Financial data analysis using R

May 18, 2016

Contents

1 "Quantmod" package

Figure 1: http://www.quantmod.com

1.1 How to get financial data using R ("Quantmod" package)

We consider a useful R package for downloading financial data directly from some open sources, including Yahoo Finance, Google Finance, and the Federal Reserve Economic Data (FRED) of Federal Reserve Bank of St. Louis. The package is quantmod which requires three additional packages: TTR, xts and zoo. Once installed, the quantmod package allows users, with internet connection, to use tick symbols to access daily stock data from Yahoo and Google Finance and to use series name to access over 1000 economic and financial time series from FRED. Also, see the web page http://www.quantmod.com. install.packages("quantmod")

```
library("quantmod")
#Extract
getSymbols("^GSPC",src="yahoo",from="1994-1-1",to=Sys.Date())
print(head(GSPC));print(tail(GSPC))
print(class(GSPC))
print(is.OHLC(GSPC))
print(is.OHLCV(GSPC))
print(has.OHLC(GSPC,which=FALSE))
print(has.OHLC(GSPC,which=TURE))
has.Vo(GSPC)
Vo(GSPC)
Ad(GSPC)
```

```
first(GSPC,5)
last(GSPC,5)
Next(GSPC,1)
#Calculate
Delt(Op(GSPC),type=("arithmetic"))
Delt(Op(GSPC),type=("log"))
Delt(Op(GSPC),Cl(GSPC))
#Translate
first(GSPC,10)
head(to.weekly(GSPC))
to.monthly(GSPC)
```

It is possible with one quantmod function to load data from a variety of sources, including...

- Yahoo! Finance (OHLC data)
- Federal Reserve Bank of St. Louis FRED (11,000 economic series)
- Google Finance (OHLC data)
- Oanda, The Currency Site (FX and Metals)[1]

```
getSymbols("YAHOO",src="google") # from google finance
getSymbols("DEXJPUS",src="FRED") # FX rates from FRED
```

Use a function to store the stock code:

```
setSymbolLookup(CJSY=list(name="0001.HK",src="yahoo"))
getSymbols("CJSY",from="1900-1-1",to=Sys.Date())
print(head(CJSY));print(tail(CJSY))
```

Download several different stock prices:

```
szSymbols <- c("MSFT","ORCL","GOOG","INTL","AAPL","CSCO","SYMC","TSLA")
getSymbols(szSymbols,src="yahoo",from="2008-1-1",to=Sys.Date())
```

Bond:

```
getSymbols("^TNX",src="yahoo",from="1900-1-1",to=Sys.Date())
print(head(TNX));print(tail(TNX))
```

Fund:

```
getSymbols("ACWI",src="yahoo",from="1900-1-1",to=Sys.Date())
print(head(ACWI));print(tail(ACWI))
```

Exchange rate:

```
getFX("USD/JPY")
print(head(USDJPY));print(tail(USDJPY))
getSymbols("EUR/USD",src="oanda")
print(head(EURUSD));print(tail(EURUSD))
```

[1] www.oanda.com

Get the stock price right now:

```
tmp <- getQuote("AAPL");print(tmp);print(class(tmp))
```

Get the earnings information of a company:

```
getFinancials("TSLA")
viewFin(TSLA.f)
viewFin(TSLA.f,"CF","A")
```

Get the history of stock dividends:

```
getDividends("AAPL")
```

Stock split information:

```
getSplits("BIDU")
```

Get the financial statement:

```
getFinancials("AAPL")
viewFinancials(AAPL.f)
```

1.2 Charting with quantmod

1.2.1 Plot

```
#suit for one index
getSymbols("^GSPC",src="yahoo",from="2013-1-1",to="2014-1-1")
plot(GSPC)
```

1.2.2 A Series of Chart

```
windows()
#chartSeries
chartSeries(GSPC)
chartSeries(GSPC,name="GSPC BARCHART",subset="2013-10-01::2013-10-23",type="bars")
chartSeries(GSPC,name="GSPC LINECHART",subset="2013-10-01::2013-10-23",type="line")
chartSeries(GSPC,name="GSPC LINECHART",subset="2013-10-01::2013-10-23",type="candlesticks")
#rechart
reChart(type="bars",subset="2013-10-05::2013-10-29", show.grid=TRUE)
#barChart
barChart(GSPC,theme="black",subset="first 10 weeks",bar.type="ohlc")
barChart(GSPC,theme="black",subset="first 10 weeks",bar.type="hlc")
#candleChart
candleChart(GSPC,theme="white",subset="2013-10-05::2013-10-30")
candleChart(GSPC,theme="white",subset="2013-10-05::2013-10-30",multi.col=T)
#lineChart
lineChart(GSPC,theme="white",subset="2013-10-05::2013-10-30")
```

```
lineChart(GSPC,theme="white",subset="2013-10-05::2013-10-30",line.type="l") lineChart(GSPC,theme="whi
#Technical index
#ADX: Average Directional Index
chartSeries(GSPC,name="GSPC CANDLECHART",subset="2013-06::2013-10-23",type="candlesticks")
addADX()
```

1.2.3 Details of chartSeries

```
windows()
chartSeries(GSPC, name="GSPC", type="candlesticks",
            subset="2012-6/2013-6", TA=NULL, theme=chartTheme("white"))
```

1.2.4 Change theme parameter

```
theme.white <- chartTheme("white")
names(theme.white)
theme.white$up.col <- "red"
theme.white$dn.col <- "white"
theme.white$border <- "lightgray"
windows()
chartSeries(GSPC, name="GSPC", type="candlesticks",
            subset="2013-6/", TA=NULL, theme=theme.white)
```

1.2.5 Choose subset using words

```
windows()
chartSeries(GSPC, name="GSPC", show.grid = T, type="candlesticks",
            subset="last 3 months", TA="addVo()", theme=theme.white)
```

1.2.6 Rechart

Rechart is using for fix the plot already made

```
reChart(theme=chartTheme("black"), subset="last 6 months")
```

1.2.7 Add several technical indexes

```
windows()
chartSeries(GSPC, name="GSPC", show.grid = T, type="candlesticks",
            subset="last 2 quarters", TA="addVo();addSMA(20);
            addBBands(20,3)", theme=theme.white)
addMACD()
zoomChart("2013-9")
addCCI(20)
```

1.2.8 Saving

```
jpeg("GSPC.jpeg")
chartSeries(GSPC, name="GSPC", show.grid = T, type="candlesticks",
            subset="last 2 quarters", TA="addVo();addSMA(20);
            addBBands(20,3)", theme=theme.white)
dev.off()
```

1.3 Example

```
getSymbols("AAPL",src="yahoo",from="1994-1-1",to=Sys.Date())
AAPL.Cl<-Delt(Cl(get("AAPL")),type=("arithmetic"))
AAPL.Cl[which(abs(AAPL.Cl)>0.02),]
plot(AAPL.Cl)
periodReturn(AAPL,period="daily")
dailyReturn(AAPL)
periodReturn(AAPL,period="weekly")
weeklyReturn(AAPL)
first(allReturns(AAPL),15)
```

2 Finance analysis

ASSET RETURNS

Most financial studies involve returns, instead of prices, of assets. First, for average investors, return of an asset is a complete and scale-free summary of the investment opportunity. Second, return series are easier to handle than price series because the former have more attractive statistical properties.

Daily log returns are simply the change series of log prices. In R, a change series can easily be obtained by taking the difference of the log prices. Specifically,

$$r_t = \ln(P_1) - \ln(P_{t-1})$$

where P_t is the stock price at time t. Adjusted daily price was used to compute log returns because adjusted price takes into consideration the stock splits.

```
library(quantmod)
getSymbols("DEXUSEU",src="FRED") #Obtain exchange rates from FRED
head(DEXUSEU)
tail(DEXUSEU)
USEU.rtn=diff(log(DEXUSEU$DEXUSEU)) # Compute changes
chartSeries(DEXUSEU,theme="white")
chartSeries(USEU.rtn,theme="white")
```

VISUALIZATION OF FINANCIAL DATA

Graphs are useful tools in analyzing financial data. Besides the time series plot shown before, we discuss some additional plots to display financial data. For instance, the daily simple returns of 3M stock from January 2, 2001 to September 30, 2011 for 2704 observations.

```
library(fBasics)
da=read.table("d-mmm-0111.txt",header=T) # Load data
mmm=da[,2] # Locate 3M simple returns
#1
hist(mmm,nclass=30) # Histogram bin=30
#2
d1=density(mmm)  # Obtain density estimate
range(mmm)  # Range of 3M returns
x=seq(-.1,.1,.001) # Create a sequence of x with increment 0.001.
y1=dnorm(x,mean(mmm),stdev(mmm))
plot(d1$x,d1$y,xlab='rtn',ylab='density',type='l')
lines(x,y1,lty=2)
#3
library(quantmod)
getSymbols("AAPL",from="2011-01-03",to="2011-06-30")
X=AAPL[,1:4] # Locate open, high, low, and close prices
xx=cbind(as.numeric(X[,1]),as.numeric(X[,2]),as.numeric(X[,3]),as.numeric(X[,4]))
source("ohlc.R")# Compile the R script
ohlc_plot(xx,xl="days",yl="price",title="Apple Stock")
#4
source("ma.R")  # Compile R script
getSymbols("AAPL",from="2010-01-02",to="2011-12-08")
x1=as.numeric(AAPL$AAPL.Close) # Locate close price ma(x1,21)
#5
da=read.table("m-ibmsp-2611.txt",header=T)
head(da)
ibm=log(da$ibm+1) # Transform to log returns
sp=log(da$sp+1)
tdx=c(1:nrow(da))/12+1926 # Create time index
opar<-par(no.readonly = TRUE)
par(mfcol=c(2,1))
plot(tdx,ibm,xlab='year',ylab='lrtn',type='l')
title(main='(a) IBM returns')
plot(tdx,sp,xlab='year',ylab='lrtn',type='l') # X-axis first.
title(main='(b) SP index')
par(opar)
```

STATIONARITY

The foundation of statistical inference in time series analysis is the concept of weak stationarity. In statistics, this phenomenon suggests that the mean of the returns and the variance of the log returns are constant over time or simply the expected return is time invariant. Formally, a time series x_t is weakly stationary if its first two moments (mean and variance) are time invariant. The weak stationarity is important because they provide the basic framework for prediction.

CORRELATION AND AUTOCORRELATION FUNCTION

In statistics, the correlation coefficient between two random variables X and Y is defined as

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}$$

The correlation can be consistently estimated by its sample counterpart

$$\hat{\rho}_{x,y} = \frac{\sum(x_t - \bar{x})(y_t - \bar{y})}{\sqrt{\sum(x_t - \bar{x})^2(y_t - \bar{y})^2}}$$

In theory, the Pearson correlation coefficient is between -1 and 1. However, for some random variables, the actual range of the coefficient can be shorter.

The correlation coefficient between x_t and x_{t-k} is

$$\rho_k = \frac{\text{cov}(x_t, x_{t-k})}{\sqrt{\text{var}(x_t)\text{var}(x_{t-k})}} = \frac{\text{cov}(x_t, x_{t-k})}{\text{var}(x_t)}$$

where $\text{var}(x_{t-k}) = \text{var}(x_t)$, because x_t is weakly stationary.

```
### correlations
da=read.table("C:\\Users\\XXXHHF\\Documents\\LYX\\9. Financial data analysis using R\\data\\m-ibmsp67
head(da)
```

```
##       date       ibm        sp
## 1 19670131  0.075370  0.078178
## 2 19670228  0.079099  0.001963
## 3 19670331  0.048837  0.039410
## 4 19670428  0.100887  0.042239
## 5 19670531 -0.035234 -0.052441
## 6 19670630  0.067024  0.017512
```

```
ibm=da$ibm
sp5=da$sp
cor(sp5,ibm)
```

```
## [1] 0.5856544
```

```
cor(sp5,ibm,method='spearman')
```

```
## [1] 0.5860817
```

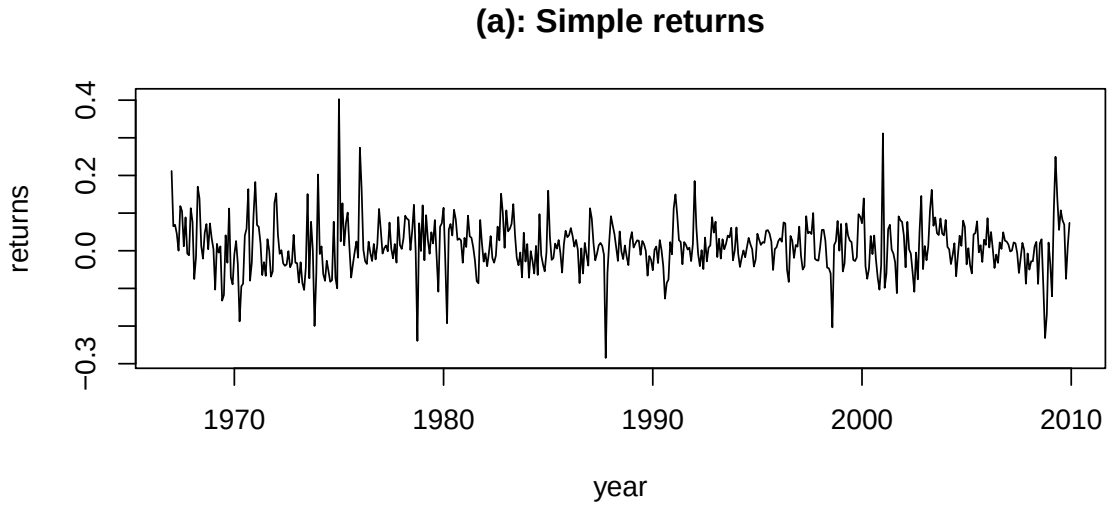
```
cor(sp5,ibm,method='kendall')
```

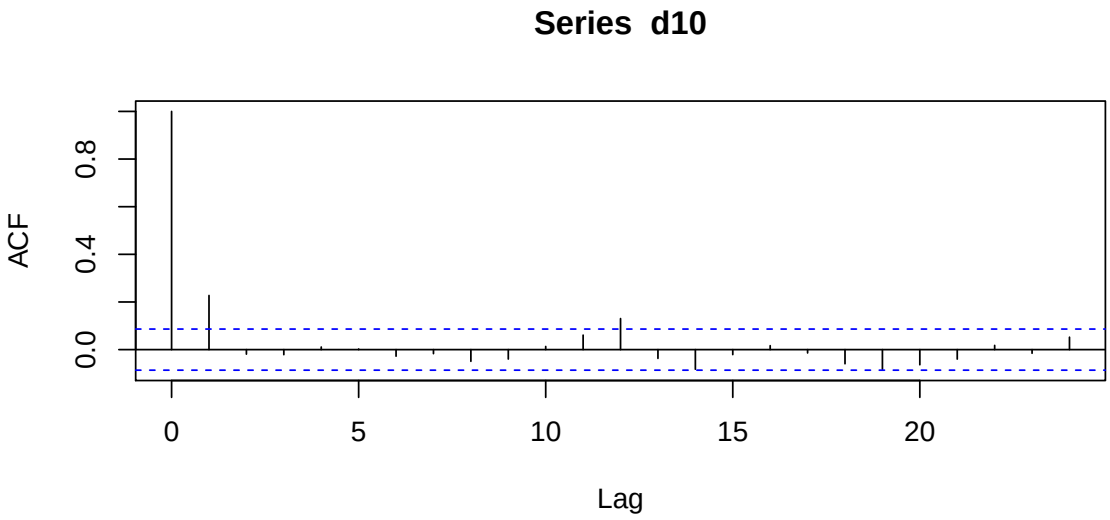
```
## [1] 0.4196587
```

```
### sample ACF
da=read.table("C:\\Users\\XXXHHF\\Documents\\LYX\\9. Financial data analysis using R\\data\\m-dec1291
head(da)
```

```
##       date      dec1      dec2      dec9    dec10
## 1 19670131  0.068568  0.080373  0.180843 0.211806
## 2 19670228  0.008735  0.011044  0.048767 0.064911
## 3 19670331  0.039698  0.035364  0.067494 0.068904
## 4 19670428  0.044030  0.037541  0.040785 0.044602
## 5 19670531 -0.050631 -0.036233 -0.002191 0.000295
## 6 19670630  0.014998  0.018870  0.102075 0.118678
```

```
d10=da$dec10  # select the Decile 10 returns
dec10=ts(d10,frequency=12,start=c(1967,1))
opar<-par(no.readonly = TRUE)
par(mfcol=c(2,1))
plot(dec10,xlab='year',ylab='returns')
title(main='(a): Simple returns')
acf(d10,lag=24) # command to obtain sample ACF of the data
```






```
par(opar)
```

WHITE NOISE

A time series x_t is called a white noise if x_t is a sequence of iid random variables with finite mean and variance. In particular, if x_t is normally distributed with mean 0 and variance σ^2, the series is called a Gaussian white noise. For a white noise series, all the ACFs are 0. In practice, if all sample ACFs are close to 0, then the series is a white noise series.

LINEAR TIME SERIES

$$x_t = \mu + \sum_{i=0}^{\infty} \psi_i a_{t-i} \tag{1}$$

where μ is the mean of x_t ,$\psi_0 = 1$, and a_t is a sequence of iid random variables (i.e. white noise series). a_t denotes the new information at time t of the time series and is often referred to as the innovation or shock at time t .Thus, a time series is linear if it can be written as a linear combination of past innovations. For a linear time series in Equation 1, the dynamic structure of x_t is governed by the coefficients ψ_i , which are called the ψ-weights of x_t in the time series literature. If x_t is weakly stationary, we can obtain its mean and variance easily by using properties of a_t as

$$E(x_t) = \mu, \qquad Var(x_t) = \sigma_a^2 \sum \psi_i^2 < \infty$$

where σ_a^2 a is the variance of a_t . Because Var(x_t) $< \infty$, ψ_i^2 must be a convergent sequence, implying that $\psi_i^2 \to 0$ as $i \to \infty$. Consequently, for a stationary series, impact of the remote shock a_{t-i} on the return x_t vanishes as i increases.

The lag-l autocovariance of x_t is

$$r_l = \text{Cov}(x_t, x_{t-l}) = E[(\sum \psi_i a_{t-i})(\sum \psi_j a_{t-l-j})] = \sigma_a^2 \sum \psi_j \psi_{j+l}$$

Consequently, the ψ-weights are related to the autocorrelations of x_t as follows:

$$\rho_l = \frac{r_l}{r_0} = \frac{\sum \psi_i \psi_{i+l}}{1 + \sum \psi_i^2}$$

where $\psi_0 = 1$. Linear time series models are econometric and statistical models employed to describe the pattern of the ψ-weights of x_t . For a weakly stationary time series, $\psi_i \to 0$ as $i \to \infty$ and, hence, ρ_l converges to 0 as l increases. For asset returns, this means that, as expected, the linear dependence of the current return x_t on the remote past return x_{t-l} diminishes for large l.

2.1 Autoregressive process (AR models)

2.1.1 AR(1) model

When x_t has a statistically significant lag-1 autocorrelation, the lagged value x_{t-1} might be useful in predicting x_t . A simple model that makes use of such predictive power is

$$x_t = \phi_0 + \phi_1 x_{t-1} + a_t \tag{2}$$

where a_t is assumed to be a white noise series with mean 0 and variance σ_a^2. The AR(1) model has several properties similar to those of the simple linear regression model. However, there are some significant differences between the two models, which we discuss later. Here, it suffices to note that an AR(1) model implies that, conditional on the past return x_{t-1},

$$E(x_t|x_{t-1}) = \phi_0 + \phi_1 x_{t-1}, \qquad Var(x_t|x_{t-1}) = \sigma_a^2$$

We begin with the sufficient and necessary condition for weak stationarity of the AR(1) model. Assuming that the series is weakly stationary, we have $E(x_t) = \mu$, $Var(x_t) = \gamma_0$, and $Cov(x_t, x_{t-j}) = \gamma_j$, where $\mu$and γ_0 are constants and γ_j is a function of j, not t. We can easily obtain the mean, variance, and autocorrelations of the series as follows.

$$E(x_t) = \phi_0 + \phi_1 E(x_{t-1})$$

Under the stationarity condition, $E(x_t) = E(x_{t-1}) = \mu$ and hence

$$\mu = \phi_0 + \phi_1 \mu$$

and

$$E(x_t) = \mu = \frac{\phi_0}{1 - \phi_1}$$

Thus, for a stationary AR(1) process, the constant term ϕ_0 is related to the mean of x_t via $\phi_0 = (1-\phi_1)\mu$, and $\phi_0 = 0$ implies that $E(x_t) = 0$. Next, using $\phi_0 = (1-\phi_1)\mu$, the AR(1) model can be rewritten as

$$x_t - \mu = \phi_1(x_{t-1} - \mu) + a_t \tag{3}$$

By repeated substitutions, the prior equation implies that

$$x_t - \mu = a_t + \phi_1 a_{t-1} + \phi_1^2 a_{t-1} + \cdots = \sum_{i=0}^{\infty} \phi_1^i a_{t-i} \tag{4}$$

Thus, $x_t - \mu$ is a linear function of a_{t-i} for $i \geq 0$. Taking the square and the expectation of aboved Equation, we obtain

$$Var(x_t) = \phi_1^2 Var(x_{t-1}) + \sigma_a^2$$

where σ_a^2 is the variance of a_t, and we make use of the fact that the covariance between x_{t-1} and at is 0. Under the stationarity assumption, $Var(x_t) = Var(x_{t-1})$, so that

$$Var(x_t) = \frac{\sigma_a^2}{1-\phi_1^2}$$

provided that $\phi_1^2 < 1$. The requirement of $\phi_1^2 < 1$ results from the fact that the variance of a random variable is nonnegative and x_t is weakly stationary. Consequently, the weak stationarity of an AR(1) model implies that $-1 < \phi_1 < 1$, that is, $|\phi_1| < 1$.

2.1.2 AR(p) model

Obviously, there are situations in which x_{t-1} alone cannot determine the conditional expectation of x_t and a more flexible model must be sought. A straightforward generalization of the AR(1) model is the AR(p) model

$$x_t = \phi_0 + \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + a_t$$

where p is a nonnegative integer. This model says that, given the past data, the first p lagged variables x_{t-i} (i = 1, . . . , p) jointly determine the conditional expectation of x_t. The AR(p) model is in the same form as a multiple linear regression model with lagged values serving as the explanatory variables.

The stationarity condition of an AR(2) time series is that the absolute values of its two characteristic roots are less than 1, that is, its two characteristic roots are less than 1 in modulus. Equivalently, the two solutions of the characteristic equation are greater than 1 in modulus.

The results of AR(1) and AR(2) models can readily be generalized to the general AR(p) model. The mean of a stationary series is

$$E(x_t) = \frac{\phi_0}{1-\phi_1-\cdots-\phi_p}$$

provided that the denominator is not 0. The associated characteristic equation of the model is

$$1 - \phi_1 z - \phi_2 z^2 - \cdots - \phi_p z^p = 0$$

If all the solutions of this equation are greater than 1 in modulus, then the series x_t is stationary. Again, inverses of the solutions are the characteristic roots of the model. Thus, stationarity requires that all characteristic roots are less than 1 in modulus.

Partial Autocorrelation Function (PACF)

The PACF of a stationary time series is a function of its ACF and is a useful tool for determining the order p of an AR model. A simple, yet effective way to introduce PACF is to consider the following AR models in consecutive orders:

$$\begin{aligned} x_t &= \phi_{0,1} + \phi_{1,1}x_{t-1} + e_{1t}, \\ x_t &= \phi_{0,1} + \phi_{1,1}x_{t-1} + \phi_{2,2}x_{t-2} + e_{1t}, \\ &\vdots \end{aligned}$$

where $\varphi_{0,j}, \varphi_{i,j}$, and e_{jt} are, respectively, the constant term, the coefficient of x_{t-i} , and the error term of an AR(j) model. These models are in the form of a multiple linear regression and can be estimated by the least squares (LS) method. From the definition, the lag-2 PACF $\hat{\varphi}_{2,2}$ shows the added contribution of x_{t-2} to x_t over the AR(1) model $x_t = \varphi_0 + \varphi_1 x_{t-1} + e_{1t}$. The lag-3 PACF shows the added contribution of x_{t-3} to x_t over an AR(2) model, and so on. Therefore, for an AR(p) model, the lag-p sample PACF should not be 0, but $\varphi_{i,j}$ should be close to 0 for all $j > p$. We make use of this property to determine the order p. For a stationary Gaussian AR(p) model, it can be shown that the sample PACF has the following properties:

- $\hat{\varphi}_{p,p}$converges to φ_p as the sample size T goes to infinity.
- $\hat{\varphi}_{l,l}$ converges to 0 for all $l > p$.
- The asymptotic variance of $\hat{\varphi}_{ll}$, is $1/T$ for $l > p$.

These results say that, for an AR(p) series, the sample PACF cuts off at lag p.

Information Criteria

There are several information criteria available to determine the order p of an AR process. All of them are likelihood based. For example, the well-known Akaike Information Criterion (AIC) (Akaike, 1973) is defined as

$$AIC = \frac{-2}{T}\ln(\text{likelihood}) + \frac{2}{T}(\text{number of parameters}),$$

where the likelihood function is evaluated at the maximum likelihood estimates and T is the sample size. For a Gaussian AR() model, AIC reduces to

$$\ln \text{AIC} = (\frac{2l}{T}) + \ln(\tilde{\sigma}_a^2)$$

where where $\tilde{\sigma}_a^2$ is the maximum likelihood estimate of σ_a^2 , which is the variance of a_t,T is the sample size.

Another commonly used criterion function is the Schwarz–Bayesian criterion (BIC, Bayesian information criterion). For a Gaussian AR() model, the criterion is

$$\ln \text{SIC} = \frac{l}{T}\ln T + \ln(\tilde{\sigma}_a^2)$$

The penalty for each parameter used is 2 for AIC and $\ln(T)$ for BIC. Thus, compared with AIC, BIC tends to select a lower AR model when the sample size is moderate or large.

Selection Rule: To use AIC to select an AR model in practice, one computes AIC() for $= 0, ..., P$, where P is a prespecified positive integer and selects the order k that has the minimum AIC value. The same rule applies to BIC.

Parameter Estimation

For a specified AR(p) model, the conditional LS method, which starts with the $(p+1)$th observation, is often used to estimate the parameters. Specifically, conditioning on the first p observations, we have

$$x_t = \phi_0 + \phi_1 x_{t-1} + \ldots + \phi_p x_{t-p} + a_t$$

which is in the form of a multiple linear regression and can be estimated by the LS method. Denote the estimate of ϕ_i by $\hat{\phi}_i$. The fitted model is

$$\hat{x}_t = \hat{\phi}_0 + \hat{\phi}_1 x_{t-1} + \ldots + \hat{\phi}_p x_{t-p}$$

and the associated residual is

$$\hat{a}_t = x_t - \hat{x}_t$$

The series $\hat{a}_t$ is called the residual series, from which we obtain

$$\hat{\sigma}_a^2 = \frac{\sum_{t=p+1}^{T} \hat{a}_t^2}{T - 2p - 1}$$

If the conditional Gaussian likelihood method is used, the estimates of ϕ_i remain unchanged, but the estimate of σ_a^2 a becomes $\tilde{\sigma}_a^2 = \hat{\sigma}_a^2 \times (T - 2p - 1)/(T - p)$.

2.2 Statistical Significance of Autocorrelation Coefficients

One simple test of stationarity is based on the so-called autocorrelation function (ACF). The ACF at lag k, denoted by ρ_k, is defined as

$$\rho_k = \frac{\gamma_k}{\gamma_0} = \frac{\text{covariance at lag } k}{\text{variance}}$$

Therefore, the sample autocorrelation function at lag k is

$$\hat{\rho}_k = \frac{\hat{\gamma}_k}{\hat{\gamma}_0}$$

where

$$\hat{\gamma}_k = \frac{\sum (x_t - \bar{x})(x_{t+k} - \bar{x})}{n}$$

and

$$\hat{\gamma}_0 = \frac{\sum (x_t - \bar{x})^2}{n}$$

where n is the sample size and $\bar{x}$ is the sample mean.

Bartlett has shown that if a time series is purely random, that is, it exhibits white noise, the sample autocorrelation coefficients $\hat{\rho}_k$ are approximately[2].

$$\hat{\rho}_k \sim N(0, 1/n)$$

that is, in large samples the sample autocorrelation coefficients are normally distributed with zero mean and variance equal to one over the sample size.

[2]M. S. Bartlett, "On the Theoretical Specification of Sampling Properties of Autocorrelated Time Series," Journal of the Royal Statistical Society, Series B, vol. 27, 1946, pp. 27–41.

$$H_0 \quad : \quad \rho(1) = \rho(2) = \ldots = \rho(m)$$
$$H_A \quad : \quad \rho(1) \neq \rho(2) \neq \ldots \neq \rho(m)$$

Instead of testing the statistical significance of any individual autocorrelation coefficient, we can test the joint hypothesis that all the ρ_k up to certain lags are simultaneously equal to zero. This can be done by using the Q statistic developed by Box and Pierce, which is defined as

$$Q = n \sum_{k=1}^{m} \hat{\rho}_k^2$$

where T = sample size and m = lag length. The Q statistic is often used as a test of whether a time series is white noise. In large samples, it is approximately distributed as the chi-square distribution with m df.

A variant of the Box–Pierce Q statistic is the Ljung–Box (LB) statistic, which is defined as

$$\text{LB} = n(n+2) \sum_{k=1}^{m} (\frac{\hat{\rho}_k^2}{n-k}) \sim \chi^2(m)$$

Although in large samples both Q and LB statistics follow the chi-square distribution with m df, the LB statistic has been found to have better (more powerful, in the statistical sense) small-sample properties than the Q statistic.

Model Checking

A fitted model must be examined carefully to check for possible model inadequacy. If the model is adequate, then the residual series should behave as a white noise. The ACF and the Ljung–Box statisticsof the residuals can be used to check the closeness of $\hat{a}_t$ to a white noise. For an AR(p) model, the Ljung–Box statistic $Q(m)$ follows asymptotically a chi-squared distribution with $m - g$ degrees of freedom, where g denotes the number of AR coefficients used in the model.

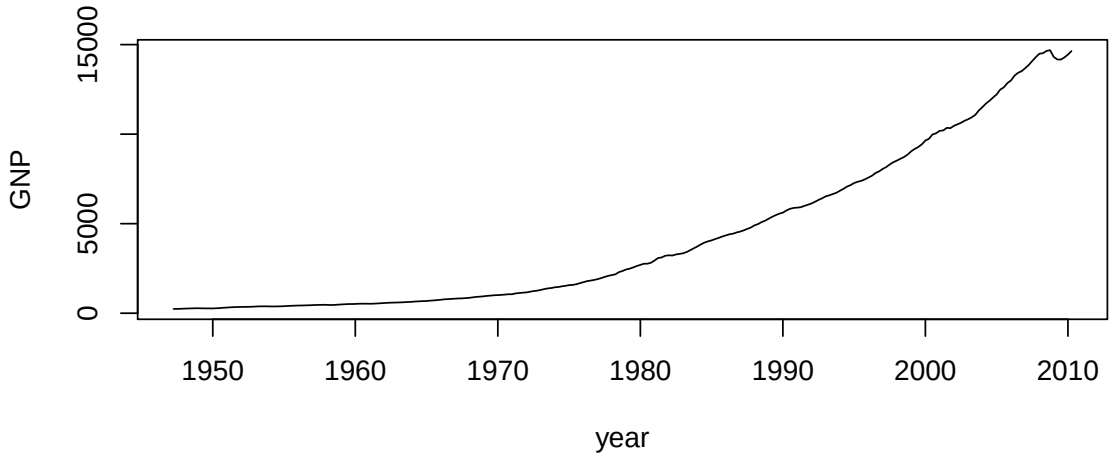
```
da=read.table("q-gnp4710.txt",header=T)
head(da)
```

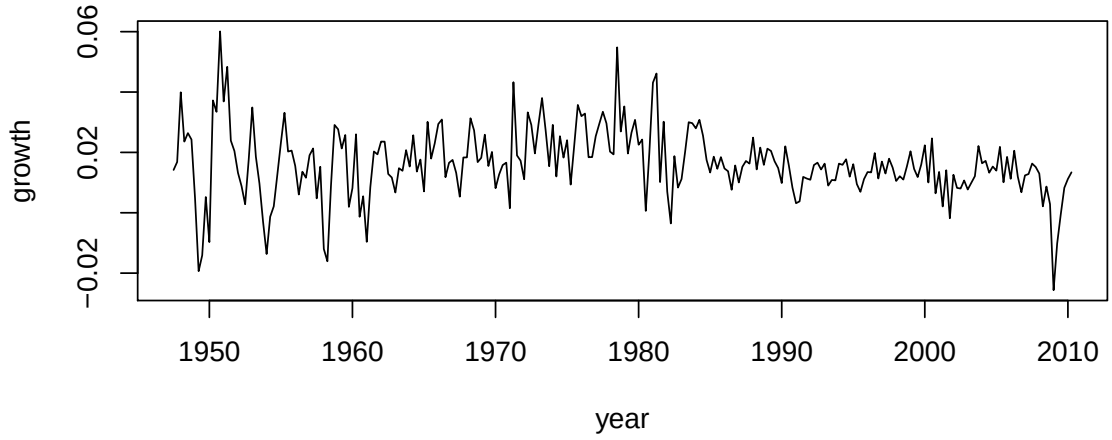
```
##   Year Mon Dat VALUE
## 1 1947   1   1 238.1
## 2 1947   4   1 241.5
## 3 1947   7   1 245.6
## 4 1947  10   1 255.6
## 5 1948   1   1 261.7
## 6 1948   4   1 268.7
```

```
G=da$VALUE
LG=log(G)
gnp=diff(LG)
dim(da)
```

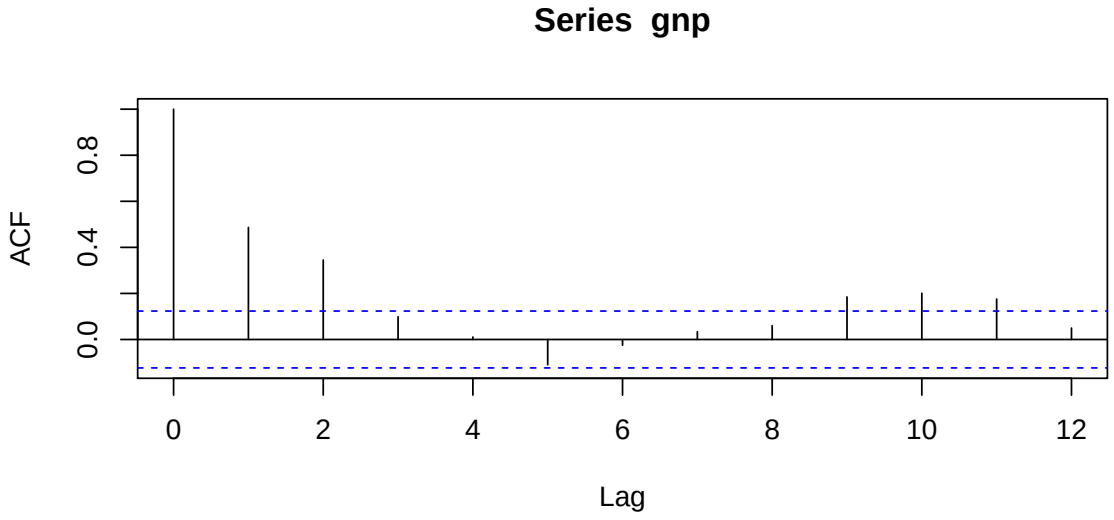
```
## [1] 253   4
```

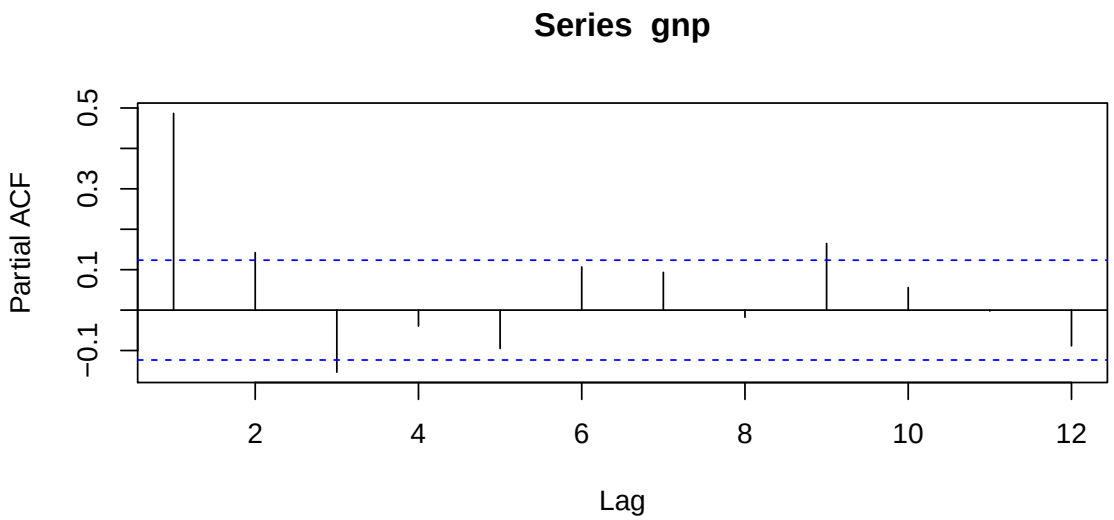
```
tdx=c(1:253)/4+1947 # create the time index
opar<-par(no.readonly = TRUE)
par(mfcol=c(2,1))
plot(tdx,G,xlab='year',ylab='GNP',type='l')
plot(tdx[2:253],gnp,type='l',xlab='year',ylab='growth')
```






```
#ACF and PACF
acf(gnp,lag=12)
pacf(gnp,lag=12) # compute PACF
```






```
par(opar)
#AIC
mm1=ar(gnp,method='mle')
mm1$order # Find the identified order
```

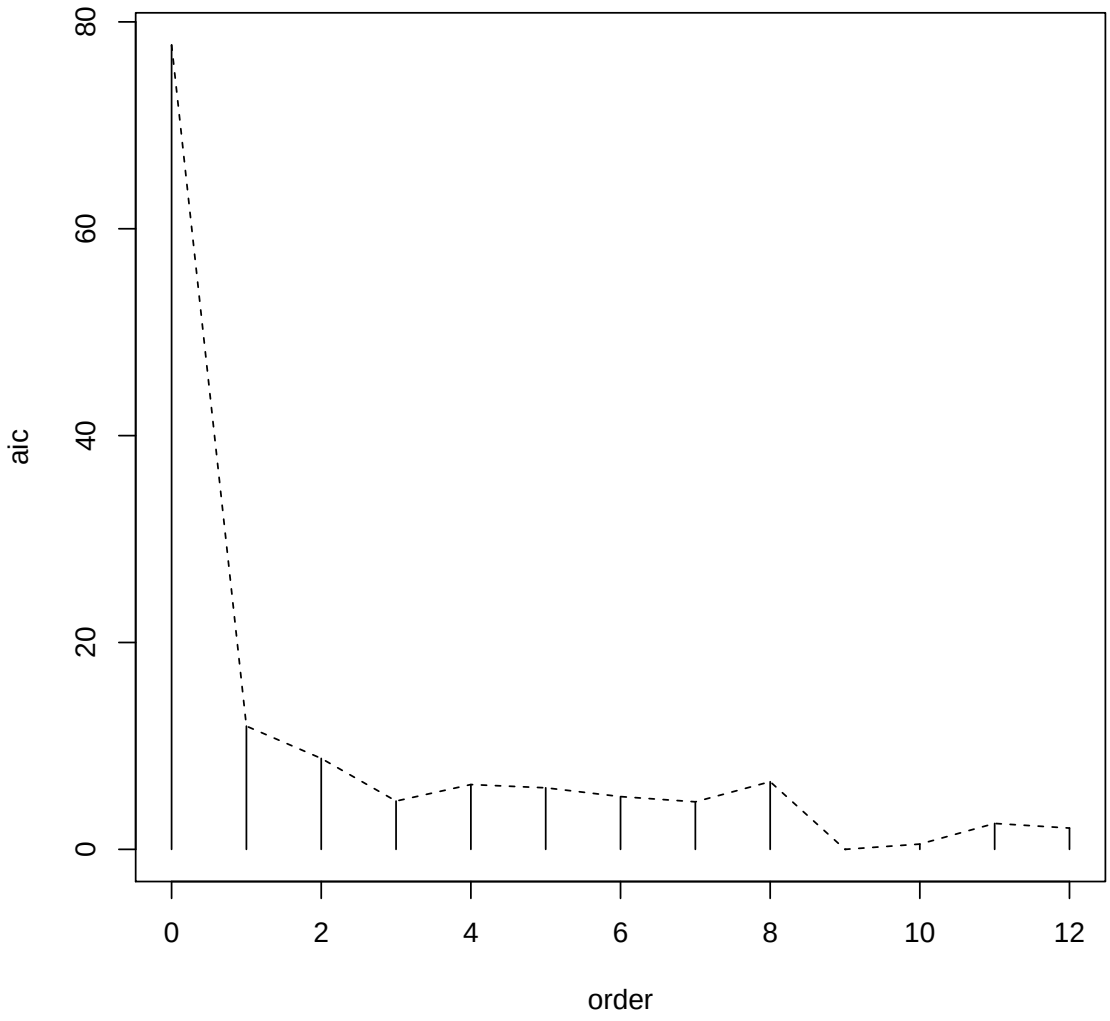
```
## [1] 9
```

```
names(mm1)
```

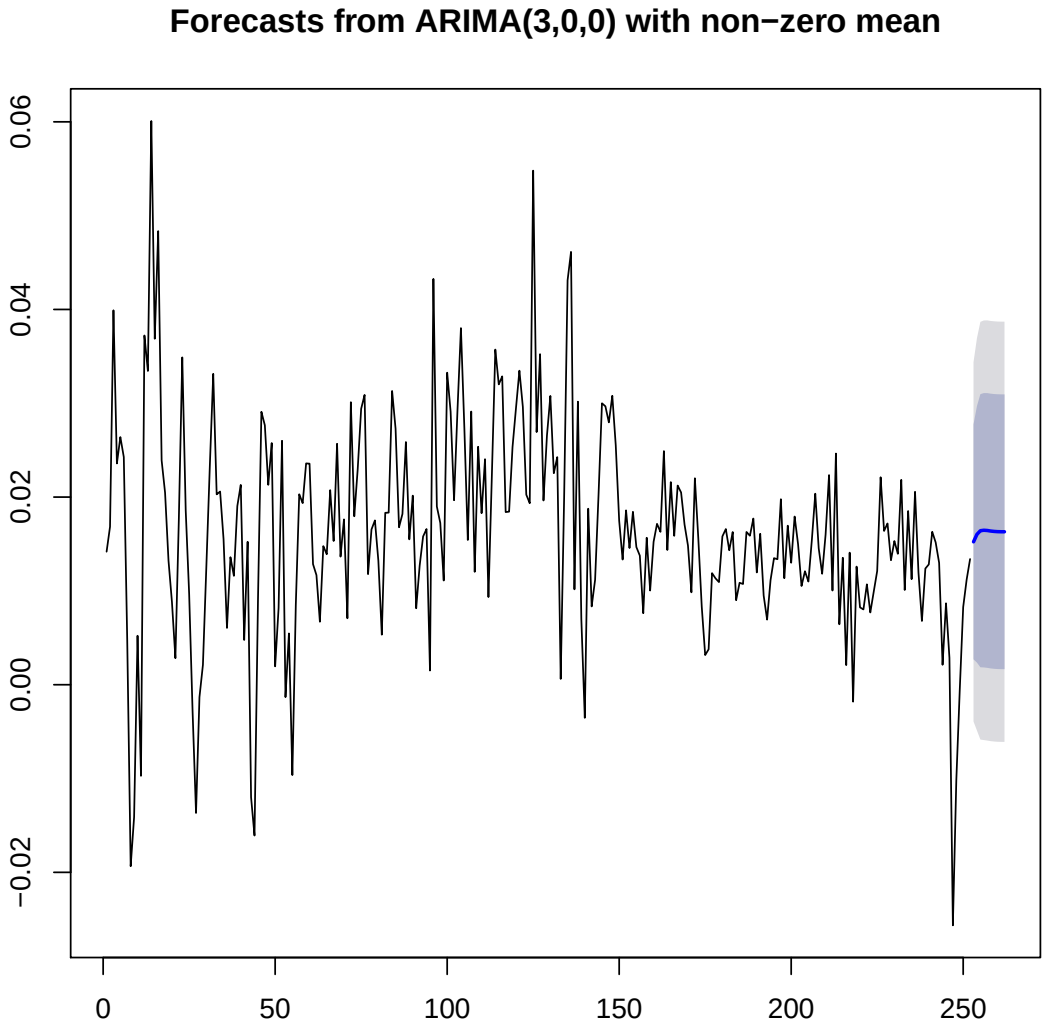
```
##  [1] "order"        "ar"           "var.pred"     "x.mean"
##  [5] "aic"          "n.used"       "order.max"    "partialacf"
##  [9] "resid"        "method"       "series"       "frequency"
## [13] "call"         "asy.var.coef"
```

```
print(mm1$aic,digits=3)
```

```
##      0      1      2      3      4      5      6      7      8      9
## 77.767 11.915  8.792  4.669  6.265  5.950  5.101  4.596  6.541  0.000
##     10     11     12
##  0.509  2.504  2.057

aic=mm1$aic  # For plotting below.
length(aic)

## [1] 13

plot(c(0:12),aic,type='h',xlab='order',ylab='aic')
lines(0:12,aic,lty=2)
#Evaluation
m1=arima(gnp,order=c(3,0,0)); m1

##
## Call:
## arima(x = gnp, order = c(3, 0, 0))
##
## Coefficients:
##          ar1     ar2      ar3  intercept
##       0.4386  0.2063  -0.1559     0.0163
## s.e.  0.0620  0.0666   0.0626     0.0012
##
## sigma^2 estimated as 9.549e-05:  log likelihood = 808.56,  aic = -1607.12

#prediction
#1
kt1=predict(m1,10)
kt1_pred=kt1$pred
#2
#install.packages("timeDate")
#install.packages("fracdiff")
library(forecast)

## Loading required package:  zoo
##
## Attaching package:  'zoo'
## The following objects are masked from 'package:base':
##
##    as.Date, as.Date.numeric
## Loading required package:  timeDate
## Loading required package:  methods
## This is forecast 6.2
```




```
kt2=forecast(m1,263,level=c(80,95))
plot(forecast(m1))
```




```
kt2_pred=kt2$mean
kt1_pred-kt2_pred

## Time Series:
## Start = 253
## End = 262
## Frequency = 1
##  [1] 0 0 0 0 0 0 0 0 0 0
```

2.3 Moving average models (MA Models)

We now turn to the class of MA models. These models are useful in modeling asset returns in finance. There are several ways to introduce MA models. One approach is to treat the model as a simple

extension of white noise series. Another approach is to treat the model as an infinite-order AR model, with some parameter constraints.

There are several ways to introduce MA models. One approach is to treat the model as a simple extension of white noise series. Another approach is to treat the model as an infinite-order AR model, with some parameter constraints. The general form of an MA(1) model is

$$x_t = c_0 + a_t - \theta_1 a_{t-1}$$

where c_0 is a constant and a_t is a white noise series. Similarly, an MA(2) model is in the form

$$x_t = c_0 + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2}$$

and an MA(q) model is

$$x_t = c_0 + a_t - \theta_1 a_{t-1} - \ldots - \theta_q a_{t-q}$$

for MA models, ACF is useful in specifying the order because ACF cuts off at lag q for an MA(q) series.

Stationarity. MA models are always weakly stationary because they are finite linear combinations of a white noise sequence for which the first two moments are time invariant.

The prior discussion applies to general MA(q) models, and we obtain two general properties. First, the constant term of an MA model is the mean of the series (i.e., $E(x_t) = c_0$). Second, the variance of an MA(q) model is

$$\text{Var}(x_t) = (1 + \theta_1^2 + \theta_2^2 + \ldots + \theta_q^2)\sigma_a^2$$

Invertibility. AR(1) model:

$$x_t = \phi x_{t-1} + a_t = -\phi L x_t + a_t$$

Then, we have

$$x_t = \frac{a_t}{1 - \phi L} = \sum_{i=0}^{\infty} (\phi L)^i a_t = \sum_{i=0}^{\infty} \phi^i a_{t-i}$$

Rewriting a zero-mean MA(1) model as $a_t = x_t + \theta_1 a_{t-1}$, we have

$$a_t = \frac{1}{1 - \theta L} x_t = \sum_{i=0}^{\infty} (\theta L)^i x_t = \sum_{i=0}^{\infty} \theta^i x_{t-i}$$

2.4 General ARMA Models

In some applications, the AR or MA models discussed in the previous sections become cumbersome because one may need a high order model with many parameters to adequately describe the dynamic structure of the data. To overcome this difficulty, the ARMA models are introduced (Box et al.,1994). Basically, an ARMA model combines the ideas of AR and MA models into a compact form so that the number of parameters used is kept small, achieving parsimony in parameterization. The model is useful in modeling business, economic, and engineering time series. A general ARMA(p, q) model is in the form

$$x_t = \phi_0 + \sum_{i=1}^{p} \phi_i x_{t-i} + a_t - \sum_{i=1}^{q} \theta_i a_{t-i}$$

where a_t is a white noise series and p and q are nonnegative integers. The AR and MA models are special cases of the ARMA(p, q) model.

TABLE 2.4. Theoretical EACF Table For an ARMA(1,1) Model, Where "X" Denotes Nonzero, "O" Denotes Zero, and "*" Denotes Either Zero or Nonzero[a]

	MA							
AR	0	1	2	3	4	5	6	7
0	X	X	X	X	X	X	X	X
1	X	O	O	O	O	O	O	O
2	*	X	O	O	O	O	O	O
3	*	*	X	O	O	O	O	O
4	*	*	*	X	O	O	O	O
5	*	*	*	*	X	O	O	O

Identifying ARMA Models

The ACF and PACF are not informative in determining the order of an ARMA model. Tsay and Tiao (1984) propose a new approach that uses the extended autocorrelation function (EACF) to specify the order of an ARMA process[3].

The output of EACF is a two-way table, where the rows correspond to AR order p and the columns to MA order q. The theoretical version of EACF for an ARMA(1,1) model is given in Table 2.4. The key feature of the table is that it contains a triangle of "O" with the upper left vertex located at the order (1,1). This is the characteristic we use to identify the order of an ARMA process. In general, for an ARMA(p, q) model, the triangle of "O" will have its upper left vertex at the (p, q) position.

The simplified table is constructed using the following notation:

1. "X" denotes that the absolute value of the corresponding EACF is greater than or equal to twice of its asymptotic standard error;

2. "O" denotes that the corresponding EACF is less than twice of its standard error in modulus.

```
da=read.table("m-3m4608.txt",header=T)
head(da)

##       date       rtn
## 1 19460228 -0.077922
## 2 19460330  0.018592
## 3 19460430 -0.100000
## 4 19460531  0.209877
## 5 19460628  0.005128
## 6 19460731  0.076531

mmm=log(da$rtn+1)
library(TSA)     # Load the package

## Loading required package:  leaps
## Loading required package:  locfit
## locfit 1.5-9.1 2013-03-22
```

[3]Tsay, R. and Tiao, G. (1984). "Consistent Estimates of Autoregressive Parameters and Extended Sample Autocorrelation Function for Stationary and Nonstationary ARMA Models." Journal of the American Statistical Association, 79 (385), pp. 84-96

```
## Loading required package:  mgcv
## Loading required package:  nlme
##
## Attaching package:  'nlme'
## The following object is masked from 'package:forecast':
##
##    getResponse
## This is mgcv 1.8-9.  For overview type 'help("mgcv-package")'.
## Loading required package:  tseries
##
## Attaching package:  'TSA'
## The following objects are masked from 'package:forecast':
##
##    fitted.Arima, plot.Arima
## The following objects are masked from 'package:timeDate':
##
##    kurtosis, skewness
## The following objects are masked from 'package:stats':
##
##    acf, arima
## The following object is masked from 'package:utils':
##
##    tar
```

```
m1=eacf(mmm,6,12)      # Simplified table
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12
## 0 o o x o o x o o o x o  x  o
## 1 x o x o o x o o o o o  x  o
## 2 x x x o o x o o o o o  o  o
## 3 x x x o o o o o o o o  o  o
## 4 x o x o o o o o o o o  o  o
## 5 x x x o x o o o o o o  o  o
## 6 x x x x x o o o o o o  o  o
```

```
print(m1$eacf,digits=2)
```

```
##        [,1]    [,2]   [,3]    [,4]     [,5]    [,6]    [,7]   [,8]
## [1,] -0.056 -0.0380 -0.082 -0.0046  0.01774  0.0821  0.0080 0.0127
## [2,] -0.474  0.0096 -0.074 -0.0209  0.00196  0.0772 -0.0288 0.0026
## [3,] -0.383 -0.3476 -0.074  0.0160 -0.00553  0.0772  0.0269 0.0120
## [4,] -0.177  0.1381  0.384 -0.0224  0.00232  0.0419 -0.0232 0.0154
## [5,]  0.421  0.0287  0.454 -0.0079  0.00071  0.0025 -0.0140 0.0305
## [6,] -0.114  0.2135  0.449  0.0096  0.20242 -0.0063 -0.0038 0.0403
## [7,] -0.208 -0.2504  0.243  0.3111  0.16745 -0.0388 -0.0034 0.0429
##          [,9]   [,10]  [,11]   [,12]  [,13]
## [1,] -0.03014 -0.0778 0.0488  0.0909 -0.011
## [2,] -0.00683 -0.0694 0.0372  0.0938 -0.024
## [3,]  0.00045 -0.0268 0.0221  0.0428  0.042
## [4,] -0.00440 -0.0254 0.0185  0.0100  0.043
## [5,]  0.01159  0.0042 0.0191 -0.0043  0.013
```

```
## [6,] -0.01294 -0.0123 0.0315  0.0117  0.028
## [7,] -0.01009 -0.0260 0.0078  0.0106  0.037
```

Consequently, the EACF suggests that the monthly log returns of 3M stock follow an ARMA(0,0) model (i.e., a white noise series).

3 Unit-root nonstationarity

So far, we have focused on the return series that are stationary. In some studies, interest rates, foreign exchange rates, or the price series of an asset are of interest. These series tend to be nonstationary. For a price series, the nonstationarity is mainly due to the fact that there is no fixed level for the price. In the time series literature, such a nonstationary series is called unit-root nonstationary time series. The best-known example of unit-root nonstationary time series is the random walk model.

3.1 Random Walk

A time series p_t is a random walk if it satisfies

$$p_t = p_{t-1} + a_t$$

where p_0 is a real number denoting the starting value of the process and a_t is a white noise series. If p_t is the log price of a particular stock at date t , then p_0 could be the log price of the stock at its initial public offering (i.e., the logged IPO price). If at has a symmetric distribution around 0, then conditional on p_{t-1}, pt has a 50–50 chance to go up or down, implying that pt would go up or down at random. If we treat the random walk model as a special AR(1) model, then the coefficient of p_{t-1} is unity, which does not satisfy the weak stationarity condition of an AR(1) model. A random walk series is, therefore, not weakly stationary, and we call it a unit-root nonstationary time series. The random walk model has been widely considered as a statistical model for the movement of logged stock prices.

The log return series of a market index tends to have a small and positive mean. This implies that the model for the log price is

$$p_t = \mu + p_{t-1} + a_t$$

where $\mu = E(p_t - p_{t-1})$ and a_t is a zero-mean white noise series. The constant term μ is very important in financial study. It represents the time trend of the log price p_t and is often referred to as the drift of the model. To see this, assume that the initial log price is p_0.

$$\begin{aligned} p_1 &= \mu + p_0 + a_t \\ p_2 &= \mu + p_1 + a_t = 2\mu + p_0 + a_2 + a_1 \\ &\vdots \\ p_t &= t\mu + p_0 + a_t + a_{t-1} + \cdots + a_1 \end{aligned}$$

From the representation, $\phi = 1$, for all i . Thus, the impact of any past shock a_{t-i} on p_t does not decay over time. Consequently, the series has a strong memory as it remembers all of the past shocks. In economics, the shocks are said to have a permanent effect on the series.

Interpretation of the Constant Term

It is important to understand the meaning of a constant term in a time series model. First, for an MA(q) model, the constant term is simply the mean of the series. Second, for a stationary AR(p)

model or ARMA(p, q) model, the constant term is related to the mean via $\mu = \phi_0/(1 - \phi_1 - \cdots - \phi_p)$. Third, for a random walk with drift, the constant term becomes the time slope of the series.

Consider an ARMA model. If one extends the model by allowing the AR polynomial to have 1 as a characteristic root, then the model becomes the well-known autoregressive integrated moving average (ARIMA) model. An ARIMA model is said to be unit-root nonstationary because its AR polynomial has a unit root. Similar to a random walk model, an ARIMA model has strong memory because the θ_i coefficients in its MA representation do not decay over time to 0, implying that the past shock a_{t-i} of the model has a permanent effect on the series. A conventional approach for handling unit-root nonstationarity is to use differencing.

3.2 The phenomenon of spurious regression

To see why stationary time series are so important, consider the following two random walk models:

$$\begin{aligned} y_t &= y_{t-1} + u_t \\ x_t &= x_{t-1} + v_t \end{aligned}$$

where we generated 500 observations of u_t from $u_t \sim N(0, 1)$ and 500 observations of v_t from $v_t \sim N(0, 1)$ and assumed that the initial values of both Y and X were zero. We also assumed that u_t and v_t are serially uncorrelated as well as mutually uncorrelated. As you know by now, both these time series are nonstationary; that is, they are $I(1)$ or exhibit stochastic trends.

```
#generate y
y<-numeric(501)
for (i in 2:501) {
        y[1]<-0
        y[i]<-y[i-1]+rnorm(n=1)
}
y<-y[-1]
#generate x
x<-numeric(501)
for (i in 2:501) {
        x[1]<-0
        x[i]<-x[i-1]+rnorm(n=1)
}
x<-x[-1]
lm_xy<-lm(y~x)
summary(lm_xy)

##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.949  -8.117   0.023  11.603  23.450
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.94394    1.41579   2.079   0.0381 *
## x           -1.68335    0.07641 -22.031   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.07 on 498 degrees of freedom
## Multiple R-squared:  0.4936,Adjusted R-squared:  0.4925
## F-statistic: 485.3 on 1 and 498 DF,  p-value: < 2.2e-16
```

As you can see, the coefficient of X is highly statistically significant, and, although the R2 value is low, it is statistically significantly different from zero. From these results, you may be tempted to conclude that there is a significant statistical relationship between Y and X, whereas a priori there should be none. This is in a nutshell the phenomenon of spurious or nonsense regression, first discovered by Yule. Yule showed that (spurious) correlation could persist in nonstationary time series even if the sample is very large.

That the regression results presented above are meaningless can be easily seen from regressing the first differences of $y_t (= \Delta y_t)$ on the first differences of $x_t (= \Delta x_t)$; remember that although y_t and x_t are nonstationary, their first differences are stationary. In such a regression you will find that R^2 is practically zero, as it should be.

```
diff_xy<-lm(diff(y)~diff(x))
summary(diff_xy)

##
## Call:
## lm(formula = diff(y) ~ diff(x))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.90968 -0.70434  0.04657  0.70662  2.80407
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.11607    0.04687  -2.476   0.0136 *
## diff(x)     -0.03953    0.04491  -0.880   0.3792
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.045 on 497 degrees of freedom
## Multiple R-squared:  0.001556,Adjusted R-squared:  -0.0004525
## F-statistic: 0.7748 on 1 and 497 DF,  p-value: 0.3792
```

3.3 Unit-Root Test

AN IDEA

We employ the following models:

$$p_t = p_{t-1} + a_t$$

Then, we have

$$p_t = p_0 + \sum a_t$$

and the expectation of p_t is

$$E(p_t) = E(p_0 + \sum a_t) = p_0$$

The variance of p_t are

$$V(p_t) = V(\sum a_t) = t\sigma^2, \qquad V(p_{t-s}) = V(\sum a_t) = (t-s)\sigma^2$$

and covariance is

$$\gamma(s) = E[(p_t - p_0)(p_{t-s} - p_0)] = (t-s)\sigma^2$$

Finally, we have ACF

$$\rho(s) = \frac{(t-s)\sigma^2}{\sqrt{t\sigma^2}\sqrt{(t-s)\sigma^2}} = \sqrt{\frac{t-s}{t}}$$

It means that ACF will decrease as s increases.

THE UNIT ROOT TEST

A test of stationarity (or nonstationarity) that has become widely popular over the past several years is the unit root test. We will first explain it, then illustrate it and then consider some limitations of this test. The starting point is the unit root (stochastic) process. We start with

$$x_t = \rho x_{t-1} + a_t, \qquad -1 \leq \rho \leq 1 \tag{5}$$

where a_t is a white noise error term. We know that if $\rho = 1$, that is, in the case of the unit root, becomes a random walk model without drift, which we know is a nonstationary stochastic process. Therefore, why not simply regress x_t on its (oneperiod) lagged value x_{t-1} and find out if the estimated ρ is statistically equal to 1? If it is, then x_t is nonstationary. This is the general idea behind the unit root test of stationarity.

For theoretical reasons, we manipulate (5) as follows: Subtract Y_{t-1} from both sides of (5) to obtain:

$$Y_t - Y_{t-1} = \rho Y_{t-1} - Y_{t-1} + a_t = (\rho - 1)Y_{t-1} + a_t$$

which can be alternatively written as:

$$\Delta Y_t = \delta Y_{t-1} + a_t \tag{6}$$

where $\delta = (\rho - 1)$ and , as usual, is the first-difference operator.

In practice, therefore, we estimate (6) and test the (null) hypothesis that $\delta = 0$. If $\delta = 0$, then $\rho = 1$, that is we have a unit root, meaning the time series under consideration is nonstationary. If $\delta = 0$, we have

$$\Delta Y_t = (Y_t - Y_{t-1}) = a_t$$

Since a_t is a white noise error term, it is stationary, which means that the first differences of a random walk time series are stationary.

Now let us turn to the estimation of (6). This is simple enough; all we have to do is to take the first differences of Y_t and regress them on Y_{t-1} and see if the estimated slope coefficient in this regression $(= \hat{\delta})$ is zero or not. If it is zero, we conclude that Y_t is nonstationary. But if it is negative, we conclude that Y_t is stationary. The only question is which test we use to find out if the estimated coefficient of Y_{t-1} in (6) is zero or not. You might be tempted to say, why not use the usual t test? Unfortunately, under the null hypothesis that $\delta = 0$ (i.e., $\rho = 1$), the t value of the estimated coefficient of Y_{t-1} does

not follow the t distribution even in large samples; that is, it does not have an asymptotic normal distribution.

What is the alternative? Dickey and Fuller have shown that under the null hypothesis that $\delta = 0$, the estimated t value of the coefficient of Y_{t-1} in (6) follows the τ (tau) statistic. In the literature the tau statistic or test is known as the Dickey–Fuller (DF) test. Interestingly, if the hypothesis that $\delta = 0$ is rejected (i.e., the time series is stationary), we can use the usual (Student's) t test.

The DF test is estimated in three different forms, that is, under three different null hypotheses.

- Y_t is a random walk: $\Delta Y_t = \delta Y_{t-1} + a_t$
- Y_t is a random walk with drift: $\Delta Y_t = \beta_1 + \delta Y_{t-1} + a_t$
- Y_t is a random walk with drift around a stochastic trend: $\Delta Y_t = \beta_1 + \beta_2 t + \delta Y_{t-1} + a_t$

where t is the time or trend variable. In each case, the null hypothesis is that $\delta = 0$; that is, there is a unit root—the time series is nonstationary. The alternative hypothesis is that δ is less than zero; that is, the time series is stationary. If the null hypothesis is rejected, it means that Y_t is a stationary time series with zero mean in the case of (6), that Y_t is stationary with a nonzero mean $[= \beta 1/(1-\rho)]$[4].

If the computed absolute value of the tau statistic ($|\tau|$) exceeds the DF or MacKinnon critical tau values, we reject the hypothesis that $\delta = 0$, in which case the time series is stationary. On the other hand, if the computed $|\tau|$ does not exceed the critical tau value, we do not reject the null hypothesis, in which case the time series is nonstationary. Make sure that you use the appropriate critical τ values.

The Augmented Dickey–Fuller (ADF) Test

In conducting the DF test as in above equations, it was assumed that the error term a_t was uncorrelated. But in case the a_t are correlated, Dickey and Fuller have developed a test, known as the augmented Dickey–Fuller (ADF) test.

The ADF test here consists of estimating the following regression:

$$\Delta y_t = \beta y_{t-1} + \sum_{i=1}^{p} \gamma_i \Delta y_{t-i} + a_t \tag{7}$$

$$\Delta y_t = \mu + \beta y_{t-1} + \sum_{i=1}^{p} \gamma_i \Delta y_{t-i} + a_t \tag{8}$$

$$\Delta y_t = \mu + \delta t + \beta y_{t-1} + \sum_{i=1}^{p} \gamma_i \Delta y_{t-i} + a_t \tag{9}$$

where a_t is a pure white noise error term and where $Y_{t-1} = (Y_{t-1} - Y_{t-2})$, $Y_{t-2} = (Y_{t-2} - Y_{t-3})$, etc. The number of lagged difference terms to include is often determined empirically, the idea being to include enough terms so that the error term is serially uncorrelated. In ADF we still test whether $\delta = 0$ and the ADF test follows the same asymptotic distribution as the DF statistic, so the same critical values can be used.

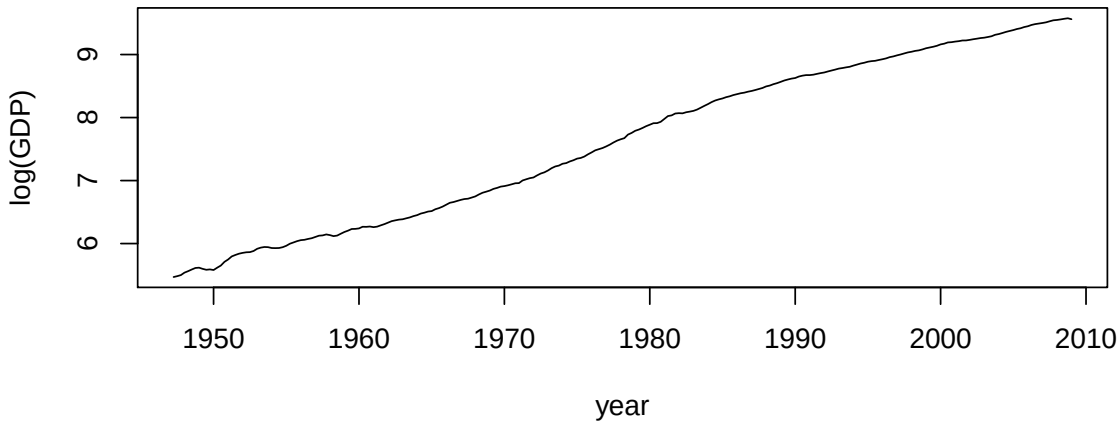
The Phillips–Perron (PP) Unit Root Tests/KPSS Tests

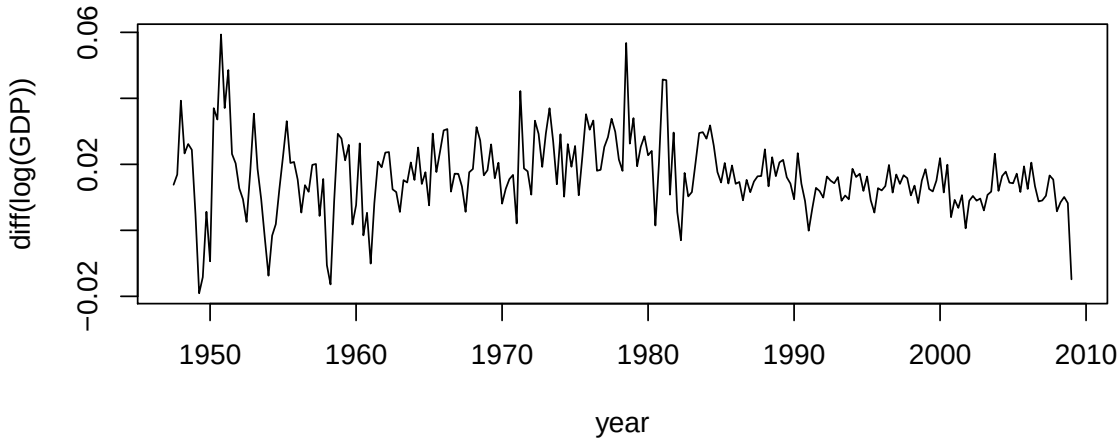
Examples:

1. Consider the log series of US quarterly GDP from 1947.I to 2008.IV. The series exhibits an upward trend, showing the growth of US economy and has high sample serial correlations.

[4]Note: Specification error problem. Some trial and error is inevitable, data mining notwithstanding.

```
library(fUnitRoots)
da=read.table("q-gdp4708.txt",header=T)
tdx=c(1:248)/4+1947
opar<-par(no.readonly = TRUE)
par(mfcol=c(2,1))
plot(tdx,log(da$gdp),xlab='year',ylab='log(GDP)',type='l')
plot(tdx[2:248],diff(log(da$gdp)),xlab='year',ylab='diff(log(GDP))',type='l')
```



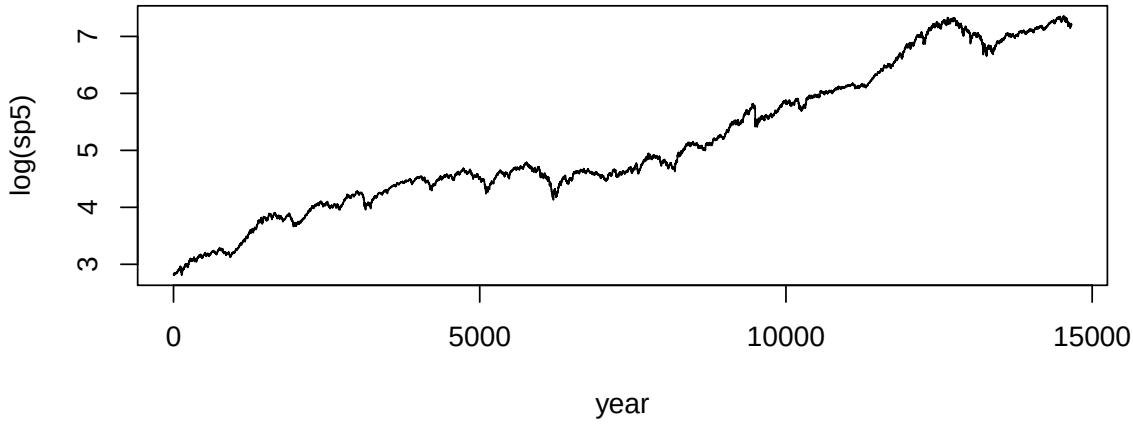


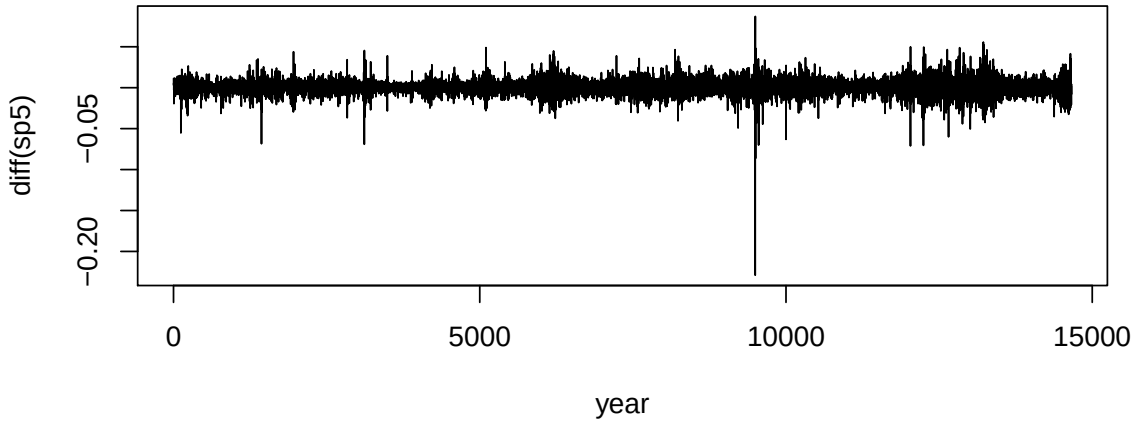

```
par(opar)
gdp=log(da[,4])
m1=ar(diff(gdp),method='mle')
m1$order

## [1] 10
```

```
adfTest(gdp,lags=10,type=c("c"))

##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 10
##   STATISTIC:
##     Dickey-Fuller: -1.6109
##   P VALUE:
##     0.4569
##
## Description:
##  Wed May 18 00:18:29 2016 by user: XXXHHF
```

2. Consider the log series of the S&P 500 index from January 3, 1950 to April 16, 2008 for 14,462 observations.

```
library(fUnitRoots)
da=read.table("d-sp55008.txt",header=T)
sp5=log(da[,7])
dsp5=diff(sp5)
opar<-par(no.readonly = TRUE)
par(mfcol=c(2,1))
tdx=c(1:length(sp5))
plot(tdx,sp5,xlab='year',ylab='log(sp5)',type='l')
tdx=c(1:length(dsp5))
plot(tdx,diff(sp5),xlab='year',ylab='diff(sp5)',type='l')
```






```
par(opar)
m2=ar(diff(sp5),method='mle') # Based on AIC
m2$order

## [1] 2

adfTest(sp5,lags=2,type=("ct"))

##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
```

```
##     Lag Order: 2
##   STATISTIC:
##     Dickey-Fuller: -2.0179
##   P VALUE:
##     0.5708
##
## Description:
##  Wed May 18 00:18:31 2016 by user: XXXHHF
```

```
adfTest(sp5,lags=15,type=("ct")) # Based on PACF
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 15
##   STATISTIC:
##     Dickey-Fuller: -1.9946
##   P VALUE:
##     0.5807
##
## Description:
##  Wed May 18 00:18:31 2016 by user: XXXHHF
```

```
m3=arima(dsp5,order=c(2,0,0))
m3
```

```
##
## Call:
## arima(x = dsp5, order = c(2, 0, 0))
##
## Coefficients:
##          ar1      ar2  intercept
##       0.0722  -0.0387      3e-04
## s.e.  0.0083   0.0083      1e-04
##
## sigma^2 estimated as 8.068e-05:  log likelihood = 48286.88,  aic = -96567.75
```

```
adfTest(dsp5,lags=2,type=("ct"))
```

```
## Warning in adfTest(dsp5, lags = 2, type = ("ct")):  p-value smaller than printed p-value
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 2
##   STATISTIC:
##     Dickey-Fuller: -70.5501
```

```
##   P VALUE:
##     0.01
##
## Description:
##  Wed May 18 00:18:31 2016 by user: XXXHHF

adfTest(dsp5,lags=15,type=("ct"))

## Warning in adfTest(dsp5, lags = 15, type = ("ct")):  p-value smaller than printed p-value

##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 15
##   STATISTIC:
##     Dickey-Fuller: -29.5117
##   P VALUE:
##     0.01
##
## Description:
##  Wed May 18 00:18:31 2016 by user: XXXHHF
```

3. A Notice
   Is there a difference between the following Command?

```
arima(x, order = c(2, 1, 0))
arima(diff(x), order = c(2, 0, 0))
```

For example:

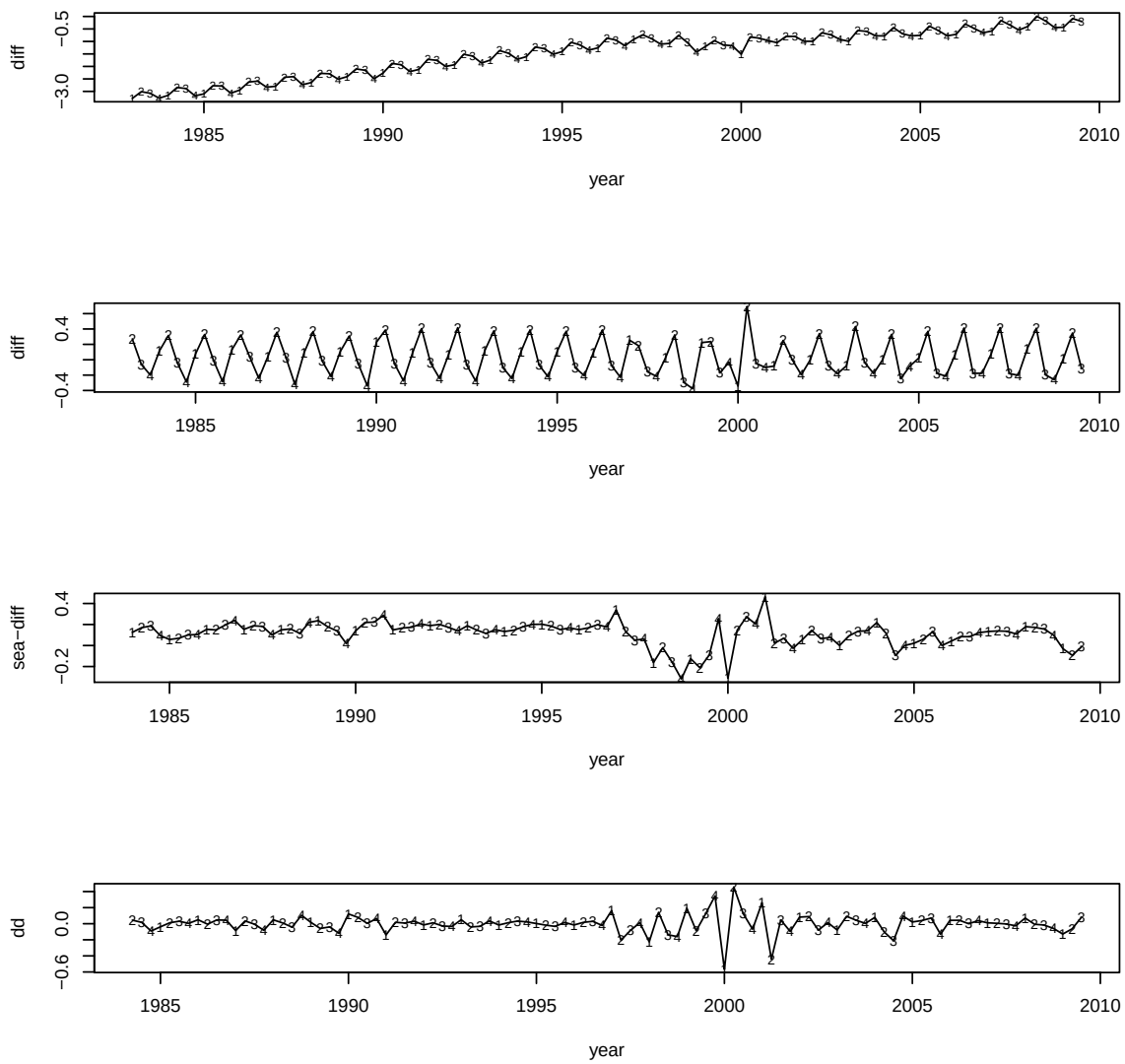
```
da=read.table("d-sp55008.txt",header=T)
sp5=log(da[,7])
dsp5=diff(sp5)
#1
arima(sp5, order = c(2, 1, 0))

##
## Call:
## arima(x = sp5, order = c(2, 1, 0))
##
## Coefficients:
##          ar1      ar2
##       0.0731  -0.0377
## s.e.  0.0083   0.0083
##
## sigma^2 estimated as 8.077e-05:  log likelihood = 48279.31,  aic = -96554.63

#2
arima(dsp5, order = c(2, 0, 0))
```

```
##
## Call:
## arima(x = dsp5, order = c(2, 0, 0))
##
## Coefficients:
##          ar1      ar2  intercept
##       0.0722  -0.0387      3e-04
## s.e.  0.0083   0.0083      1e-04
##
## sigma^2 estimated as 8.068e-05:  log likelihood = 48286.88,  aic = -96567.75

#3
tdx=c(1:length(sp5))
arima(sp5, order = c(2, 1, 0),xreg=tdx)

##
## Call:
## arima(x = sp5, order = c(2, 1, 0), xreg = tdx)
##
## Coefficients:
##          ar1      ar2   xreg
##       0.0722  -0.0387  3e-04
## s.e.  0.0083   0.0083  1e-04
##
## sigma^2 estimated as 8.068e-05:  log likelihood = 48286.88,  aic = -96567.75
```

Seasonal Differencing

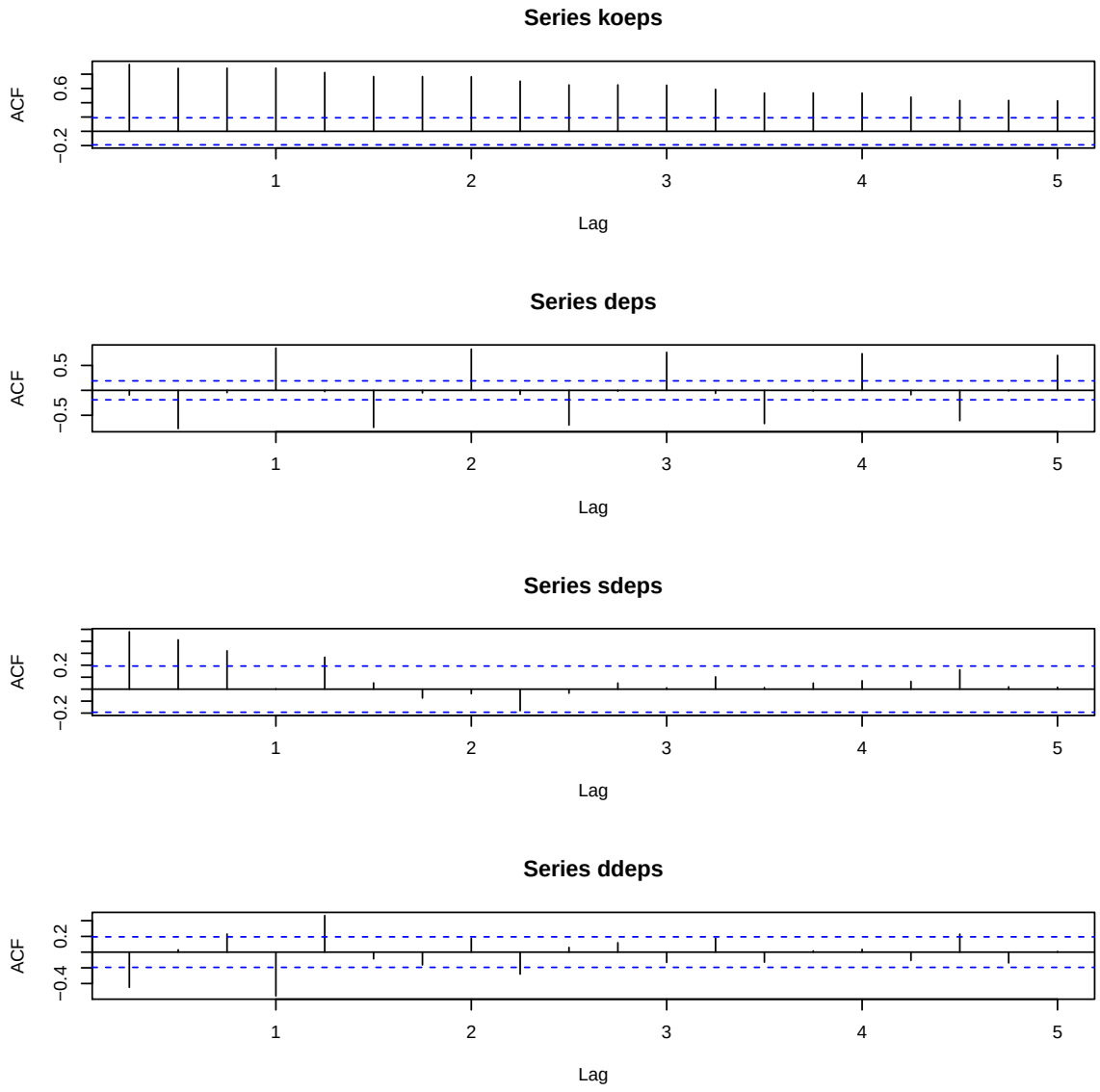
```
da=read.table("q-ko-earns8309.txt",header=T)
head(da) #index

##      pends  anntime  value
## 1 19830331 19830426 0.0375
## 2 19830630 19830725 0.0492
## 3 19830930 19831102 0.0463
## 4 19831231 19840214 0.0379
## 5 19840331 19840419 0.0425
## 6 19840630 19840720 0.0583

eps=log(da$value)
koeps=ts(eps,frequency=4,start=c(1983,1))
deps=diff(koeps)
sdeps=diff(koeps,4)
ddeps=diff(sdeps)#Time plot
opar<-par(no.readonly = TRUE)
par(mfcol=c(4,1))
c1=c("2","3","4","1")
c2=c("1","2","3","4")
#1
plot(koeps,xlab='year',ylab='diff',type='l')
```

```
points(koeps,pch=c2,cex=0.6)
#2
plot(deps,xlab='year',ylab='diff',type='l')
points(deps,pch=c1,cex=0.7)
#3
plot(sdeps,xlab='year',ylab='sea-diff',type='l')
points(sdeps,pch=c2,cex=0.7)
#4
plot(ddeps,xlab='year',ylab='dd',type='l')
points(ddeps,pch=c1,cex=0.7)
```




```
#ACF
acf(koeps,lag=20)
acf(deps,lag=20)
```

```
acf(sdeps,lag=20)
acf(ddeps,lag=20)
```




```
par(opar)
```

4 Asset volatility and volatility models

An important measure in finance is the risk associated with an asset and asset volatility is perhaps the most commonly used risk measure. There are several types of asset volatility. Volatility has many important applications in finance. It is a key factor in options pricing and asset allocation. It plays an important role in value at risk calculation for risk management.

Although asset volatility is well defined, it is not directly observable in practice. What we observe are the prices of an asset and its derivatives. One must estimate the volatility from these observed prices. The fact that volatility is not directly observable has several important implications in studying and modeling volatility.

There are many volatility models available in the literature. The univariate models discussed in this chapter include the autoregressive conditional heteroscedastic (ARCH) model of Engle (1982), the generalized autoregressive conditional heteroscedastic (GARCH) model of Bollerslev (1986), the exponential generalized autoregressive conditional heteroscedastic (EGARCH) model of Nelson (1991), the threshold generalized autoregressive conditional heteroscedastic (TGARCH) model of Glosten et al. (1993) and Zakoian (1994), the nonsymmetric generalized autoregressive conditional heteroscedastic (NGARCH) model of Engle and Ng (1993) and Duan (1995), and the stochastic volatility (SV) models of Melino and Turnbull (1990), Taylor (1994), Harvey et al. (1994), and Jacquier et al. (1994). We discuss advantages and weaknesses of each volatility model and consider some applications of volatility.

4.1 Characteristics of volatility

Although volatility is not directly observable, it has some characteristics that are commonly seen in asset returns. First, there exist volatility clusters (i.e., volatility is high for certain time periods and low for other periods). Second, volatility evolves over time in a continuous manner–that is, volatility jumps are rare. Third, volatility does not diverge to infinity–that is, volatility varies within some fixed range. Statistically speaking, this means that volatility is often stationary. Fourth, volatility seems to react differently to a big price increase and a big price drop with the latter having a greater impact. This phenomenon is referred to as the leverage effect. These properties play an important role in the development of volatility models. Some volatility models were proposed specifically to correct the weaknesses of the existing ones for their inability to capture the characteristics mentioned earlier. For example, the EGARCH and TGARCH models were developed to capture the asymmetry in volatility induced by big "positive" and "negative" asset returns.

In practice, we typically estimate the volatility of an asset using the prices of its stock or derivatives or both. Consider the daily volatility of IBM stock. What we observe are (i) the daily return for each trading day, (ii) tick-by-tick data for intraday transactions and quotes, and (iii) the prices of options contingent on IBM stock. These three data sources give rise to three types of volatility measures for IBM stock. They are as follows:

- Volatility as the conditional standard deviation of daily returns: This is the usual definition of volatility and is the focus of volatility models discussed in this chapter.
- Implied volatility: Using prices from options markets, one can use a pricing formula, for example, the Black–Scholes pricing formula, to deduce the volatility of the stock price. This volatility measure is called the implied volatility. Because implied volatility is derived under certain assumptions that relate the price of an option to that of the underlying stock, it is often criticized to be model dependent. Experience shows that implied volatility of an asset return tends to be larger than that obtained by using daily returns and a volatility model. This might be due to the risk premium for volatility in options markets or to the way daily returns are calculated.
- Realized volatility: With the availability of high frequency financial data, one can use intraday returns, for example, 5-min returns, to estimate the daily volatility. This volatility measure is called realized volatility.

4.2 Structure of a model

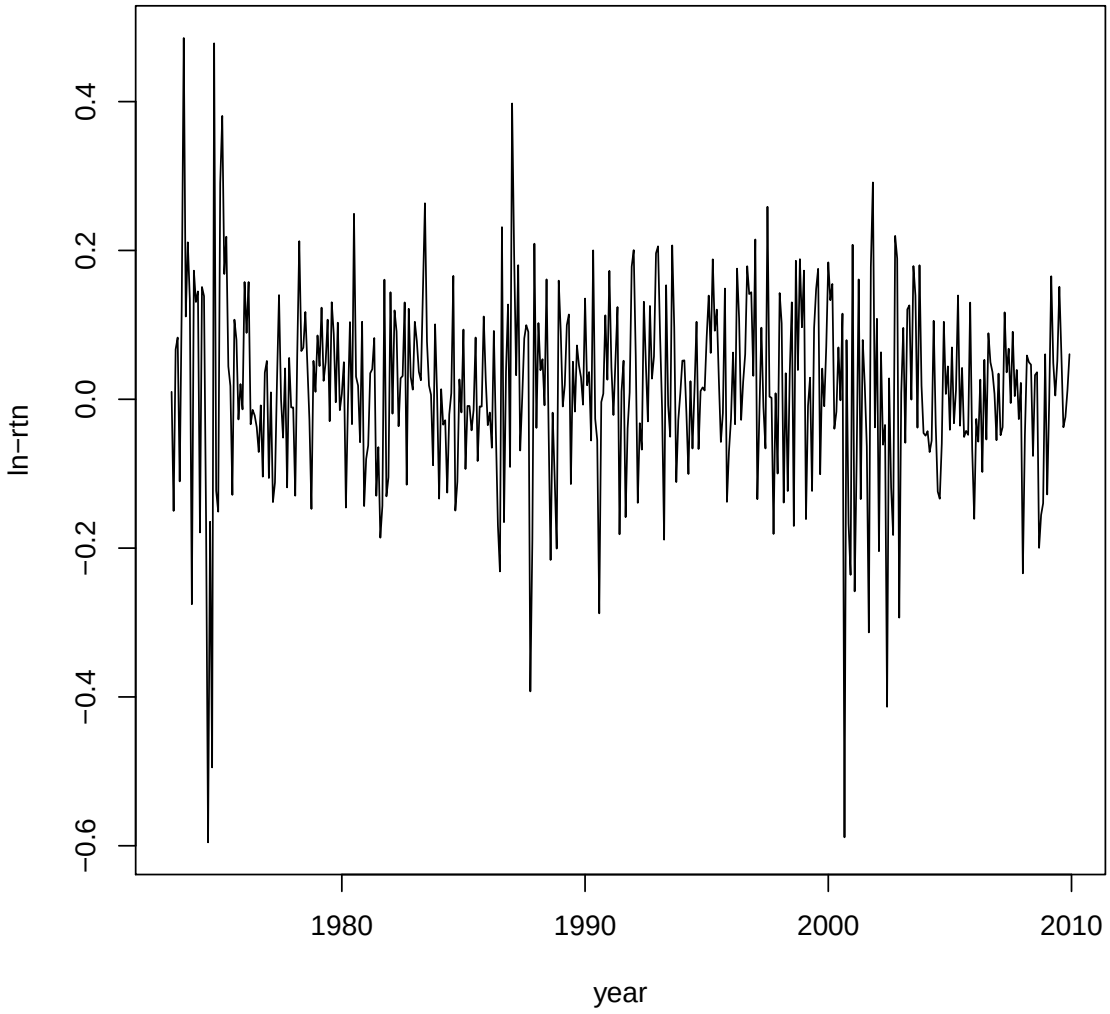
Let r_t be the log return of an asset at time index t . The basic idea behind volatility study is that the series r_t is either serially uncorrelated or with minor lower order serial correlations, but it is a

dependent series. To put the volatility models in proper perspective, it is informative to consider the conditional mean and variance of r_t given F_{t-1}; that is,

$$\mu_t = E(r_t|F_{t-1}), \qquad \sigma_t^2 = Var(r_t|F_{t-1}) = E[(r_t - \mu_t)^2|F_{t-1}].$$

where F_{t-1} denotes the information set available at time $t-1$. Typically, F_{t-1} consists of all linear functions of the past returns.

```
da=read.table("m-intcsp7309.txt",header=T)
head(da)

##       date      intc        sp
## 1 19730131  0.010050 -0.017111
## 2 19730228 -0.139303 -0.037490
## 3 19730330  0.069364 -0.001433
## 4 19730430  0.086486 -0.040800
## 5 19730531 -0.104478 -0.018884
## 6 19730629  0.133333 -0.006575

intc=log(da$intc+1)
rtn=ts(intc,frequency=12,start=c(1973,1))
plot(rtn,type='l',xlab='year',ylab='ln-rtn')# time plot
```

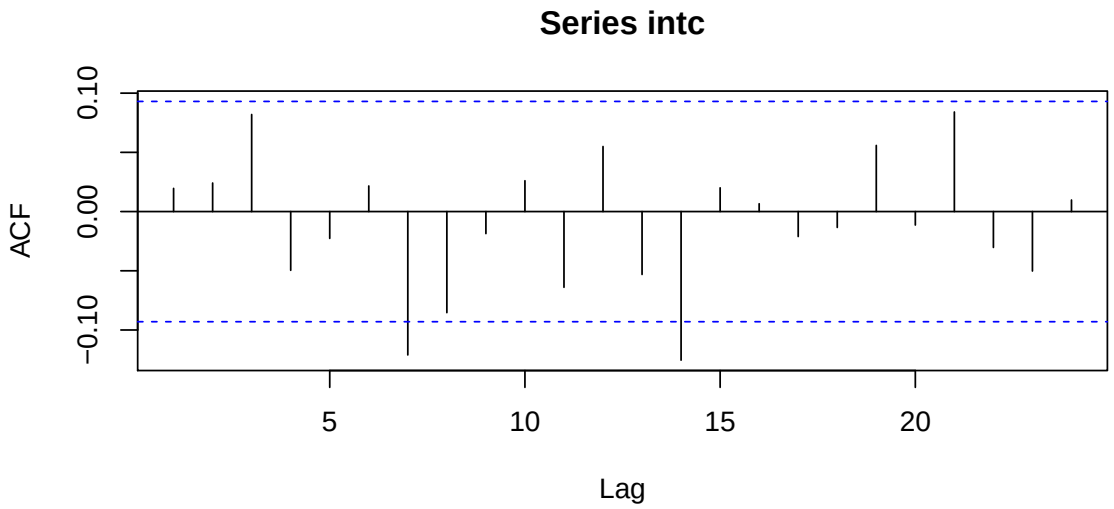


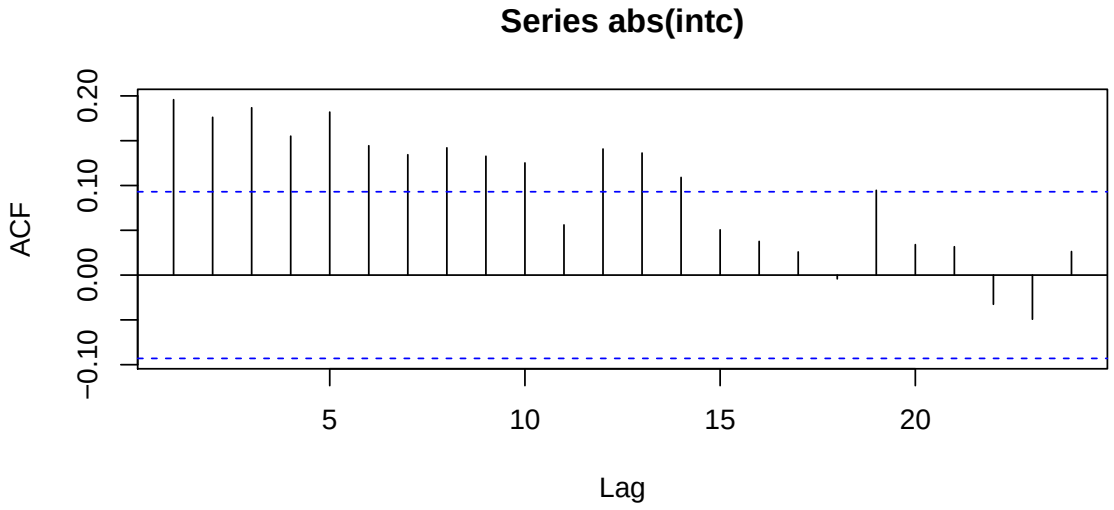

```
t.test(intc)  # testing the mean of returns
```

```
##
##  One Sample t-test
##
## data:  intc
## t = 2.3788, df = 443, p-value = 0.01779
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.00249032 0.02616428
## sample estimates:
## mean of x
## 0.0143273
```

```
Box.test(intc,lag=12,type='Ljung')
```

```
##
##  Box-Ljung test
##
## data:  intc
## X-squared = 18.676, df = 12, p-value = 0.09665

par(mfcol=c(2,1))
acf(intc,lag=24) # ACF plots
acf(abs(intc),lag=24)
```






```
Box.test(abs(intc),lag=12,type='Ljung')

##
##  Box-Ljung test
```

```
##
## data:  abs(intc)
## X-squared = 124.91, df = 12, p-value < 2.2e-16
```

We assume that r_t follows a simple time series model such as a stationary ARMA(p, q) model. For example, consider the monthly log returns of Intel stock. As shown before, the Ljung–Box statistics show that the returns have no serial correlations and a simple one-sample test confirms that the mean of r_t is significantly different from zero. More specifically, the t -ratio of testing $H_0 : \mu = 0$ versus $H_a : \mu = 0$ is 2.38 with p value 0.018.

In general, we assume that r_t follows an ARMA(p, q) model so that $r_t = \mu_t + a_t$, where μ_t is given by

$$\mu_t = \phi_0 + \sum_{i=1}^{p} \phi_i r_{t-i} - \sum_{j=1}^{q} \theta_j a_{t-j}.$$

we have

$$\sigma_t^2 = Var(r_t|F_{t-1}) = Var(a_t|F_{t-1}).$$

where the positive square root σt is the volatility.

Conditional heteroscedastic models can be classified into two general categories. Those in the first category use an exact function to govern the evolution of σ_t^2 , whereas those in the second category use a stochastic equation to describe σ_t^2. The GARCH model belongs to the first category, whereas the SV model is in the second category.

a_t is referred to as the shock or innovation of an asset return at time t. The model for μ_t is referred to as the mean equation for r_t and the model for σ_t^2 is the volatility equation for r_t. Therefore, modeling conditional heteroscedasticity amounts to augmenting a dynamic equation, which governs the time evolution of the conditional variance of the asset return to a time series model.

4.3 Modle building

Building a volatility model for an asset return series consists of four steps:

1. Specify a mean equation by testing for serial dependence in the data and, if necessary, building an econometric model (e.g., an ARMA model) for the return series to remove any linear dependence.
2. Use the residuals of the mean equation to test for ARCH effects.
3. Specify a volatility model if ARCH effects are statistically significant, and perform a joint estimation of the mean and volatility equations.
4. Check the fitted model carefully and refine it if necessary.

4.4 The ARCH model

4.4.1 Testing for ARCH effect

For ease in notation, let $a_t = r_t - \mu_t$ be the residuals of the mean equation. The squared series a_t^2 is then used to check for conditional heteroscedasticity, which is also known as the ARCH effects. Two tests are available. The first test is to apply the usual Ljung–Box statistics $Q(m)$ to the a_t^2 series; see McLeod and Li (1983). The null hypothesis of the test statistic is that the first m lags of ACF of the a_t^2 series are zero. The second test for conditional heteroscedasticity is the Lagrange multiplier test of Engle (1982). This test is equivalent to the usual F statistic for testing $\alpha_i = 0(i = 1, ..., m)$ in the linear regression

$$a_t^2 = \alpha_0 + \alpha_1 a_{t-1}^2 + \cdots + \alpha_m a_{t-m}^2 + e_t, \quad t = m+1, \ldots, T,$$

where e_t denotes the error term, m is a prespecified positive integer, and T is the sample size. Specifically, the null hypothesis is $H_0 : \alpha_1 = \cdots = \alpha_m = 0$ and the alternative hypothesis is $H_a : \alpha_i = 0$ for some i between 1 and m. Let SSR0 $= \sum_{t=m+1}^{T}(a_t^2 - \bar{\omega})^2$, where $\bar{\omega} = (1/T)\sum_{t=1}^{T} a_t^2$ is the sample mean of a_t^2 , and SSR1 $= \sum_{t=m+1}^{T} \hat{e}_t^2$, where $\hat{e}_t^2$ is the least squares residual of the prior linear regression. Then we have

$$F = \frac{(SSR_0 - SSR_1)/m}{SSR_1/(T-2m-1)},$$

which follows an F distribution with degrees of freedom m and $T - 2m - 1$ under H_0. When T is sufficiently large, one can use mF as the test statistic, which is asymptotically a chi-squared distribution with m degrees of freedom under the null hypothesis. The decision rule is to reject the null hypothesis if $mF > \chi_m^2(\alpha)$, where χ_m^2 (α) is the upper $100(1-\alpha)$th percentile of χ_m^2 , or the p value of mF is less than α, type I error.

Example 1

To demonstrate, we consider the monthly log stock returns of Intel Corporation from 1973 to 2009.

```
y=intc-mean(intc)
Box.test(y^2,lag=12,type='Ljung')

##
##  Box-Ljung test
##
## data:  y^2
## X-squared = 92.939, df = 12, p-value = 1.332e-14

source("archTest.R")
archTest(y,12)   # output edited.

##
## Call:
## lm(formula = atsq ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.07440 -0.01153 -0.00658  0.00395  0.35255
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.005977   0.002249   2.658 0.008162 **
## x1           0.093817   0.048147   1.949 0.052013 .
## x2           0.153085   0.048102   3.183 0.001569 **
## x3           0.146087   0.048614   3.005 0.002815 **
## x4           0.023539   0.049126   0.479 0.632075
## x5           0.007347   0.049107   0.150 0.881139
## x6           0.010342   0.047027   0.220 0.826050
## x7           0.057183   0.047027   1.216 0.224681
## x8           0.014320   0.047079   0.304 0.761149
## x9           0.007157   0.046968   0.152 0.878965
```

```
## x10         -0.019742   0.046566  -0.424 0.671810
## x11         -0.057537   0.046041  -1.250 0.212116
## x12          0.161945   0.045965   3.523 0.000473 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03365 on 419 degrees of freedom
## Multiple R-squared:  0.1248,Adjusted R-squared:  0.0997
## F-statistic: 4.978 on 12 and 419 DF,  p-value: 9.742e-08
```

Example 2

The ARCH effect also occurs in other financial time series. The data is the daily exchange rate between US Dollar and Euro from January 4, 1999, to August 20, 2010.

```
fx=read.table("d-useu9910.txt",header=T)
fxeu=log(fx$rate)
eu=diff(fxeu)
Box.test(eu,lag=20,type='Ljung')

##
##  Box-Ljung test
##
## data:  eu
## X-squared = 30.585, df = 20, p-value = 0.06091

t.test(eu)

##
##  One Sample t-test
##
## data:  eu
## t = 0.20217, df = 2928, p-value = 0.8398
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.0002122342  0.0002610303
## sample estimates:
##    mean of x
## 2.439805e-05

Box.test(eu^2,lag=20,type='Ljung')

##
##  Box-Ljung test
##
## data:  eu^2
## X-squared = 661.45, df = 20, p-value < 2.2e-16

archTest(eu,20)

##
## Call:
```

```
## lm(formula = atsq ~ x)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -3.229e-04 -3.369e-05 -1.949e-05  9.360e-06  2.100e-03
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.281e-05  2.535e-06   5.054 4.60e-07 ***
## x1          -3.022e-02  1.858e-02  -1.626 0.103966
## x2           9.441e-02  1.859e-02   5.080 4.02e-07 ***
## x3          -1.226e-02  1.867e-02  -0.657 0.511513
## x4           5.309e-02  1.864e-02   2.848 0.004428 **
## x5           2.668e-03  1.864e-02   0.143 0.886202
## x6           7.200e-02  1.862e-02   3.868 0.000112 ***
## x7           5.625e-02  1.866e-02   3.015 0.002594 **
## x8          -1.599e-03  1.869e-02  -0.086 0.931828
## x9           6.060e-02  1.867e-02   3.245 0.001188 **
## x10          2.794e-02  1.867e-02   1.497 0.134592
## x11          6.413e-02  1.867e-02   3.435 0.000602 ***
## x12          4.020e-02  1.867e-02   2.153 0.031439 *
## x13          4.375e-02  1.869e-02   2.341 0.019299 *
## x14          2.900e-02  1.867e-02   1.553 0.120458
## x15          4.927e-02  1.863e-02   2.645 0.008222 **
## x16          5.349e-02  1.865e-02   2.868 0.004163 **
## x17          5.702e-02  1.865e-02   3.058 0.002251 **
## x18          1.873e-03  1.868e-02   0.100 0.920136
## x19         -1.836e-02  1.859e-02  -0.987 0.323583
## x20          5.844e-02  1.859e-02   3.144 0.001683 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.483e-05 on 2888 degrees of freedom
## Multiple R-squared:  0.09265,Adjusted R-squared:  0.08636
## F-statistic: 14.74 on 20 and 2888 DF,  p-value: < 2.2e-16
```

4.4.2 The ARCH model

The first model that provides a systematic framework for volatility modeling is the ARCH model of Engle (1982). The basic idea of ARCH models is that (i) the shock at of an asset return is serially uncorrelated, but dependent, and (ii) the dependence of at can be described by a simple quadratic function of its lagged values. Specifically, an ARCH(m) model assumes that

$$a_t = \sigma_t \epsilon_t, \quad \sigma_t^2 = \alpha_0 + \alpha_1 a_{t-1}^2 + \cdots + \alpha_m a_{t-m}^2,$$

where t is a sequence of independent and identically distributed (iid) random variables with mean zero and variance 1, $\alpha_0 > 0$, and $\alpha_i \geq 0$ for $i > 0$. The coefficients α_i must satisfy some regularity conditions to ensure that the unconditional variance of at is finite. In practice, t is often assumed to follow the standard normal or a standardized Student t distribution or a generalized error distribution (GED).

Order Determination

If an ARCH effect is found to be significant, one can use the PACF of a_t^2 to determine the ARCH order.

Model Checking

For a properly specified ARCH model, the standardized residuals

$$\tilde{a}_t = \frac{a_t}{\sigma_t}$$

form a sequence of iid random variables. Therefore, one can check the adequacy of a fitted ARCH model by examining the series $\tilde{a}_t$. In particular, the Ljung–Box statistics of $\tilde{a}_t$ can be used to check the adequacy of the mean equation and that of $\tilde{a}_t^2$ can be used to test the validity of the volatility equation. The skewness, kurtosis, and quantile-to-quantile plot (i.e., QQ plot) of $\tilde{a}_t$ can be used to check the validity of the distribution assumption. The fGarch package provides many plots for a fitted volatility model.

4.4.3 Building an ARCH Model

Example 1

We continue to demonstrate the volatility modeling by using the monthly log returns of Intel stock from 1973 to 2009.

```
#1 Intel
library(fGarch) # Load package
```

```
## Warning:  package 'fGarch' was built under R version 3.2.4
```

```
da=read.table("m-intcsp7309.txt",header=T)
head(da)
```

```
##       date      intc        sp
## 1 19730131  0.010050 -0.017111
## 2 19730228 -0.139303 -0.037490
## 3 19730330  0.069364 -0.001433
## 4 19730430  0.086486 -0.040800
## 5 19730531 -0.104478 -0.018884
## 6 19730629  0.133333 -0.006575
```

```
intc=log(da$intc+1)
m1=garchFit(~1+garch(3,0),data=intc,trace=F) # Fit an ARCH(3) model
summary(m1)
```

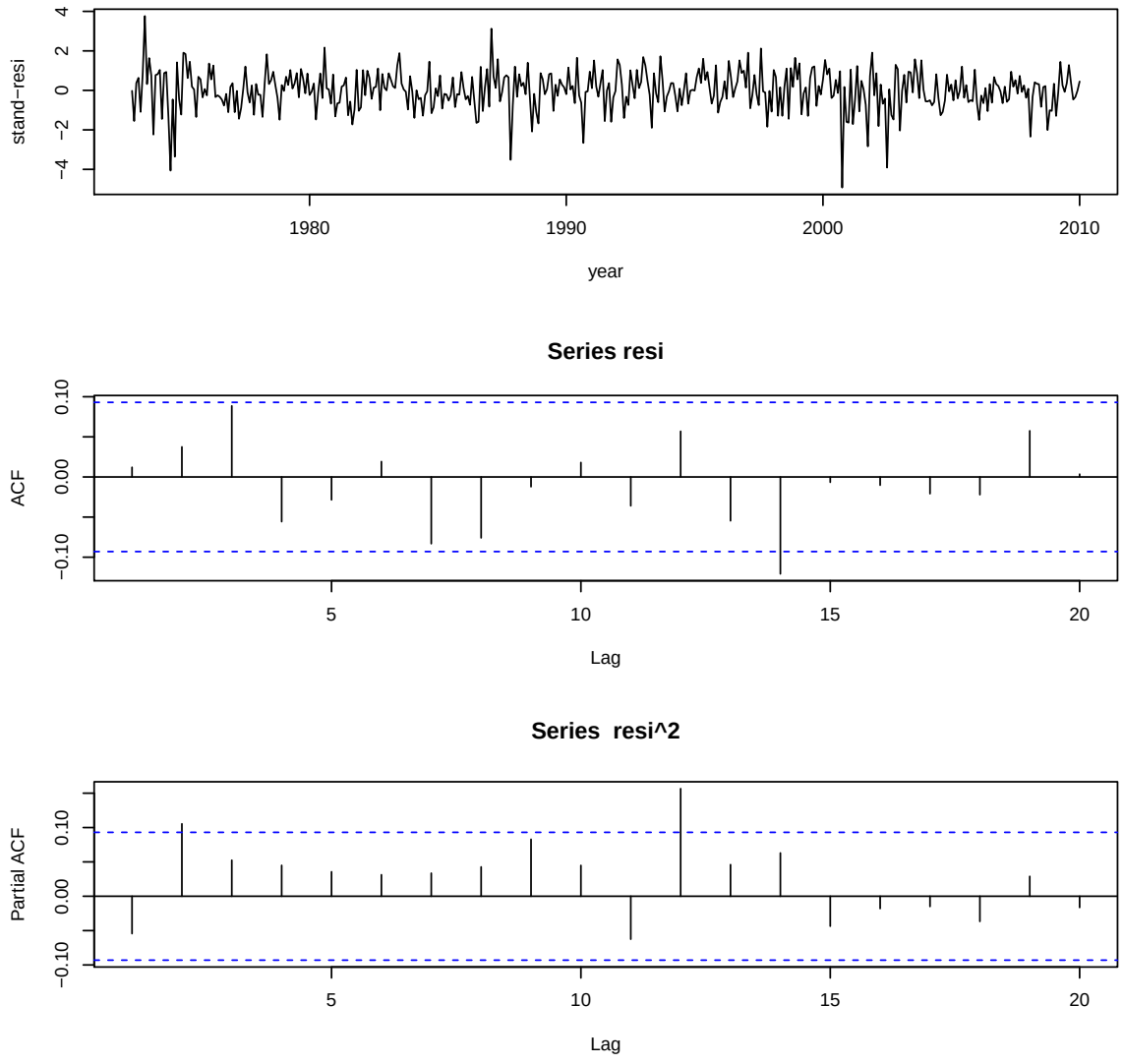
```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~1 + garch(3, 0), data = intc, trace = F)
##
## Mean and Variance Equation:
##  data ~ 1 + garch(3, 0)
```

```
## <environment: 0x000000000e986978>
##  [data = intc]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##       mu     omega    alpha1    alpha2    alpha3
## 0.012567  0.010421  0.232889  0.075069  0.051994
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.012567    0.005515    2.279   0.0227 *
## omega   0.010421    0.001238    8.418   <2e-16 ***
## alpha1  0.232889    0.111541    2.088   0.0368 *
## alpha2  0.075069    0.047305    1.587   0.1125
## alpha3  0.051994    0.045139    1.152   0.2494
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  303.9607    normalized:  0.6845963
##
## Description:
##  Wed May 18 00:18:33 2016 by user: XXXHHF
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  203.362   0
##  Shapiro-Wilk Test  R    W      0.9635971 4.898647e-09
##  Ljung-Box Test     R    Q(10)  9.260782  0.5075463
##  Ljung-Box Test     R    Q(15)  19.36748  0.1975619
##  Ljung-Box Test     R    Q(20)  20.46983  0.4289059
##  Ljung-Box Test     R^2  Q(10)  7.322136  0.6947234
##  Ljung-Box Test     R^2  Q(15)  27.41532  0.02552908
##  Ljung-Box Test     R^2  Q(20)  28.15113  0.1058698
##  LM Arch Test       R    TR^2   25.23347  0.01375447
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -1.346670 -1.300546 -1.346920 -1.328481

m2=garchFit(~1+garch(1,0),data=intc,trace=F)
summary(m2)

##
## Title:
##  GARCH Modelling
```

```
##
## Call:
##  garchFit(formula = ~1 + garch(1, 0), data = intc, trace = F)
##
## Mean and Variance Equation:
##  data ~ 1 + garch(1, 0)
## <environment: 0x000000001b62b748>
##  [data = intc]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##       mu     omega    alpha1
## 0.013130  0.011046  0.374976
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.013130    0.005318    2.469  0.01355 *
## omega   0.011046    0.001196    9.238  < 2e-16 ***
## alpha1  0.374976    0.112620    3.330  0.00087 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  299.9247    normalized:  0.675506
##
## Description:
##  Wed May 18 00:18:33 2016 by user: XXXHHF
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  144.3783  0
##  Shapiro-Wilk Test  R    W      0.9678175 2.670321e-08
##  Ljung-Box Test     R    Q(10)  12.12248  0.2769429
##  Ljung-Box Test     R    Q(15)  22.30705  0.1000019
##  Ljung-Box Test     R    Q(20)  24.33412  0.2281016
##  Ljung-Box Test     R^2  Q(10)  16.57807  0.08423723
##  Ljung-Box Test     R^2  Q(15)  37.44349  0.001089733
##  Ljung-Box Test     R^2  Q(20)  38.81395  0.007031558
##  LM Arch Test       R    TR^2   27.32897  0.006926821
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -1.337499 -1.309824 -1.337589 -1.326585
```

```
resi=residuals(m2,standardize=T)
tdx=c(1:444)/12+1973
```

```
par(mfcol=c(3,1))
plot(tdx,resi,xlab="year",ylab="stand-resi",type="l")
acf(resi,lag=20)
pacf(resi^2,lag=20)
```




```
#plot(m2)
```

We entertain an ARCH(3) model for the volatility, the standard errors of the parameters are 0.0055, 0.0012, 0.1115, 0.0473, and 0.0451, respectively. While the estimates meet the general requirement of an ARCH(3) model, the estimates of $\alpha 2$ and $\alpha 3$ appear to be statistically insignificant at the 5% level. Dropping the two insignificant parameters, we obtain the model

$$r_t = 0.0131 + a_t, \qquad \sigma_t^2 = 0.0110 + 0.3750 a_{t-1}^2,$$

where the standard errors of the parameters are 0.0053, 0.0012, and 0.1126, respectively. All the estimates are statistically significant. The ACF plot indicates that the standardized residuals have no serial correlations, but the PACF plot suggests that certain serial dependence at higher lags remains in the squared standardized residuals. Consequently, the ARCH(1) model is adequate for describing the conditional heteroscedasticity of the data at the 5% significance level if one focuses only on the lower order models.

GED

ϵ_t may assume a Generalized Error Distribution (GED) with probability density function

$$f(x) = \frac{v \exp(-\frac{1}{2}|x/\lambda|^v)}{\lambda 2^{(1+1/v)}\Gamma(1/v)}, \qquad -\infty < x < \infty, \quad 0 < v \leq \infty,$$

where $\Gamma(.)$ is the gamma function and

$$\lambda = [2^{(-2/v)}\Gamma(1/v)/\Gamma(3/v)]^{1/2}.$$

This distribution reduces to a Gaussian distribution if v = 2 and it has heavy tails when v < 2.

```
#With Student-t innovations
m3=garchFit(~1+garch(1,0),data=intc,trace=F,cond.dist="std")
summary(m3)

##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~1 + garch(1, 0), data = intc, cond.dist = "std",
##     trace = F)
##
## Mean and Variance Equation:
##  data ~ 1 + garch(1, 0)
## <environment: 0x000000000f686738>
##  [data = intc]
##
## Conditional Distribution:
##  std
##
## Coefficient(s):
##       mu     omega    alpha1     shape
## 0.017202  0.011816  0.277476  5.970266
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.017202    0.005195    3.311 0.000929 ***
## omega   0.011816    0.001560    7.574 3.62e-14 ***
## alpha1  0.277476    0.107183    2.589 0.009631 **
## shape   5.970266    1.529524    3.903 9.49e-05 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  315.0899    normalized:  0.709662
##
## Description:
##  Wed May 18 00:18:33 2016 by user: XXXHHF
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  157.7799  0
##  Shapiro-Wilk Test  R    W      0.9663975 1.488224e-08
##  Ljung-Box Test     R    Q(10)  12.8594   0.2316396
##  Ljung-Box Test     R    Q(15)  23.40632  0.07588561
##  Ljung-Box Test     R    Q(20)  25.374    0.1874956
##  Ljung-Box Test     R^2  Q(10)  19.96092  0.02962445
##  Ljung-Box Test     R^2  Q(15)  42.55549  0.0001845089
##  Ljung-Box Test     R^2  Q(20)  44.06739  0.00147397
##  LM Arch Test       R    TR^2   29.76071  0.003033508
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -1.401306 -1.364407 -1.401466 -1.386755
```

Remark. In fGarch package, the command garchFit allows for several conditional distributions, including Student t and skew Student t distributions. They are specified by cond.dist = "std" or "sstd", respectively.

Example 2

Consider the log returns of daily exchange rate between US Dollar and Euro from January 4, 1999, to August 20, 2010. As shown in previous Section, the mean equation of the log returns is $r_t = a_t$ and there exist strong ARCH effects in the data.

```
mm1=garchFit(~1+garch(11,0),data=eu,trace=F)
summary(mm1)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~1 + garch(11, 0), data = eu, trace = F)
##
## Mean and Variance Equation:
##  data ~ 1 + garch(11, 0)
## <environment: 0x000000001b2a5758>
##  [data = eu]
##
```

```
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         mu       omega      alpha1      alpha2      alpha3      alpha4
## 1.2646e-04  1.8902e-05  1.6608e-02  4.4562e-02  2.7212e-02  8.0372e-02
##     alpha5      alpha6      alpha7      alpha8      alpha9     alpha10
## 5.0110e-02  9.2191e-02  7.5282e-02  6.9537e-02  3.3467e-02  2.7823e-02
##    alpha11
## 3.8773e-02
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##          Estimate  Std. Error  t value Pr(>|t|)
## mu      1.265e-04   1.110e-04    1.140 0.254434
## omega   1.890e-05   1.727e-06   10.944  < 2e-16 ***
## alpha1  1.661e-02   1.575e-02    1.055 0.291566
## alpha2  4.456e-02   2.085e-02    2.137 0.032595 *
## alpha3  2.721e-02   1.700e-02    1.601 0.109351
## alpha4  8.037e-02   2.363e-02    3.402 0.000669 ***
## alpha5  5.011e-02   2.127e-02    2.355 0.018501 *
## alpha6  9.219e-02   2.274e-02    4.053 5.05e-05 ***
## alpha7  7.528e-02   2.406e-02    3.129 0.001755 **
## alpha8  6.954e-02   2.455e-02    2.832 0.004622 **
## alpha9  3.347e-02   2.022e-02    1.655 0.097828 .
## alpha10 2.782e-02   1.820e-02    1.528 0.126410
## alpha11 3.877e-02   1.906e-02    2.035 0.041894 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  10698.47    normalized:  3.652603
##
## Description:
##  Wed May 18 00:18:41 2016 by user: XXXHHF
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  360.8012  0
##  Shapiro-Wilk Test  R    W      0.9891735 3.894158e-14
##  Ljung-Box Test     R    Q(10)  15.77628  0.1062183
##  Ljung-Box Test     R    Q(15)  21.50105  0.12157
##  Ljung-Box Test     R    Q(20)  24.77444  0.2101969
##  Ljung-Box Test     R^2  Q(10)  4.801156  0.9040589
##  Ljung-Box Test     R^2  Q(15)  16.73088  0.3352053
##  Ljung-Box Test     R^2  Q(20)  27.56081  0.1202104
##  LM Arch Test       R    TR^2   11.96806  0.4482485
##
```

```
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -7.296329 -7.269777 -7.296368 -7.286766
```

4.5 The GARCH model

Although the ARCH model is simple, it often requires many parameters to adequately describe the volatility process of an asset return. To keep the model simple, some alternative model must be sought. Bollerslev (1986) proposes a useful extension known as the generalized ARCH (GARCH) model. For a log return series r_t, let $a_t = r_t - \mu_t$ be the innovation at time t . Then, at follows a GARCH(m, s) model if

$$a_t = \sigma_t \epsilon_t, \qquad \sigma_t^2 = \alpha_0 + \sum_{i=1}^{m} \alpha_i a_{t-i}^2 + \sum_{j=1}^{s} \beta_j \sigma_{t-j}^2,$$

where again t is a sequence of iid random variables with mean 0 and variance 1.0, $\alpha_0 > 0$, $\alpha_i \geq 0$, $\beta_j \geq 0$, and $\sum_{i=1}^{\max(m,s)}(\alpha_i + \beta_i) < 1$. Here, it is understood that $\alpha_i = 0$ for $i > m$ and $\beta_j = 0$ for $j > s$. The latter constraint on $\alpha_i + \beta_i$ implies that the unconditional variance of at is finite, whereas its conditional variance σ_t^2 evolves over time. As before, t is often assumed to follow a standard normal or standardized Student t distribution or GED. Equation reduces to a pure ARCH(m) model if $s = 0$. The α_i and β_j are referred to as ARCH and GARCH parameters, respectively.

The implied unconditional variance of r_t is

$$E(a_t^2) = \frac{\alpha_0}{1 - \sum_{i=1}^{\max(m,s)}(\alpha_i + \beta_i)}$$

provided the denominator of the prior fraction is positive.

The strengths and weaknesses of GARCH models can be easily seen:

1. a large a_{t-1}^2 or σ_{t-1}^2 gives rise to a large σ_t^2. This means that a large a_{t-1}^2 tends to be followed by another large a_t^2 , generating, again, the well-known behavior of volatility clustering in financial time series.

2. Similar to ARCH models, the tail distribution of a GARCH(1,1) process is heavier than that of a normal distribution.

3. The model provides a simple parametric function that can be used to describe the volatility evolution.

The literature on GARCH models is enormous; see Bollerslev et al. (1992, 1994), and references therein. The model encounters the same weaknesses as the ARCH model. For instance, it responds equally to positive and negative shocks. In addition, recent empirical studies of high frequency financial time series indicate that the tail behavior of GARCH models remains too short even with standardized Student t innovations. For further information about kurtosis of GARCH models, see Tsay (2010).

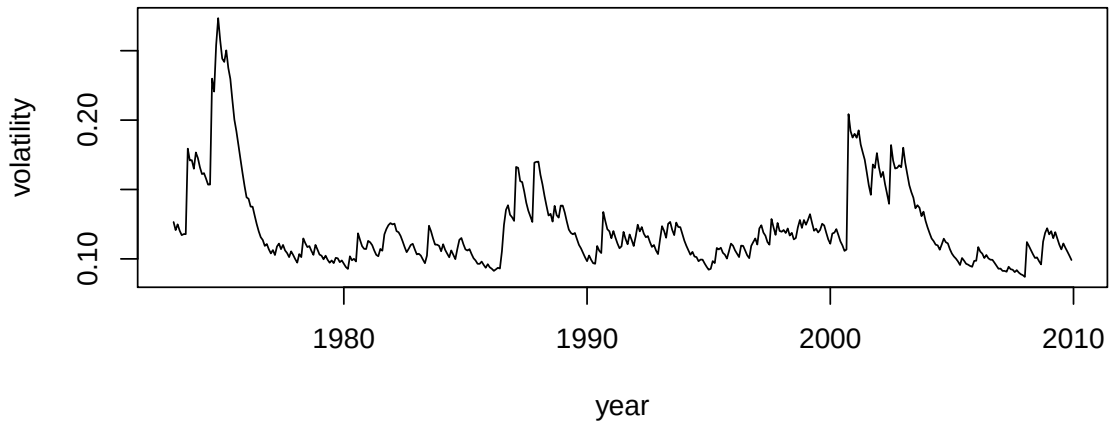
An Illustrative Example

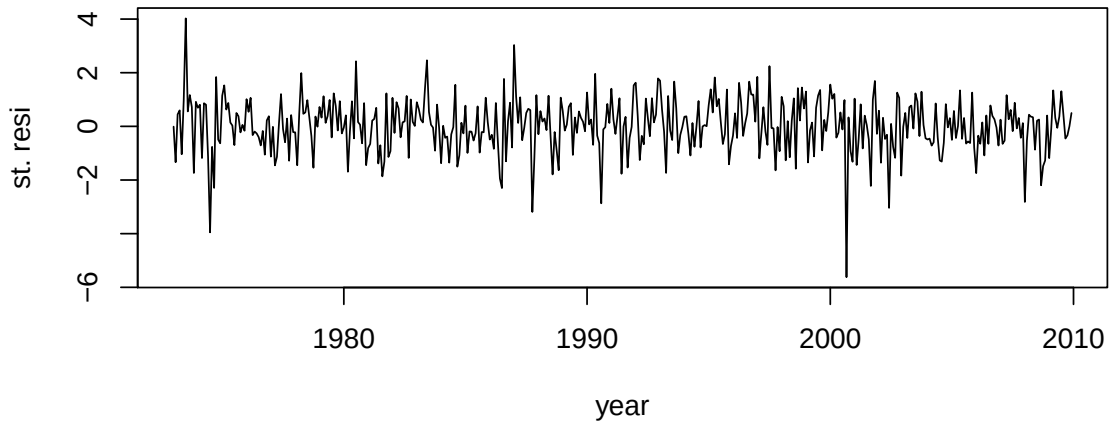
Model checking for the ARCH(1) model with Gaussian innovations of Example 1 shows that the model needs some refinement, for example, the Ljung–Box statistics of the squared standardized residuals give Q(20) = 38.81 with p value 0.007. Here, we entertain a GARCH(1,1) model for the monthly log returns of Intel stock. We employ different innovations to provide a better understanding of the return series. Again, let r_t be the monthly log return, and using Gaussian innovations, we obtain the model

```
m4=garchFit(~1+garch(1,1),data=intc,trace=F)
summary(m4)
```

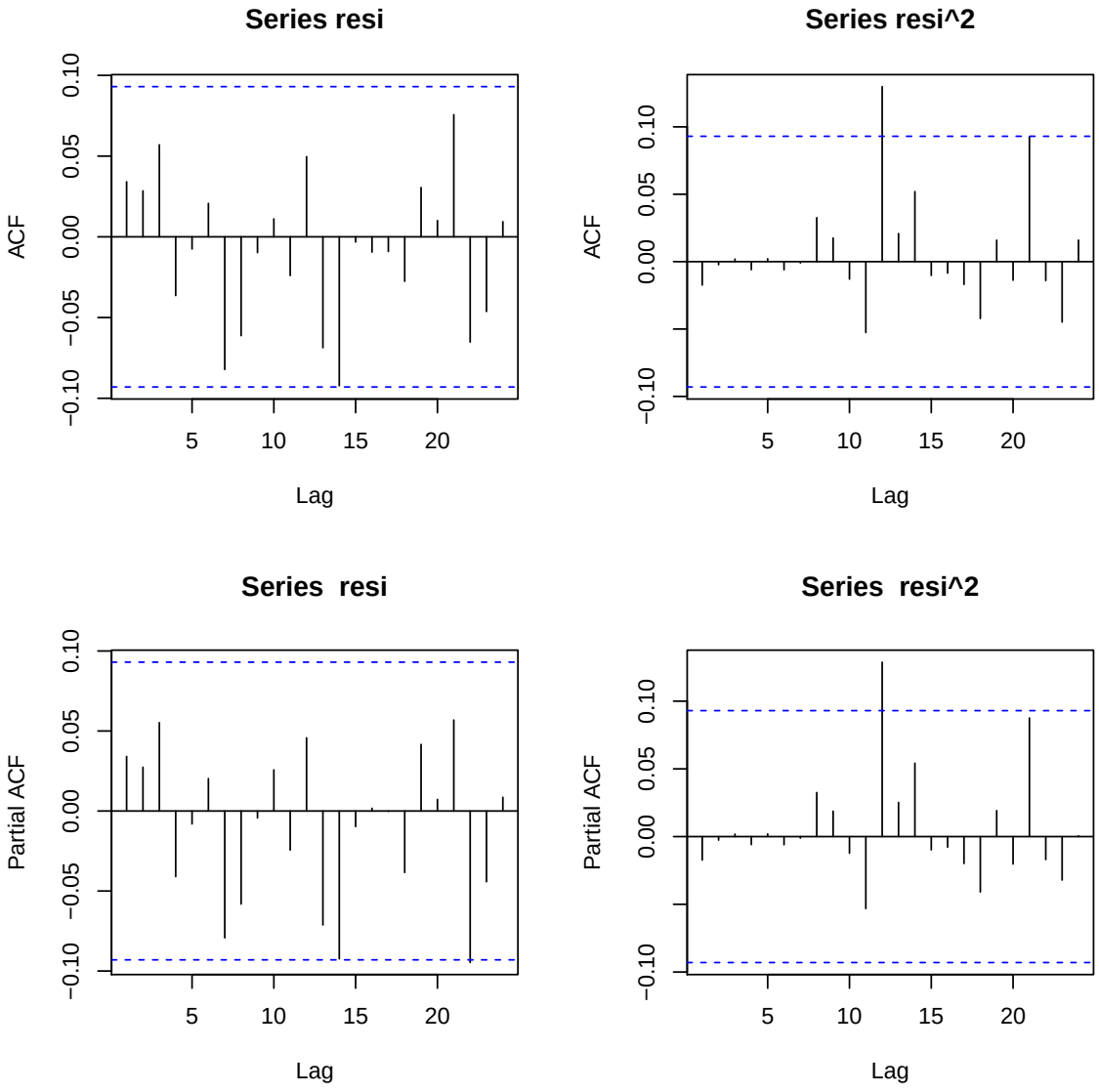
```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~1 + garch(1, 1), data = intc, trace = F)
##
## Mean and Variance Equation:
##  data ~ 1 + garch(1, 1)
## <environment: 0x000000001ac54570>
##  [data = intc]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         mu       omega      alpha1       beta1
## 0.01126568  0.00091902  0.08643831  0.85258554
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     0.0112657   0.0053931    2.089  0.03672 *
## omega  0.0009190   0.0003888    2.364  0.01808 *
## alpha1 0.0864383   0.0265439    3.256  0.00113 **
## beta1  0.8525855   0.0394322   21.622  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  312.3307    normalized:  0.7034475
##
## Description:
##  Wed May 18 00:18:41 2016 by user: XXXHHF
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  174.904   0
##  Shapiro-Wilk Test  R    W      0.9709615 1.030282e-07
##  Ljung-Box Test     R    Q(10)  8.016844  0.6271916
##  Ljung-Box Test     R    Q(15)  15.5006   0.4159946
##  Ljung-Box Test     R    Q(20)  16.41549  0.6905368
##  Ljung-Box Test     R^2  Q(10)  0.8746345 0.9999072
##  Ljung-Box Test     R^2  Q(15)  11.35935  0.7267295
##  Ljung-Box Test     R^2  Q(20)  12.55994  0.8954573
##  LM Arch Test       R    TR^2   10.51401  0.5709617
```

```
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -1.388877 -1.351978 -1.389037 -1.374326

v1=volatility(m4)  # Obtain volatility
resi=residuals(m4,standardize=T) # Standardized residuals
vol=ts(v1,frequency=12,start=c(1973,1))
res=ts(resi,frequency=12,start=c(1973,1))
par(mfcol=c(2,1))  # Show volatility and residuals
plot(vol,xlab='year',ylab='volatility',type='l')
plot(res,xlab='year',ylab='st. resi',type='l')
```





```
par(mfcol=c(2,2)) # Obtain ACF & PACF
acf(resi,lag=24)
pacf(resi,lag=24)
acf(resi^2,lag=24)
pacf(resi^2,lag=24)
```

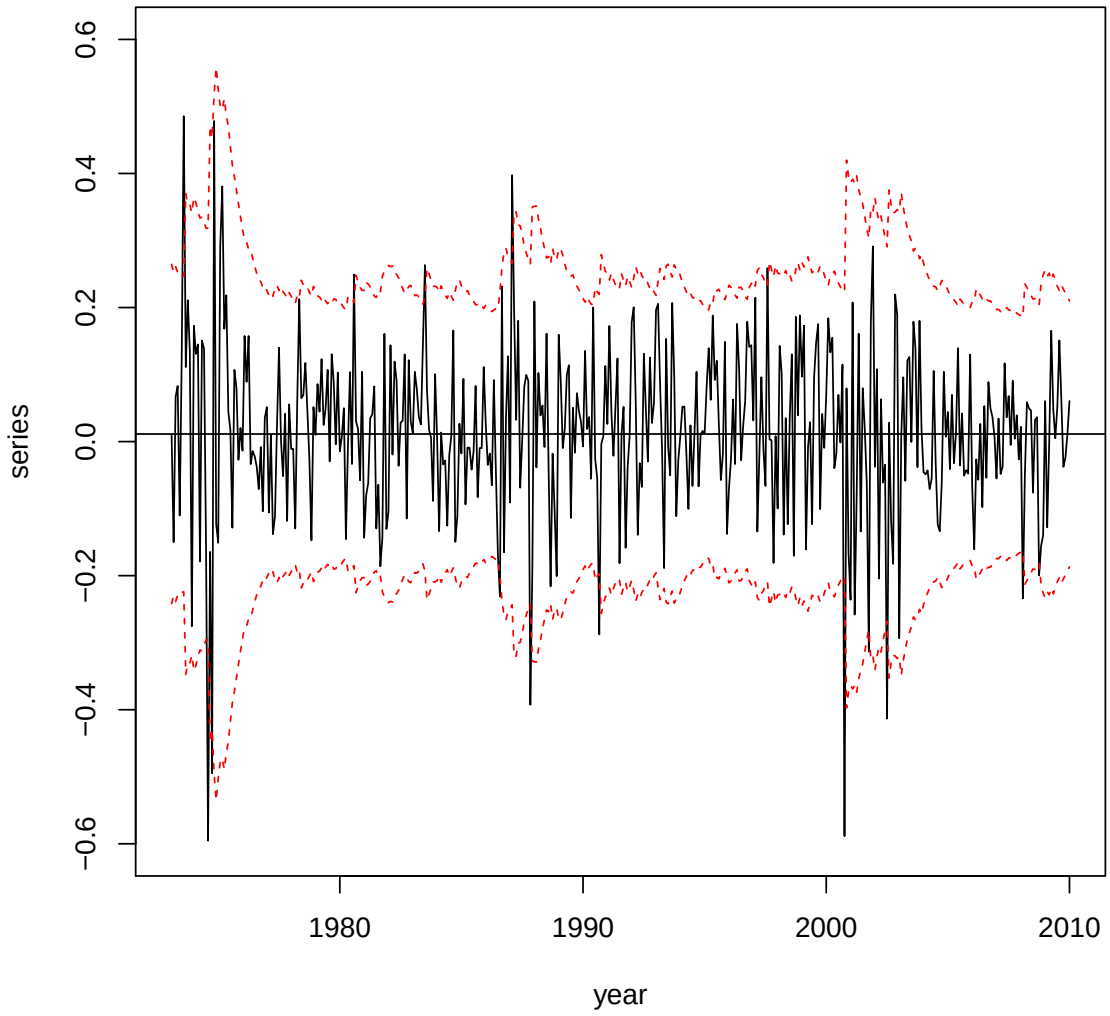


where all estimates are significant at the 5% level. Except for the normality tests, model checking statistics indicate that this Gaussian GARCH(1,1) model is adequate for r_t . AIC for the model is -1.3889. Let $\tilde{a}_t = \hat{a}_t/\sigma_t$ be the standardized residuals of the model. Except for a marginal correlation at lag 12 of ${\tilde{a}_t}^2$, these ACF and PACF confirm that the fitted model is adequate in describing the conditional mean and variance of the log return series.

Finally, the below Figure shows the time plot of the log returns with 95% pointwise predictive intervals. The intervals are calculated by $\hat{\mu} \pm \hat{\sigma}_t$, where $\hat{\mu}$= 0.0113 is the constant term of the mean equation. With some extreme exceptions, all returns are within the 95% predictive intervals. The implied unconditional variance for r_t is $0.000919/(1 - 0.0864 - 9.8526) = 0.0151$, which is slightly

smaller than the sample variance 0.0161 of the data.

```
par(mfcol=c(1,1))
upp=0.0113+2*v1
low=0.0113-2*v1
tdx=c(1:444)/12+1973
plot(tdx,intc,xlab='year',ylab='series',type='l',ylim=c(-0.6,0.6))
lines(tdx,upp,lty=2,col='red')
lines(tdx,low,lty=2,col='red')
abline(h=c(0.0113))
```



With Student-t innovations

```
m5=garchFit(~1+garch(1,1),data=intc,trace=F,cond.dist="std")
summary(m5)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~1 + garch(1, 1), data = intc, cond.dist = "std",
##     trace = F)
##
## Mean and Variance Equation:
##  data ~ 1 + garch(1, 1)
## <environment: 0x00000000198424d8>
##  [data = intc]
##
## Conditional Distribution:
##  std
##
## Coefficient(s):
##        mu      omega     alpha1      beta1      shape
## 0.0165075  0.0011576  0.1059030  0.8171313  6.7723503
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     0.0165075   0.0051031    3.235 0.001217 **
## omega  0.0011576   0.0005782    2.002 0.045286 *
## alpha1 0.1059030   0.0372047    2.846 0.004420 **
## beta1  0.8171313   0.0580141   14.085  < 2e-16 ***
## shape  6.7723503   1.8572388    3.646 0.000266 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  326.2264    normalized:  0.734744
##
## Description:
##  Wed May 18 00:18:42 2016 by user: XXXHHF
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  203.4933  0
##  Shapiro-Wilk Test  R    W      0.9687607 3.970603e-08
##  Ljung-Box Test     R    Q(10)  7.877778  0.6407741
##  Ljung-Box Test     R    Q(15)  15.5522   0.4124197
##  Ljung-Box Test     R    Q(20)  16.50475  0.6848581
##  Ljung-Box Test     R^2  Q(10)  1.066054  0.9997694
##  Ljung-Box Test     R^2  Q(15)  11.49875  0.7165045
```

```
##  Ljung-Box Test     R^2  Q(20)  12.61496  0.8932865
##  LM Arch Test       R    TR^2   10.80739  0.5454935
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -1.446966 -1.400841 -1.447215 -1.428776

v2=volatility(m5)
```

Using Student t innovations, we obtain the model

$$\begin{aligned} r_t &= 0.0165 + a_t, \quad a_t = \sigma_t \epsilon_t, \quad \epsilon_t \sim t^*_{6.77} \\ \sigma_t^2 &= 0.00116 + 0.1059 a_{t-1}^2 + 0.8171 \sigma_{t-1}^2, \end{aligned}$$

all estimates are significantly different from zero at the 5% level, and t_d^* denotes a standardized Student t distribution with d degrees of freedom. Model checking statistics show that this fitted model is adequate for the log return series. The AIC of the model in Equation (4.19) is -1.4470, and the implied unconditional variance of rt is $0.0011576/(1 - 0.0159 - 0.8171) = 0.01503$.

The sample skewness of the log returns is -0.5526, which has a t -ratio of -4.75 so that the monthly log returns of Intel stock are negatively skew. To model this skewness, we employ a skew Student t distribution for the innovations t .

```
m6=garchFit(~1+garch(1,1),data=intc,trace=F,cond.dist='sstd')
summary(m6)

##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~1 + garch(1, 1), data = intc, cond.dist = "sstd",
##     trace = F)
##
## Mean and Variance Equation:
##  data ~ 1 + garch(1, 1)
## <environment: 0x000000000e65db18>
##  [data = intc]
##
## Conditional Distribution:
##  sstd
##
## Coefficient(s):
##        mu      omega     alpha1      beta1       skew      shape
## 0.0133343  0.0011621  0.1049289  0.8177875  0.8717220  7.2344225
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     0.0133343   0.0053430    2.496 0.012572 *
```

```
## omega  0.0011621   0.0005587    2.080 0.037519 *
## alpha1 0.1049289   0.0358860    2.924 0.003456 **
## beta1  0.8177875   0.0559863   14.607  < 2e-16 ***
## skew   0.8717220   0.0629129   13.856  < 2e-16 ***
## shape  7.2344225   2.1018042    3.442 0.000577 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  328.0995    normalized:  0.7389628
##
## Description:
##  Wed May 18 00:18:42 2016 by user: XXXHHF
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  195.2178  0
##  Shapiro-Wilk Test  R    W      0.9692506 4.892686e-08
##  Ljung-Box Test     R    Q(10)  7.882126  0.6403496
##  Ljung-Box Test     R    Q(15)  15.62496  0.4074054
##  Ljung-Box Test     R    Q(20)  16.5774   0.6802193
##  Ljung-Box Test     R^2  Q(10)  1.078429  0.9997569
##  Ljung-Box Test     R^2  Q(15)  11.95155  0.6826923
##  Ljung-Box Test     R^2  Q(20)  13.03792  0.8757513
##  LM Arch Test       R    TR^2   11.18826  0.5128574
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -1.450899 -1.395550 -1.451257 -1.429071

v3=volatility(m6)
```

The resulting model is

$$\begin{aligned} r_t &= 0.0133 + a_t, \quad a_t = \sigma_t \epsilon_t, \quad \epsilon_t \sim t^*_{0.87,7.23} \\ \sigma_t^2 &= 0.00116 + 0.1049 a_{t-1}^2 + 0.8178 \sigma_{t-1}^2, \end{aligned}$$

where $t^*_{\xi,d}$ denotes a standardized skew Student t distribution with d degrees of freedom and skew parameter ξ, and all estimates are significant at the 5% level. Model checking statistics also fail to indicate any inadequacy of the fitted model in above Equation. AIC of the model is -1.4509. Note that the estimate of the skew parameter is 0.8717 with standard error 0.0629. The hypothesis of interest here is $H_0 : \xi = 1$ versus the alternative $H_a : \xi = 1$. In this particular case, the t -ratio is $t = (0.8717 - 1)/0.0629 = -2.04$ with a two-sided p value 0.041. Consequently, the null hypothesis of no skewness is rejected at the 5% level.

```
library(fBasics)
basicStats(intc)

##                   intc
## nobs        444.000000
```

```
## NAs           0.000000
## Minimum      -0.595420
## Maximum       0.485508
## 1. Quartile  -0.048741
## 3. Quartile   0.095849
## Mean          0.014327
## Median        0.018419
## Sum           6.361321
## SE Mean       0.006023
## LCL Mean      0.002490
## UCL Mean      0.026164
## Variance      0.016106
## Stdev         0.126910
## Skewness     -0.552618
## Kurtosis      3.124026
```

```
tt=-0.5526/sqrt(6/444) # Testing skewness of the data
tt
```

```
## [1] -4.753645
```

```
tt=(0.8717-1)/0.0629 # Testing skewness of the model.
tt
```

```
## [1] -2.039746
```

```
pv=2*pnorm(tt)  # Compute p-value
pv
```

```
## [1] 0.04137567
```

```
#plot(m6) 12
```

Discussion and Comparison. We have applied three GARCH(1,1) models to the monthly log returns of Intel stock from January 1973 to December 2009. All three models fit the data well. If AIC is used in model selection, one selects the one with skew Student t innovations as the best model for the data. This selection is also supported by the preliminary analysis that shows significant skewness in the returns. On the other hand, if BIC is used, then one selects the model with Student t innovations as the best one. This is not surprising because BIC, in this particular case, puts a heavier penalty for each parameter used and the p value for testing no skewness is 0.041, which is only slightly smaller than 0.05. In other words, under BIC, the penalty is heavier than the contribution of the skew parameter. This example illustrates that different criteria may select different models in volatility modeling.

```
#plot
par(mfcol=c(3,1))
plot(tdx,v1,xlab='year',ylab='volatility',type='l',ylim=c(0.06,0.3))
title(main='(a) Gaussian')
plot(tdx,v2,xlab='year',ylab='volatility',type='l',ylim=c(0.06,0.3))
title(main='(b) Student-t')
plot(tdx,v3,xlab='year',ylab='volatility',type='l',ylim=c(0.06,0.3))
title(main='(c) Skew Student-t')
```

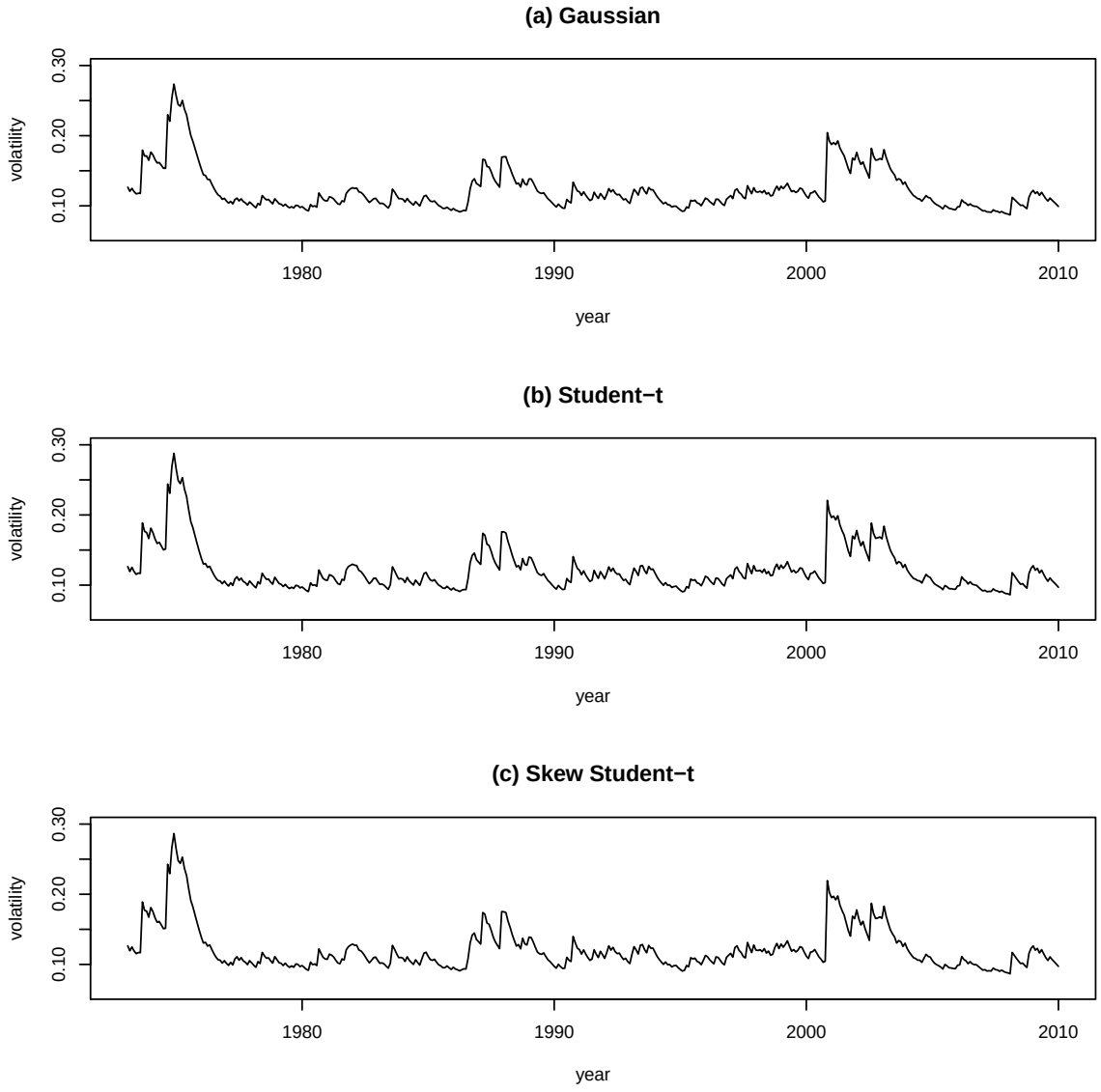


Figure provides time plots of volatility of the three models entertained. The plots are in the same scale so that a direct comparison is possible. From the plots, the three estimated volatility series are essentially the same; it is hard to see any major difference between the three volatility series. As a matter of fact, the correlation coefficients between the three volatility series are all close to one; see the attached R output. Thus, the three entertained models are close to each other.

```
cor(cbind(v1,v2,v3))

##           v1        v2        v3
## v1 1.0000000 0.9936777 0.9944356
## v2 0.9936777 1.0000000 0.9998430
## v3 0.9944356 0.9998430 1.0000000
```

THE INTEGRATED GARCH MODEL

If the AR polynomial of the GARCH representation in has a unit root, then we have an IGARCH (integrated generalized autoregressive conditional heteroscedastic) model. Thus, IGARCH models are unit-root GARCH models. Similar to ARIMA models, a key feature of IGARCH models is that the impact of past squared shocks $\eta_{t-i} = a_{t-i}^2 - \sigma_{t-i}^2$ for $i > 0$ on a_t^2 is persistent.

An IGARCH(1,1) model can be written as

$$a_t = \sigma_t^2 \varepsilon_t, \quad \sigma_t^2 = \alpha_0 + \beta_1 \sigma_{t-1}^2 + (1 - \beta_1) a_{t-1}^2 \tag{10}$$

where ε_t is defined as before and $1 > \beta_1 > 0$.

THE GARCH-M MODEL

In finance, the return of an asset may depend on its volatility. To model such a phenomenon, one may consider the GARCH-M model, where "M" stands for GARCH in the mean. A simple GARCH(1,1)-M model can be written as

$$\begin{aligned} r_t &= \mu + c\sigma_t^2 + a_t, \quad a_t = \sigma_t \varepsilon_t, \\ \sigma_t^2 &= \alpha_0 + \alpha_{t-1}^2 + \beta_1 \sigma_{t-1}^2, \end{aligned}$$

where μ and c are constants. The parameter c is called the risk premium parameter. A positive c indicates that the return is positively related to its past volatility. The formulation of the GARCH-M model implies that there are serial correlations in the return series r_t. These serial correlations are introduced by those in the volatility process σ_t^2. The existence of risk premium is, therefore, another reason that some historical stock returns have serial correlations.

4.6 An Example

In this section, we apply the GARCH model to improve the modeling and forecasting of a time series. The increase in oil prices of Summer 2008 and Spring 2011 had substantial impacts on the global economy. Predicting oil price is, therefore, an interesting and important topic. Oil prices, however, are influenced by many factors and external shocks, and are not easy to analyze. In this application, we employ the weekly crude oil prices of the US from January 3, 1997 to September 24, 2010 with 717 observations. The prices are in dollars per barrel and are the spot price FOB (freight on board) weighted by estimated import volume. The data are downloaded from the US Energy Information Administration.

```
library(fGarch)
da=read.table("w-petroprice.txt",header=T)
head(da)
```

```
##   Mon Day Year World    US
## 1   1   3 1997 23.18 22.90
## 2   1  10 1997 23.84 23.56
## 3   1  17 1997 22.99 22.79
## 4   1  24 1997 22.05 21.83
## 5   1  31 1997 21.87 21.16
## 6   2   7 1997 21.56 20.97
```

```
str(da)
```

```
## 'data.frame': 717 obs. of  5 variables:
##  $ Mon  : int  1 1 1 1 1 2 2 2 2 3 ...
##  $ Day  : int  3 10 17 24 31 7 14 21 28 7 ...
##  $ Year : int  1997 1997 1997 1997 1997 1997 1997 1997 1997 1997 ...
##  $ World: num  23.2 23.8 23 22.1 21.9 ...
##  $ US   : num  22.9 23.6 22.8 21.8 21.2 ...

price=ts(da$US,frequency=52,start=c(1997,1))
dp=ts(diff(price),frequency=52,start=c(1997,2))
par(mfcol=c(2,1))
plot(price,xlab='year',ylab='price')
plot(dp,xlab='year',ylab='changes')
```

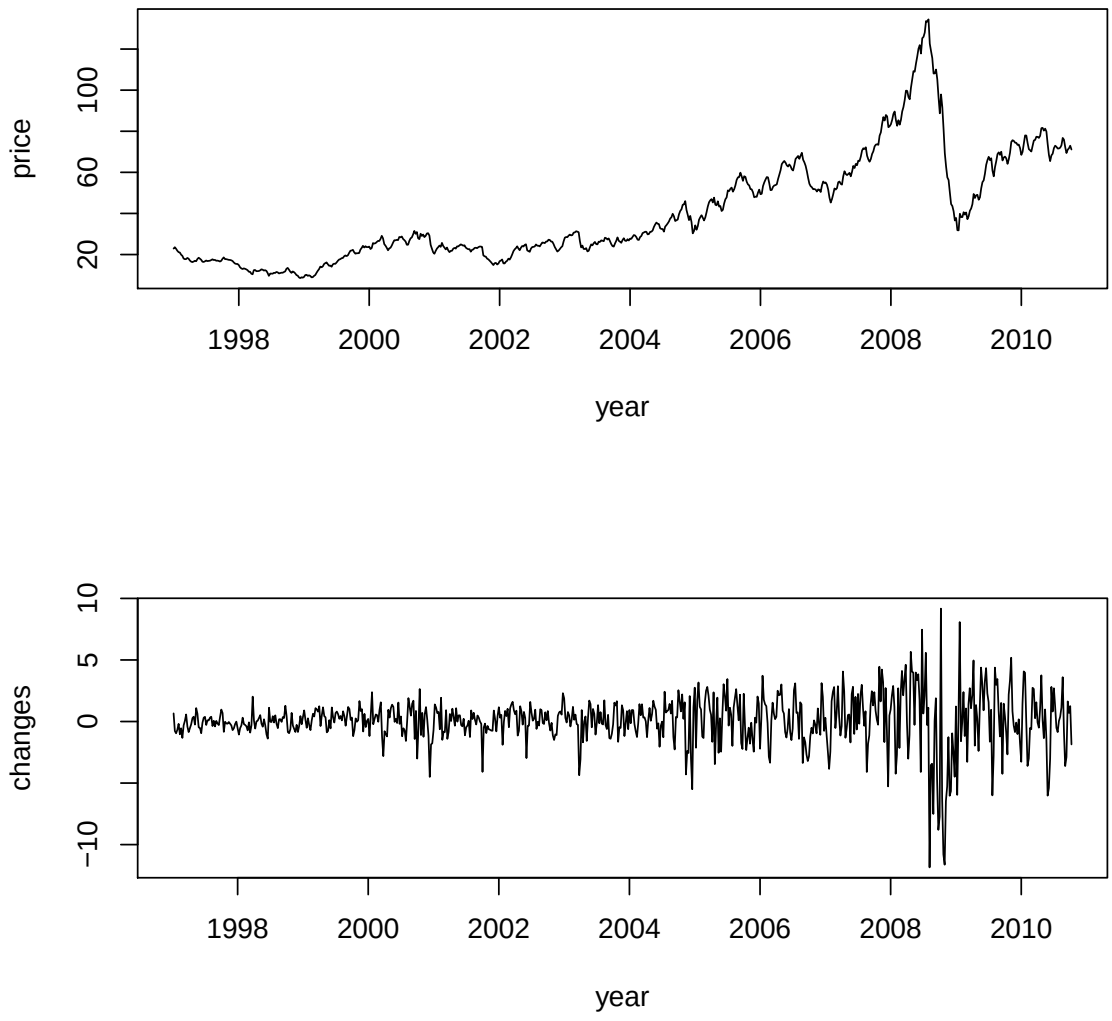
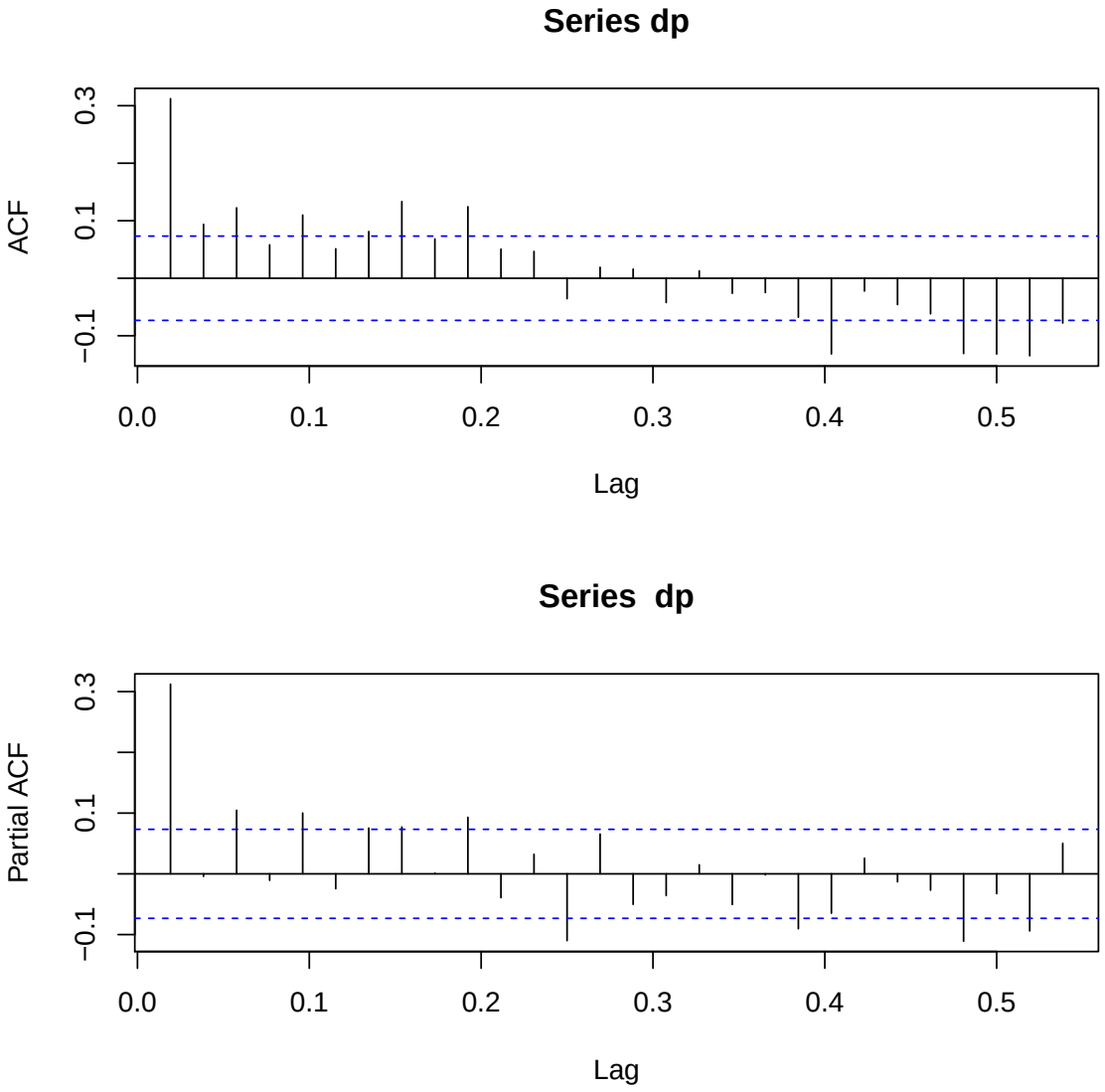


Figure a shows the time plot of the oil prices. The increase in oil prices during Summer 2008 is clearly seen. The prices also exhibit an increasing trend so that the price series is nonstationary. Figure

b shows the changes in the weekly price. This differenced series shows volatility clusters, but it has no obvious violation of being weakly stationary. We shall focus our analysis on the change series. Our analysis demonstrates that a pure ARMA model for the price change is not adequate because, among other reasons, ARMA models cannot handle volatility clusters. On the other hand, an ARMA-GARCH model can adequately handle the complexity of the data and produce improvement in out-of-sample prediction.

```
par(mfcol=c(2,1))
acf(dp,lag.max  = 28)
pacf(dp,lag.max = 28)
```



Denote the price change series by C_t. Aboved Figure gives the sample ACF and PACF of the C_t series. These correlations confirm the weak stationarity of C_t. They also show that there exists certain periodic behavior in the crude oil prices. The ACF and PACF have significant values at lags 5, 10, 20, and 25. We regard this periodic behavior as a weekly pattern because there are five working days in

a week. Consequently, we employ a seasonal model for C_t. Focusing on the PACF of C_t in Figure, we see that, except for the seasonal lags, PACFs are significant mainly at lags 1 and 3. This implies that an AR(3) would be sufficient for the regular component of the change series. Turn to seasonal pattern. The ACF and PACF at the seasonal lags are not large, even though they are outside the asymptotic two standard-error limits. Therefore, it suffices to start with a lower order seasonal model.

```
m2=arima(cprice,order=c(3,0,0),seasonal=list(order=c(2,0,0),period=5))
m2
m2=arima(cprice,order=c(3,0,0),seasonal=list(order=c(2,0,0),period=5),include.mean=F)
m2
length(cprice)
m2=arima(cprice,seasonal=list(order=c(2,0,0),period=5),include.mean=F)
m2
```

The model appears to be reasonable except the higher volatilities from 2008 to 2010. Some of the standardized residuals have magnitudes around 6, which are rather high compared with standard normal distribution. Further improvement is needed and we turn to GARCH modeling. As the fGarch package does not specifically handle seasonal mean equations, we decide to remove the weak seasonality from the change series. To this end, we fit a pure seasonal model to C_t.

```
adjcp=cprice[11:716]-0.0983*cprice[6:711]-0.1152*cprice[1:706]
par(mfcol=c(2,1))
acf(adjcp)
pacf(adjcp)
```

This model is not adequate, but it provides a simple filter that can be used to remove seasonality from C_t. In fact, for this particular instance, the residual series C_t^*has no significant serial correlations at the seasonal lags. Figure shows the sample ACF and PACF of the C_t^* series. Except for lag 25, there are no significant correlations at the seasonal lags. With the removal of seasonal component, we apply an AR(3)-GARCH(1,1) model to the adjusted series C_t^* of the price change.

```
m3=garchFit(~arma(3,0)+garch(1,1),data=adjcp,trace=F,include.mean=F)
summary(m3)
#plot(m3)
#3,13
m4=garchFit(~arma(3,0)+garch(1,1),data=adjcp,trace=F,include.mean=F,cond.dist="std")
summary(m4)
#plot(m4)
#3,13
m5=garchFit(~arma(1,0)+garch(1,1),data=adjcp,trace=F,include.mean=F,cond.dist="sstd")
summary(m5)
#plot(m5)
#3,13
```