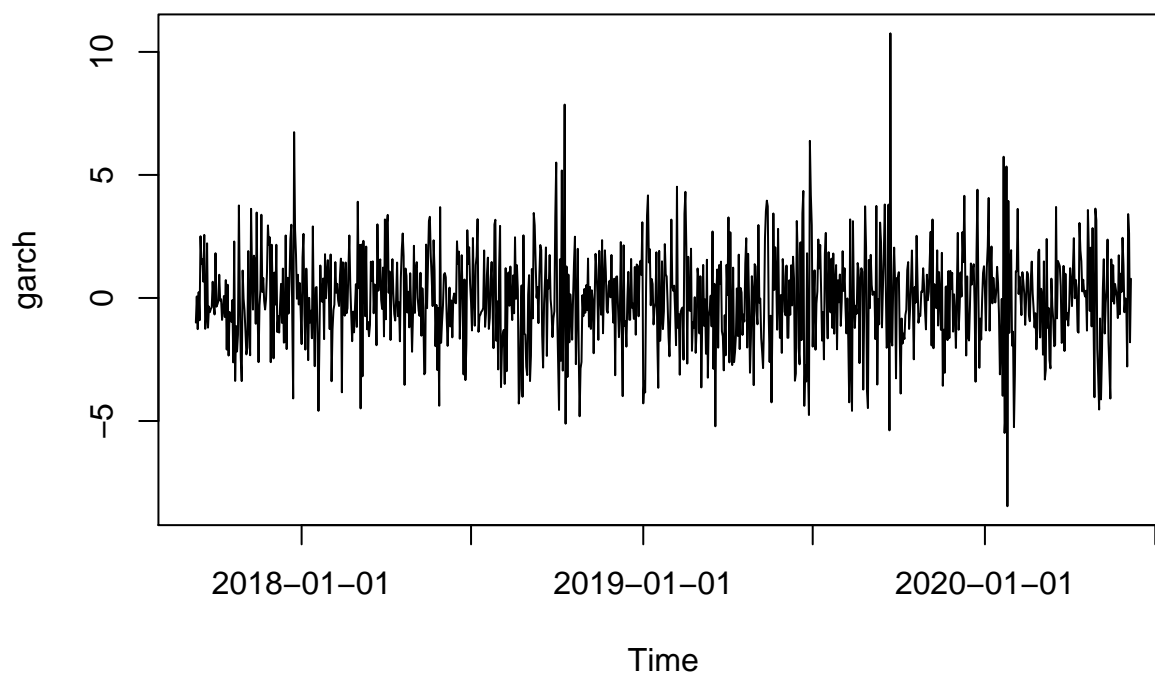
김민국

2020-06-06

5번

오차가 정규분포를 따를 때

```
g_1 <- garchSpec(model = list(omega=1, alpha=0.3, beta=0.4), rseed=1)
data_1 <- garchSim(spec = g_1, n=1000)
plot(data_1)
```

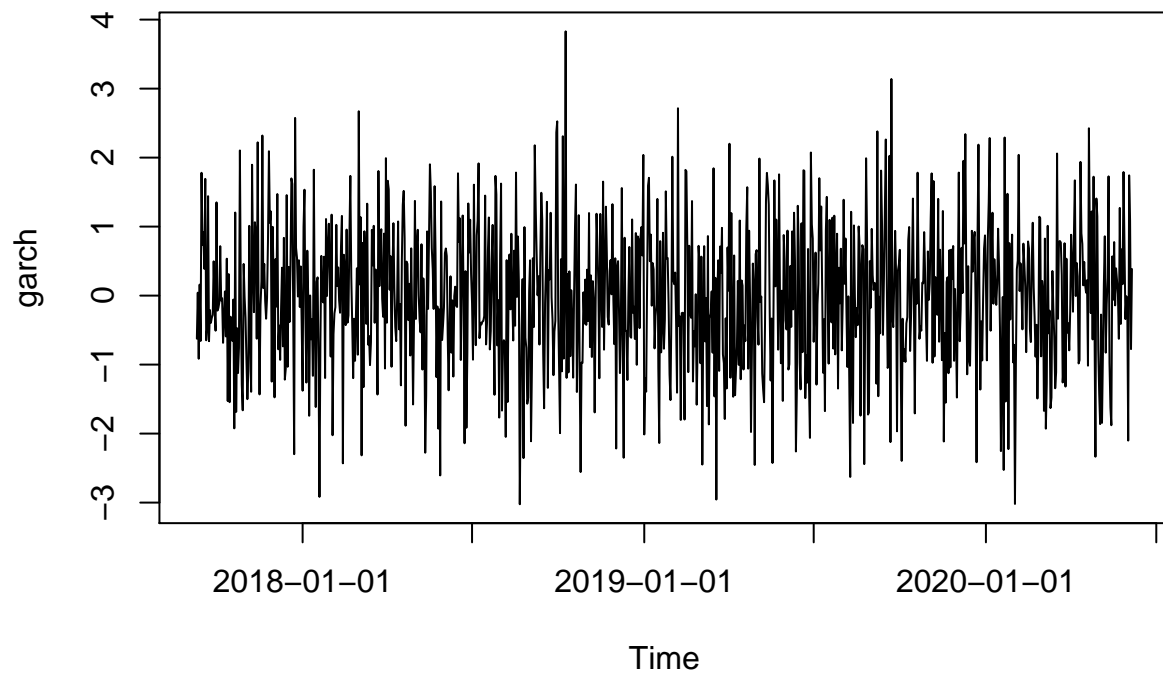


-> alpha와 beta가 적당한 수준이며 둘의 합이 1이 넘지 않는 경우이다.

-> 중간에 값이 심하게 커지거나 작아지는 부분을 제외하고서는 일반 시계열도와 큰 차이를 느낄 수는 없다.

-> 분산이 조금은 변하는 것처럼 보이지만 그 외에는 정상성도 만족하는 것처럼 보인다.

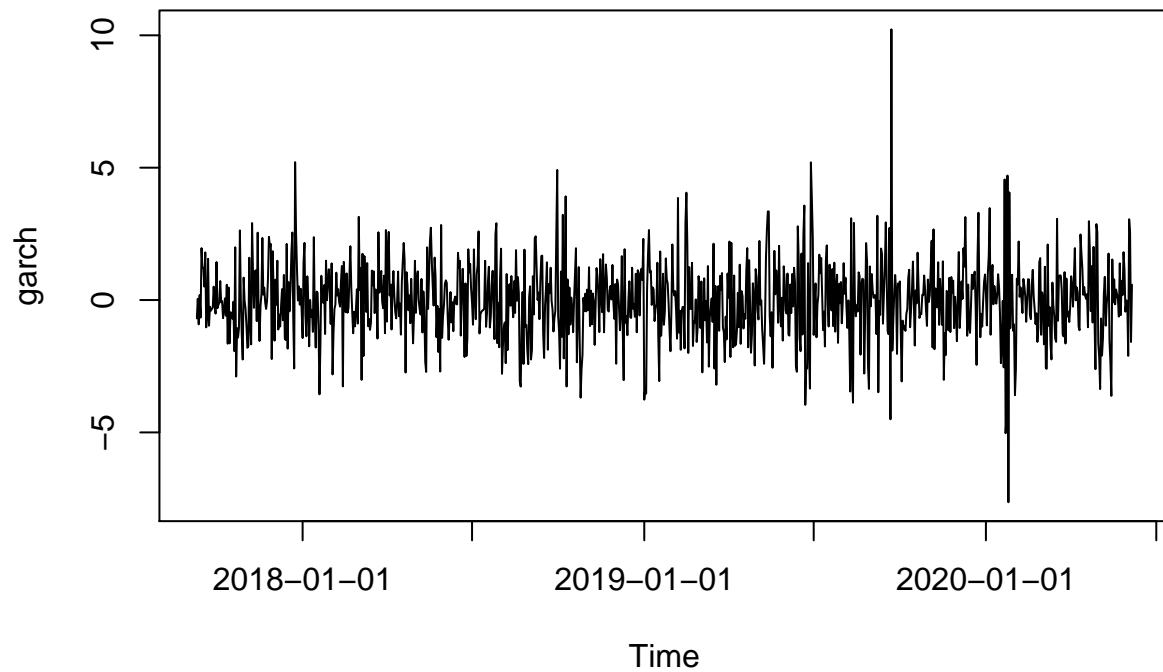
```
g_2 <- garchSpec(model = list(omega=1, alpha=0.01, beta=0.01), rseed=1)
data_2 <- garchSim(spec = g_2, n=1000)
plot(data_2)
```



->-> alpha와 beta가 매우 작은 경우이다.

-> 정상성을 만족하는 것처럼 보이며 데이터가 일정 수준을 벗어나는 것은 거의 없다고 볼 수 있다.

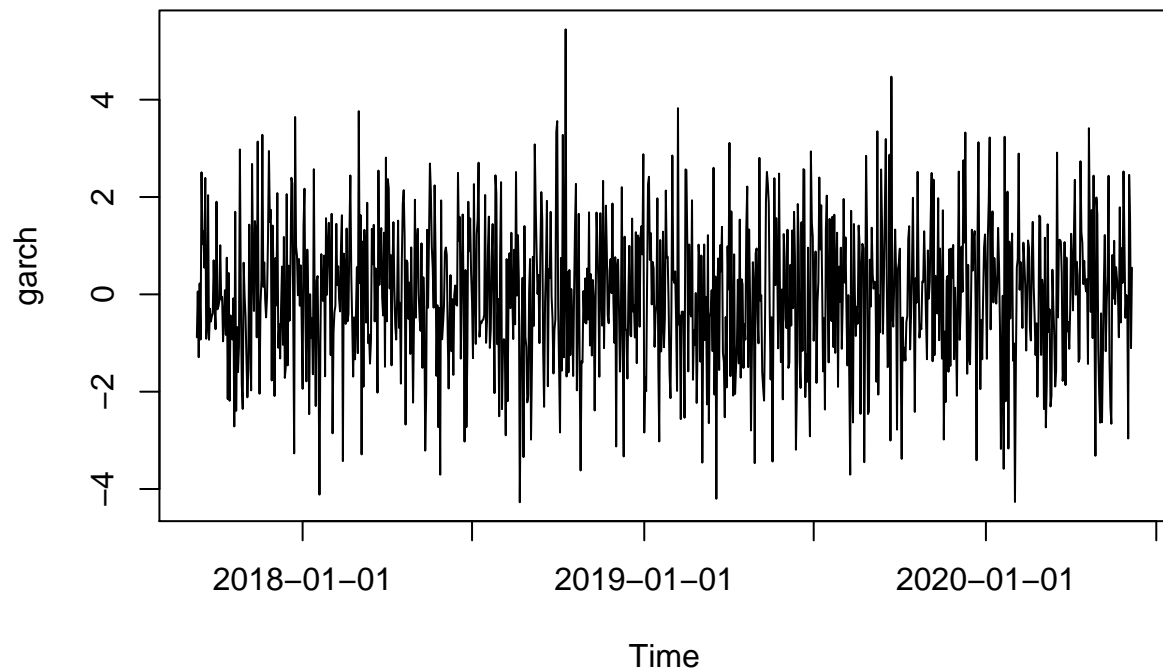
```
g_3 <- garchSpec(model = list(omega=1, alpha=0.5, beta=0.01), rseed=1)
data_3 <- garchSim(spec = g_3, n=1000)
plot(data_3)
```



-> alpha는 적당한 수준이며 beta가 매우 작을 때이다. -> 중간에 값이 급격히 커지거나 작아지는 부분이 존재하지만 그 외에는 큰 문제는 없어 보인다.

-> 중간에 값이 될 때 분산이 변하는 것을 제외하고는 정상성을 만족하는 것처럼 보인다.

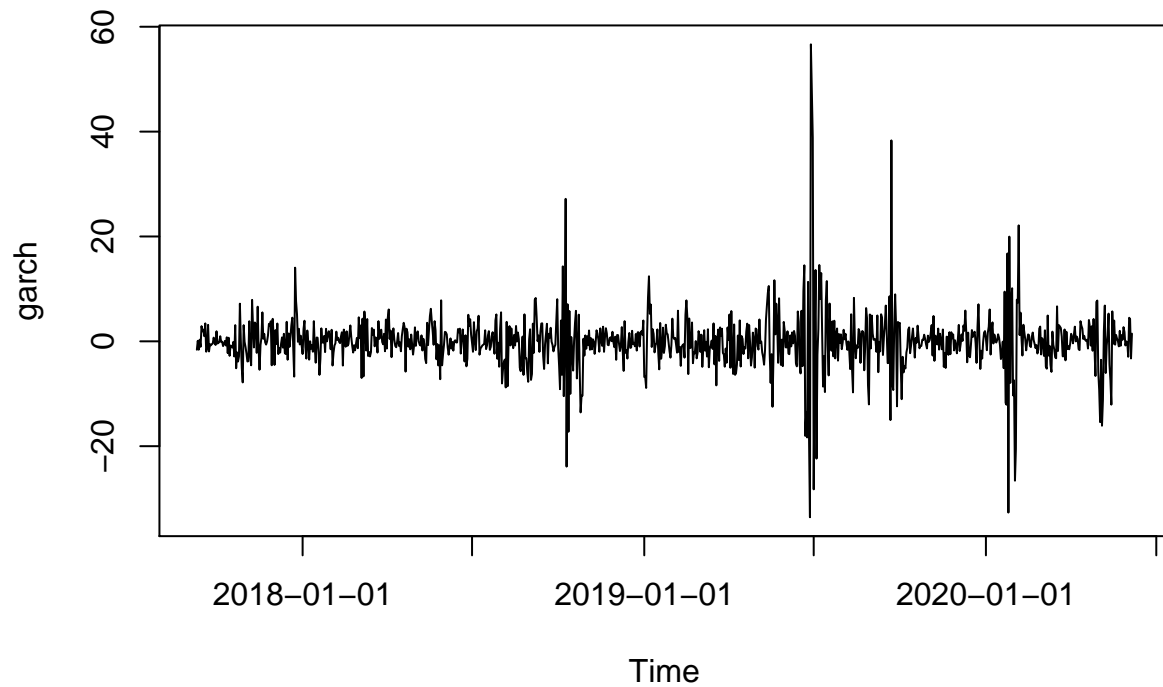
```
g_4 <- garchSpec(model = list(omega=1, alpha=0.01, beta=0.5), rseed=1)
data_4 <- garchSim(spec = g_4, n=1000)
plot(data_4)
```



-> alpha가 매우 작으며 beta가 적당한 수준인 경우이다.

-> 크게 범위를 벗어나는 값들이 보이지 않으며 정상성을 만족하는 시계열처럼 보인다.

```
g_5 <- garchSpec(model = list(omega=1, alpha=0.6, beta=0.3999999), rseed=1)
data_5 <- garchSim(spec = g_5, n=1000)
plot(data_5)
```



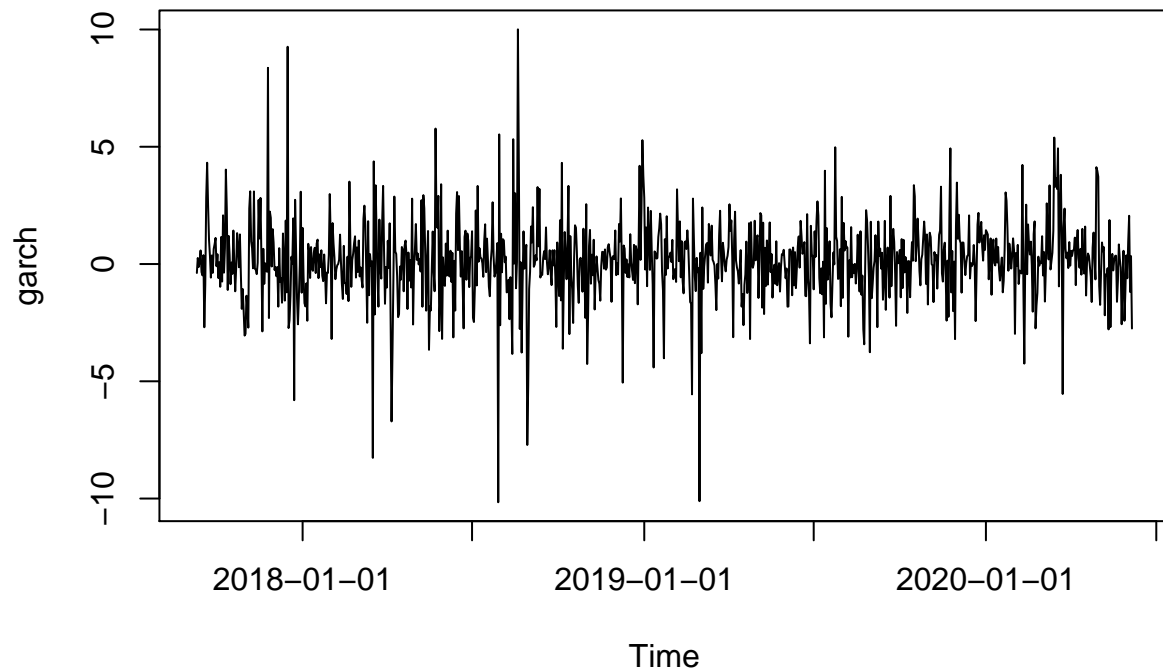
-> alpha와 beta의 합이 1에 가까운 경우이다.

-> 데이터의 값들이 범위가 매우 넓으며 분산 자체의 변화가 매우 크다.

-> 분산이 어느정도 일정해서 계속 커지는 것처럼 보이기도 한다.

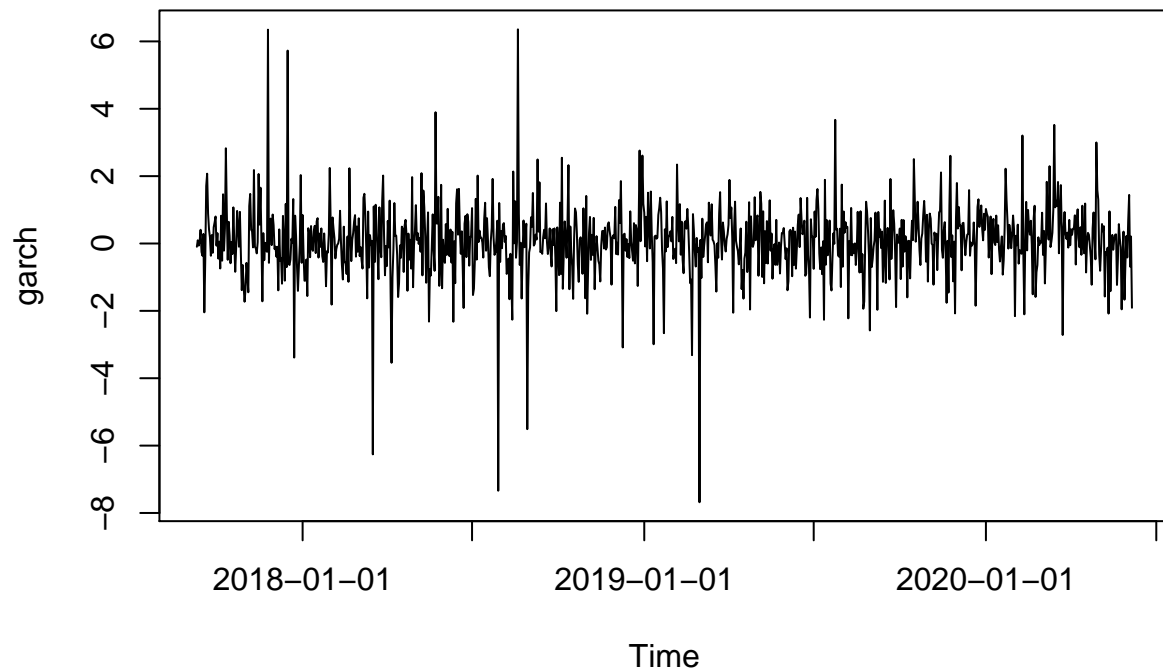
오차가 $t(4)$ 분포를 따를 때

```
g_t_1 <- garchSpec(model = list(omega=1, alpha=0.3, beta=0.4, shape = 4),
                   rseed=1, cond.dist="std")
data_t_1 <- garchSim(spec = g_t_1, n=1000)
plot(data_t_1)
```



- > alpha와 beta가 적당한 수준이며 둘의 합이 1이 넘지 않는 경우이다.
- > 중간중간 값이 비정상적으로 커지는 경우들이 존재한다.
- > 비정상적으로 변화하는 점들을 제외하고서는 정상성을 만족하는 것처럼 보인다.

```
g_t_2 <- garchSpec(model = list(omega=1, alpha=0.01, beta=0.01, shape = 4),  
                  rseed=1, cond.dist="std")  
data_t_2 <- garchSim(spec = g_t_2, n=1000)  
plot(data_t_2)
```

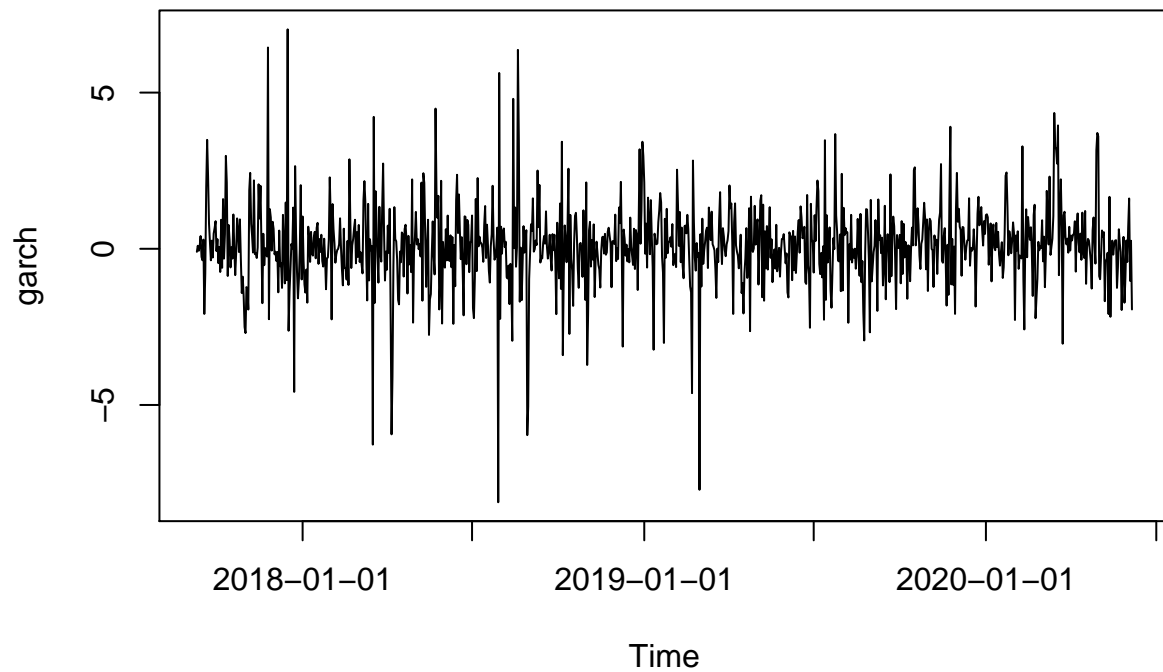


-> alpha와 beta가 매우 작은 경우이다.

-> 중간중간 값이 비정상적으로 커지는 부분이 존재한다.

-> 값이 비정상적으로 변하는 부분을 제외하고서는 정상성이 만족하는 것처럼 보인다.

```
g_t_3 <- garchSpec(model = list(omega=1, alpha=0.5, beta=0.01, shape = 4),  
                  rseed=1, cond.dist="std")  
data_t_3 <- garchSim(spec = g_t_3, n=1000)  
plot(data_t_3)
```

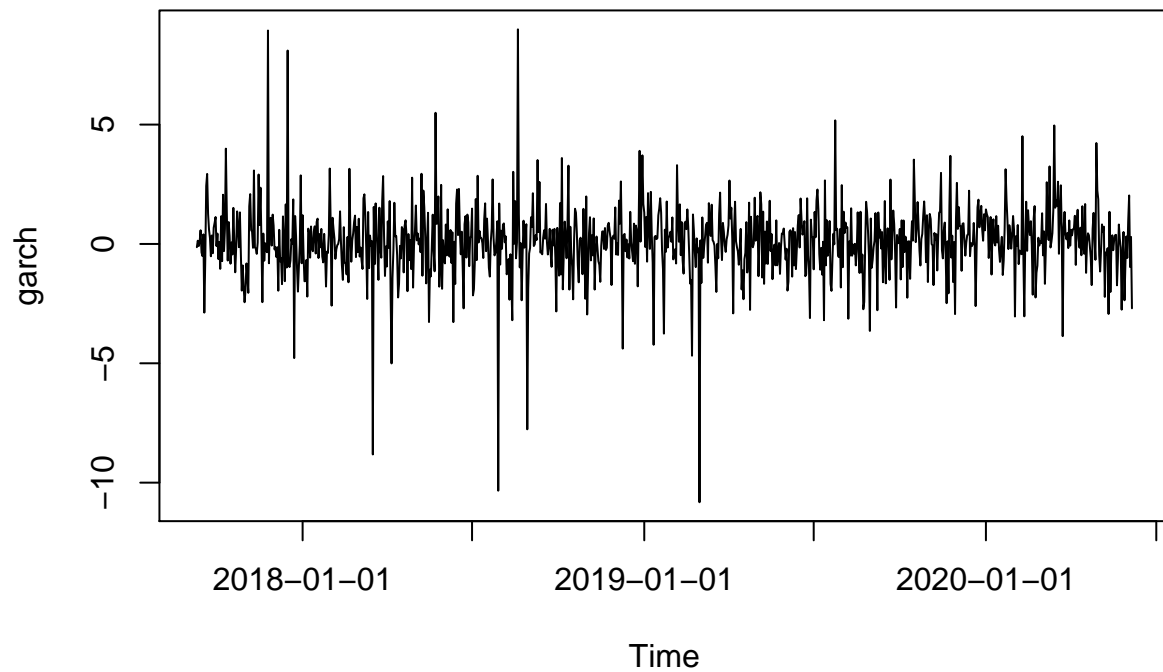


-> alpha는 적당하고 beta가 매우 작은 경우이다.

-> 중간중간 값이 비정상적으로 커지는 부분이 존재한다.

-> 값이 비정상적으로 변하는 부분을 제외하고서는 정상성이 만족하는 것처럼 보인다.

```
g_t_4 <- garchSpec(model = list(omega=1, alpha=0.01, beta=0.5, shape = 4),  
                   rseed=1, cond.dist="std")  
data_t_4 <- garchSim(spec = g_t_4, n=1000)  
plot(data_t_4)
```

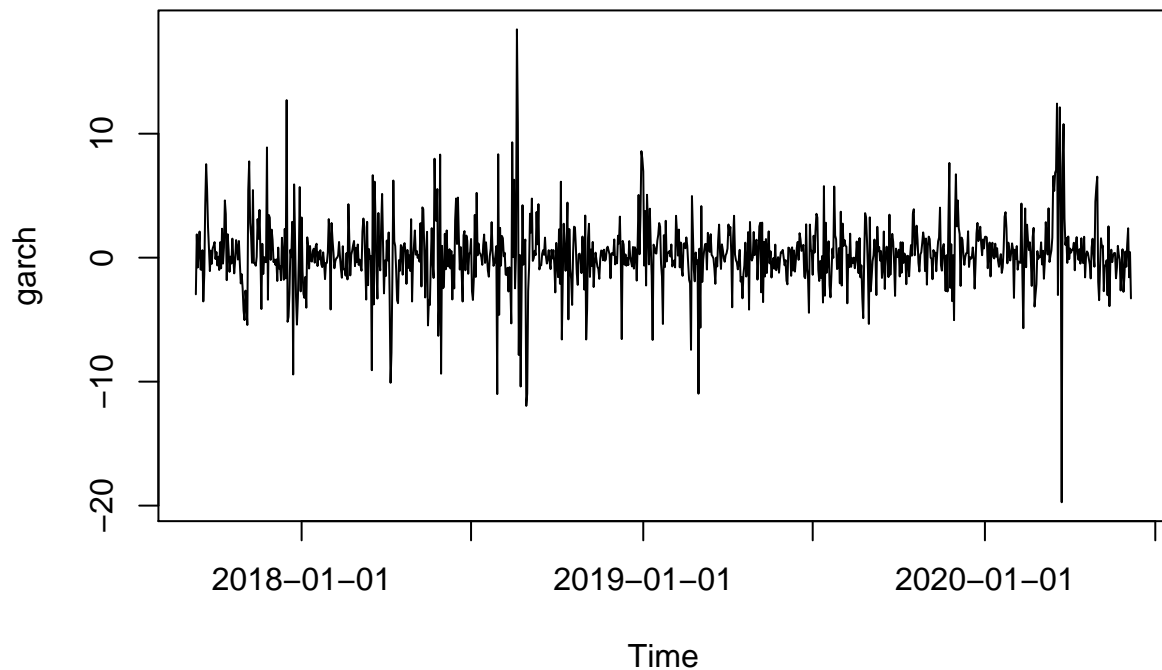



-> alpha가 매우 작고 beta는 적당한 크기인 경우이다.

-> 중간중간 값이 비정상적으로 커지는 부분이 존재한다.

-> 값이 비정상적으로 변하는 부분을 제외하고서는 정상성이 만족하는 것처럼 보인다.

```
g_t_5 <- garchSpec(model = list(omega=1, alpha=0.6, beta=0.3999999, shape = 4),  
                  rseed=1, cond.dist="std")  
data_t_5 <- garchSim(spec = g_t_5, n=1000)  
plot(data_t_5)
```



- > alpha와 beta의 합이 1에 매우 가까운 경우이다.
- > 중간중간 값이 비정상적으로 커지는 부분이 존재한다.
- > 오차가 정규분포를 따를 때보다는 값이 극단적으로 변하는 경우는 드물다.
- > 하지만 마찬가지로 값이 극단적으로 변화하는 경우가 있어서 분산 크게 변한다.

14번

```
data_14 <- read.table("ex_ch6_14.txt", stringsAsFactors = F)
data_14 <- data_14[-1,]
data_14 <- as.numeric(data_14[,2])
```

```
fit_14 <- garchFit(formula = ~ garch(1,1), data = data_14)
```

```
##
## Series Initialization:
## ARMA Model:          arma
## Formula Mean:        ~ arma(0, 0)
## GARCH Model:         garch
## Formula Variance:    ~ garch(1, 1)
## ARMA Order:          0 0
```

```

## Max ARMA Order:          0
## GARCH Order:             1 1
## Max GARCH Order:        1
## Maximum Order:          1
## Conditional Dist:        norm
## h.start:                 2
## llh.start:               1
## Length of Series:       1823
## Recursion Init:         mci
## Series Scale:           260.5341
##
## Parameter Initialization:
## Initial Parameters:      $params
## Limits of Transformations: $U, $V
## Which Parameters are Fixed? $includes
## Parameter Matrix:
##           U           V  params includes
## mu      -39.27140350  39.2714  3.92714      TRUE
## omega    0.00000100 100.0000  0.10000      TRUE
## alpha1   0.00000001   1.0000  0.10000      TRUE
## gamma1  -0.99999999   1.0000  0.10000      FALSE
## beta1    0.00000001   1.0000  0.80000      TRUE
## delta    0.00000000   2.0000  2.00000      FALSE
## skew     0.10000000  10.0000  1.00000      FALSE
## shape    1.00000000  10.0000  4.00000      FALSE
## Index List of Parameters to be Optimized:
## mu  omega alpha1 beta1
##   1    2    3    5
## Persistence:          0.9
##
##
## --- START OF TRACE ---
## Selected Algorithm: nlminb
##
## R coded nlminb Solver:
##
## 0:      2336.1382:  3.92714 0.100000 0.100000 0.800000
## 1:      2167.1206:  3.68259 0.0353926 0.0956572 0.764256
## 2:      1479.5962:  3.38732 1.00000e-06 0.128529 0.758988
## 3:      1461.0038:  3.38735 0.000597127 0.128534 0.758990
## 4:      1179.1442:  3.21333 0.000876834 0.185125 0.784630

```

```

## 5:      1082.5779:  3.08346 0.00108315 0.226968 0.803518
## 6:      820.09900:  3.02890 1.00000e-06 0.223371 0.796734
## 7:      745.58777:  3.02652 1.00000e-06 0.429238 0.605705
## 8:      743.71382:  3.02652 1.90520e-05 0.429238 0.605705
## 9:      743.11897:  3.02652 1.63482e-05 0.429244 0.605699
## 10:     742.32423:  3.02651 7.00654e-06 0.429244 0.605699
## 11:     742.26101:  3.02651 1.05536e-05 0.429250 0.605693
## 12:     742.20108:  3.02651 9.21326e-06 0.429257 0.605687
## 13:     742.18927:  3.02651 8.40386e-06 0.429264 0.605681
## 14:     742.17345:  3.02650 9.15338e-06 0.429277 0.605668
## 15:     742.15535:  3.02650 8.41540e-06 0.429305 0.605643
## 16:     742.13410:  3.02650 9.15242e-06 0.429360 0.605592
## 17:     742.10445:  3.02649 8.41617e-06 0.429470 0.605490
## 18:     724.31722:  3.02793 1.55213e-05 0.636500 0.414110
## 19:     708.56480:  2.97693 1.00000e-06 0.637218 0.413201
## 20:     674.07714:  2.96045 5.98748e-05 0.635294 0.414900
## 21:     671.14843:  2.95883 5.26597e-05 0.635105 0.415067
## 22:     637.94774:  2.95700 1.00000e-06 0.634802 0.415331
## 23:     636.68889:  2.95700 5.94199e-06 0.634802 0.415331
## 24:     636.16184:  2.95701 5.04234e-06 0.634803 0.415331
## 25:     635.65544:  2.95701 3.34904e-06 0.634803 0.415331
## 26:     635.65419:  2.95702 3.29113e-06 0.634804 0.415330
## 27:     635.62978:  2.95727 3.37021e-06 0.634829 0.415308
## 28:     635.62958:  2.95727 3.34011e-06 0.634829 0.415308
## 29:     635.62703:  2.95737 3.35340e-06 0.634820 0.415301
## 30:     634.01962:  2.95725 2.58698e-06 0.587086 0.422032
## 31:     631.66394:  2.95769 4.88165e-06 0.620601 0.387384
## 32:     626.39732:  2.95913 5.54518e-06 0.769101 0.264382
## 33:     625.15178:  2.95901 5.37293e-06 0.796853 0.238545
## 34:     622.36473:  2.95906 5.06175e-06 1.00000 0.0320043
## 35:     620.51795:  2.95870 5.17072e-06 0.913356 0.110783
## 36:     620.22422:  2.95845 5.76673e-06 0.913830 0.102574
## 37:     620.08213:  2.95854 6.20929e-06 0.921265 0.0872317
## 38:     620.06518:  2.95852 6.11230e-06 0.926412 0.0829887
## 39:     620.06028:  2.95856 5.99028e-06 0.929597 0.0815721
## 40:     620.06021:  2.95856 5.98035e-06 0.929563 0.0818657
## 41:     620.06021:  2.95856 5.98367e-06 0.929603 0.0818324
## 42:     620.06021:  2.95856 5.98469e-06 0.929626 0.0818207
## 43:     620.06021:  2.95856 5.98475e-06 0.929628 0.0818210
##

```

```
## Final Estimate of the Negative LLH:
```

```
## LLH: 10760.92    norm LLH: 5.902865
##          mu      omega      alpha1      beta1
## 770.8055403    0.4062328    0.9296281    0.0818210
##
## R-optimhess Difference Approximated Hessian Matrix:
##          mu      omega      alpha1      beta1
## mu      -22.2734018  9.917378e-01    0.6794942  -32.5534
## omega    0.9917378 -1.769846e-02 -4656.0377232  1284.6246
## alpha1   0.6794942 -4.656038e+03 -1021.6435499 -1099.8557
## beta1   -32.5533992  1.284625e+03 -1099.8556643 -2049.7120
## attr("time")
## Time difference of 0.01799321 secs
##
## --- END OF TRACE ---

## Warning in sqrt(diag(fit$cvar)): NaN이 생성되었습니다
##
## Time to Estimate Parameters:
## Time difference of 0.3159668 secs
```

```
para <- fit_14@fit$par
para
```

```
##          mu      omega      alpha1      beta1
## 770.8055403    0.4062328    0.9296281    0.0818210
```

-> GARCH(1,1)을 적합한 결과는 다음과 같았다.

-> omega는 0.4062328, alpha는 0.9296281, beta는 0.0818210이다.

```
cov_cal <- function(data, h) {
  n <- length(data)
  m <- mean(data)
  x <- data[1:(n-h)] - m
  y <- data[(h+1):n] - m
  return(sum(x * y/n))
}

var_cal <- function(data, max_h=length(data)^(1/3)) {
  n <- length(data)
  sd2_hat <- cov_cal(data, 0)
  for(i in 1:max_h) {
    sd2_hat <- sd2_hat + 2*(cov_cal(data, i))
  }
}
```

```

    return(sd2_hat)
}

CUSUM_cal <- function(data) {
  ## return : maximum cusum test statistics and change point
  n <- length(data)
  cusum <- abs((cumsum(data) - (1:n)/n*sum(data) ) / ( sqrt(n) * sqrt(var_cal(data))))
  argmax <- which.max(cusum)
  if(max(cusum)>1.358) return(list("CUSUM_statistics" = max(cusum), "change_point"=argmax))
  else return(print("no change"))
}

```

```
CUSUM_cal(data_14)
```

```

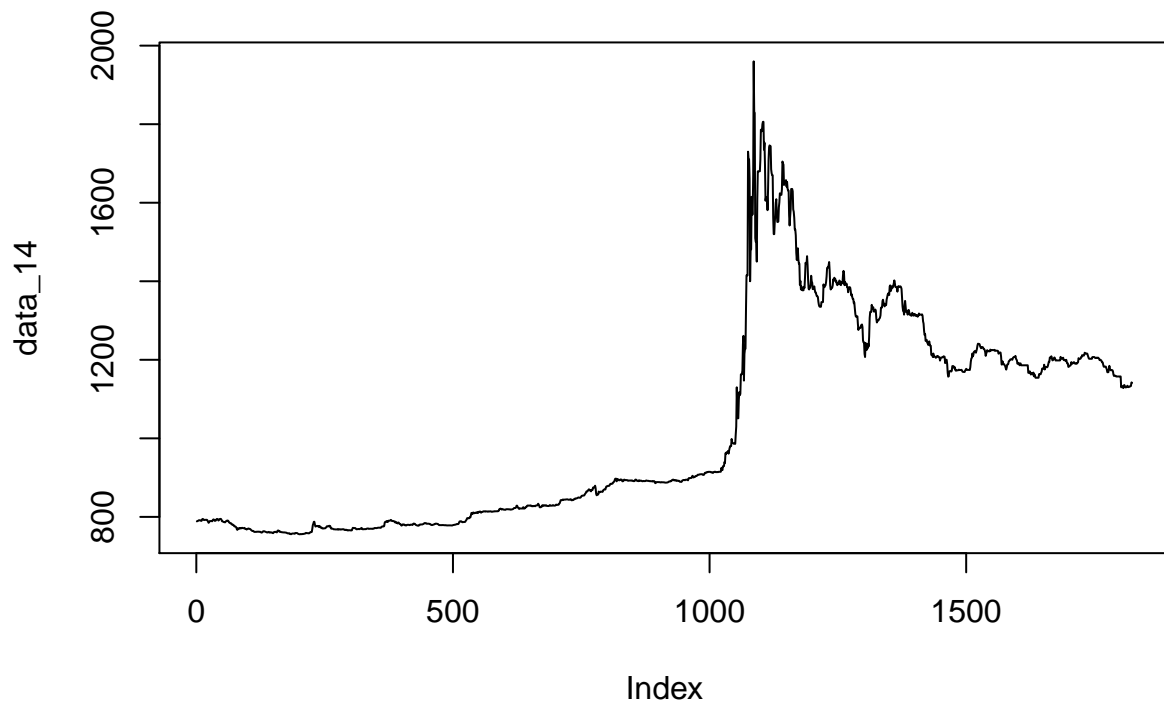
## $CUSUM_statistics
## [1] 3.854989
##
## $change_point
## [1] 1051

```

-> 실습시간에 이용한 cumsum test를 이용한 결과이다.

-> 유의수준 5% 수준에서 1051번째 데이터에서 모수가 변화하는 것을 알 수 있다.

```
plot(data_14, type = "l")
```



-> 실제 data를 plotting 해본 결과 1000번 쯤 이후에서 급격히 값이 증가하는 부분이 존재.

-> 이 때 모수의 변화가 있는 것인데 이 지점이 1051번째 데이터임을 검정을 통해 확인할 수 있었다.