HW - 이분산성.

1. 1973-2008 월별 Uncertainty Index data (UI_HW_1.csv) 에대해 다음의 분석을 수행하자.

모형 iqr_gip ~ recession을 추정하고 recession 때 불확실성 지수가 얼마 더 높아지는지 (δ) 를 추정하고자한다.

(1) 이 추정치의 OLSE $\bar{\delta}$ 를 구하여라

```
UI.data = read.csv("UI_HW_1.csv", header = T)
attach(UI.data)
ols.fit = lm(iqr_gip ~ recession)    #0.04649
```

(1-1) $\bar{\delta}$ 의 OLS 표준오차를 구하여라

```
summary(ols.fit)$coef[2,2]   #0.0035
```

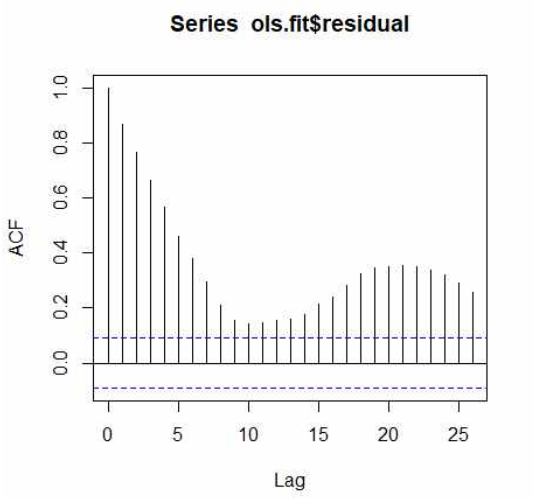
(1-2) $\bar{\delta}$ 의 HC 표준오차를 구하여라

```
library(sandwich)
sqrt(vcovHC(ols.fit)[2,2])    #0.00466
```

(1-3) $\bar{\delta}$ 의 HAC 표준오차를 구하여라

```
sqrt(vcovHAC(ols.fit)[2,2])    #0.0127
```

(1-4) 위 (1-1), (1-2), (1-3) 중 어느것이 가장 적절한지를 설명하시오



1. OLS 잔차의 acf를 그려보면 좌측과 같이 ±2 표분오차 밴드를 벗어나는 것이 많아 회귀모형의 오차항은 매우 강한 자기상관을 가짐을 알수 있다.
acf(ols.fit$residual)
2. HAC.se 가 HC.se 보다 훨씬 큰것은 이분산성보다는 자기상관성(과 조건부이분산성)이 훨씬 더 뚜렷함을 시사한다.

위 1,2 의 이유로 HAC.se 가 가장 적절하다.

(2) (2-1)recession 기간과 expansion 기간 분산 이 차이가 나는지 분산비 검정 (F 검정)을 수행하자.

```
var.U = var(iqr_gip[recession==0]); n.U = length(iqr_gip[recession==0])
var.D = var(iqr_gip[recession==1]); n.D = length(iqr_gip[recession==1])
F=var.U / var.D          #0.4259
qf(0.025,n.U-1,n.D-1,lower.tail=TRUE)       #0.71,  critial value
qf(0.975,n.U-1,n.D-1,lower.tail=TRUE)       #1.47,  critical value
```

F 값이 critical value 범위를 벗어나므로 등분산 가설 기각됨.

(2-2) 위 (2-1) 의 이분산성을 감안하여 그룹핑 방법으로 이분산성하 GLS 추정치 $\hat{\delta}$ 를 구하고 이의 GLS standard error를 구하시오

```
Y.str = ifelse(recession==0, iqr_gip/sqrt(var.U), iqr_gip/sqrt(var.D))
X.str = ifelse(recession==0, recession/sqrt(var.U), recession/sqrt(var.D))
I.str = ifelse(recession==0, 1/sqrt(var.U), 1/sqrt(var.D))
lm.fit = lm(Y.str ~ X.str + I.str -1)       #GLSE =0.046   GLS.SE=0.004634
```

2. 시뮬레이션

다음 모형을 고려하자.

$y_t = \beta_0 + x_t\beta + u_t, t = 1, \ldots, n, n = 2000$

$u_t = x_t^{\alpha} a_t, \quad a_t \sim \ iid \ N(0,1)$

$x_t \sim \quad iid \quad N(0,1)$

$\beta_0 = 0, \ \beta_1 = 0, \quad \alpha = 1$

(1) 모든학생이 동일한 난수를 발생시키기 위해 set.seed(1)을 수행한다.

(2) 위 모형에서 데이터 $\{x_t, y_t, t = 1, \ldots, n\}$ 을 발생시켜 β_1 의 OLSE b_1 과 OLS 표준오차 ols.se(b_1), HC 표준오차 HC.se(b_1) 을 구한다.

(3) 위 (2)를 1000 번 반복하여

(3-1) 1000 개 OLSE b_1 의 평균을 구한다.

평균: -0.001898 의미: 불편이다.

(3-2) 1000 개 OLSE b_1 의 표준편차를 구한다.

표준편차 0.03868 의미: OLSE b_1 의 정밀도

(3-3) 1000 개 ols.se(b_1) 의 평균을 구한다.

평균: 0.0223 의미: OLSE b_1 의 표준편차 추정치 의평균

(3-4) 1000 개 HC.se(b_1) 의 평균을 구한다.

평균: 0.0386 의미: 이분산성을 감안하여 추정한 OLSE b_1 의 표준편차의평균

(4) α 를 모른다고 가정하고 $var(u_t) \propto |x_t|$ 이라는 가정 하에 GLSE (즉 weighted least squares estimator) $\widehat{\beta_1}$ 를 1000 번 구하라. 그후 1000 개 GLSE $\widehat{\beta_1}$ 의 평균 과 표준편차 를 구한다. (등분산 변환은 데이터를 $\sqrt{|x_t|}$ 로 나누는 것이다.)

평균: -0.000456 의미: GLSE 는 불편이다

표준편차: 0.0279 의미0.0279<0.0386이므로 GLSE 가 OLSE 보다 더 정밀하다.

힌트: 위 (3-1) 과 (4)에서 평균의 의미를 불편성 측면에서 설명하시오.

힌트: 위 (4)에서 표준편차를 위 (3-2) 에서의 표준편차와 비교하여 효율성 측면에서 설명하시오.

set.seed(1)을 수행하지 않으면 오답처리됨.

```
library(sandwich)

n.data = 2000 # 발생시킬 데이터 수
n.sim = 1000 # simulation 반복 수


alpha = 1
 beta = 0
 ols.beta = c(); ols.se = c(); HC.se=c(); t.value = c()
 x.s = c(); y.s=c(); gls.beta=c()
 for(i in 1:n.sim){
 x = rnorm(n.data)
 a = rnorm(n.data)
 u = x^alpha * a
 y = x*beta + u
 lm.fit = lm(y ~ x)
 ols.beta[i] = summary(lm.fit)$coef[2,1]
 ols.se[i] = summary(lm.fit)$coef[2,2]
 HC.se[i]=sqrt(vcovHC(lm.fit)[2,2])
 x.s = x/sqrt(abs(x)); y.s=y/sqrt(abs(x))
 gls.beta[i]=lm(y.s ~ x.s)$coef[2]
 }

mean(ols.beta); sd(ols.beta)
mean(ols.se); mean(HC.se)
mean(gls.beta); sd(gls.beta)
```