hawks

Dingxian Cao

November 3, 2015

library(quantmod)

## Loading required package: xts  
## Loading required package: zoo  
##   
## Attaching package: 'zoo'  
##   
## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric  
##   
## Loading required package: TTR  
## Version 0.4-0 included new data defaults. See ?getSymbols.

getSymbols("^GSPC",from="2013-01-01",to="2015-09-30")

## As of 0.4-0, 'getSymbols' uses env=parent.frame() and  
## auto.assign=TRUE by default.  
##   
## This behavior will be phased out in 0.5-0 when the call will  
## default to use auto.assign=FALSE. getOption("getSymbols.env") and   
## getOptions("getSymbols.auto.assign") are now checked for alternate defaults  
##   
## This message is shown once per session and may be disabled by setting   
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for more details.

## [1] "GSPC"

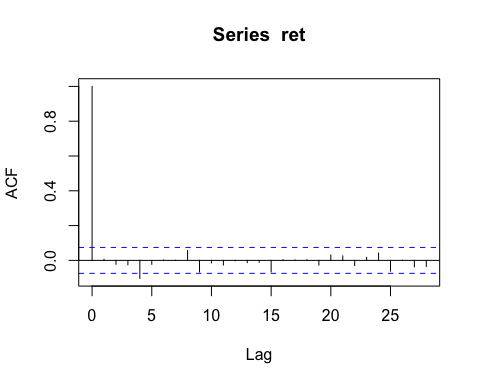
range(index(GSPC))

## [1] "2013-01-02" "2015-09-30"

GSPC<-Ad(GSPC)  
ret<- dailyReturn(GSPC,type = "log")

## a

acf(ret)



ma<-arima(ret,order = c(0,0,4),include.mean = F)  
se<-sqrt(diag(ma$var.coef))  
t<-ma$coef/se  
which( abs(t) <1.645)

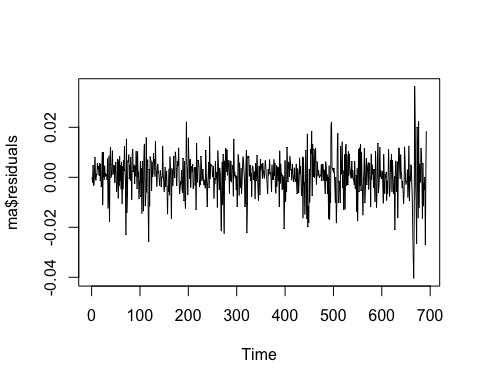
## ma1 ma2 ma3   
## 1 2 3

## b

ma<-arima(ret,order = c(0,0,4),include.mean = F,fixed = c(0,0,0,NA))  
ma

##   
## Call:  
## arima(x = ret, order = c(0, 0, 4), include.mean = F, fixed = c(0, 0, 0, NA))  
##   
## Coefficients:  
## ma1 ma2 ma3 ma4  
## 0 0 0 -0.092558159692089997  
## s.e. 0 0 0 0.036109101305220000  
##   
## sigma^2 estimated as 6.1840160399728e-05: log likelihood = 2371.1500000000001, aic = -4738.3000000000002

plot(ma$residuals)



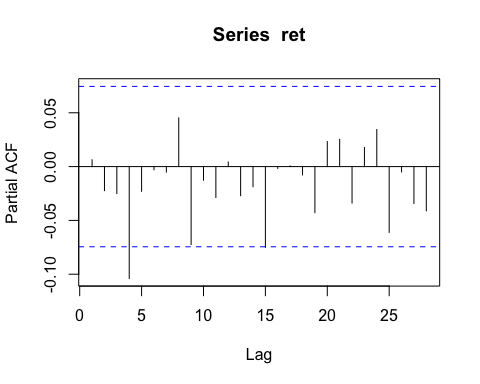
Box.test(ma$residuals,lag = 10,type = "Ljung-Box",fitdf = 4)

##   
## Box-Ljung test  
##   
## data: ma$residuals  
## X-squared = 7.3694760534138402, df = 6, p-value = 0.28802499774276

according to the LB test, the ma(4) model is adequate.

## c

pacf(ret)



AR<-arima(ret,order = c(4,0,0),include.mean = FALSE)  
se<-sqrt(diag(AR$var.coef))  
t<-AR$coef/se  
which( abs(t) <1.645)

## ar1 ar2 ar3   
## 1 2 3

AR<-arima(ret,order = c(4,0,0),include.mean = FALSE,fixed = c(0,0,0,NA))

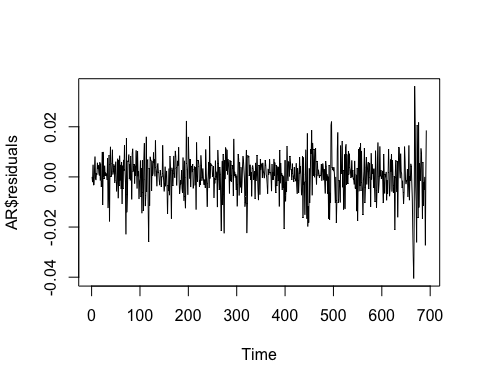
## Warning in arima(ret, order = c(4, 0, 0), include.mean = FALSE, fixed =  
## c(0, : some AR parameters were fixed: setting transform.pars = FALSE

AR

##   
## Call:  
## arima(x = ret, order = c(4, 0, 0), include.mean = FALSE, fixed = c(0, 0, 0,   
## NA))  
##   
## Coefficients:  
## ar1 ar2 ar3 ar4  
## 0 0 0 -0.102722419661389994  
## s.e. 0 0 0 0.038178903047429998  
##   
## sigma^2 estimated as 6.1775379168412e-05: log likelihood = 2371.5100000000002, aic = -4739.0100000000002

## d

plot(AR$residuals)



Box.test(AR$residuals,lag = 10,type = "Ljung-Box",fitdf = 4)

##   
## Box-Ljung test  
##   
## data: AR$residuals  
## X-squared = 6.6975788761690804, df = 6, p-value = 0.3497224002087

according to the LB test, the ar(4) model is adequate.

## e

ma

##   
## Call:  
## arima(x = ret, order = c(0, 0, 4), include.mean = F, fixed = c(0, 0, 0, NA))  
##   
## Coefficients:  
## ma1 ma2 ma3 ma4  
## 0 0 0 -0.092558159692089997  
## s.e. 0 0 0 0.036109101305220000  
##   
## sigma^2 estimated as 6.1840160399728e-05: log likelihood = 2371.1500000000001, aic = -4738.3000000000002

since the log likelihood and aic value are more preferrable for ar model. I am going to take ar(4) model for in-sample fitting.

## f

pre\_ma<-c()  
for(i in((length(ret)-49):length(ret))){  
 ma<-arima(ret[1:i],order = c(0,0,4),include.mean = F,fixed = c(0,0,0,NA))  
 pre<-predict(ma,1)$pred  
 pre<-unclass(pre)[1]  
 pre\_ma<-c(pre\_ma,pre)  
}  
  
  
pre\_ar<-c()  
for(i in((length(ret)-49):length(ret))){  
 AR<-arima(ret[1:i],order = c(4,0,0),include.mean = F,fixed = c(0,0,0,NA))  
 pre<-predict(AR,1)$pred  
 pre<-unclass(pre)[1]  
 pre\_ar<-c(pre\_ar,pre)  
}

(mse\_pre\_ar<-sum((ret[(length(ret)-48):length(ret)]- pre\_ar[-50])^2))

## [1] 0.010032161940163577

(mse\_pre\_ma<-sum((ret[(length(ret)-48):length(ret)]- pre\_ma[-50])^2))

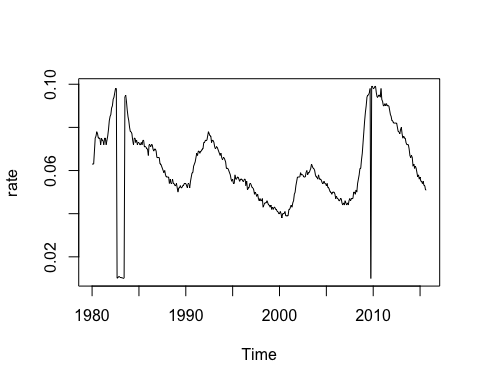
## [1] 0.010052901563711024

since the mse of ar model's out sample prediction is smaller, I would use ar(4) for prediction.

# 4

## a

rate<-read.csv("rate1.csv")#1980-1 2015-9/monthly rate  
rate<-as.matrix(rate)  
rate<-as.vector(rate)  
rate<-rate-10\*(1:length(rate))  
rate<-rate/100  
rate<-ts(rate,deltat = 1/12,start = c(1980,1))#monthly data  
  
plot(rate)



library(tseries)  
adf.test(rate,k = 2)

##   
## Augmented Dickey-Fuller Test  
##   
## data: rate  
## Dickey-Fuller = -3.9768897902559699, Lag order = 2, p-value =  
## 0.010299510487201  
## alternative hypothesis: stationary

According to the result of Augmented Dickey-Fuller Test, since the p value is greater than 0.05, we should accept the null hypothesis which means the rate series does have a unit root.

## b

(out<-which(rate<0.02))#outliers

## [1] 33 34 35 36 37 38 39 40 41 42 358

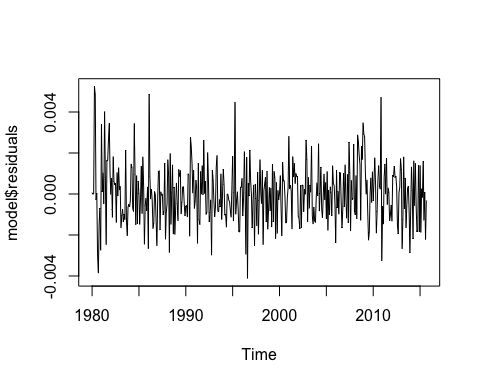
X<-matrix(0,nrow = length(rate),ncol = length(out))  
loc<-cbind(out,1:length(out))  
for(i in seq(length(out))){  
 X[loc[i],i]<-1  
}  
  
  
# rate<-rate[-out]  
# ts.plot(rate)  
  
library(forecast)

## Loading required package: timeDate  
## This is forecast 6.2

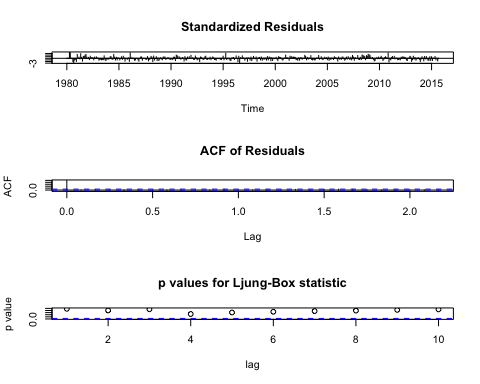
(model<- auto.arima(rate,xreg =X ))

## Series: rate   
## ARIMA(2,1,2)(2,0,2)[12]   
##   
## Coefficients:  
## ar1 ar2 ma1  
## 1.57659924903444004 -0.63699600760332997 -1.58003344528219003  
## s.e. 0.10838929054816999 0.11377280153567000 0.10089766594372999  
## ma2 sar1 sar2  
## 0.74853485520416996 0.80184205581665002 -0.29416372044675998  
## s.e. 0.10041828847419999 0.26388305636843001 0.17976701875439000  
## sma1 sma2 X1  
## -0.99128851609877000 0.17286535277862 -0.0904592810829600025  
## s.e. 0.27005068573761998 0.24628956024224 0.0011930620141300001  
## X2 X3 X4  
## -0.090637315939659999 -0.0918210328994600034 -0.092838573730469998  
## s.e. 0.001447527780860000 0.0015852567988199999 0.001657495708870000  
## X5 X6  
## -0.0932286962333700037 -0.0921538733013199940  
## s.e. 0.0016824609380599999 0.0016363575160100001  
## X7 X8  
## -0.091742999857310006 -0.0904769699845300035  
## s.e. 0.001605666138050000 0.0015393803987900001  
## X9 X10 X11  
## -0.0873888307900600042 -0.0857183618587000068 -0.08944530825923  
## s.e. 0.0014073325689899999 0.0011570263886800001 0.00090370187836  
##   
## sigma^2 estimated as 2.0357103143178e-06: log likelihood=2193.9499999999998  
## AIC=-4347.9099999999999 AICc=-4345.8400000000001 BIC=-4266.7299999999996

plot(model$residuals)



tsdiag(model)



X\_NEW<- matrix(0,nrow = 1,ncol = length(out))  
X\_NEW

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]  
## [1,] 0 0 0 0 0 0 0 0 0 0 0

oct<-predict(model,n.ahead = 1,newxreg = X\_NEW)  
oct$pred

## Oct  
## 2015 0.05090697531909056

According to the box test, the residuals of the model can be seen as white noise which means the seasonal ARIMA(2,1,2)(2,0,2)[12] is adequate for the data.

## c

To calculate the ar part polynomial function,we can get complex roots. So the model shows the business cycle.

p<-c(1,-model$coef[1:2])  
(a1<- polyroot(p))

## [1] 1.2375267899765474+0.19594896262881539i  
## [2] 1.2375267899765474-0.19594896262881539i