**Boston University Questrom School of Business**

**MF840 – Fall 2021**

Eric Jacquier

**Problem Set 1**

**Due Wednesday February 10th 10:00pm**

* Do the Problem Set in groups of one or two
* Submit the problem via GradeScope – indications will be emailed.
* **To get a check, you need to answer all the questions.**
* **All discussion and theoretical questions must be hand written to count**.
* **All R code must be shown at the end of the Problem Set**

Use this template – make sure to write in the blanks and not modify template

LAST NAME FIRST NAME

1 Wang Zuhua

2 Wang Chenwei

**Problem 1:** Feasible GLS

The file sizebtm-week.csv contains the weekly returns on 6 US stock portfolios “*double sorted*” by 2 size and 3 book-to-market ratios groups. See Ken French’s web site for explanations. For this problem you will use the portfolio smbm3 (small size/ high book to market). We want to estimate the portfolio beta and alpha by regressing the **excess** return of the portfolio on the excess return of the cap-weighted market. Use the weekly returns from 1981 to 1990 included. Don’t forget that KF’s data are in %. I added Rm-Rf and Rf to the file.

a) Estimate the portfolio’s (α , β) by OLS. In **Table 1** put the estimates and OLS standard errors. In a second row, add (both types of) HAC standard errors.

**Table 1: OLS and Standard Errors**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Estimate | OLS std. err. | HET std. err. | HAC std. err. |
| β | 0.00067124  0.70650089 | 0.00039071  0.01830426 | 0.00041133  0.05014938 | 0.00046913  0.05049130 |

On the view of the OLS and HAC standard errors do you suspect a potential GLS situation?

On the view of the OLS and HAC standard errors, I think proper GLS should have a better estimate because the OLS standard error for β is much smaller than HAC’s.

b) Autocorrelation

* Using the “Acf” command in the forecast package, plot in **Figure1** the ACF and PACF of the residuals.

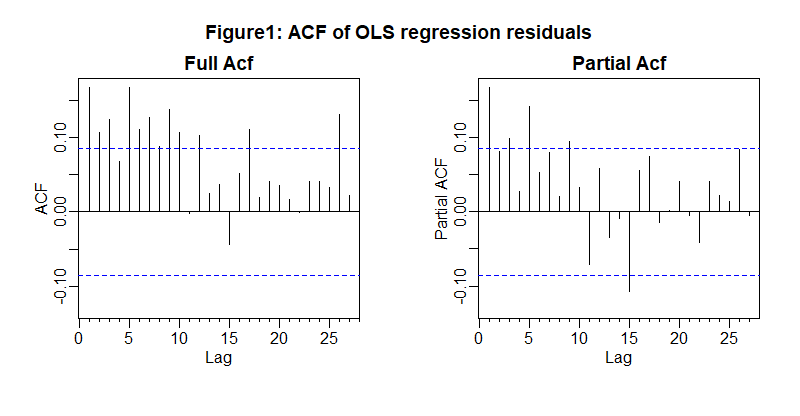


Figure 1 here

* What would be a likely time series model for the residuals?(no need for a formal model search).

It could be AR(3)

* In **Table 2**, report the first four autocorrelations and their standard errors.

|  |  |  |  |
| --- | --- | --- | --- |
| Lag 1 | Lag 2 | Lag 3 | Lag 4 |
| 0.17 | 0.11 | 0.13 | 0.07 |

**Table 2: First 4 ACF**

The standard error is: T-0.5= 522-0.5=0.438.

* Construct the for a feasible GLS allowing for autocorrelation. Use an AR(3). That is, just use directly the autocorrelations you estimated to compute a TxT correlation matrix C. and then the TxT covariance matrix of the residuals
* Write in your report what the first row of C is, and the residual variance you will use.

The first row of C is:

The first row of C is [1.0000 0.1676 0.1075 0.1253 and 518 zeros]

The residual variance is:

Residual variance is 7.9267e-05.

Tip: To get the covariance matrix, you also need to estimate the standard deviation of the noise and put it in a diagonal matrix D, then you can use the relation:  Ω = D C D.

Tip: To create C, you need to write the ACF up to lag created by an MA3. You can do this with the command "toeplitz". If you forgot what a Toeplitz matrix is, look it up.

xx <-c( values of the first row of the matrix )

cc <-toeplitz(xx) # Voila! Check that it works, read the manual

* Write the 1st GLS estimates and standard errors in **Table 3**
* Set up the iterated GLS you will stop when the slope estimate changes by less than 0.01
* Write the first 3 estimates and standard errors in **Table 3**, as well as the final estimates and standard errors
* How many iterations did you need?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | α | σα | β | σβ |
| First GLS | 0.0430 | 0.0442 | 0.6742 | 0.0178 |
| After 1 iter. | -0.0006 | 0.0004 | 0.6956 | 0.0181 |
| After 2 iter. | -0.0345 | 0.0445 | 0.7076 | 0.0183 |
| After 3 iter. | 0.0001 | 0.0004 | 0.6939 | 0.0183 |

**Table 3: GLS estimate for 3 iterations**

We needed \_\_\_\_\_5\_\_\_\_\_\_\_ iterations

c) Heteroskedasticity

* For your diagnostic of heteroskedasticity, plot in **Figure 2** the acf of the **absolute values** of the residuals and a Normal Probability plot of the residuals. Conclusion?

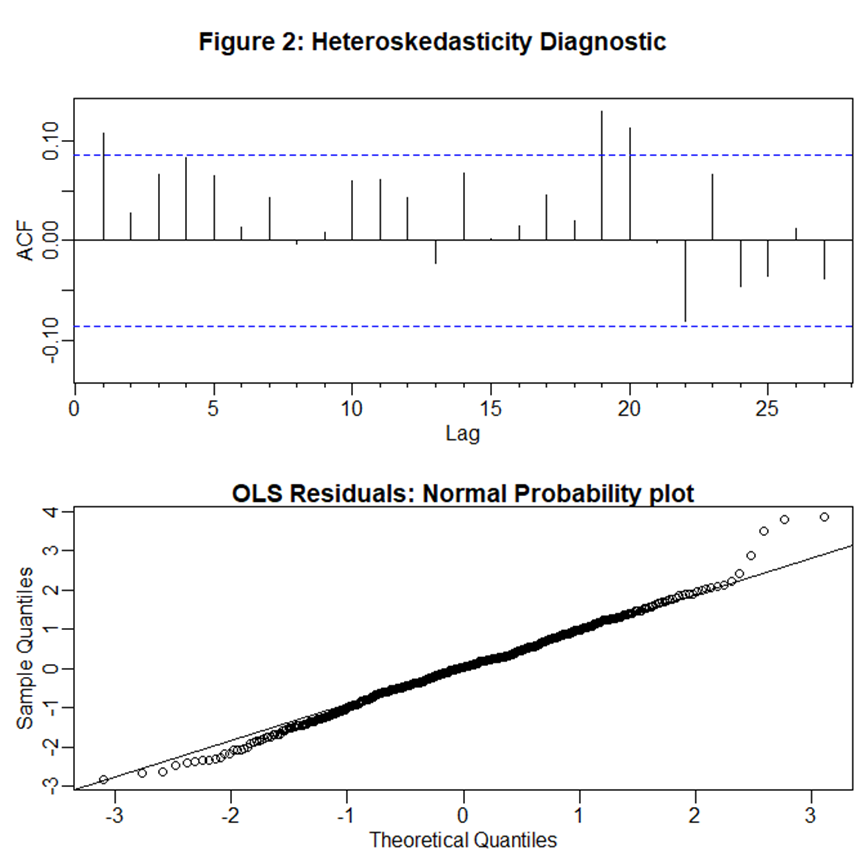


Figure 2 here

Plot a) shows a very weak persistence of absolute values which is not a significant sign of predictable time varying variance.

Plot b) shows that the strong nonnormality of residuals.

* Estimate a GARCH(1,1) on the residuals of the OLS regression. Report the parameter estimates.

ht = ω + α e2t-1 + β ht-1

Parameter estimate: \_0.00001293\_\_\_\_\_ 0.24 0.61

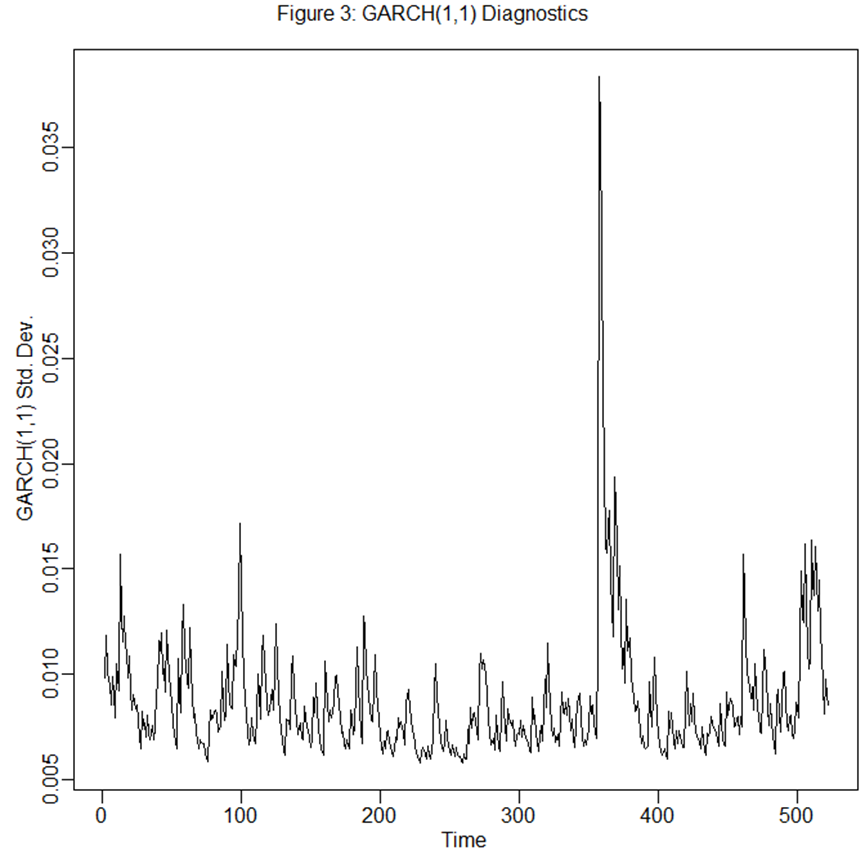
* In **Figure 3**, plot the time varying standard deviations of the residuals.

Figure 3 here

* Use these {σt}s to construct the Ω matrix and a feasible GLS with GARCH(1,1) errors.
* Write the β estimates and standard errors to the OLS results table.
* Iterate as in question b).

Tip: Use the command garchFit from the fGarch package. garchFit uses the @ symbol not the $ symbol for elements of a list. For example, after saving your garch in “mymodel”, you use: [mymodel@sigma.t](mailto:mymodel@sigma.t) or [mymodel@h.t](mailto:mymodel@h.t). You can also use the garch command from the tseries package.

Tip: Recall that the GARCH(1,1) can't estimate a variance for the first observation since it is akin to an ARMA(1,1) in the squares. For the first observation, use the unconditional variance instead. That allows you to have a TxT matrix. Since GARCH gives you variance estimates, you directly have the diagonal TxT covariance matrix.

* Now implement the iterated GLS. Save the estimate at each iteration. Stop the iteration when the β estimates changes by less than 0.5%. Write a **Table 4** similar to Table 3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | α | σα | β | σβ |
| Iter. 1 | 0.000666972 | 0.000349632 | 0.6800181 | 0.01739321 |
| Iter. 2 | 0.0007 | 0.0003 | 0.6736 | 0.0173 |
| Iter. 3 | 0.0007 | 0.0003 | 0.6722 | 0.0172 |
|  | 0.0007 | 0.0003 | 0.6717 | 0.0172 |

**Table 4: GLS estimate for 3 iterations of heterokedasticity**

We needed \_\_\_\_12\_\_\_\_\_ iterations

d) Explain but do not do it, how you would implement a feasible GLS accounting for both autocorrelation and heteroskedasticity.

I think we can do autocorrelation and heteroskedasticity in two different steps. For example, we can do MA(3) iteration and then do GARCH(1,1) iteration.

**Problem 2:** MLE estimation with heteroskedasticity

Stock returns are independently distributed: rt ~ N(μ ,σt), where σt is a simple function of an observable strictly positive variable xt: = σ2 xt.

a) You collect T independent returns rt’s and the corresponding xt’s. Write the log-likelihood of the parameters given the data: l(μ , σ | R , X).

b) Maximize it to compute , and then. Show your proof and result

c) Explain in a few words how this is like a weighted least squares estimator. What are the weights? What observations are down- or over- weighted in the estimator, relative to the basic sample average?

d) Use the Cramer-Rao lower bound to compute the asymptotic variance of .

e) Compute the variance of the basic equal-weighted sample mean, , given the heteroskedasticity in the data.

f)Compare the variances of and the basic equal-weighted . You can use the results in the handout AM-GM-HM-inequalities.pdf to conclude.

g) In words, how would you efficiently estimate a mean stock return if you knew that it has time varying variance that follows a GARCH(1,1) process?

**R CODE**

**Problem 1**

**install.packages("expm", repos="http://R-Forge.R-project.org")**

**library(expm)**

**library(matlib)**

**install.packages("fGarch")**

**library(sandwich)**

**library(lmtest)**

**install.packages("tseries")**

**data<-read.csv('sizebtm-week.csv')**

**data<-data/100**

**model<-lm((smbm3-RF) ~ Mkt.RF , data)**

**#OLS**

**coeftest(model)**

**# BOTH Heteroskedasticity and Autocorrelation**

**coeftest(model,vcov=vcovHC) # Newey and West**

**coeftest(model,vcov=vcovHAC) # Andrews (1991)**

**require(forecast) # A better ACF plot**

**par(mfrow=c(1,2),mgp=c(1.5,0.5,0))**

**Acf(model$residuals,main="");title(line=0.5,"Full Acf")**

**Acf(model$residuals,main="",type="partial");title(line=0.5,main="Partial Acf")**

**title(outer=T,"Figure1: ACF of OLS regression residuals",line=-2)**

**acf(model$residuals,plot=F)$acf[1:5]**

**olsres=model$residuals**

**sdres <- sd(olsres)\*sqrt(521/522)**

**cormat<- toeplitz(c(1,acf(olsres,plot=F)$acf[2:4],rep(0,518)))**

**dmat <- diag(rep(sdres,522))**

**omga <- dmat%\*% cormat %\*% dmat**

**ev <- eigen(omga)**

**# extract components**

**(L <- ev$values)**

**(p <- ev$vectors)**

**d=diag(L)**

**p <- t(p)**

**dnzerofive=solve(diag(sqrt(L)))**

**a=dnzerofive %\*% d %\*% dnzerofive**

**a[1:5,1:5]**

**pstar <- dnzerofive %\*% p**

**a=pstar %\*% omga %\*% t(pstar)**

**modelgls<-lm(pstar %\*% (smbm3-RF) ~ pstar %\*% Mkt.RF , data)**

**coeftest(modelgls)**

**record\_beta=coeftest(modelgls)[2]**

**difference=1**

**coeficientalpha=c()**

**coeficientbeta=c()**

**std=c()**

**stdalpha=c()**

**while (difference>0.01){**

**count=count+1**

**olsres=modelgls$residuals**

**sdres <- sd(olsres)\*sqrt(521/522)**

**cormat<- toeplitz(c(1,acf(olsres,plot=F)$acf[2:4],rep(0,518)))**

**dmat <- diag(rep(sdres,522))**

**omga <- dmat%\*% cormat %\*% dmat**

**ev <- eigen(omga)**

**# extract components**

**L <- ev$values**

**p <- ev$vectors**

**d=diag(L)**

**p <- t(p)**

**dnzerofive=solve(diag(sqrt(L)))**

**a=dnzerofive %\*% d %\*% dnzerofive**

**a[1:5,1:5]**

**pstar <- dnzerofive %\*% p**

**a=pstar %\*% omga %\*% t(pstar)**

**modelgls<-lm(pstar %\*% (smbm3-RF) ~ pstar %\*% Mkt.RF , data)**

**coeftest(modelgls)**

**difference=abs(record\_beta-coeftest(modelgls)[2])**

**std=append(std,coeftest(modelgls)[4])**

**stdalpha=append(stdalpha,coeftest(modelgls)[3])**

**record\_beta=coeftest(modelgls)[2]**

**coeficientbeta=append(coeficientbeta,record\_beta)**

**coeficientalpha=append(coeficientalpha,coeftest(modelgls)[1])**

**}**

**coeficientalpha**

**coeficientbeta**

**std**

**stdalpha**

**par(mfrow=c(2,1),mgp=c(1.5,0.5,0),mar=c(3,3,2,0.5),oma=c(0,0,2,0))**

**#plot(data[,4],abs(olsres),xlab="RmRf",ylab="|OLS Residual|")**

**#title(line=0.2,"OLS residual: absolute value vs VW-Xrm")**

**Acf(abs(olsres),main="")**

**qqnorm(olsres,main="");qqline(olsres)**

**title(line=0.2,"OLS Residuals: Normal Probability plot")**

**title(outer=T,"Figure 2: Heteroskedasticity Diagnostic",line=0)**

**library(fGarch)**

**library(tseries)**

**garchols<-garch(olsres)**

**summary(garchols)**

**plot(garchols)**

**require(fGarch)**

**garch2<-garchFit(formula = ~garch(1,1),data = olsres,include.mean=FALSE)**

**summary(garch2)**

**par(mfrow=c(1,1),mgp=c(1.5,0.5,0),mar=c(3,3,1,0.3),oma=c(0,0,1,0))**

**ts.plot(garchols$fitted.values[,1],ylab="GARCH(1,1) Std. Dev.")**

**mtext(outer=T,"Figure 3: GARCH(1,1) Diagnostics",line=0)**

**omgainv <- diag(1/garch2@h.t)**

**xx <- cbind(rep(1,522),data[,8])**

**xomx <- solve(t(xx)%\*% omgainv %\*% xx)**

**betgarch <- xomx %\*% (t(xx) %\*% omgainv %\*% (data[,4]-data[,11]))**

**betgarch**

**sqrt(diag(xomx))**

**# Iterating**

**xx <- cbind(rep(1,522),data[,8])**

**betiter<-matrix(0,ncol=2,nrow=100)**

**res0 <- olsres**

**betiter[1,]<-model$coef**

**i <- 1**

**zcond <- 100 # Careful if you copy paste several times!**

**betiter<-matrix(0,ncol=2,nrow=100)**

**bstd<-matrix(0,ncol=2,nrow=100)**

**bstd[1,]=coeftest(model)[3:4]**

**while (zcond > 0.000001/100) {**

**i <- i+1**

**omgainv <- diag(1/ garchFit(formula=~garch(1,1), data=res0, include.mean=FALSE,trace=FALSE)@h.t)**

**garch2<-garchFit(formula=~garch(1,1), data=res0, include.mean=FALSE,trace=FALSE)**

**summary(garch2)**

**xomx <- solve(t(xx)%\*% omgainv %\*% xx)**

**betiter[i,] <- xomx %\*% (t(xx) %\*% omgainv %\*% (data[,4]-data[,11]))**

**bstd[i,] <- sqrt(diag(xomx))**

**zcond <- abs((betiter[i,2]-betiter[i-1,2])/betiter[i-1,2])**

**res0 <- (data[,4]-data[,11]) - xx %\*% betiter[i,]**

**}**

**betiter[1:(i+1),]**

**bstd[1:(i+1),]**