1\. a) empirically, the market reacts more extreme to the negative shocks compared to the positive shock. So manager's expectation on negative skewness is correct. Since the volatility in the market tends to be larger after the negative shock, no matter how big 'n' is, there will be heteroscedastic volatility movement.

b) If we reintroduce the volatility to be heteroscedastic with ARCH(p) or GARCH(p,q) model.

2\. a) $E(X_t^4|\mathcal{F}_{t-1}) = 3\left[E(X_t^2|\mathcal{F}_{t-1})\right]^2$. given $X_t = \sigma_t \varepsilon_t$

$$E(X_t^2|\mathcal{F}_{t-1}) = \sigma_t^2$$

$$= a_0 + a_1 X_{t-1}^2 + b_1 \sigma_{t-1}^2$$

$$= 3E\big[a_0 + \underbrace{a_1 X_{t-1}^2 + b_1 \varepsilon_{t-1}^2}_{\text{identical to } E(X_{t-1}^2)}\big]^2$$

Rearranging ... $= 3(a_0^2 - (a_1+b_1)^2)$

we need $a_0^2 > (a_1+b_1)^2$ (stationary).

b) kurtosis: $\dfrac{3(a_0^2-(a_1+b_1)^2)}{a_0-(a_1+b_1)^2-2a_1^2}$, numerator > denominator hence greater or equal to 3. (Heavy tail).

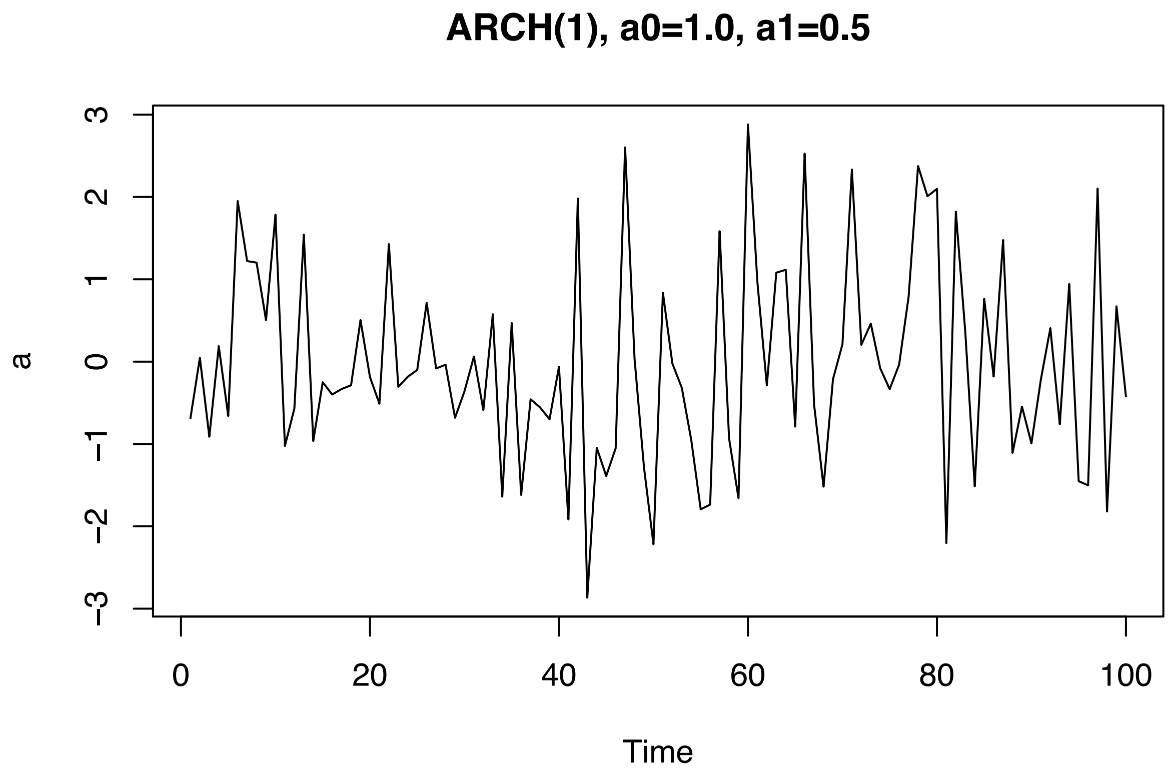
if $a_1 \to 0$, GARCH(1,1) model tends to have kurtosis of 3.

EX4.R

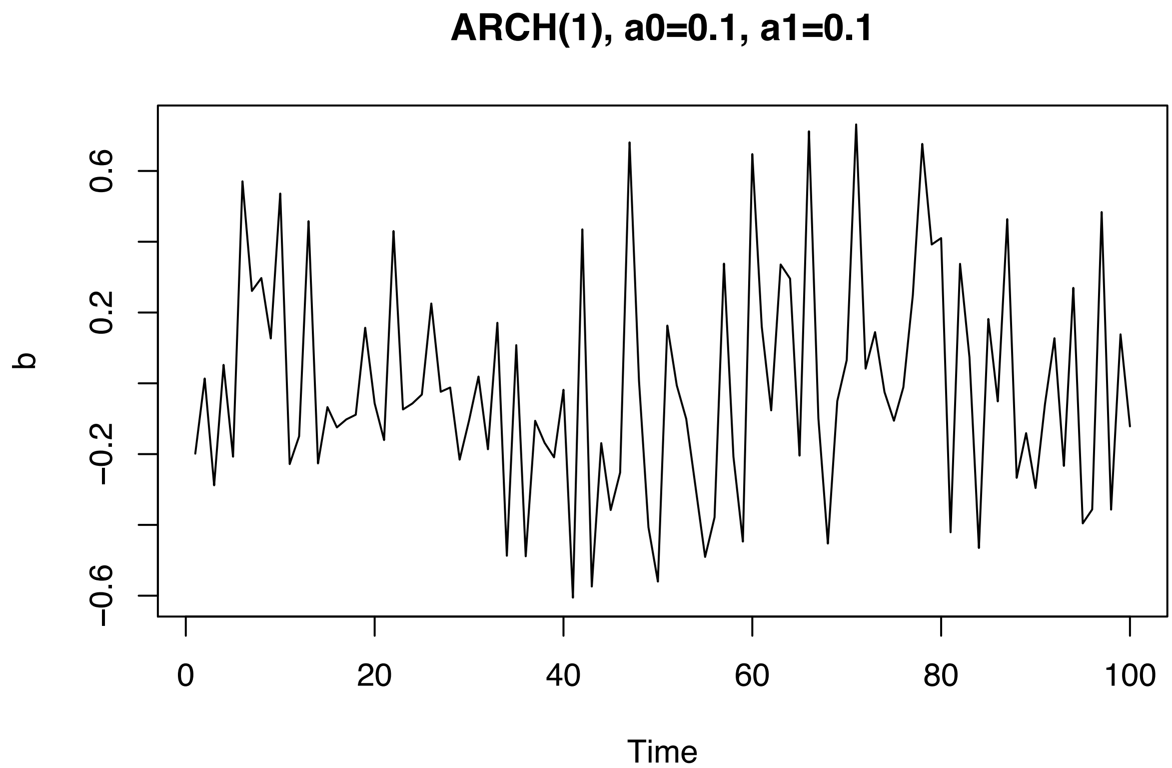
joonkang

2021-02-10

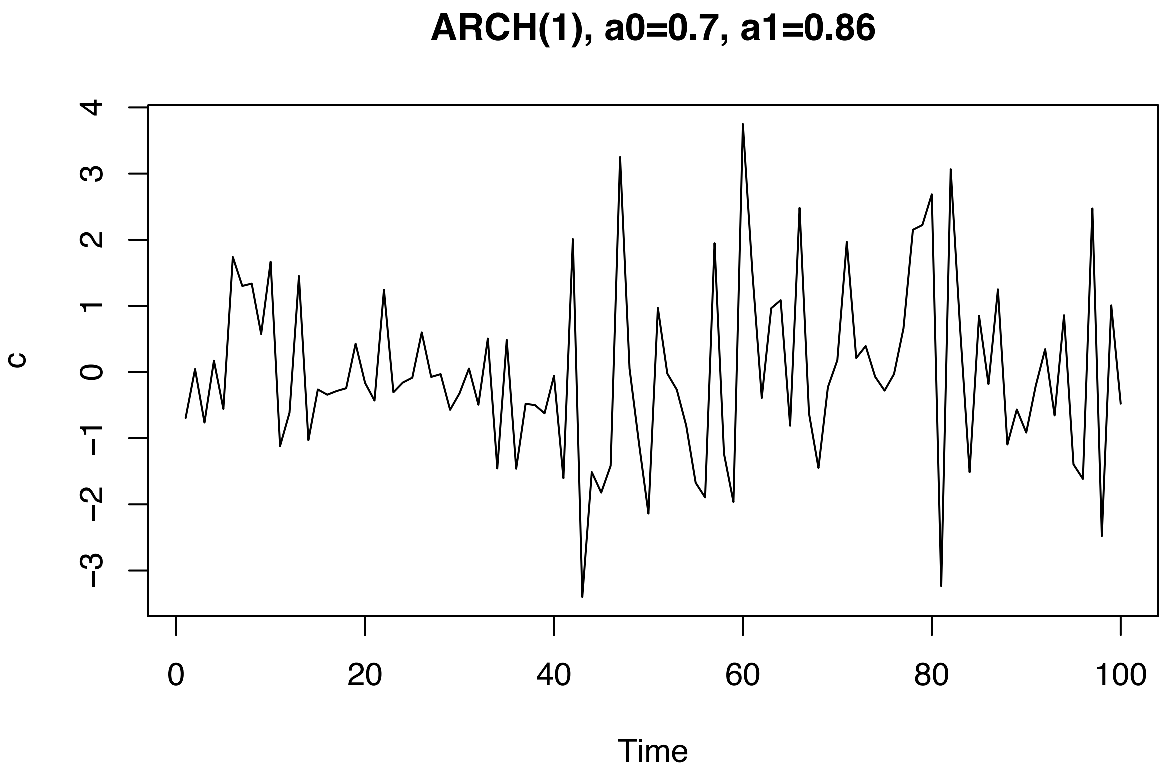
```r
#Q3.a
simulate_arch = function(alpha, n=100){
  #' ARCH process simulator
  #' @param alpha (vector): vector of ARCH coefficients including omega as the first element.
  #' @param n (int): sample size
  #' @return ARCH time series of size n

  q=length(alpha)-1
  total.n=n+100
  e=rnorm(total.n)
  x=double(total.n)
  sigt=x
  sigma2=alpha[1]/(1-sum(alpha[-1]))

  if(sum(alpha[-1])>1) stop("Infinite Variance")
  if(sigma2<=0) stop("Negative Variance")

  x[1:q]=rnorm(q,sd=sqrt(sigma2))

  for (i in (q+1):total.n)
  {
    sigt[i]=sum(alpha*c(1,x[i-(1:q)]^2))
    x[i]=e[i]*sqrt(sigt[i])
  }
  return(invisible(x[(100+1):total.n]))
}

#Q3.b
set.seed(1)
# a0 = 1, a1 = 0.5
a = simulate_arch(alpha=c(1,0.5), n=100)
plot.ts(a, main="ARCH(1), a0=1.0, a1=0.5")
```




```r
set.seed(1)
# a0 = 0.1, a1 = 0.1
b = simulate_arch(alpha=c(0.1, 0.1), n=100)
plot.ts(b, main="ARCH(1), a0=0.1, a1=0.1")
```




```r
set.seed(1)
# a0 = 0.7, a1 = 0.86
c = simulate_arch(alpha=c(0.7, 0.86), n=100)
plot.ts(c, main="ARCH(1), a0=0.7, a1=0.86")
```


ARCH(1), a0=0.7, a1=0.86
c
4
3
2
1
0
−1
−2
−3
0
20
40
60
80
100
Time