

The background of the slide is a reproduction of the painting 'The Starry Night' by the Dutch Impressionist painter J.M.W. Turner. The painting depicts a night sky filled with swirling, luminous clouds and stars, rendered in shades of blue and yellow. In the foreground, a dark, silhouetted cypress tree stands prominently on the left side, its form mirrored in the dark, wavy lines of the water below. In the distance, a small village with white buildings and a church spire is visible under the turbulent sky.

Introductory Financial Econometrics

Spring 2020

Wasin Siwasarit

Project Spring 2020, Faculty of Economics, Thammasat University.

Academic Year Spring 2020

Road Map of this class:





1. Financial Time Series and Their Characteristics

Financial time series (FTS) analysis

Financial time series (FTS) analysis is concerned with theory and practice of asset valuation over time.

What is the difference, if any, from traditional time series analysis?

Two topics are highly related, but FTS has added uncertainty, because it must deal with the ever-changing business & economic environment and the fact that volatility is not directly observed.

1.1 The Objectives of this chapter

1. to access financial data online and to process the embedded information
2. to provide basic knowledge of FTS data such as skewness, heavy tails, and measure of dependence between asset returns
3. to introduce statistical tools econometric models useful for analyzing these series.
4. to gain experience in analyzing FTS

1.2 Examples of financial time series

1. Daily log returns of Apple stock: 2007 to 2019 (13 years). Data downloaded using quantmod

2. The VIX index.

3. CDS spreads: Daily 3-year CDS spreads of JP Morgan from July 20, 2004 to September 19, 2018.

4. Quarterly earnings of Coca-Cola Company: 1983-2009 Seasonal time series useful in

- earning forecasts
- pricing weather related derivatives (e.g. energy)
- modeling intraday behavior of asset returns

5. US monthly interest rates (3m & 6m Treasury bills)

Relations between the two asset returns? Term structure of interest rates.

6. Exchange rate between US Dollar vs Euro Fixed income, hedging, carry trade.

7. Size of insurance claims.

8. High-frequency financial data:

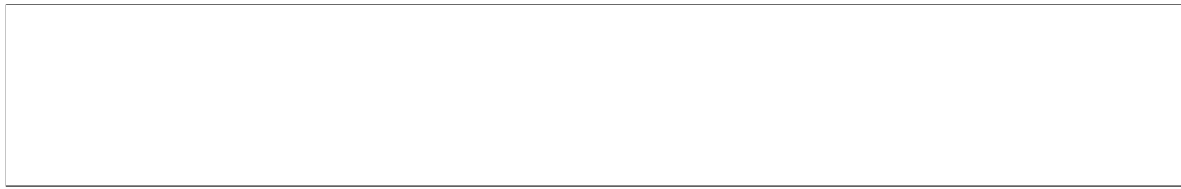
Tick-by-tick data of Caterpillars stock: January 04, 2010.

1.3 Asset Returns

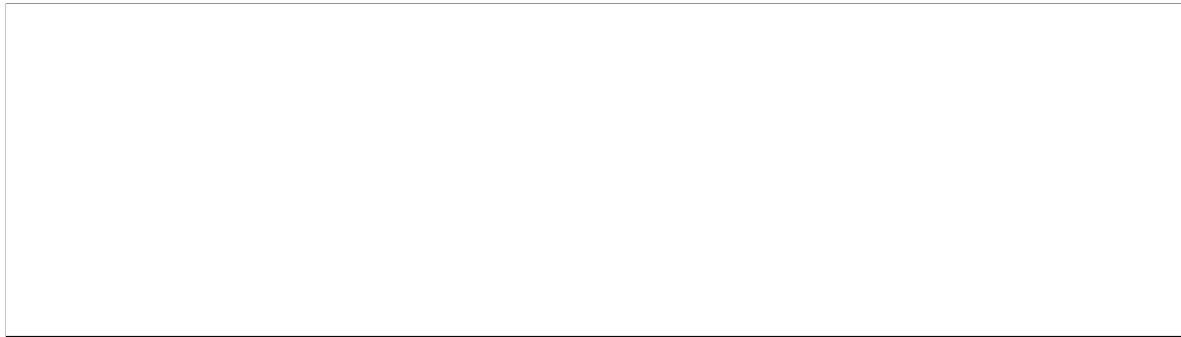
Let P_t be the price of an asset at time t , and assume no dividend. One-period simple return: Gross return

$$1 + R_t = \frac{P_t}{P_{t-1}}$$

One-Period Simple Net Return or Simple Return:



Multiperiod simple return: Gross return)




Example: Table below gives 5 daily prices of Apple stock in January 2020.

Apple Inc. (AAPL)
NasdaqGS - NasdaqGS Real Time Price. Currency in USD ☆ Add to watchlist

310.33 +0.70 (+0.23%)
At close: January 10 4:00PM EST

[Buy](#) [Sell](#)

[Summary](#) [Company Outlook](#) [Chart](#) [Conversations](#) [Statistics](#) **[Historical Data](#)** [Profile](#) [Financials](#) [Analysis](#) [Options](#)



Time Period: [Jan 12, 2019 - Jan 12, 2020](#) Show: [Historical Prices](#) Frequency: [Daily](#) [Apply](#)

Currency in USD [Download Data](#)

Date	Open	High	Low	Close*	Adj Close**	Volume
Jan 10, 2020	310.60	312.67	308.25	310.33	310.33	35,217,272
Jan 09, 2020	307.24	310.43	306.20	309.63	309.63	42,527,100
Jan 08, 2020	297.16	304.44	297.16	303.19	303.19	33,019,800
Jan 07, 2020	299.84	300.90	297.48	298.39	298.39	27,218,000
Jan 06, 2020	293.79	299.96	292.75	299.80	299.80	29,596,800

what is one-day gross return of holding the stock from 01/06 to 12/07 and the daily simple return?

what is one-day log return of holding the stock from 01/09 to 01/10 ?

Time interval is important! Default is one year. Annualized (average) return:

Besides the simple return, we can also compute the continuously compounding interest rate where r is the interest rate per annum, C is the initial capital, n is the number of years, and \exp is the exponential function.

$$A = C \times \exp(r \times n)$$

Continuously compounded (or log) return

$$r_t = \ln(1 + R_t) = \ln \frac{P_t}{P_{t-1}} = p_t - p_{t-1}$$

where $p_t = \ln(P_t)$

Multiperiod log return:



Continuously compounding: Illustration of the power of compounding (int. rate 10 % per annum)

Type	#(payment)	Int.	Net
Annual	1	0.1	\$1.10000
Semi-Annual	2	0.05	\$1.10250
Quarterly	4	0.025	\$1.10381
Monthly	12	0.0083	\$1.10471
Weekly	52	$\frac{0.1}{52}$	\$1.10506
Daily	365	$\frac{0.1}{365}$	\$1.10516
Continuously	∞		\$1.10517

Portfolio return: N assets

Dividend payment:

Excess Returns (adjusting for risk)

Remarks:

Example If the monthly log returns of an asset are 4.46 %, -7.34 % and 10.77 %, then what is the corresponding quarterly log return?

Example If the monthly simple returns of an asset are 4.46 %, -7.34 %, and 10.77 %, then what is the corresponding quarterly simple return?

1.4 Distributional Properties of Returns

What is the distribution of r_{it} where $i = 1, \dots, N$; and $t = 1, \dots, T$

Some theoretical properties:

Moments of a random variable X with density $f(x)$: l -th moment

$$m'_l = E(X^l) = \int_{-\infty}^{\infty} x^l f(x) dx$$

First Moment: mean or expectation of X .

l -th central moment

$$m_l = E[(X - \mu_x)^l] = \int_{-\infty}^{\infty} (x - \mu_x)^l f(x) dx$$

2nd central moment.

Standard deviation: square-root of variance

Skewness (Symmetry)

$$S(x) = E \left[\frac{(X - \mu_x)^3}{\sigma_x^3} \right]$$

Kurtosis (Fat-tails)

$$K(x) = E \left[\frac{(X - \mu_x)^4}{\sigma_x^4} \right]$$

Q1: Why study the mean and variance of returns?

They are concerned with long-term return and risk, respectively.

Q2: Why is symmetry important?

Symmetry has important implications in holding short or long financial positions and in risk management.

Q3: Why is kurtosis important?

Related to volatility forecasting, efficiency in estimation and tests High kurtosis implies heavy (or long) tails in distribution.

Estimation

Sample mean, Sample Variance, Sample Skewness and Sample Kurtosis



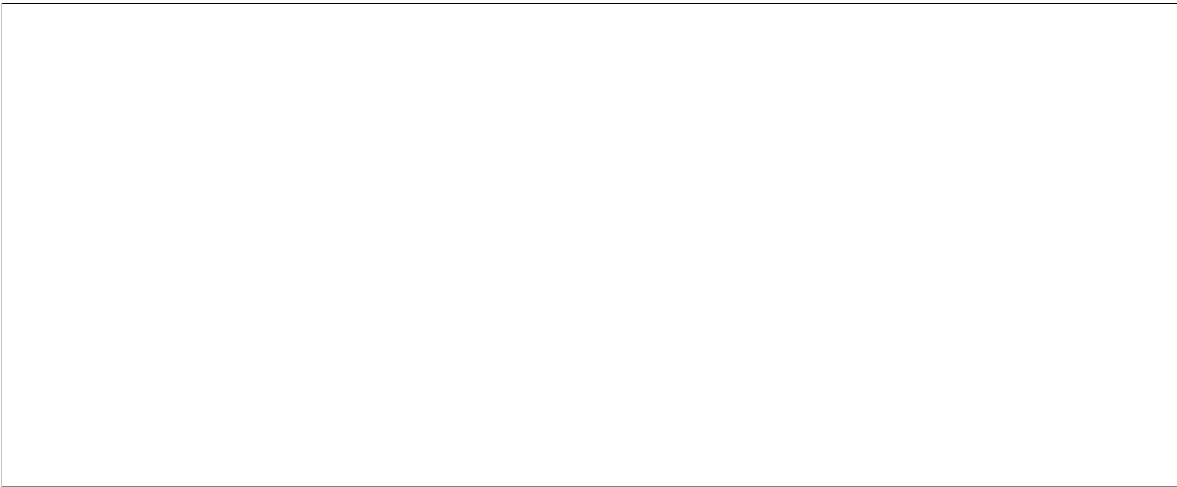
1.5 Hypothesis Testing

A random variable under the normal distribution

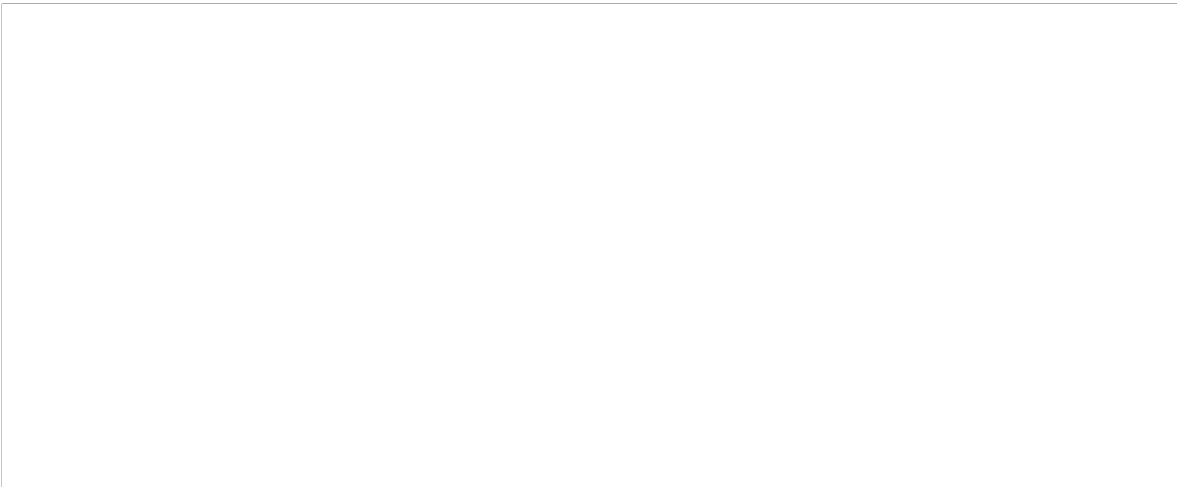
$$\widehat{S}(x) \sim N\left(0, \frac{6}{T}\right)$$

$$\widehat{K}(x) - 3 \sim N\left(0, \frac{24}{T}\right)$$

Test for symmetry



Test for tail thickness



Test for normality :(Jarque-Bera test)



1.6 Empirical work using R program

FE code

```
#EE435 Wasin Siwasarit Lecture1 Fall/2020
setwd("/Users/wasin_siwasarit/Desktop/EE435")
cat(rep("\n",50)) #clear R Console
#install.packages("quantmod")
#install.packages("fBasics")
#install.packages("sn")
#install.packages("PerformanceAnalytics")
#install.packages("car")
#install.packages("tseries")
#install.packages("forecast")
library(quantmod)
library(fBasics)
library(sn)
library(PerformanceAnalytics)
library(car)
library(tseries)
library(forecast)

getSymbols("^GSPC",from="2000-01-03",to="2020-01-10")
dim(GSPC)
head(GSPC)
tail(GSPC)
da=GSPC
chartSeries(GSPC,theme="white")
price=da[,6]
plot(price,type='l')
```

```

logprice=log(price)
plot(logprice,type='l')
logreturn=diff(log(price))
simplereturn <-exp(logreturn)-1
#1 Plot the series of log return and simple return

par(mfrow=c(1,1))
plot(logreturn,type='l')
plot(simplereturn)

newlogreturn <- logreturn[2:nrow(logreturn),]
newsimplereturn <- simplereturn[2:nrow(logreturn),]

#2 Histogram and sample statistics
hist(logreturn, breaks=100, col="slateblue")
chart.Histogram(logreturn,methods = c("add.normal"))
table.Stats(logreturn)

#3 QQ-plots and tests for normality
#
# use qqnorm function
par(mfrow=c(1,1))
qqnorm(newlogreturn)
qqline(newlogreturn, col = 2)
jarque.bera.test(newlogreturn)

```

FE Print out

```

> #install.packages("quantmod")
> #install.packages("fBasics")
> #install.packages("sn")
> #install.packages("PerformanceAnalytics")
> #install.packages("car")
> #install.packages("tseries")
> #install.packages("forecast")
> library(quantmod)
> library(fBasics)
> library(sn)
> library(PerformanceAnalytics)
> library(car)
> library(tseries)
> library(forecast)
> getSymbols("^GSPC",from="2000-01-03",to="2020-08-12")
[1] "^GSPC"
> dim(GSPC)
[1] 5185    6

```

```

> head(GSPC)
      GSPC.Open GSPC.High GSPC.Low GSPC.Close GSPC.Volume GSPC.Adjusted
2000-01-03  1469.25  1478.00  1438.36  1455.22  931800000  1455.22
2000-01-04  1455.22  1455.22  1397.43  1399.42  1009000000  1399.42
2000-01-05  1399.42  1413.27  1377.68  1402.11  1085500000  1402.11
2000-01-06  1402.11  1411.90  1392.10  1403.45  1092300000  1403.45
2000-01-07  1403.45  1441.47  1400.73  1441.47  1225200000  1441.47
2000-01-10  1441.47  1464.36  1441.47  1457.60  1064800000  1457.60
> tail(GSPC)
      GSPC.Open GSPC.High GSPC.Low GSPC.Close GSPC.Volume GSPC.Adjusted
2020-08-04  3289.92  3306.84  3286.37  3306.51  4621670000  3306.51
2020-08-05  3317.37  3330.77  3317.37  3327.77  4732220000  3327.77
2020-08-06  3323.17  3351.03  3318.14  3349.16  4267490000  3349.16
2020-08-07  3340.05  3352.54  3328.72  3351.28  4104860000  3351.28
2020-08-10  3356.04  3363.29  3335.44  3360.47  4318570000  3360.47
2020-08-11  3370.34  3381.01  3326.44  3333.69  5087650000  3333.69
> da=GSPC
> chartSeries(GSPC,theme="white")
> price=da[,6]
> plot(price,type='l')
> logprice=log(price)
> plot(logprice,type='l')
> logreturn=diff(log(price))
> simplereturn <-exp(logreturn)-1
> par(mfrow=c(1,1))
> plot(logreturn,type='l')
> plot(simplereturn)
> newlogreturn <- logreturn[2:nrow(logreturn),]
> newsimplereturn <- simplereturn[2:nrow(logreturn),]
> #2 Histogram and sample statistics
> hist(logreturn, breaks=100, col="slateblue")
> chart.Histogram(logreturn,methods = c("add.normal"))
> table.Stats(logreturn)
      GSPC.Adjusted
Observations      5184.0000
NAs                1.0000
Minimum            -0.1277
Quartile 1         -0.0048
Median              0.0006
Arithmetic Mean    0.0002
Geometric Mean     0.0001
Quartile 3         0.0057
Maximum            0.1096
SE Mean            0.0002
LCL Mean (0.95)   -0.0002
UCL Mean (0.95)   0.0005

```

```
Variance          0.0002
Stdev             0.0126
Skewness          -0.3865
Kurtosis          11.0457
> #3 QQ-plots and tests for normality
> #
> # use qqnorm function
> par(mfrow=c(1,1))
> qqnorm(newlogreturn)
> qqline(newlogreturn, col = 2)
> jarque.bera.test(newlogreturn)

          Jarque Bera Test

data:  newlogreturn
X-squared = 26483, df = 2, p-value < 2.2e-16
```

FE code (Cont.)

```
#4 Test mean = 0
t.test(newlogreturn)

#5 Test Skewness = 0
T=length(newlogreturn)
s3=skewness(newlogreturn)
tst = s3/sqrt(6/T)
tst
pv = 2*pnorm(tst)
pv

#6 Test excess kurtosis =0
k4 = kurtosis(newlogreturn)
tst = k4/sqrt(24/T)
tst
pv = 2*(1-pnorm(tst))
pv
```

FE Print out (Cont.)

```
> t.test(newlogreturn)
```

```
One Sample t-test
```

```
data: newlogreturn
```

```
t = 0.96103, df = 5035, p-value = 0.3366
```

```
alternative hypothesis: true mean is not equal to 0
```

```
95 percent confidence interval:
```

```
-0.0001674840 0.0004895925
```

```
sample estimates:
```

```
mean of x
```

```
0.0001610542
```

```
> #5 Test Skewness = 0
```

```
> T=length(newlogreturn)
```

```
> s3=skewness(newlogreturn)
```

```
> s3
```

```
[1] -0.230244
```

```
> tst = s3/sqrt(6/T)
```

```
> tst
```

```
[1] -6.670456
```

```
> pv = 2*pnorm(tst)
```

```
> pv
```

```
[1] 2.550093e-11
```

```
> #6 Test excess kurtosis =0
```

```
> k4 = kurtosis(newlogreturn)
```

```
> k4
```

```
[1] 8.652073
```

```
> tst = k4/sqrt(24/T)
```

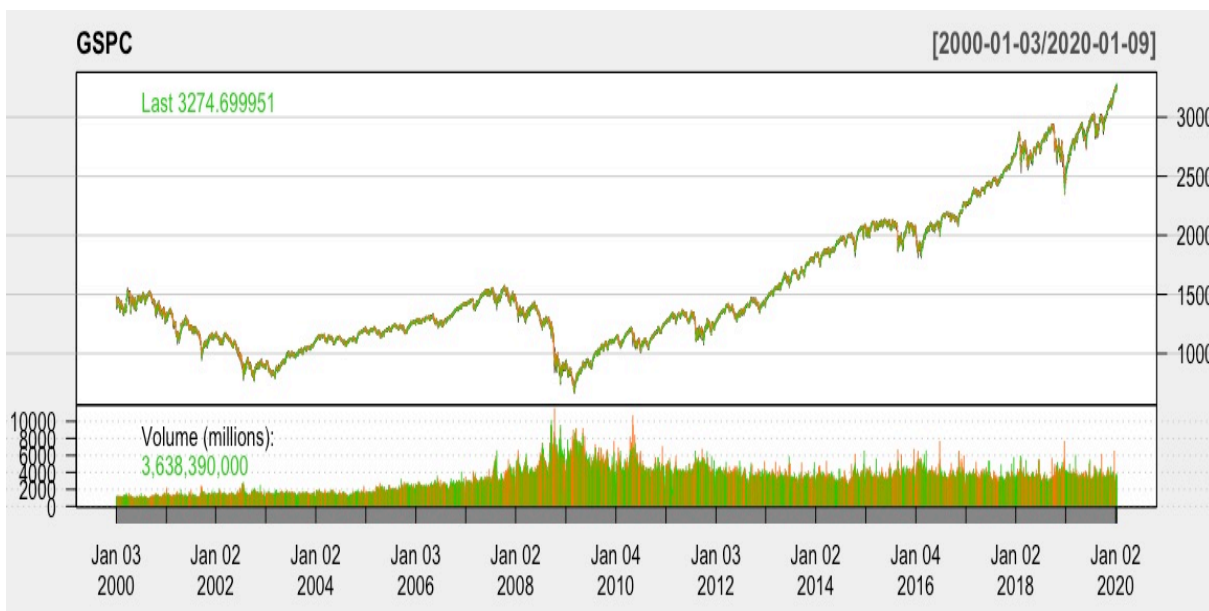
```
> tst
```

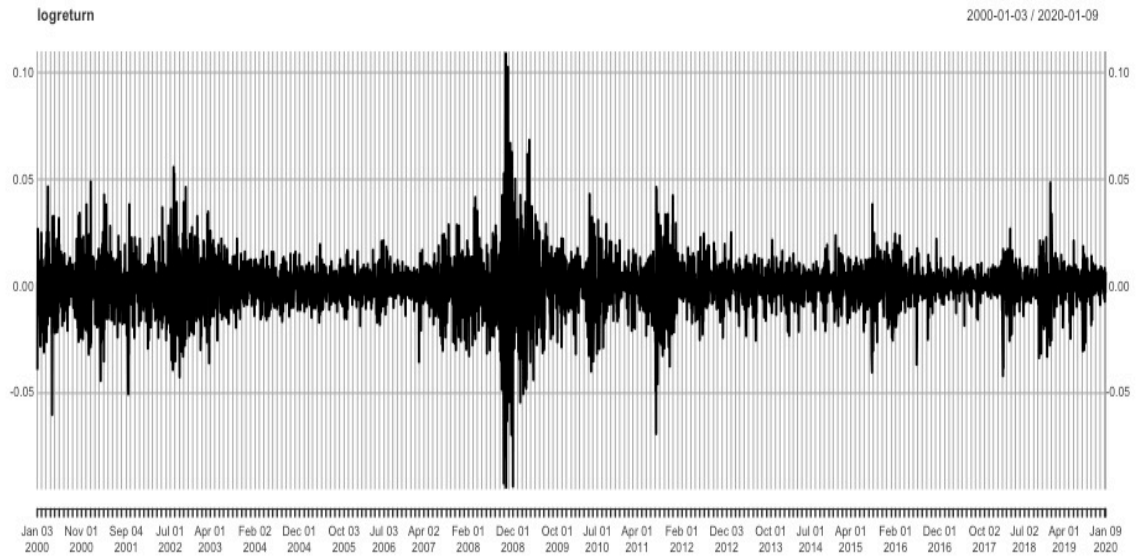
```
[1] 125.3307
```

```
> pv = 2*(1-pnorm(tst))
```

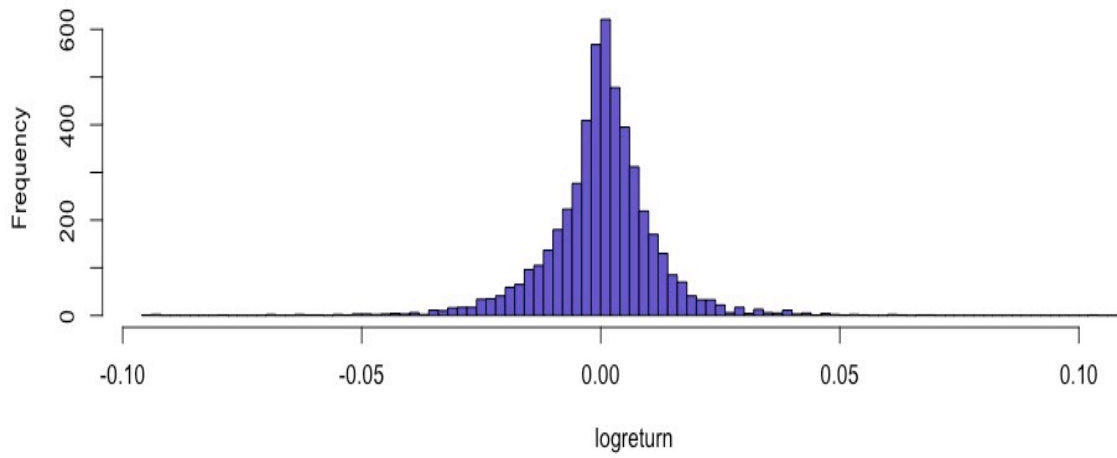
```
> pv
```

```
[1] 0
```

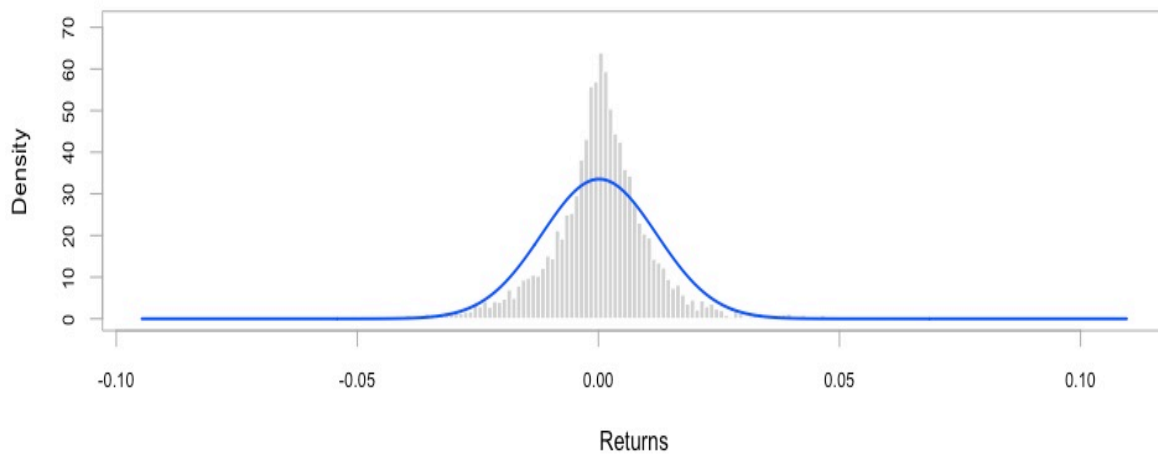


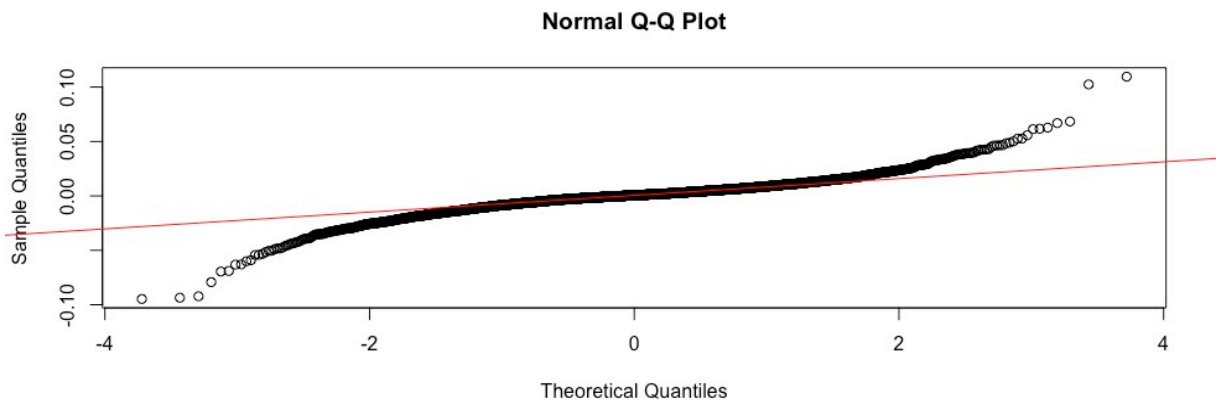
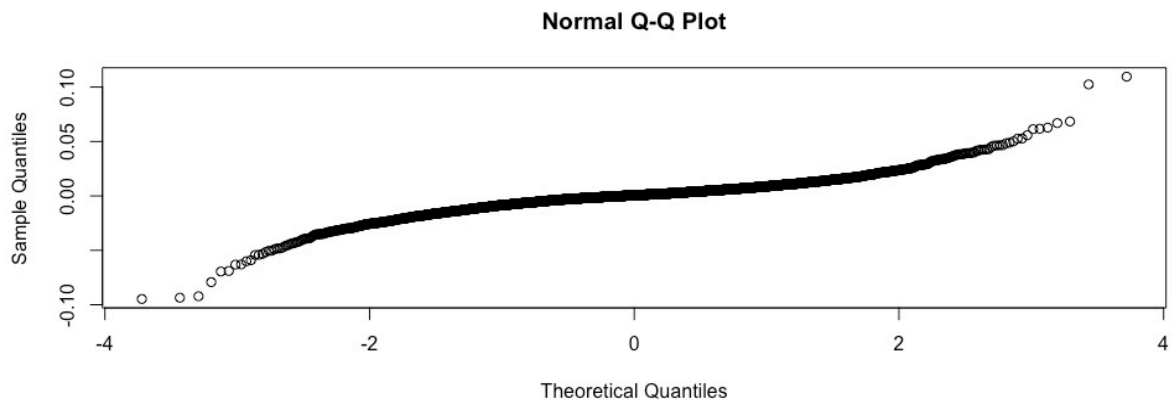


Histogram of logreturn



GSPC.Adjusted







2. Linear Time Series (TS) Models

2.1 Basic Concepts

Financial TS: collection of a financial measurement over time.

Example: log return of apple r_t

Data: $\{r_1, r_2, \dots, r_T\}$

Purpose What is the information contained in series of r_t

Definition: Stationarity

-Strict: Distributions are time-invariant

-Weak: First 2 moments are time-invariant

What does weak stationarity mean in practice?

Past: time plot of r_t varies around a fixed level within a finite range!

Future: the first 2 moments of future r_t are the same as those of the data so that meaningful inferences can be made.

Mean (or expectation) of returns

$$\mu = E(r_t)$$

Variance (variability) of returns

$$\text{Var}(r_t) = E[(r_t - \mu)^2]$$

Sample mean and Sample Variance are used to estimate the mean and variance of returns.

$$\bar{r} = \frac{1}{T} \sum_{t=1}^T r_t$$

$$\text{Var}(r_t) = \frac{1}{T-1} \sum_{t=1}^T (r_t - \bar{r})^2$$

testing the mean of r_t is different from zero or not

$$H_0 : \mu = 0$$

$$H_a : \mu \neq 0$$

$$t_{cal} =$$

Decision rule: Reject H_0 if $|t| > Z_{\frac{\alpha}{2}}$ or p-value is less than α

Lag-kk autocovariance:

$$\gamma_k = \text{Cov}(r_t, r_{t-k}) = E[(r_t - \mu)(r_{t-k} - \mu)]$$

Serial (or-auto) correlations:

$$\rho_l = \frac{\text{cov}(r_t, r_{t-l})}{\text{var}(r_t)}$$

Remark The existence of serial correlation in r_t implies that.....

Sample Autocorrelation function (AFC) can be computed by:

$$\hat{\rho}_l = \frac{\sum_{t=1}^{T-l} (r_t - \bar{r})(r_{t+l} - \bar{r})}{\sum_{t=1}^T (r_t - \bar{r})^2}$$

Test Zero Serial Correlations (Market Efficiency)

1. Individual Test

$$H_0: \rho_1 = 0$$

$$H_a: \rho_1 \neq 0$$

$$t_{cal} =$$

Decision rule: Reject the null hypothesis when $|t| > Z_{\frac{\alpha}{2}}$ or the p-value has the value less than α

2. Joint Test (Ljung-Box Statistics):

$$H_0: \rho_1 = \dots = \rho_m = 0$$

$$H_a: \rho_i \neq 0$$

$$Q(m) = T * (T + 2) \sum_{l=1}^m \frac{\rho_l^2}{T - l}$$

Decision rule: Reject the null hypothesis when $Q(m) > \chi_m^2(\alpha)$ or the p-value has the value less than α

FE toolbox 2

```
#EE435 Wasin Siwasarit
setwd("/Users/wasinsiwasarit/Desktop/EE435")
library(fBasics)
cat(rep("\n",50)) #clear R Console
da <- read.table("CRSP.txt")
log_return =da[,1]
par(mfcol=c(1,1))
length(log_return)
tdx = c(1:456)/12+1961
plot(tdx, log_return, xlab='year', ylab='log_return', type='l')
basicStats(log_return)
normalTest(log_return, method="jb")
t.test(log_return)
tt1 = skewness(log_return)/sqrt(6/546)
tt1
pv = 2*pnorm(tt1)
pv
tt2 = kurtosis(log_return)/sqrt(24/546)
tt2
```

```
pv = 2*(1-pnorm(tt2))
pv
m1=acf(log_return)
names(m1)
m1$acf
m2=pacf(log_return)
names(m2)
m2$acf
Box.test(log_return,lag=12,type='Ljung')
```

FE Analysis 2

```
> da <- read.table("CRSP.txt")
> log_return =da[,1]
> par(mfcol=c(1,1))
> length(log_return)
[1] 456
> tdx = c(1:456)/12+1961
> plot(tdx,log_return,xlab='year',ylab='log_return',type='l')
> basicStats(log_return)
      log_return
nobs      456.000000
NAs        0.000000
Minimum   -31.588000
Maximum    26.175000
1. Quartile -1.860000
3. Quartile  4.268250
Mean        1.059511
Median      1.494500
Sum         483.137000
SE Mean     0.262245
LCL Mean    0.544149
UCL Mean    1.574873
Variance    31.360270
Stdev       5.600024
Skewness    -0.673271
Kurtosis    4.122884
> normalTest(log_return,method="jb")
```

```
Title:
Jarque - Bera Normalality Test
```

```
Test Results:
STATISTIC:
X-squared: 362.5726
```

```

P VALUE:
  Asymptotic p Value: < 2.2e-16

Description:
  Sat Aug 19 22:51:45 2017 by user:

> t.test(log_return)

      One Sample t-test

data:  log_return
t = 4.0402, df = 455, p-value = 6.27e-05
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.544149 1.574873
sample estimates:
mean of x
 1.059511

> tt1 = skewness(log_return)/sqrt(6/546)
> tt1
[1] -6.443776
> pv = 2*pnorm(tt1)
> pv
[1] 1.165367e-10
> tt2 = kurtosis(log_return)/sqrt(24/546)
> tt2
[1] 19.81441
> pv = 2*(1-pnorm(tt2))
> pv
[1] 0
> m1=acf(log_return)
> names(m1)
[1] "acf"      "type"     "n.used"  "lag"     "series"  "snames"
> m1$acf
, , 1

      [,1]
[1,] 1.000000000
[2,] 0.226356832
[3,] -0.009975215
[4,] -0.038128697
[5,] -0.015760585
...
[26,] -0.012220663
[27,] 0.064944965

```

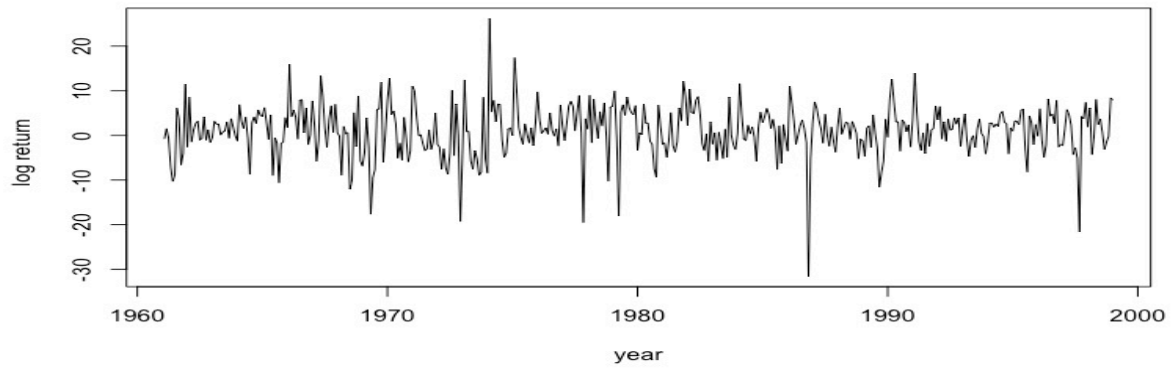
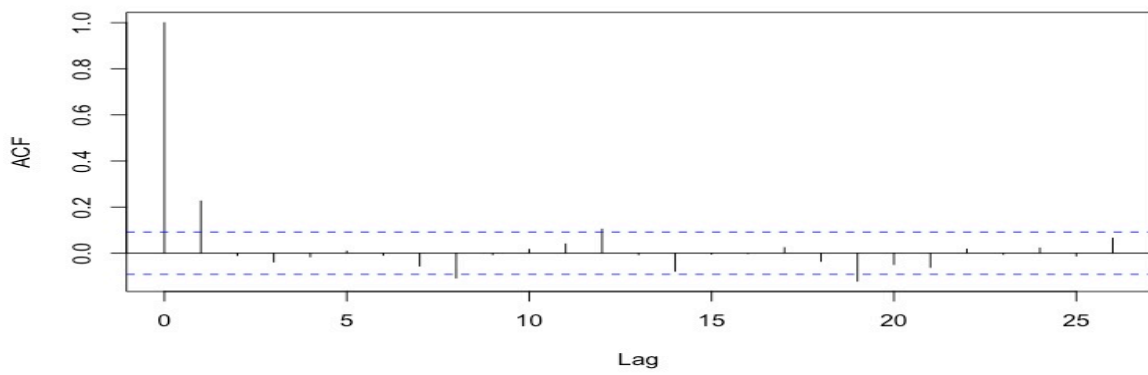
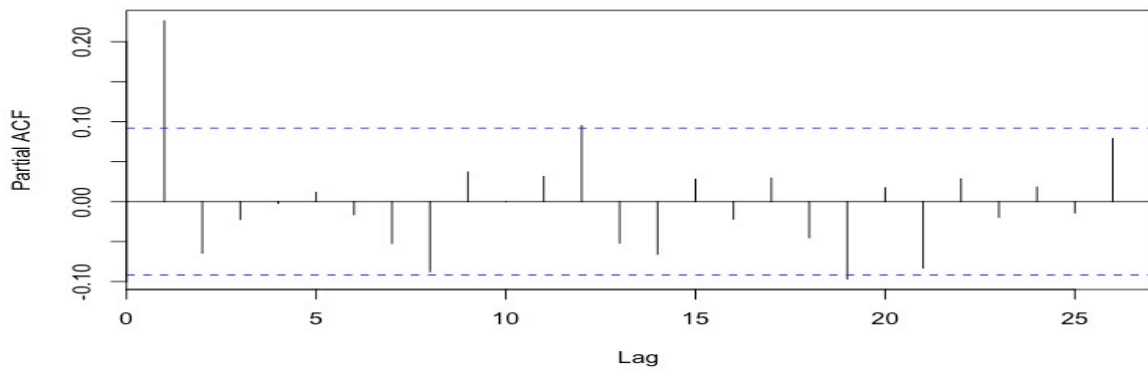
```
> m2=pacf(log_return)
> names(m2)
[1] "acf"      "type"     "n.used"  "lag"     "series"  "snames"
> m2$acf
, , 1

          [,1]
[1,]  0.2263568318
[2,] -0.0645183854
[3,] -0.0223545572
[4,] -0.0022849724
[5,]  0.0116224218
...
[25,] -0.0142841594
[26,]  0.0788941527

> Box.test(log_return,lag=12,type='Ljung')

      Box-Ljung test

data:  log_return
X-squared = 37.302, df = 12, p-value = 0.0001995
```

**Series log_return****Series log_return**



2.2 Back-Shift (lag) operator

Definition $Br_t = r_{t-1}$ or $Lr_t = r_{t-1}$

$$B^2r_t = B(Br_t) = Br_{t-1} = r_{t-1}$$

B or L means Time Shift

For example Br_t is the value of the series at time $t-1$

For example

The table of log return:

Date	r_t
1	0.025
2	0.013
3	-0.003
4	0.035

What is the following value of

$$r_2 =$$

$$Br_3 =$$

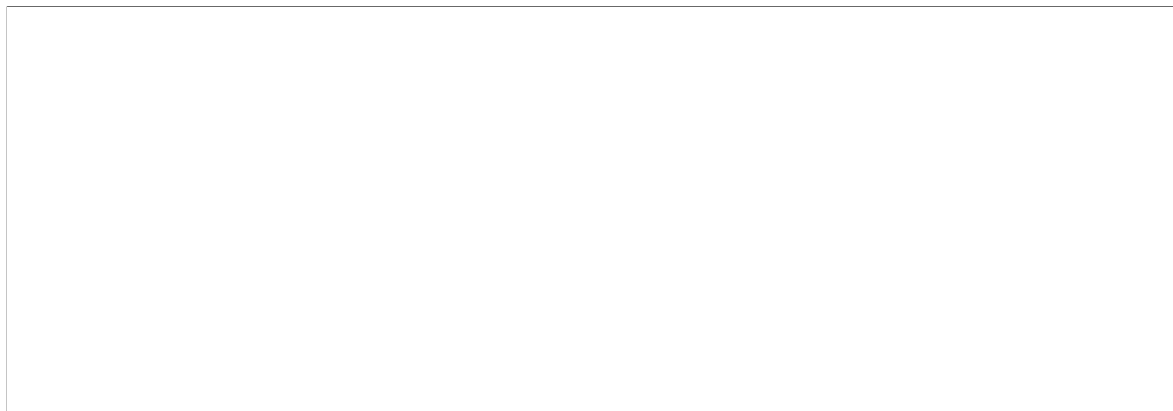
$$B^2r_5 =$$

A proper perspective: at a time point t

Available data: $\{r_1, r_2, \dots, r_{t-1}\} \equiv F_{t-1}$

The return is decomposed into two parts:

$$r_t = \text{predictable part} + \text{not predictable part}$$



Traditional TS modeling is concerned with μ_t :

Model for μ_t : mean equation

Volatility modeling concerns σ_t

Model for σ_t^2 : volatility equation

2.3 Linear Time Series

r_t is linear if

- . the predictable part is a linear function of F_{t-1}
- . $\{a_t\}$ are independent and have the same distribution (iid)

Mathematically, it means r_t can be written as

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

where μ is constant and $\psi_0 = 1$ and a_t is an iid sequence with mean 0 and well-defined distribution.

In the economic literature a_t is the shock or innovation at time t and ψ_i are the impulse responses of r_t .

White noise: iid sequence (with finite variance), which is the building block of linear TS models. White noise is not predictable, but has zero mean and finite variance.

In EE435 we will study the (Univariate linear time series models) as follow:

1. autoregressive (AR) models
2. moving-average (MA) models
3. mixed ARMA models
4. seasonal models

Important properties of a model

- Stationarity condition
- Basic properties: mean, variance, serial dependence
- Empirical model building: specification, estimation, & checking
- Forecasting

2.4 Autoregressive(AR) model

If the series r_t and r_{t-1} are correlated, we might be able to use the series r_{t-1} in forecasting r_t . Thus the linear model can be:

$$r_t = \phi_0 + \phi_1 r_{t-1} + a_t$$

where a_t are white noise series with mean equal to 0 and variance equals to σ_a^2

The above model is known as AR(1) since the variation of r_t can be explained by the variation of r_{t-1} . From this model we can calculate the conditional mean and conditional variance as the follow:



In general, we can write down the model of AR(p) as

$$r_t = \phi_0 + \phi_1 r_{t-1} + \dots + \phi_p r_{t-p} + a_t$$


2.4.1 Properties of AR models

AR (1) model

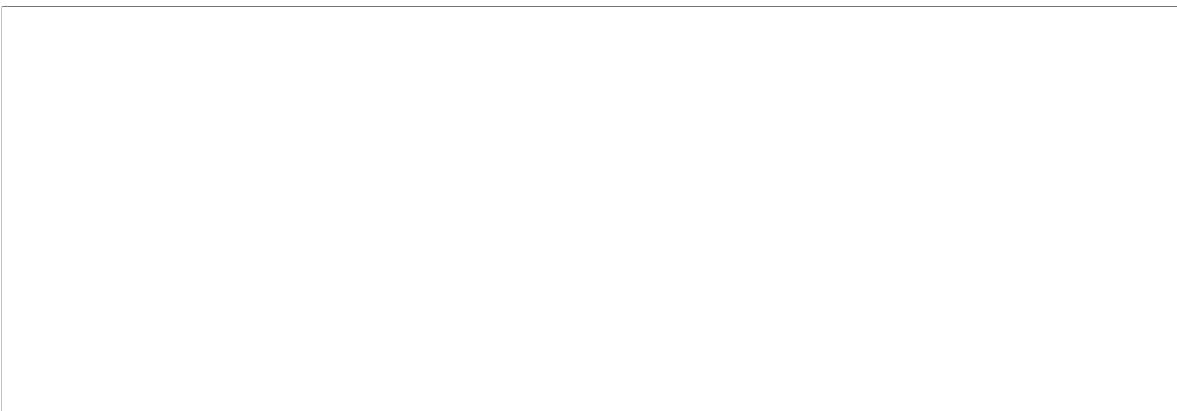
the necessary condition for AR model is that series r_t have to be weak stationarity.



Unconditional mean



Unconditional variance



Unconditional autocorrelations

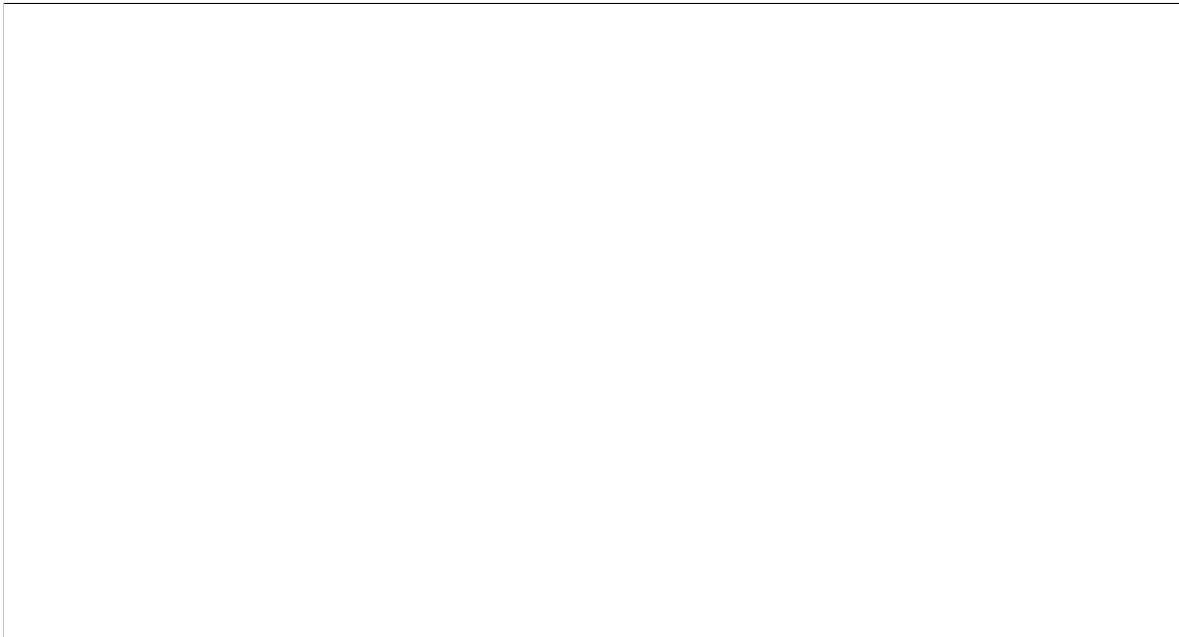
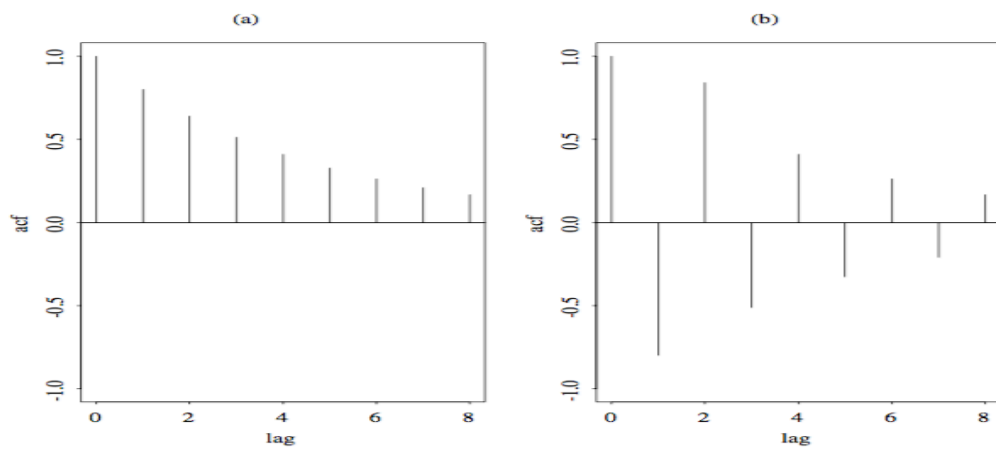


Figure The autocorrelation function of an AR(1) model: (a) for $\phi_1 = 0.8$, and (b) for $\phi_1 = -0.8$.

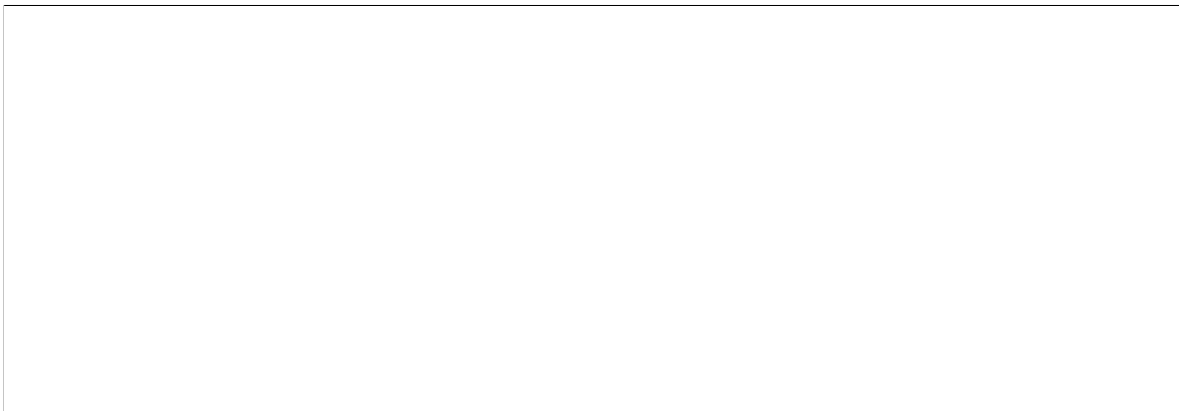


AR (2) model

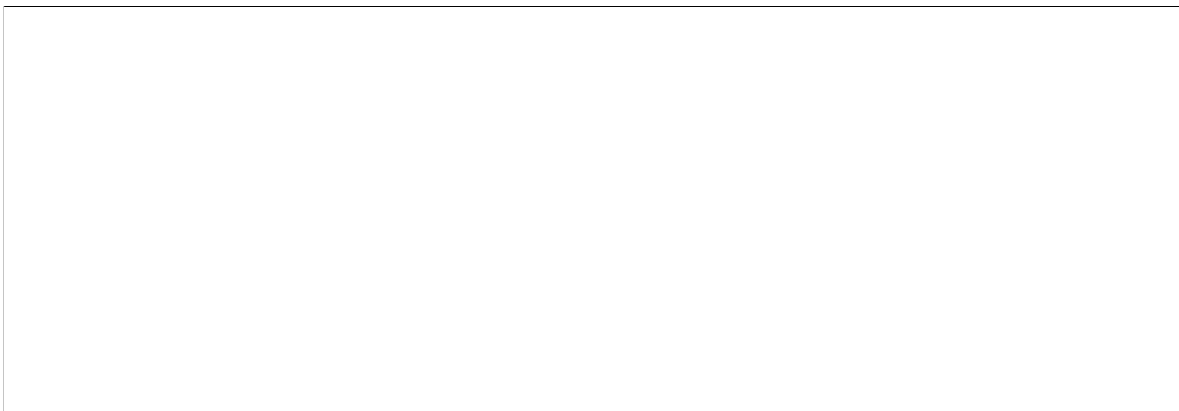
$$r_t = \phi_0 + \phi_1 r_{t-1} + \phi_2 r_{t-2} + a_t$$



Unconditional mean



Unconditional variance



Unconditional autocorrelations

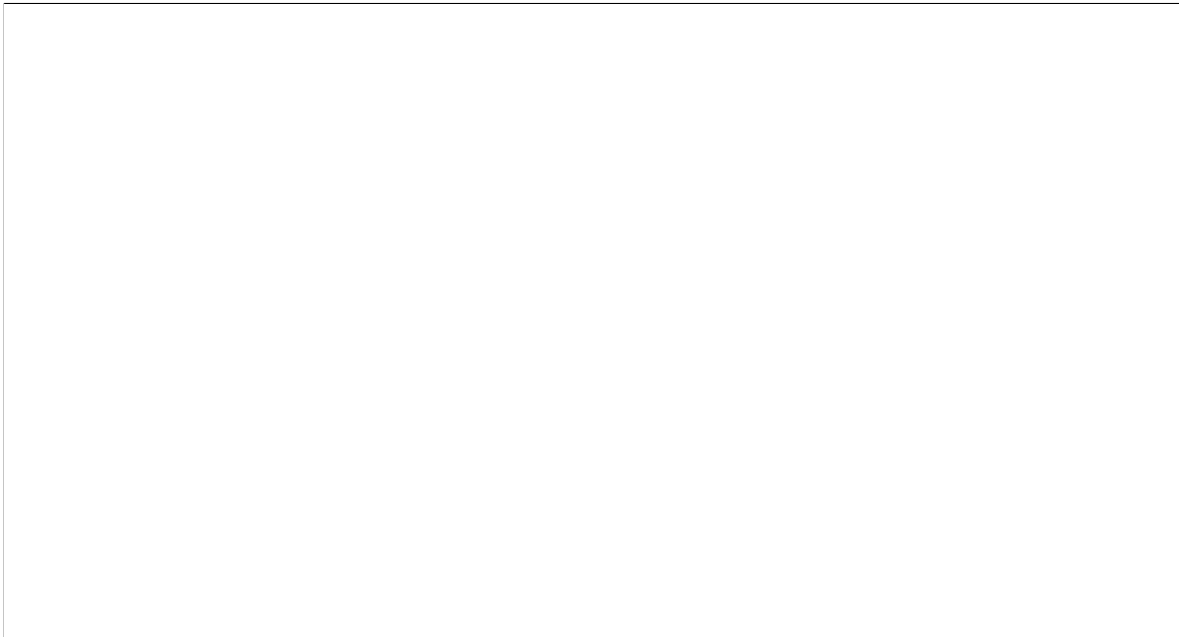
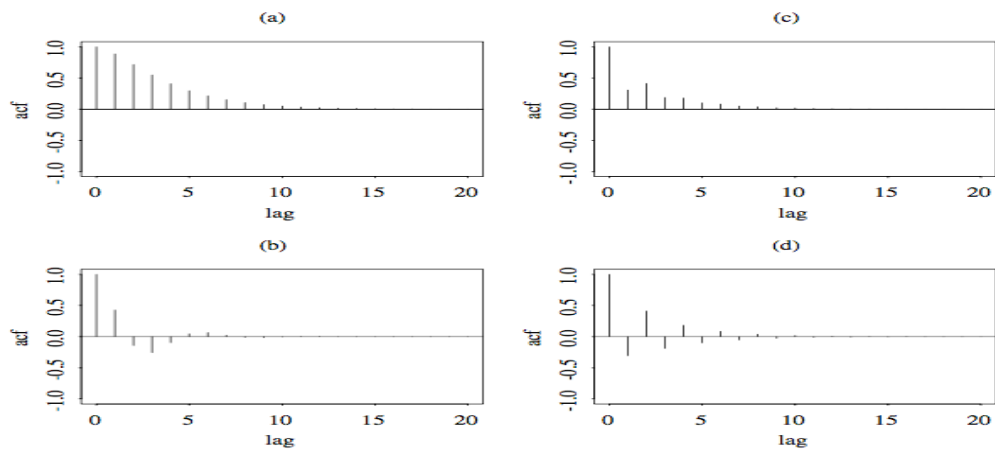


Figure The autocorrelation function of an AR(2) model: (a) $\phi_1 = 1.2$ and $\phi_2 = -0.35$, (b) $\phi_1 = 0.6$ and $\phi_2 = -0.4$, (c) $\phi_1 = 0.2$ and $\phi_2 = 0.35$, (d) $\phi_1 = -0.2$ and $\phi_2 = 0.35$.



2.4.2 Stationarity of AR(p) Model

In case of AR(p)

$$r_t = \phi_0 + \phi_1 r_{t-1} + \dots + \phi_p r_{t-p} + a_t$$

We can write down the Unconditional Mean as :

$$E(r_t) = \frac{\phi_0}{1 - \phi_1 - \dots - \phi_p},$$

which the Polynomial equation can be expressed as

$$x^p - \phi_1 x^{p-1} - \phi_2 x^{p-2} - \dots - \phi_p = 0$$

The above equation is also known as Equation in which if Characteristic roots has the value less than 1 in modulus, we can say that the model is stationary.

Moreover, AR(p) , the ACF can be written as difference equation

$$(1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p) \rho_l = 0$$

where $l > 0$

The graph of ACF of AR(p) has the pattern as the graph of sine and cosine.

2.4.3 Identifying AR Models

Partial Autocorrelation Function (PACF)

PACF is considered to be a tool to determine the order of AR(p). We can calculate the PACF from the following equations:

$$r_t = \phi_{0,1} + \phi_{1,1}r_{t-1} + e_{1t}$$

$$r_t = \phi_{0,2} + \phi_{1,2}r_{t-1} + \phi_{2,2}r_{t-2} + e_{2t}$$

$$r_t = \phi_{0,3} + \phi_{1,3}r_{t-1} + \phi_{2,3}r_{t-2} + \phi_{3,3}r_{t-3} + e_{3t}$$

In case of lag-2 PACF $\phi_{2,2}$ shows the marginal effect of r_{t-2} on r_t

In case of lag-3 PACF $\phi_{3,3}$ shows the marginal effect of r_{t-3} on r_t

Therefore, If the AR(p) is the optimal model, we then have the the lag-p PACF have to significantly different from 0, but $\phi_{j,j}$ have to be insignificant when $j > p$

Information Criteria

There are two methods to select the optimal lag AR(p)

1. Akaike Information Criterion

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

For AR(l), $\tilde{\sigma}_l^2$ is the MLE of residual variance

We select the AR(l) model that provides the minimum AIC for all $l \in [0, \dots, P]$

2. BIC Criterion

$$BIC(l) = \ln(\tilde{\sigma}_l^2) + \frac{l * \ln(T)}{T}$$

For AR(l), $\tilde{\sigma}_l^2$ is the MLE of residual variance

We select the AR(l) model that provides the minimum BIC for $l \in [0, \dots, P]$

Example: GDP Growth

```
#EE 435 Wasin Siwasarit
setwd("/Users/wasinsiwasarit/Desktop/EE435")
cat(rep("\n",50)) #clear R Console
library(fBasics)
library(quantmod)
library(sn)
library(PerformanceAnalytics)
library(car)
library(tseries)
library(forecast)
library(Matrix)
da=read.table("dgnp82.txt")
x=da[,1]
par(mfcol=c(2,2))
plot(x,type='l')
plot(x[1:175],x[2:176])
plot(x[1:174],x[3:176])
acf(x,lag=12)
par(mfcol=c(1,1))
pacf(x,lag.max=12)
```

Figure: GDP growth, ACF and PACF

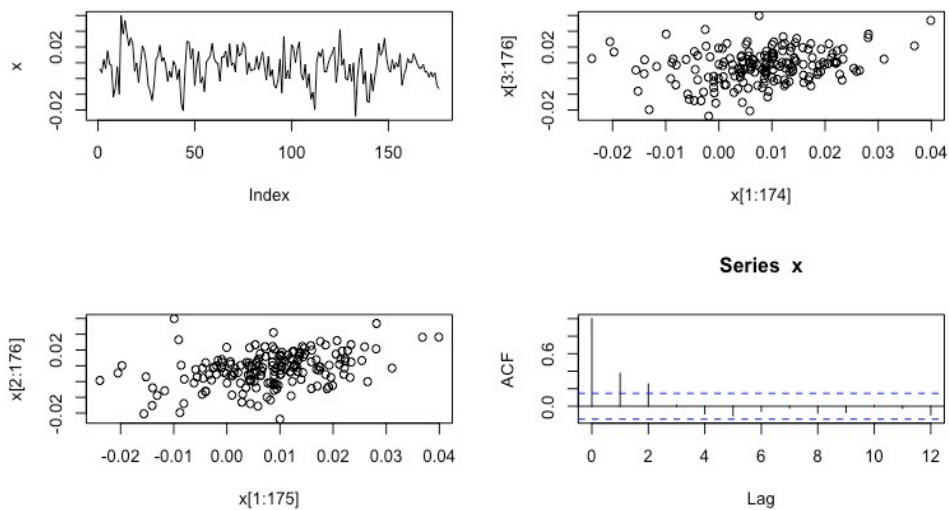
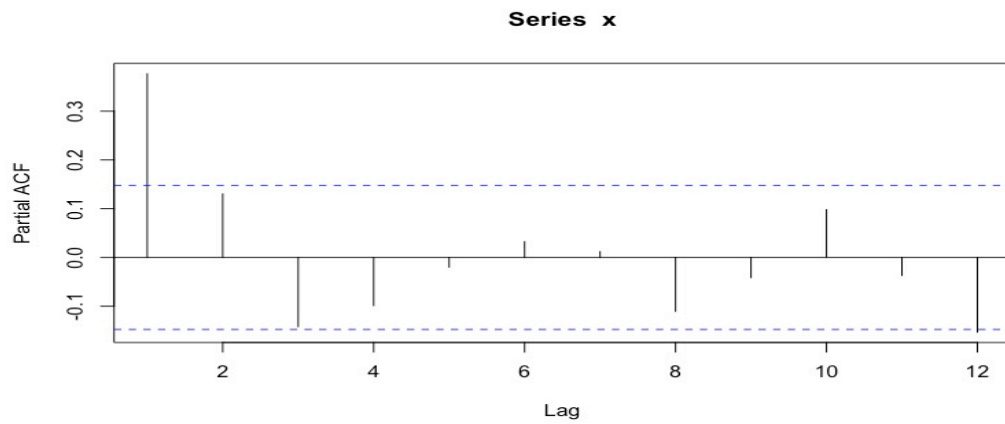


Figure: GDP growth, ACF and PACF (cont.)

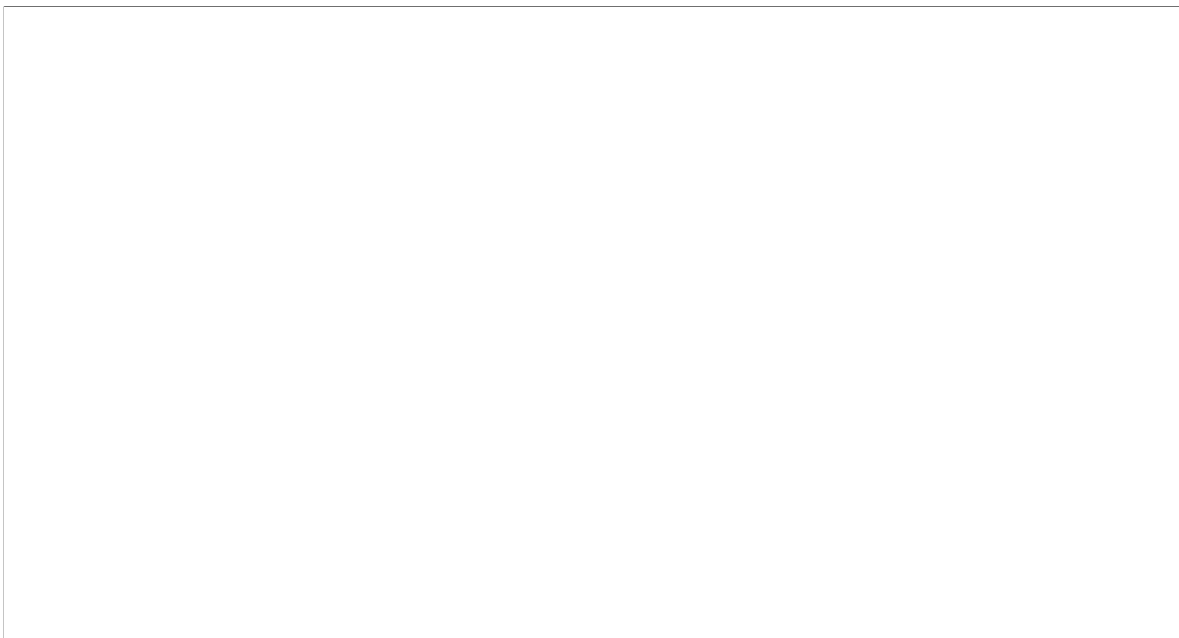


2.5 AR(P) in Lag Operator Notation

AR(1) in Lag Operator Notation

$$(r_t - \mu) = \phi_1(r_{t-1} - \mu) + a_t$$

if $|\phi_1| < 1$ then,



AR(P) model

From the Mean-Adjusted Form:

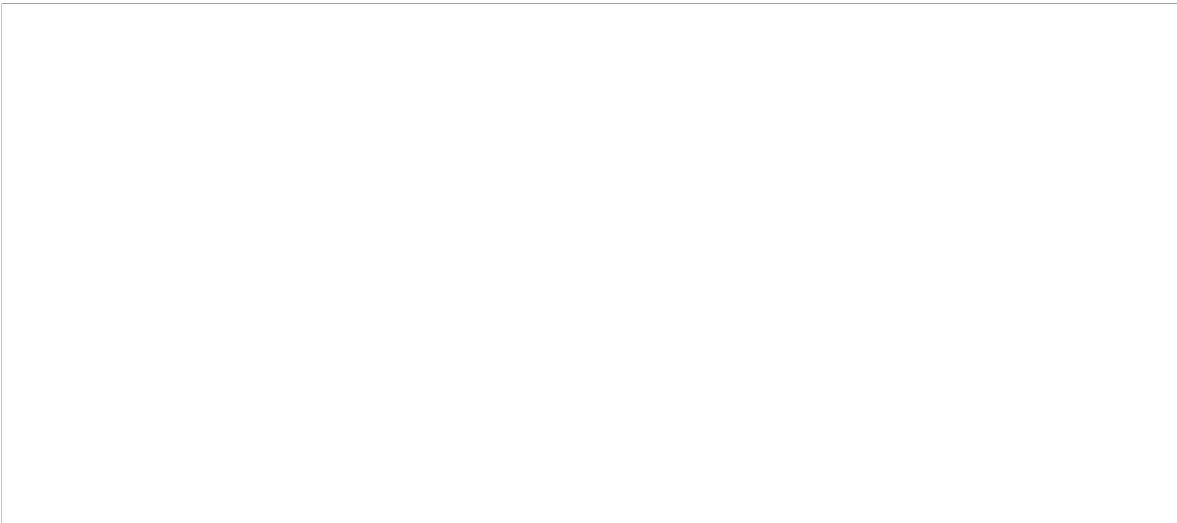
$$(r_t - \mu) = \phi_1(r_{t-1} - \mu) + \dots + \phi_p(r_{t-p} - \mu) + a_t$$

Stability and Stationarity Condition

$$\begin{bmatrix} r_t \\ r_{t-1} \\ \vdots \\ r_{t-p+1} \end{bmatrix} = \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 & \dots & \phi_p \\ 1 & 0 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{t-1} \\ r_{t-2} \\ \vdots \\ r_{t-p} \end{bmatrix} + \begin{bmatrix} a_t \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

we can write it as

$$\xi_t = F\xi_{t-1} + vt$$



Example: AR(2)

$$r_t = \phi_1 r_{t-1} + \phi_2 r_{t-2} + a_t$$



Results: The AR(p) model is weakly stationary and has Wold representation

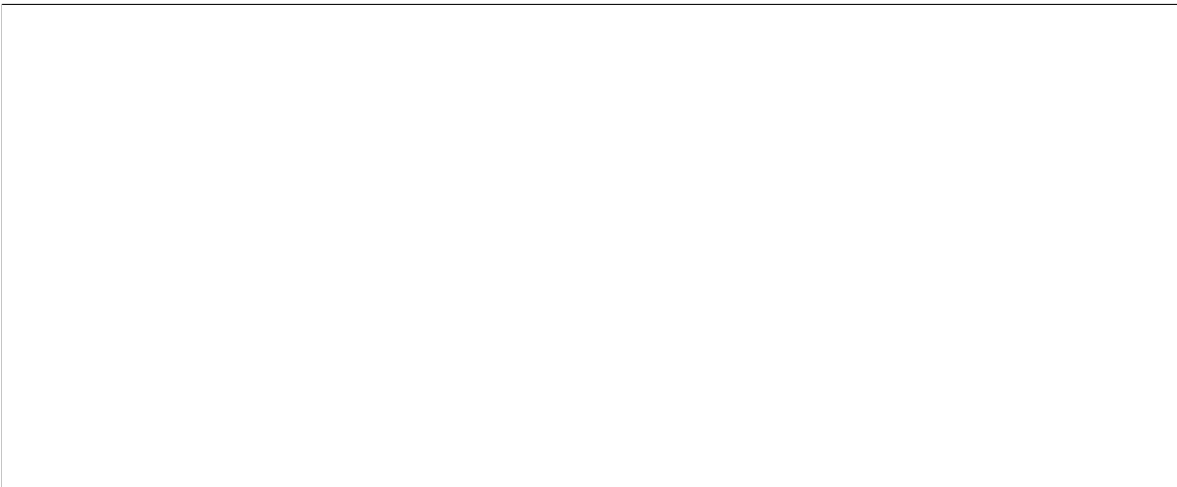
$$r_t = \mu + \sum_{j=0}^{\infty} \psi_j \epsilon_{t-j}$$

with $\psi_j = (1, 1)$ element of \mathbb{F}^j provided all of the eigenvalues of F have modulus less than 1.

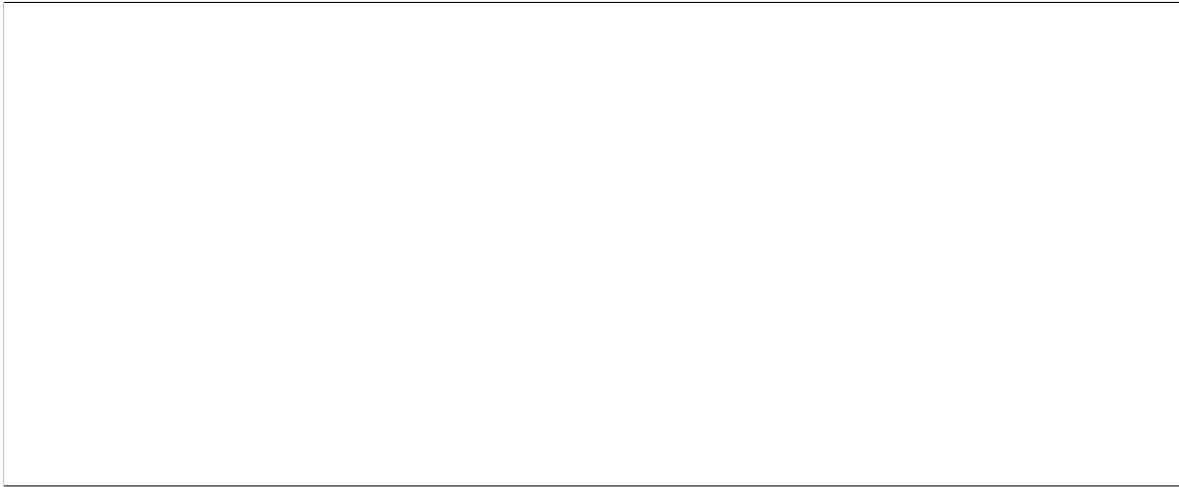
2.6 Finding Eigenvalue

λ is an eigenvalue of F and x is the eigenvector if

$$Fx = \lambda x$$



Example: AR(2)



The eigenvalues of F solve the "reverse" characteristic equation

$$\lambda^2 - \phi_1\lambda - \phi_2 = 0$$

Using the quadratic equation, the roots satisfy

$$\lambda_i = \frac{\phi_1 \pm \sqrt{\phi_1^2 + 4\phi_2}}{2}$$

These roots may be real or complex. Complex roots induce periodic behavior in y_t . If λ_i is complex then

$$\lambda_i = a + bi$$

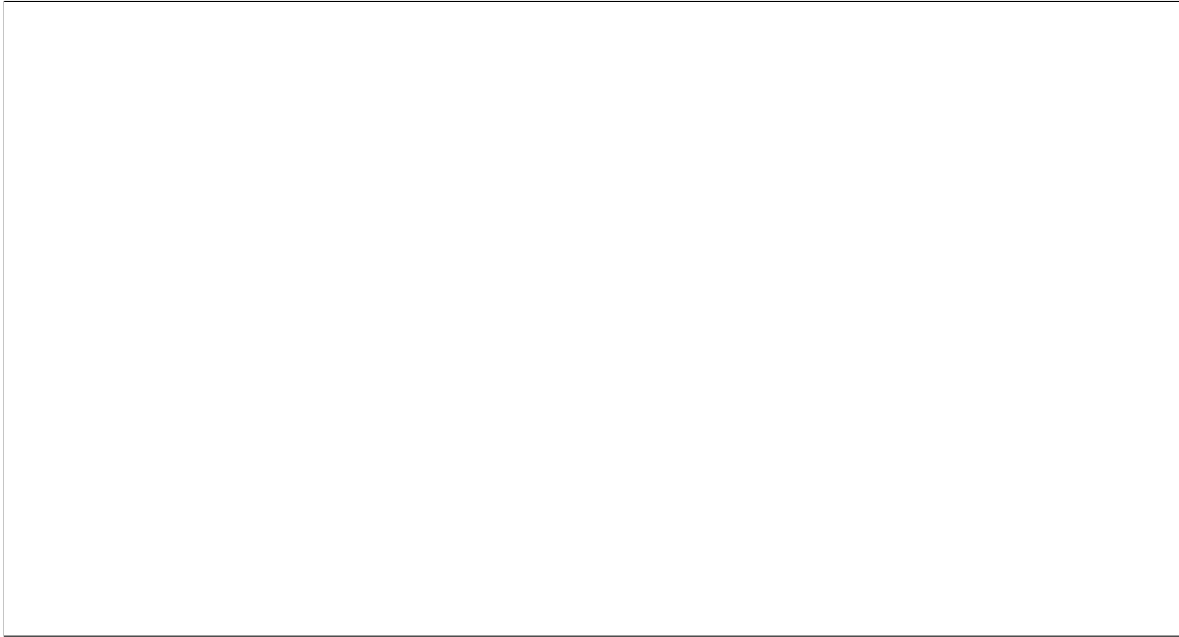
$$a = R\cos(\theta), b = R\sin(\theta)$$

$$R = \sqrt{a^2 + b^2}$$

Remark: R =modulus

Example 1: AR(2)

$$Y_t = 0.6Y_{t-1} + 0.2Y_{t-2} + \epsilon_t$$



Example 1

```
> Re(polyroot(c(-0.2, -0.6, 1)))  
[1] -0.2385165  0.8385165  
> Im(polyroot(c(-0.2, -0.6, 1)))  
[1] -1.29247e-26  1.29247e-26
```

Example 2: AR(2)

$$Y_t = 0.5Y_{t-1} - 0.8Y_{t-2} + \epsilon_t$$



Example 2

```
> Re(polyroot(c(0.8, -0.5, 1)))  
[1] 0.25 0.25  
> Im(polyroot(c(0.8, -0.5, 1)))  
[1] 0.8587782 -0.8587782
```

2.7 Parameter Estimation

For a specified AR(p) model, the conditional least squares 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

$$r_t = \phi_0 + \phi_1 r_{t-1} + \dots + \phi_p r_{t-p} + a_t$$

$$t=p+1, \dots, T$$

which can be estimated by the least squares method. Denote the estimate of ϕ by $\widehat{\phi}$.

The fitted model is

$$r_t = \widehat{\phi}_0 + \widehat{\phi}_1 r_{t-1} + \dots + \widehat{\phi}_p r_{t-p} + a_t$$

where the residual term is

$$\widehat{a}_t = r_t - \widehat{r}_t$$

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

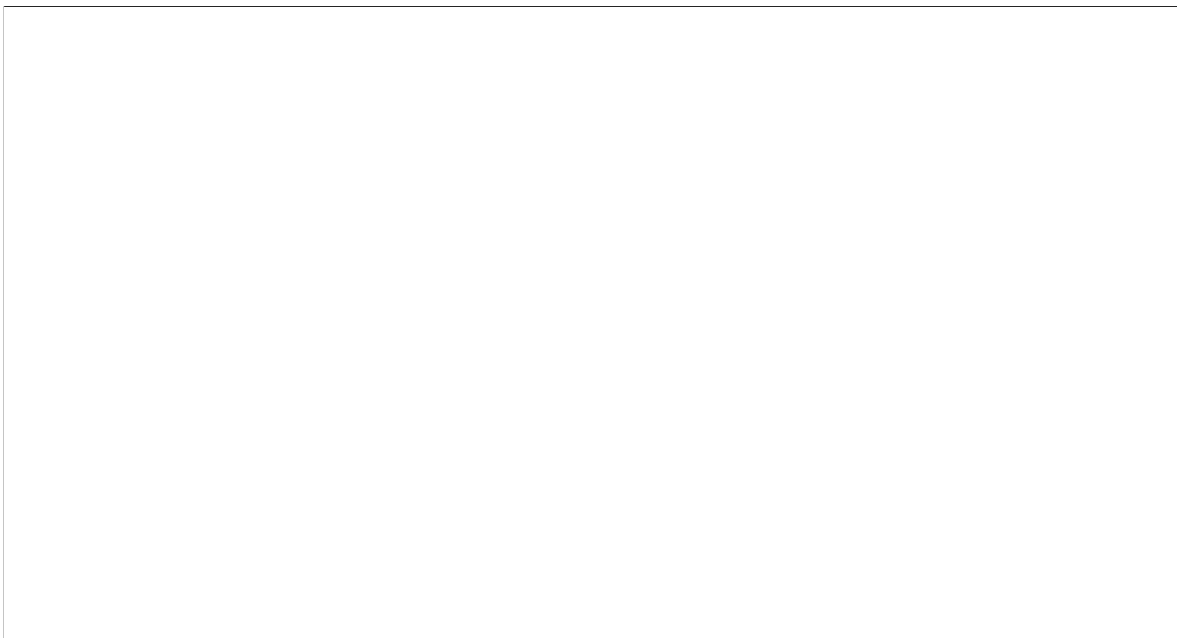
2.8 Model Checking

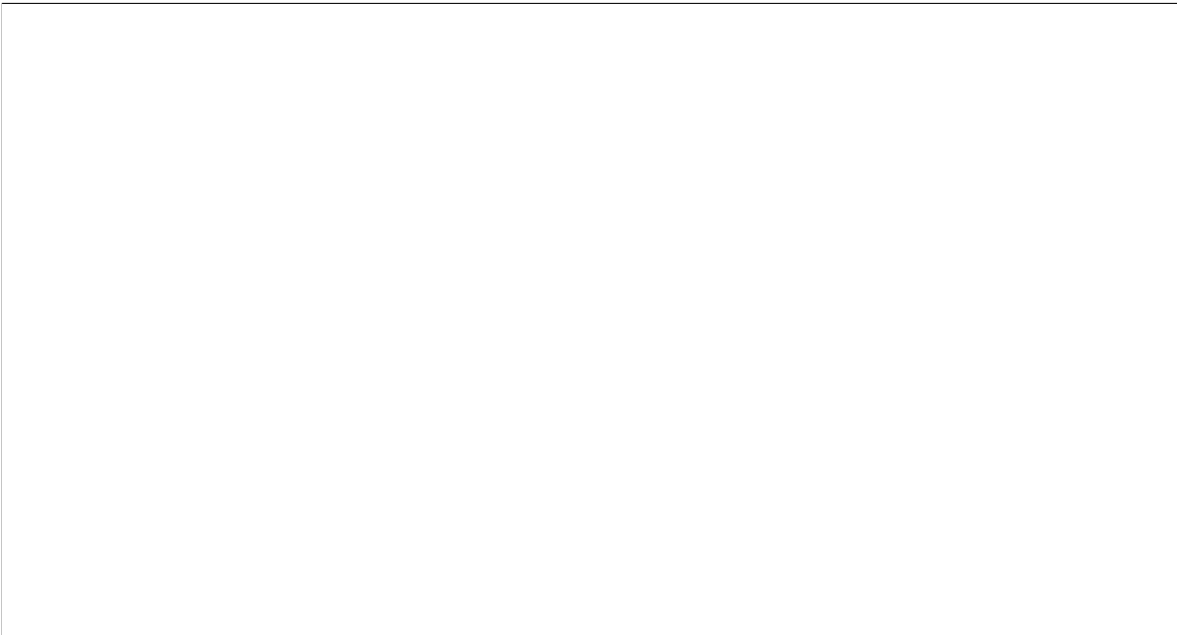
A fitted model must be examined carefully to check for possible model inadequacy. If the model is adequate, then the residual series should behave as a white noise. The ACF and the Ljung–Box statistics of the residuals can be used to check the closeness of a_t to a white noise. For an AR(p) model, the Ljung–Box statistic $Q(m)$ follows asymptotically a chi-squared distribution with $m-p$ degrees of freedom. Here the number of degrees of freedom is modified to signify that p AR coefficients are estimated. If a fitted model is found to be inadequate, it must be refined.

2.9 Forecasting

Forecasting is an important application of time series analysis. For the AR(p) model, suppose that we are at the time index h and are interested in forecasting r_{h+l} , where $l \geq 1$. The time index h is called the forecast origin and the positive integer l is the forecast horizon. Let $\hat{r}_h(l)$ be the forecast of r_{h+l} ,

2.9.1 1-Step Ahead Forecast



2.9.2 2-Step Ahead Forecast**2.9.3 3-Step Ahead Forecast**

Example: Analysis of U.S. GNP growth rate series.

```
#EE435
setwd("/Users/wasinsiwasarit/Desktop/EE435")
cat(rep("\n",50)) #clear R Console
da=read.table("dgnp82.txt")
x=da[,1]
> da=read.table("dgnp82.dat")
> x=da[,1]
> par(mfcol=c(2,2)) % put 4 plots on a page
plot(x,type='l') % first plot
plot(x[1:175],x[2:176]) % 2nd plot
plot(x[1:174],x[3:176]) % 3rd plot
acf(x,lag=12) % 4th plot
pacf(x,lag.max=12) % Compute PACF
Box.test(x,lag=10,type='Ljung') % Compute Q(10) statistics
Box-Ljung test
data: x
X-squared = 43.2345, df = 10, p-value = 4.515e-06
m1=ar(x,method='mle') % Automatic AR fitting using AIC criterion.
m1
Call: ar(x = x, method = "mle")
Coefficients:
1      2      3      % An AR(3) is specified.
0.3480 0.1793 -0.1423
Order selected 3 sigma^2 estimated as 9.427e-05
names(m1)
[1] "order" "ar" "var.pred" "x.mean" "aic"
[6] "n.used" "order.max" "partialacf" "resid" "method"
[11] "series" "frequency" "call" "asy.var.coef"

plot(m1$resid,type='l') % Plot residuals of the fitted model (not shown)

Box.test(m1$resid,lag=10,type='Ljung') % Model checking
Box-Ljung test
data: m1$resid
X-squared = 7.0808, df = 10, p-value = 0.7178

m2=arima(x,order=c(3,0,0)) % Another approach with order given.
m2
Call: arima(x = x, order = c(3, 0, 0))
Coefficients:
      ar1      ar2      ar3 intercept % Fitted model is
0.3480 0.1793 -0.1423  0.0077 % y(t)=0.348y(t-1)+0.179y(t-2)
s.e. 0.0745 0.0778 0.0745 0.0012 % -0.142y(t-3)+a(t),
```

```

% where  $y(t) = x(t) - 0.0077$ 
sigma^2 estimated as 9.427e-05: log likelihood = 565.84, aic = -1121.68
> names(m2)
[1] "coef"      "sigma2"    "var.coef"  "mask"     "loglik"    "aic"
[7] "arma"      "residuals" "call"      "series"   "code"      "n.cond"
[13] "model"
Box.test(m2$residuals, lag=10, type='Ljung')
Box-Ljung test
data:  m2$residuals
X-squared = 7.0169, df = 10, p-value = 0.7239
ts.plot(m2$residuals) % Residual plot
tsdiag(m2) % obtain 3 plots of model checking (not shown in handout).
p1=c(1,-m2$coef[1:3]) % Further analysis of the fitted model.
roots=polyroot(p1)
roots
[1] 1.590253+1.063882e+00i -1.920152-3.530887e-17i 1.590253-1.063882e+00i
Mod(roots)
[1] 1.913308 1.920152 1.913308
predict(m2,8) % Prediction 1-step to 8-step ahead.
$pred
Time Series:
Start = 177
End = 184
Frequency = 1
[1] 0.001236254 0.004555519 0.007454906 0.007958518
[5] 0.008181442 0.007936845 0.007820046 0.007703826
$se
Time Series:
Start = 177
End = 184
Frequency = 1
[1] 0.009709322 0.010280510 0.010686305 0.010688994
[5] 0.010689733 0.010694771 0.010695511 0.010696190

```

Figure: GDP growth and PACF of GDP growth

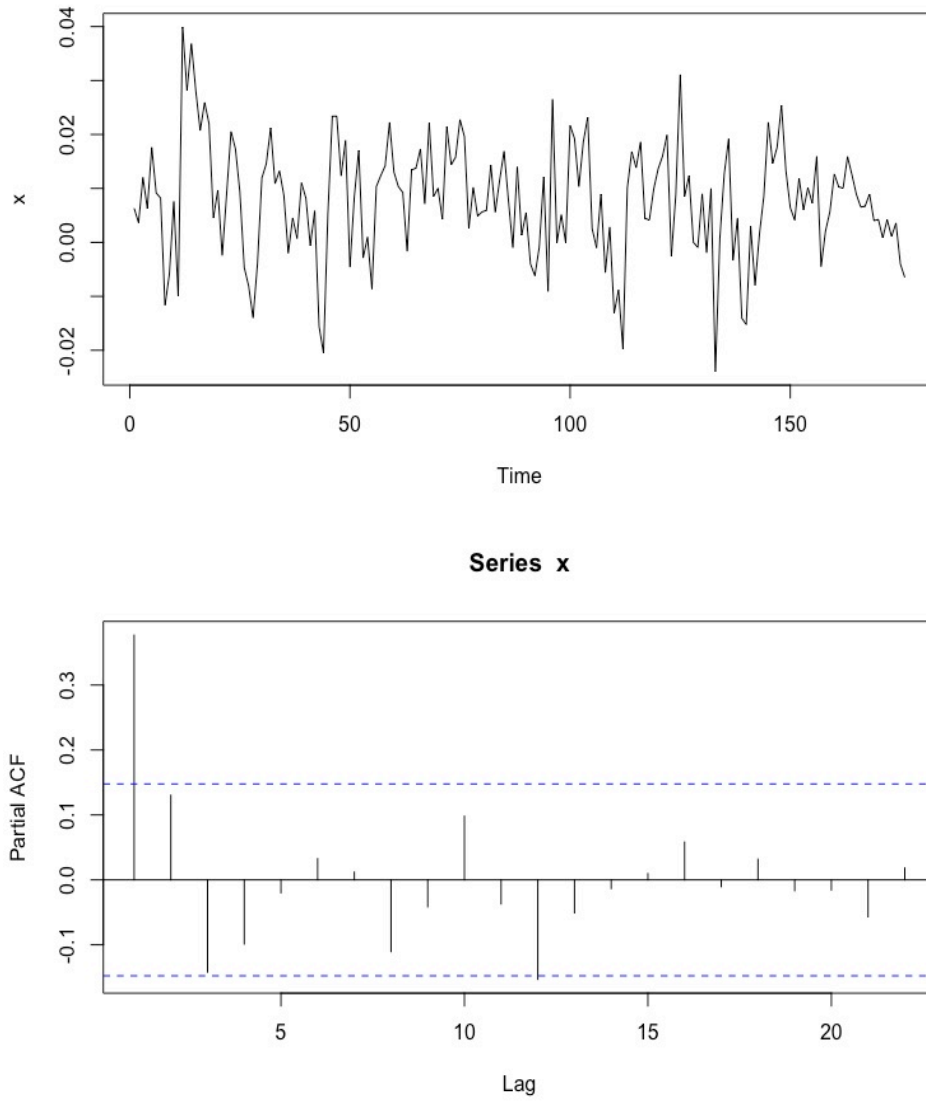
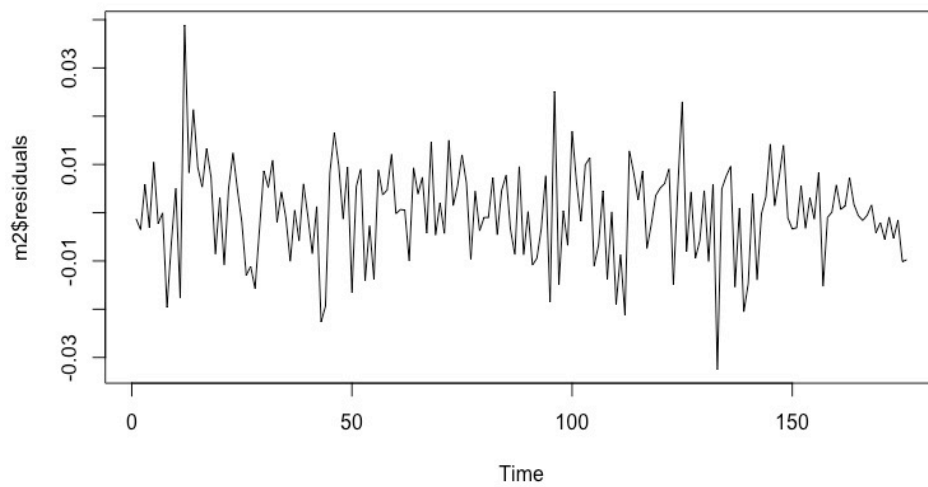
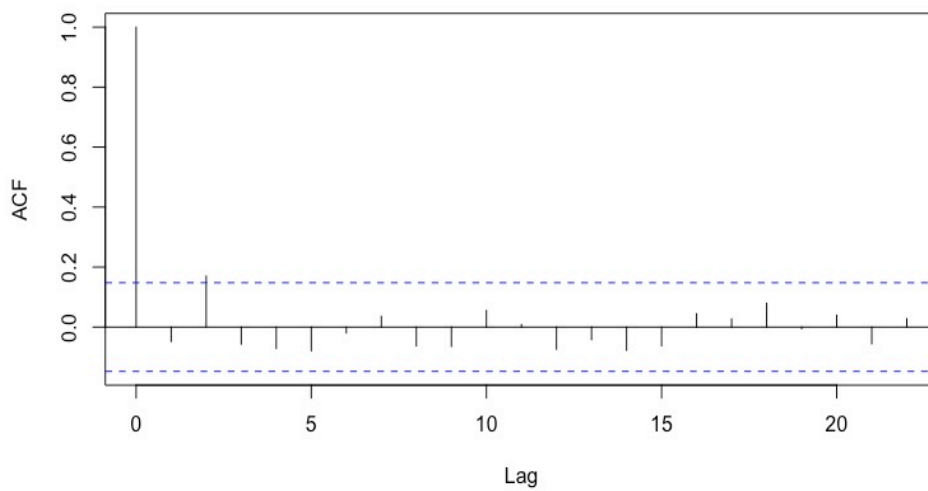


Figure: Residual Term from the estimated model and ACF of Residual Term

**Series m2\$residuals**

2.10 Moving-Average Models (MAs)

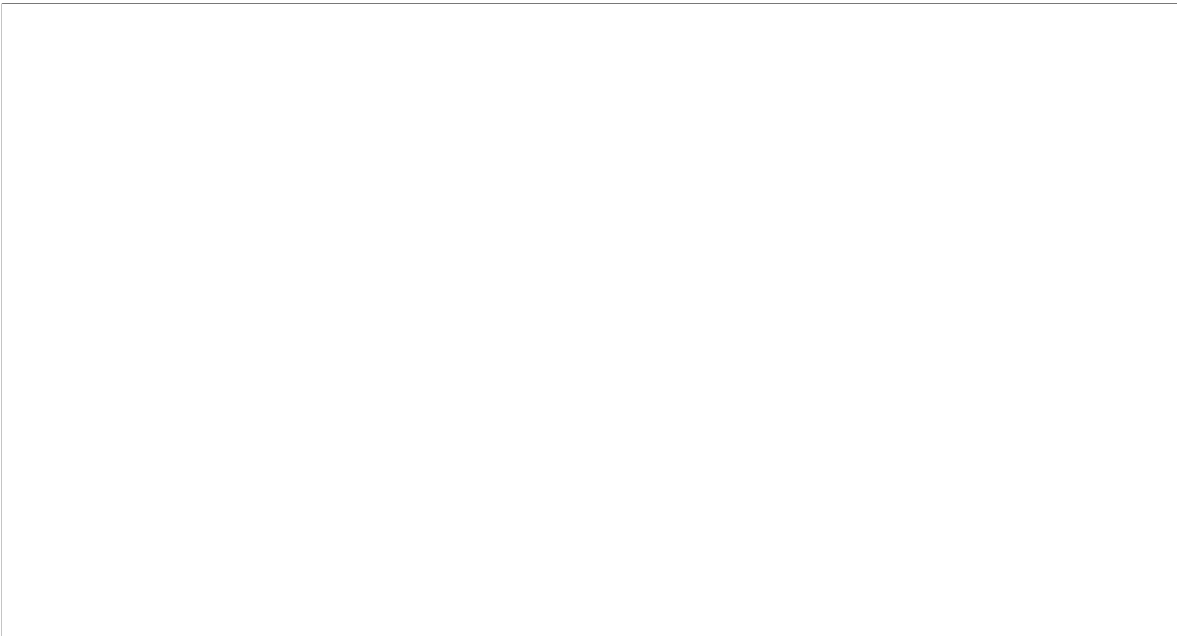
An AR model with infinite order can be written as:

$$r_t = \phi_0 + \phi_1 r_{t-1} + \phi_2 r_{t-2} + \dots + a_t$$

The above AR model is not realistic because it has infinite many parameters. One way to make the model practical is to assume that the coefficients satisfy some constraints so that they are determined by a finite number of parameters. A special case of this idea is

$$r_t = \phi_0 - \theta_1 r_{t-1} - \theta_1^2 r_{t-2} - \theta_1^3 r_{t-3} - \dots + a_t$$

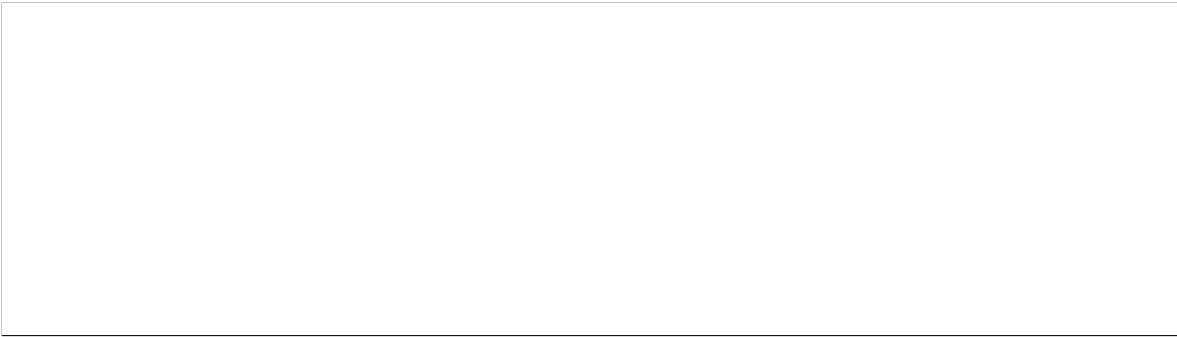
where $\phi_i = -\theta_1^i$ for all i



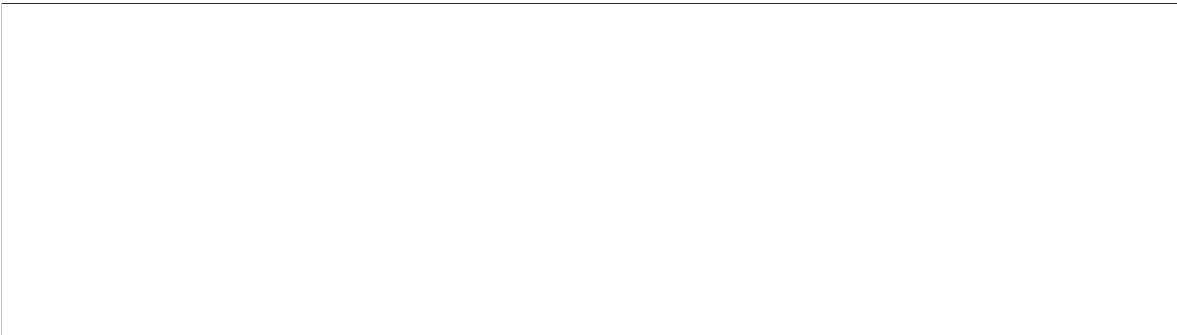
2.10.1 Properties of MA Models

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.

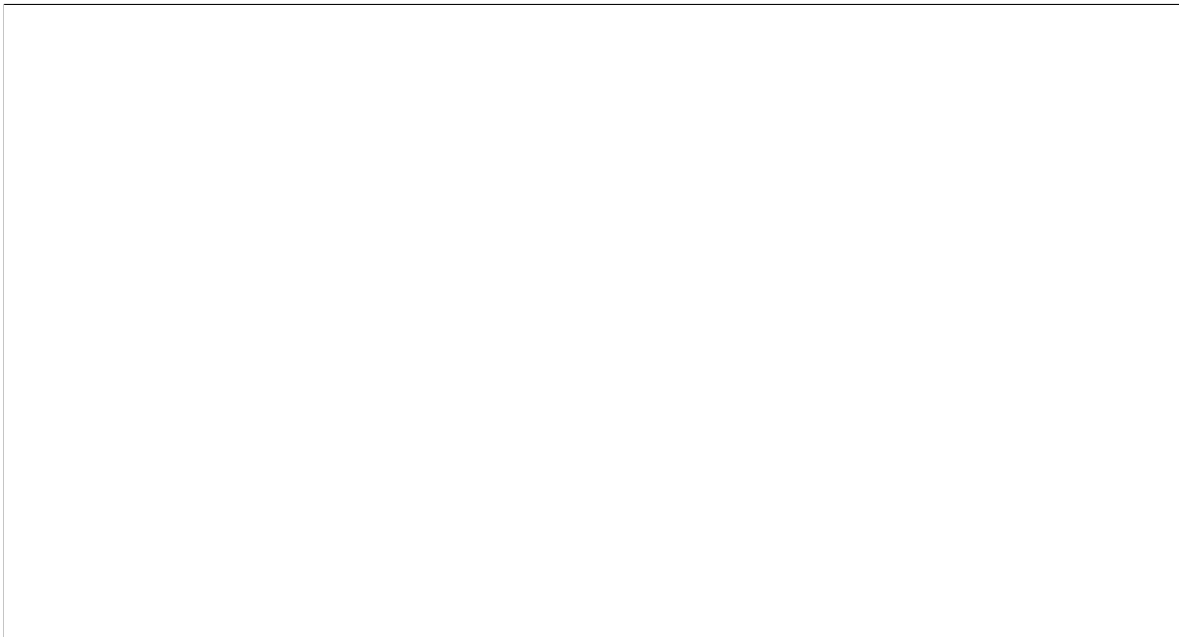
$E(r_t)$



$Var(r_t)$



Autocorrelation Function

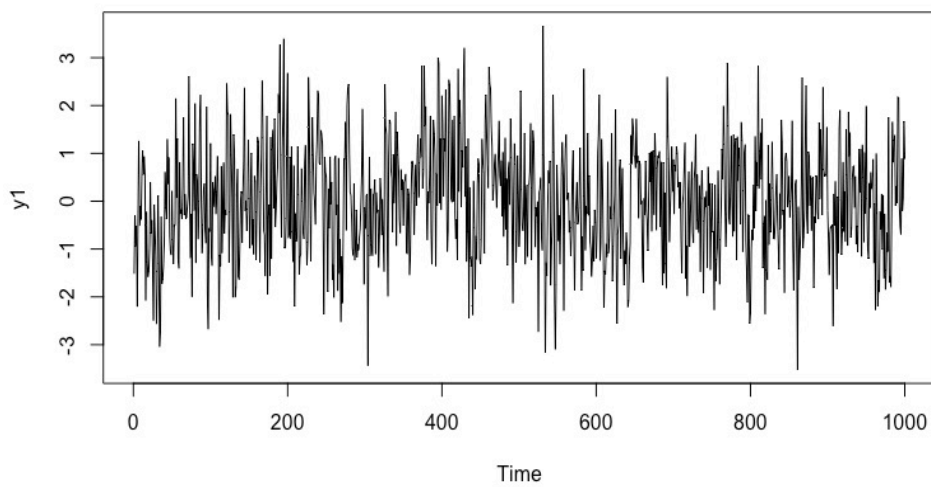


2.10.2 Identifying MA order

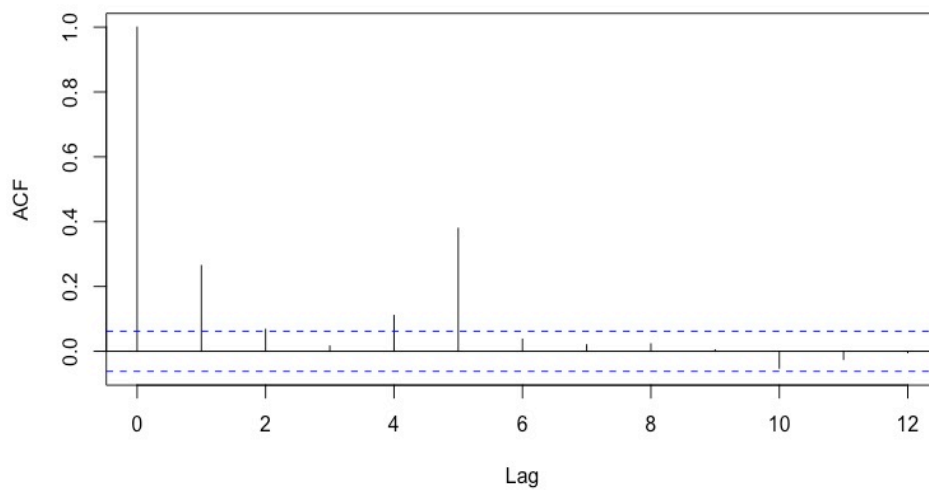
The ACF is useful in identifying the order of an MA model. For the MA(q), we find out that $\rho_q \neq 0$ but $\rho_l = 0$ for all $l > q$

The example of MA(5) can be depicted as following:

Figure of the MA(5) model and its ACF



Series y1



2.11 Parameter Estimation

For the MA(q), we can apply the Conditional Maximum Likelihood Estimation to estimate by starting from the observation (q+1). It can write down the model as following:

$$r_t = c_0 + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q}$$

$$t=q+1, \dots, T$$

The fitted model can be written as:

$$r_t = \hat{c}_0 - \hat{\theta}_1 a_{t-1} + \dots + \hat{\theta}_q a_{t-q} + \hat{a}_t$$

We can define the residual terms from the below equation:

$$\hat{a}_t = r_t - \hat{r}_t$$

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

2.12 Model Checking

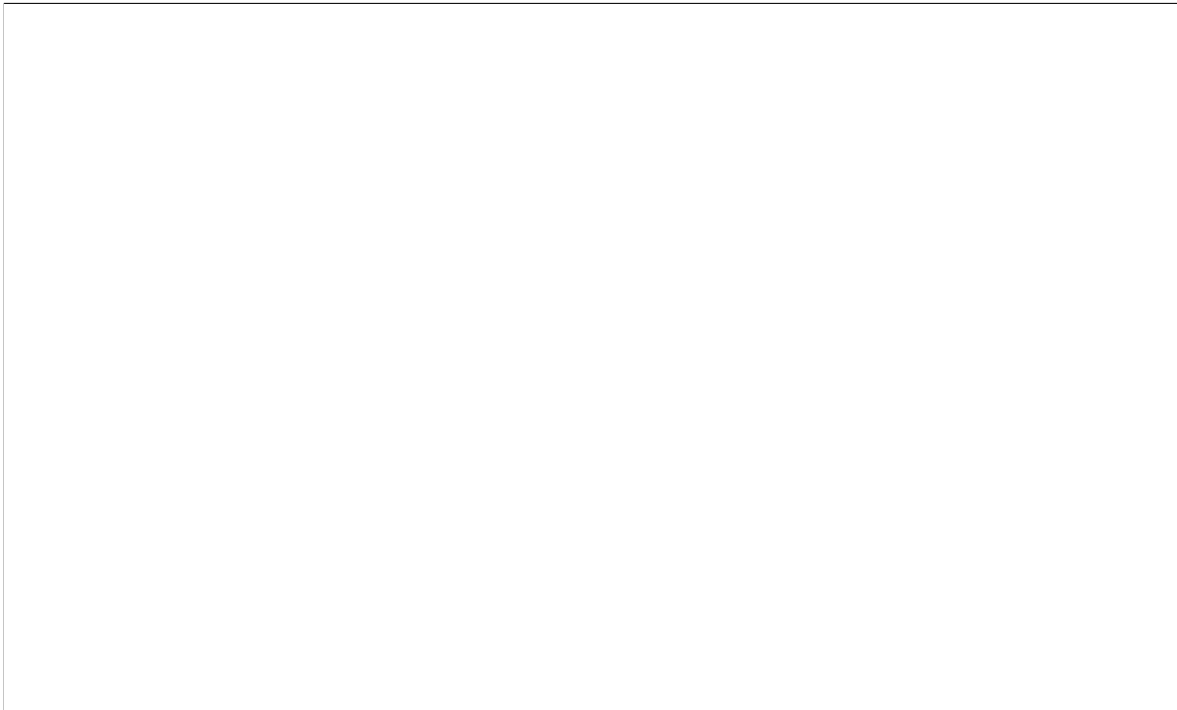
A fitted model must be examined carefully to check for possible model inadequacy. If the model is adequate, then the residual series should behave as a white noise. The ACF and the Ljung–Box statistics of the residuals can be used to check the closeness of a_t to a white noise. For an MA(q) model, the Ljung–Box statistic $Q(m)$ follows asymptotically a chi-squared distribution with $m-q$ degrees of freedom. Here the number of degrees of freedom is modified to signify that q MA coefficients are estimated. If a fitted model is found to be inadequate, it must be refined.

2.13 Forecasting

Forecasts of an MA model can easily be obtained. Because the model has finite memory, its point forecasts go to the mean of the series quickly. To see this, assume that the forecast origin is h . For the 1-step ahead forecast of an MA(1) process which is defined as

$$\hat{r}_h(l)$$

2.13.1 1-Step Ahead Forecast



2.13.2 2-Step Ahead Forecast



2.13.3 Multi-Step Ahead Forecast



The example of MA(q) model

```

> m2=arima(y1,order=c(0,0,5),include.mean = TRUE)
> m2

Call:
arima(x = y1, order = c(0, 0, 5), include.mean = TRUE)

Coefficients:
      ma1      ma2      ma3      ma4      ma5  intercept
 0.3172  0.0513 -0.0173 -0.0071  0.5308   0.0167
s.e.  0.0286  0.0285   0.0277   0.0292  0.0282   0.0580

sigma^2 estimated as 0.9601:  log likelihood = -1399.62,  aic = 2813.23
> names(m2)
 [1] "coef"      "sigma2"    "var.coef"  "mask"      "loglik"    "aic"      "
     arma"      "residuals" "call"
[10] "series"    "code"      "n.cond"    "nobs"      "model"
> Box.test(m2$residuals,lag=10,type='Ljung')

      Box-Ljung test

data:  m2$residuals
X-squared = 5.6582, df = 10, p-value = 0.8431

> ts.plot(m2$residuals)
> predict(m2,5)
$pred
Time Series:
Start = 1001
End = 1005
Frequency = 1
[1] 0.3016739 -0.4146540 0.1831065 1.0182231 0.1933976

$se
Time Series:
Start = 1001
End = 1005
Frequency = 1
[1] 0.9798251 1.0279423 1.0291705 1.0293095 1.0293327

```

2.14 ARMA Models

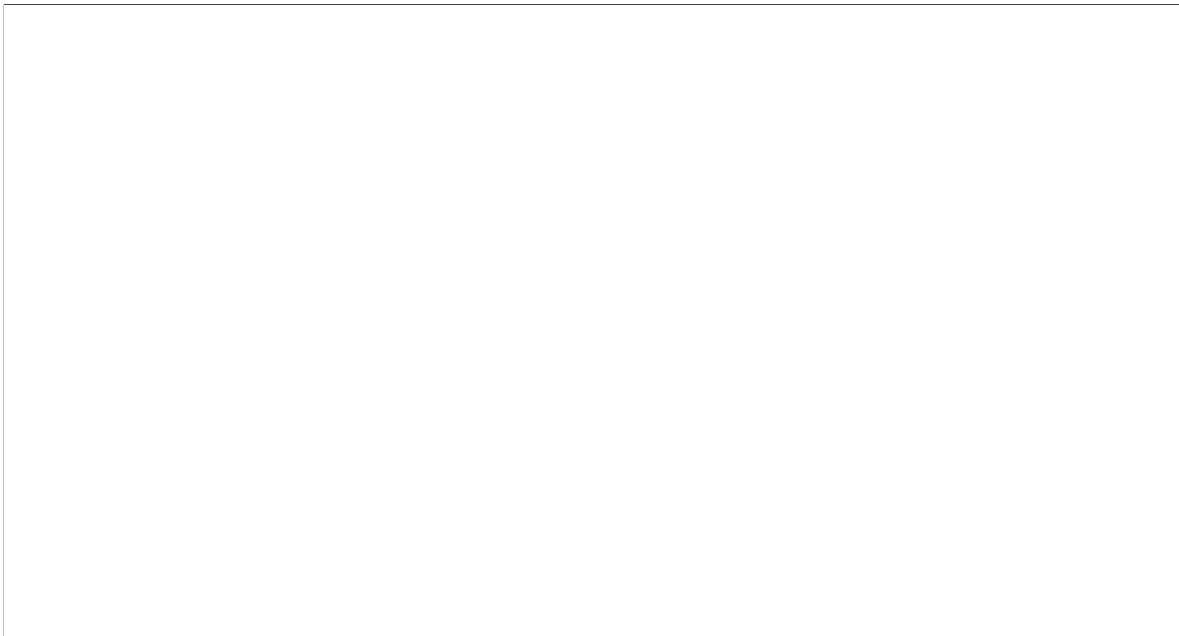
A time series r_t follows an ARMA (1,1) model if it satisfies

$$r_t - \phi_1 r_{t-1} = \phi_0 + a_t - \theta_1 a_{t-1}$$

where a_t is the White Noise Series.

2.14.1 Properties of ARMA Models

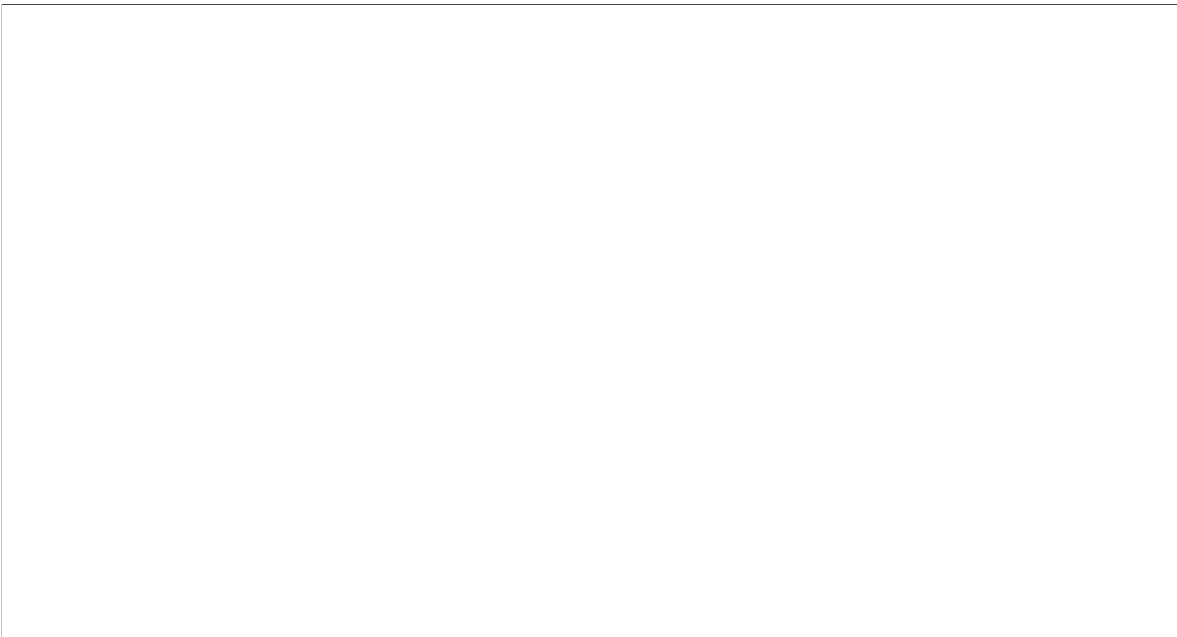
$E(r_t)$



$Var(r_t)$



Autocorrelation Function



2.15 General ARMA Model



2.16 Identifying ARMA Model



2.17 Model Checking

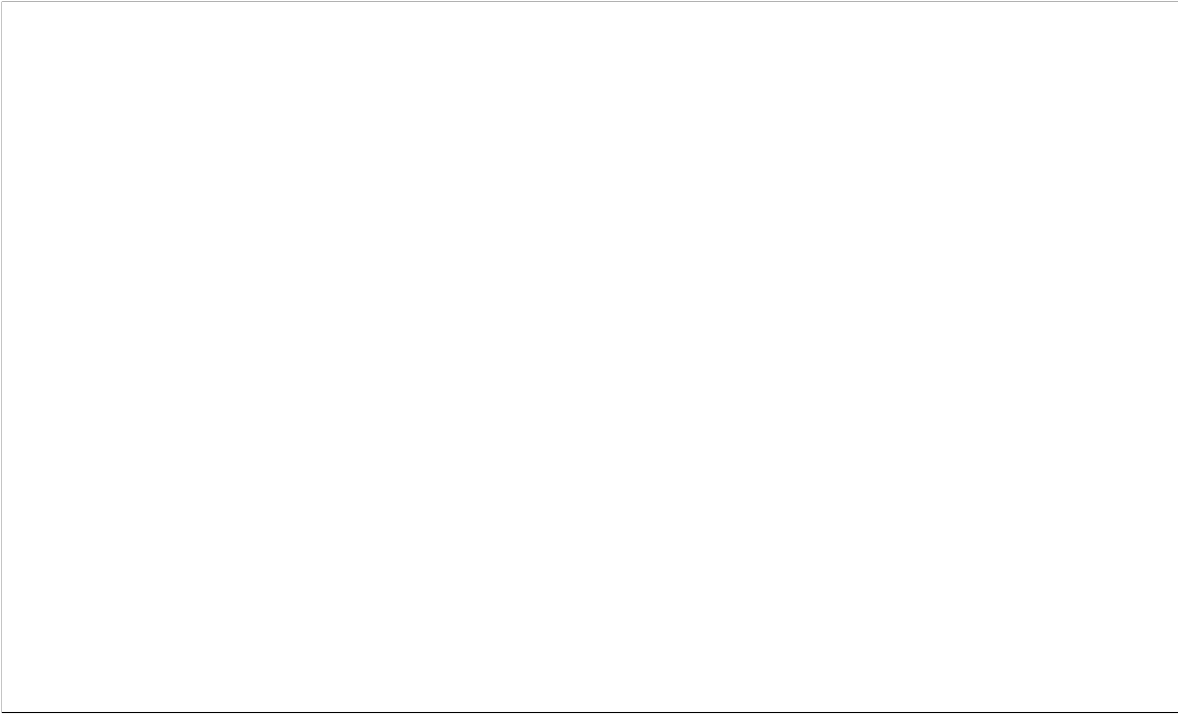


2.18 Forecasting

2.18.1 1-Step Ahead Forecast



2.18.2 2-Step Ahead Forecast



2.18.3 Multi-Step Ahead Forecast



Comparing between the AR(p) and ARMA(p,q) to estimate the Civilian Unemployment Rate

```
setwd("/Users/wasinsiwasarit/Desktop/EE435")
cat(rep("\n",50)) #clear R Console
require(quantmod)
getSymbols("UNRATE",src="FRED")
dim(UNRATE)
head(UNRATE)
rate <- as.numeric(UNRATE[,1])
ts.plot(rate)
logreturn=diff(log(rate))
m1 <- ar(logreturn,order.max=15) ## AR order selection using AIC
m1$order
pacf(logreturn)
m2 <- arima(logreturn,order=c(12,0,0))
m2
tsdiag(m2,gof=36)
### Model refinement
c1 <- c(0,NA,NA,NA,NA,0,0,0,0,NA,0,NA,NA)
m3 <- arima(logreturn,order=c(12,0,0),fixed=c1)
m3
require(forecast)
auto.arima(logreturn)
m4 <- arima(logreturn,order=c(2,0,2))
m4
tsdiag(m4,gof=36)
source("/Users/wasinsiwasarit/Desktop/EE435/backtest.R")
backtest(m3,logreturn,770,fixed=c1)
backtest(m4,logreturn,770)
```

The main results:

```

> require(quantmod)
> getSymbols("UNRATE",src="FRED")
[1] "UNRATE"
> dim(UNRATE)
[1] 836  1
> head(UNRATE)
      UNRATE
1948-01-01  3.4
1948-02-01  3.8
1948-03-01  4.0
1948-04-01  3.9
1948-05-01  3.5
1948-06-01  3.6
> rate <- as.numeric(UNRATE[,1])
> ts.plot(rate)
> logreturn=diff(log(rate))
> m1 <- ar(logreturn,order.max=15) ## AR order selection using AIC
> m1$order
[1] 12
> pacf(logreturn)
> m2 <- arima(logreturn,order=c(12,0,0))
> m2

```

Call:

```
arima(x = logreturn, order = c(12, 0, 0))
```

Coefficients:

	ar1	ar2	ar3	ar4	ar5	ar6	ar7	ar8	ar9
		ar10	ar11	ar12	intercept				
	0.0542	0.1665	0.1197	0.0818	0.1340	0.0262	-0.0244	0.0278	
		0.0033	-0.1093	0.0531	-0.1452	4e-04			
s.e.	0.0345	0.0346	0.0348	0.0354	0.0355	0.0358	0.0358	0.0357	0.0356
	0.0352	0.0349	0.0350	2e-03					

```
sigma^2 estimated as 0.001238: log likelihood = 1609.61, aic = -3191.22
```

```

> tsdiag(m2,gof=36)
> c1 <- c(0,NA,NA,NA,NA,0,0,0,0,NA,0,NA,NA)
> m3 <- arima(logreturn,order=c(12,0,0),fixed=c1)

```

Warning message:

```

In arima(logreturn, order = c(12, 0, 0), fixed = c1) :
  some AR parameters were fixed: setting transform.pars = FALSE
> m3

```

Call:

```

arima(x = logreturn, order = c(12, 0, 0), fixed = c1)

Coefficients:
      ar1      ar2      ar3      ar4      ar5 ar6 ar7 ar8 ar9      ar10 ar11
      ar12 intercept
      0  0.1693  0.1400  0.0966  0.1418  0  0  0  0  -0.1008  0
      -0.1369  0.0004
s.e.    0  0.0343  0.0336  0.0340  0.0347  0  0  0  0  0.0344  0
      0.0341  0.0018

sigma^2 estimated as 0.001249: log likelihood = 1606, aic = -3195.99
> require(forecast)
> auto.arima(logreturn)
Series: logreturn
ARIMA(2,0,2) with zero mean

Coefficients:
      ar1      ar2      ma1      ma2
      1.6382 -0.7491 -1.5965  0.7931
s.e.  0.0548  0.0554  0.0507  0.0535

sigma^2 estimated as 0.001291: log likelihood=1594.16
AIC=-3178.31 AICc=-3178.24 BIC=-3154.67
> m4 <- arima(logreturn,order=c(2,0,2))
> m4

Call:
arima(x = logreturn, order = c(2, 0, 2))

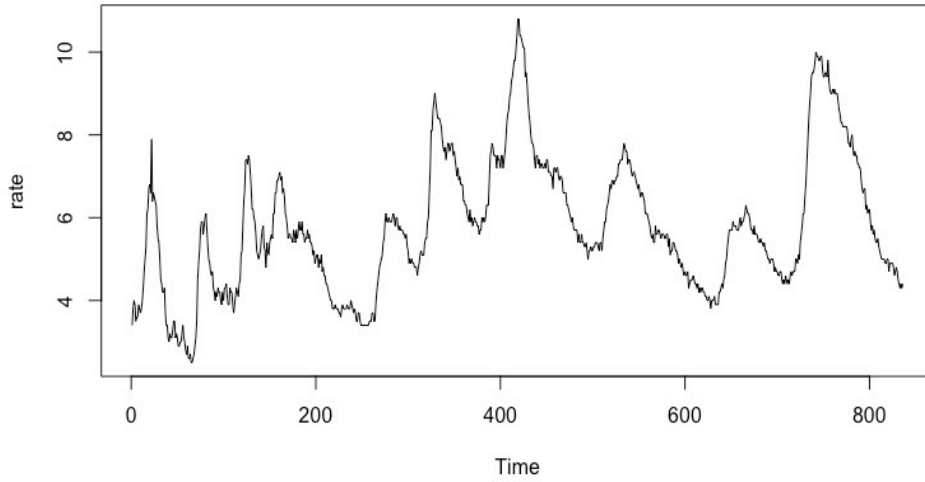
Coefficients:
      ar1      ar2      ma1      ma2 intercept
      1.6385 -0.7494 -1.5968  0.7934  0.0005
s.e.  0.0547  0.0553  0.0506  0.0534  0.0022

sigma^2 estimated as 0.001285: log likelihood = 1594.18, aic = -3176.36
> tsdiag(m4,gof=36)
> source("/Users/wasinsiwasarit/Desktop/EE435/backtest.R")
> backtest(m3,logreturn,770,fixed=c1)
[1] "RMSE of out-of-sample forecasts"
[1] 0.02535371
[1] "Mean absolute error of out-of-sample forecasts"
[1] 0.02022139
There were 50 or more warnings (use warnings() to see the first 50)
> backtest(m4,logreturn,770)
[1] "RMSE of out-of-sample forecasts"
[1] 0.02465446

```

```
[1] "Mean absolute error of out-of-sample forecasts"  
[1] 0.01972048  
>
```

Figure: Civilian Unemployment Rate



แผนภาพ PACF ของ Civilian Unemployment Rate

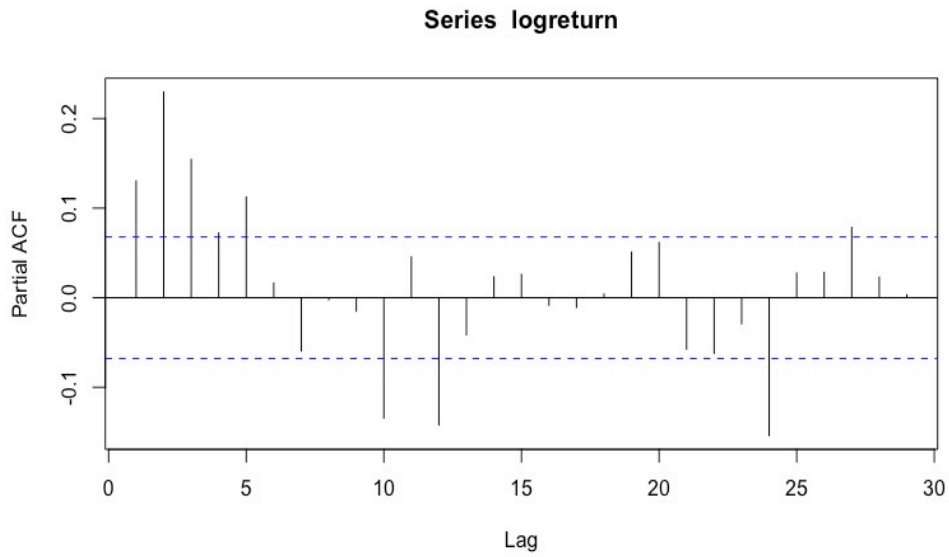


Figure: Residual Term Civilian Unemployment Rate from the AR(13) model

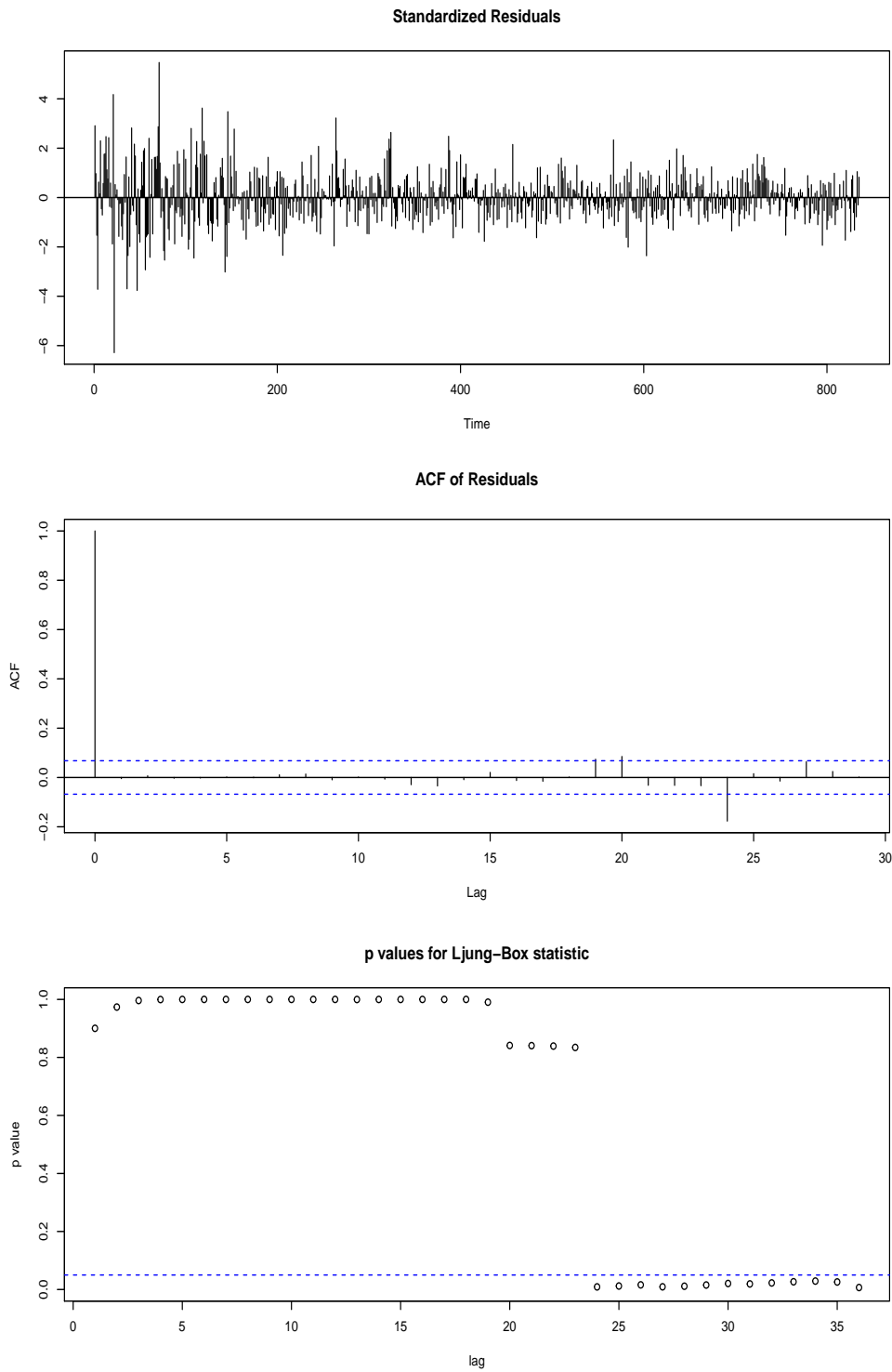
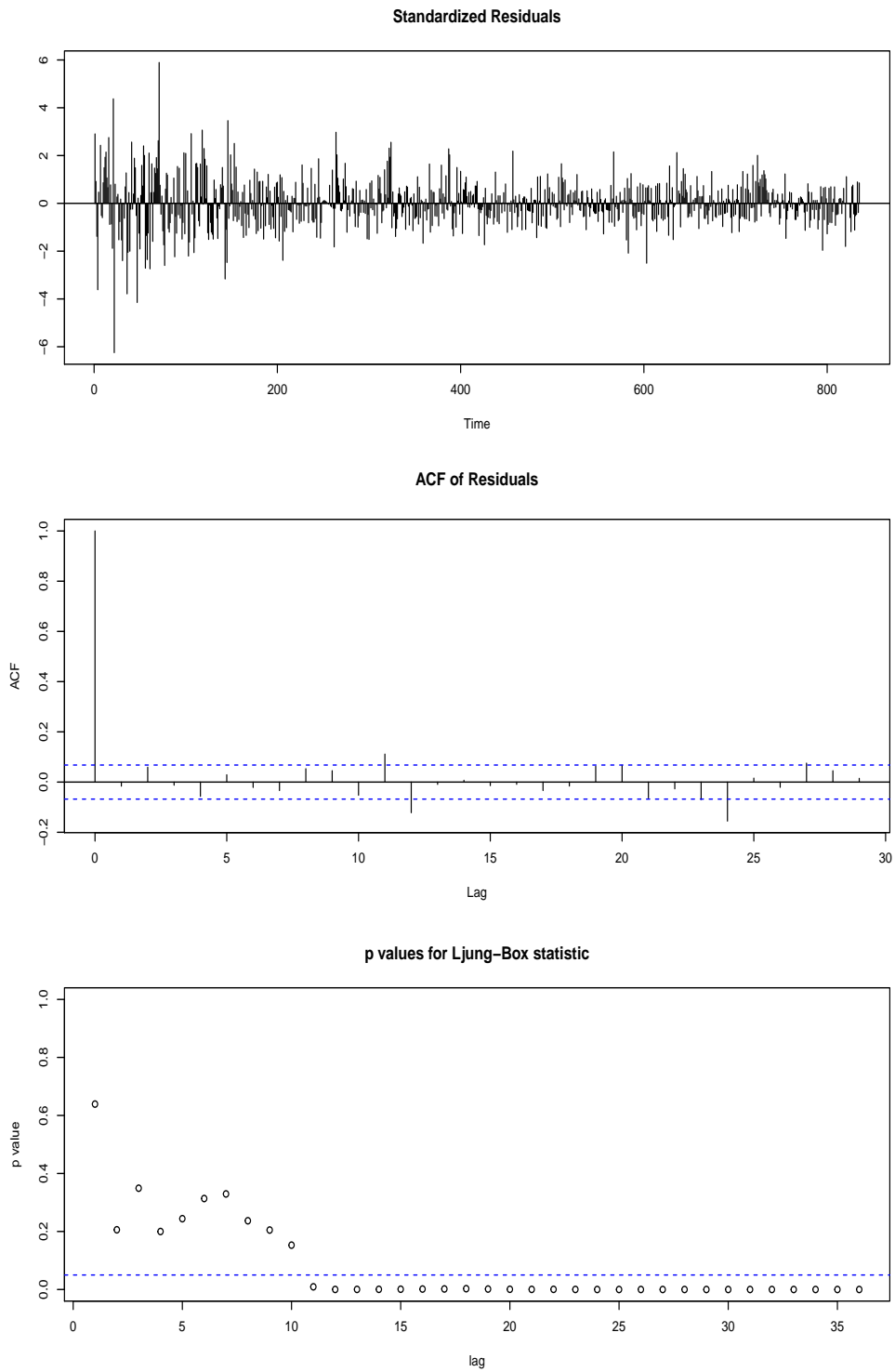


Figure: Residual Term Civilian Unemployment Rate from the ARMA(2,0,2)



2.19 Unit-root Nonstationarity

2.19.1 Random Walk

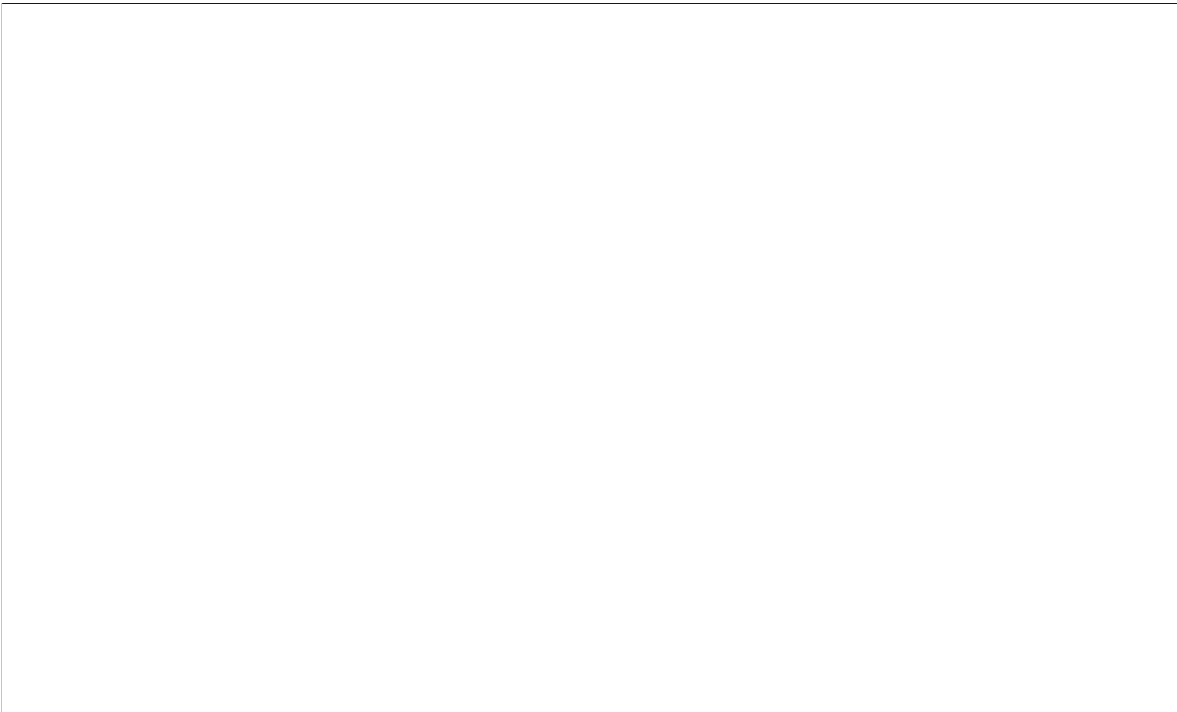
$$p_t = p_{t-1} + a_t$$

Unit root? It is an AR(1) model with coefficient $\phi_1 = 1$

Nonstationary: Why? Because the variance of r_t diverges to infinity as t increases.

Strong Memory: Sample ACF approaches 1 for any finite lag.

Repeated Substitution shows:



2.19.2 Random Walk with Drift

Form:

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

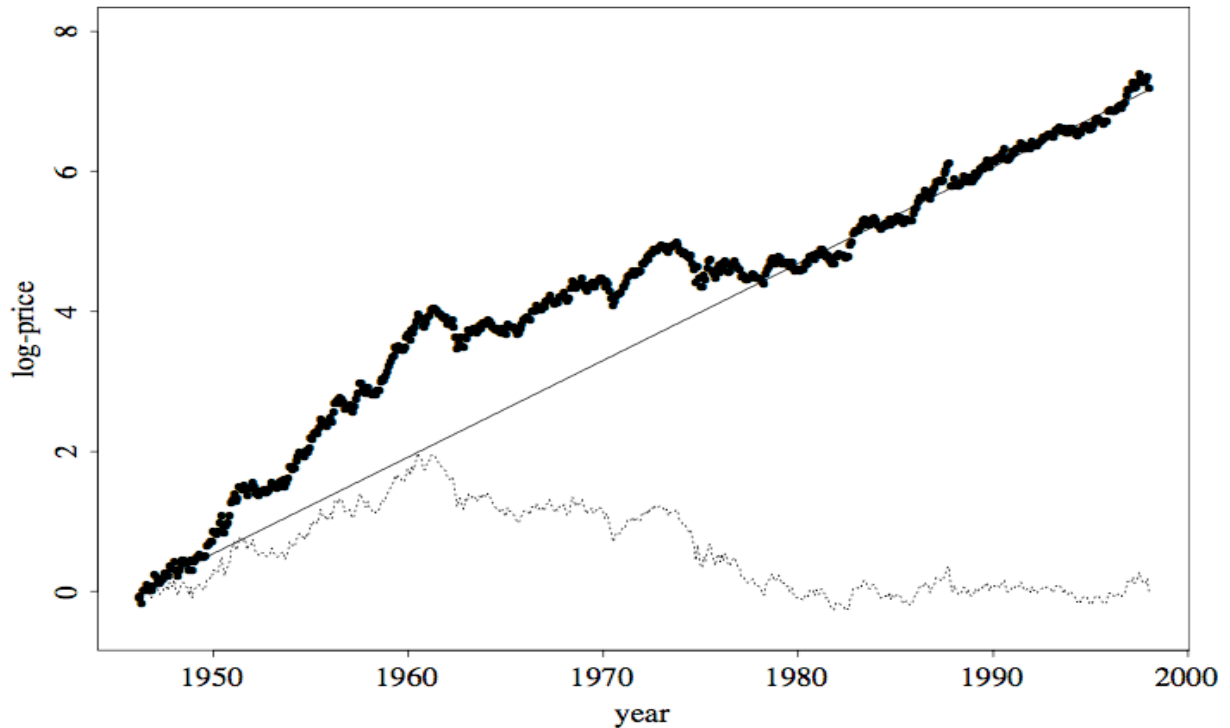
Has a unit root

Nonstationary

Strong Memory

Has a time trend with slope μ . Why?

Figure: Time plots of log prices for 3M stock from February 1946 to December 1997, assuming that the log price of January 1946 was zero. The dashed line is for log price without time trend. The straight line is $y_t = 0.0115 * t$



2.19.3 Differencing

1st difference:

$$r_t = p_t - p_{t-1}$$

If p_t is the log-price, then the 1st difference is simply the log return. Typically, 1st difference means the "Change" or "increment" of the original series.

Seasonal difference: $y_t = p_t - p_{t-s}$, where s is the periodicity, e.g. $s=4$ for the quarterly series and $s=12$ for monthly series.

If p_t denote quarterly earning, then y_t is the change in earning from the same quarter one year before.



2.19.4 Meaning of the Constant Term

♡ MA model

♡ AR model

♡ 1st differenced

2.20 Unit-Root

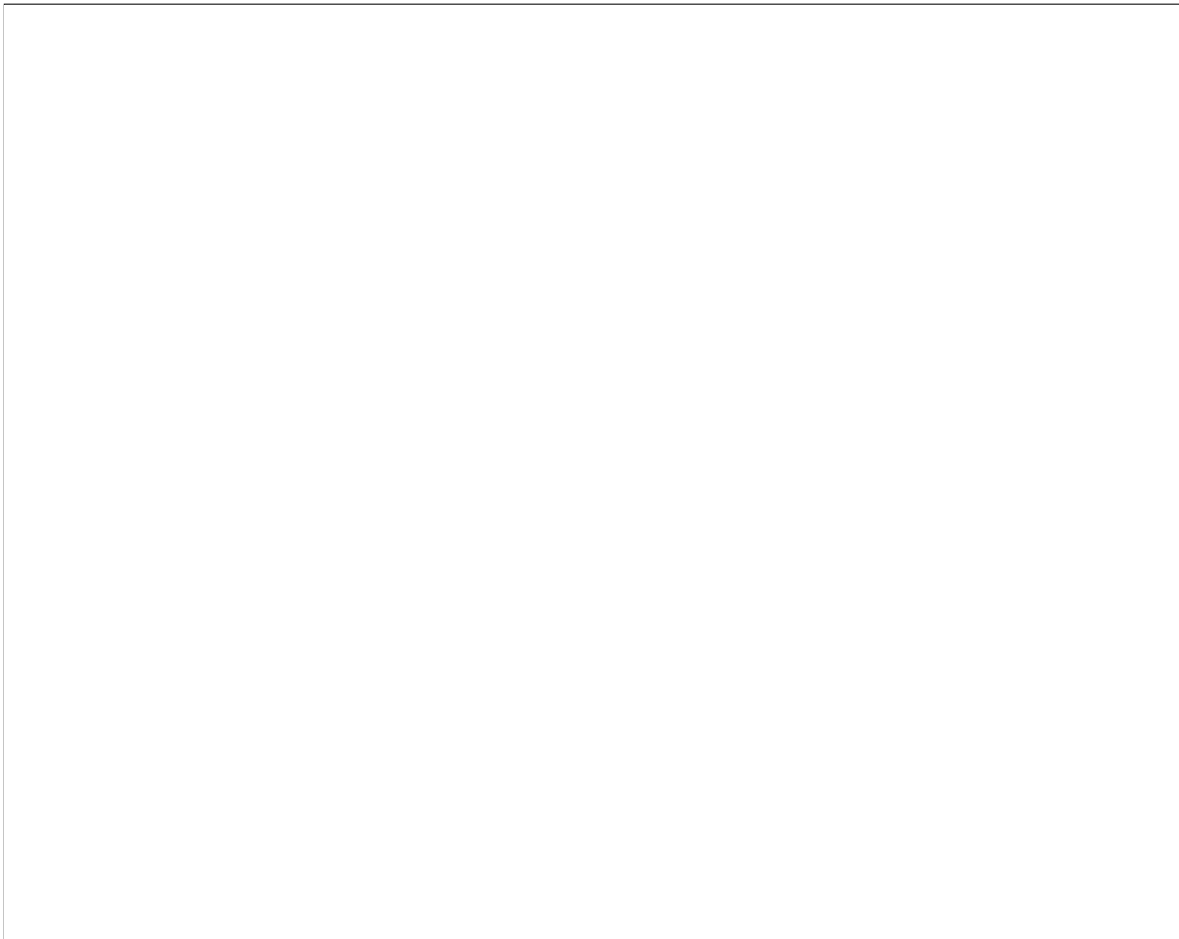
To check the series log price p_t to be random walk process:

$$p_t = \phi_1 p_{t-1} + e_t$$

or random walk with a drift:

$$p_t = \phi_0 + \phi_1 p_{t-1} + e_t$$

We can apply the following process:



The Result from Unit-root Test

```

> setwd("/Users/wasinsiwasarit/Desktop/EE435")
> cat(rep("\n",50)) #clear R Console
> library(quantmod)
Loading required package: xts
Loading required package: zoo
Attaching package: 'zoo'
The following objects are masked from 'package:base':
  as.Date, as.Date.numeric
Loading required package: TTR
Version 0.4-0 included new data defaults. See ?getSymbols.
Learn from a quantmod author: https://www.datacamp.com/courses/importing-and-
  managing-financial-data-in-r
> library(fBasics)
Loading required package: timeDate
Loading required package: timeSeries
Attaching package: 'timeSeries'
The following object is masked from 'package:zoo':

  time<-

Rmetrics Package fBasics
Analysing Markets and calculating Basic Statistics
Copyright (C) 2005-2014 Rmetrics Association Zurich
Educational Software for Financial Engineering and Computational Science
Rmetrics is free software and comes with ABSOLUTELY NO WARRANTY.
https://www.rmetrics.org --- Mail to: info@rmetrics.org

Attaching package: 'fBasics'

The following object is masked from 'package:TTR':

  volatility

> library(fUnitRoots)
Loading required package: urca

Attaching package: 'fUnitRoots'

The following objects are masked from 'package:urca':

  punitroot, qunitroot, unitrootTable

> library(forecast)
> lexrates<- read.csv(file="lexrates.csv",head=TRUE,sep=";")

```

```

> uscn.spot = lexrates[, "USCNS"]
> plot.ts(uscns.spot, main="Log of US/CN spot exchange rate")
> xx = acf(uscns.spot)
> plot.ts(diff(uscns.spot), main="First difference of Log of US/CN spot exchange
      rate")
> xx = acf(diff(uscns.spot))
> m1=adfTest(uscns.spot, lags = 2, type = c("c"), title = NULL,
+           description = NULL)
> m1@test$p.value

0.231362
> m1@test$parameter
Lag Order
      2
> m1@test$lm

Call:
lm(formula = y.diff ~ y.lag.1 + 1 + y.diff.lag)

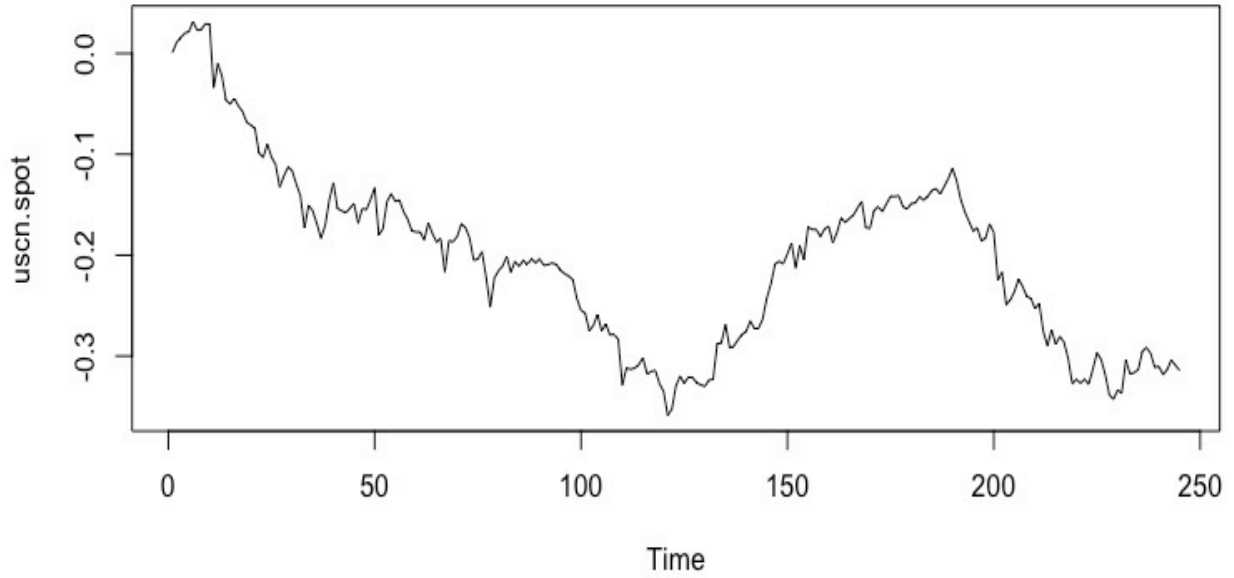
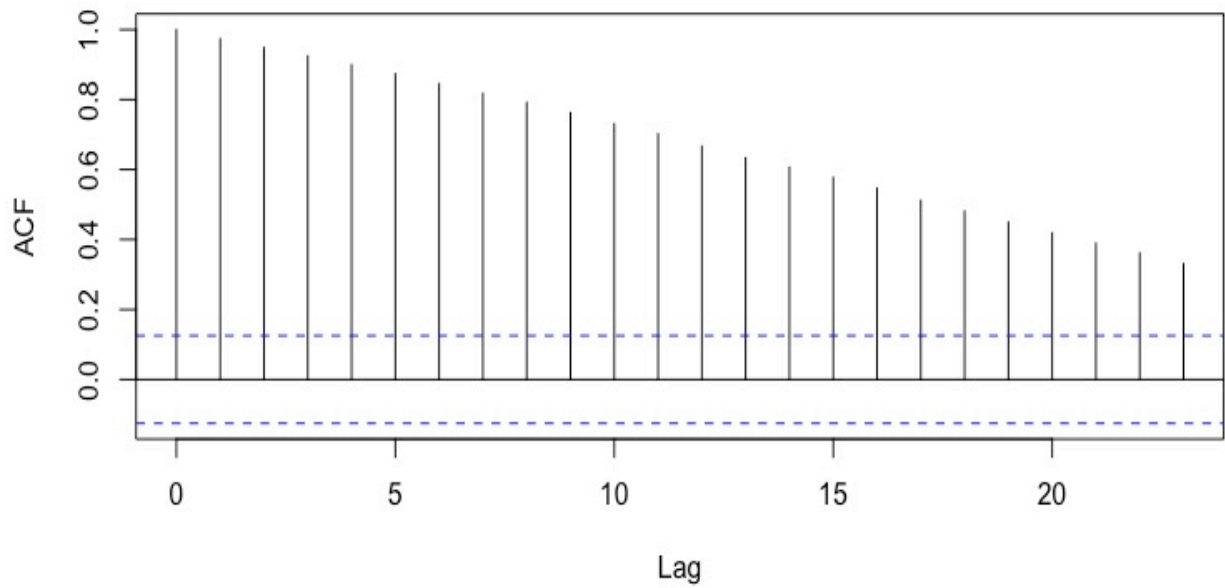
Coefficients:
(Intercept)      y.lag.1  y.diff.lag1  y.diff.lag2
   -0.006226   -0.022717   -0.112722   -0.048532

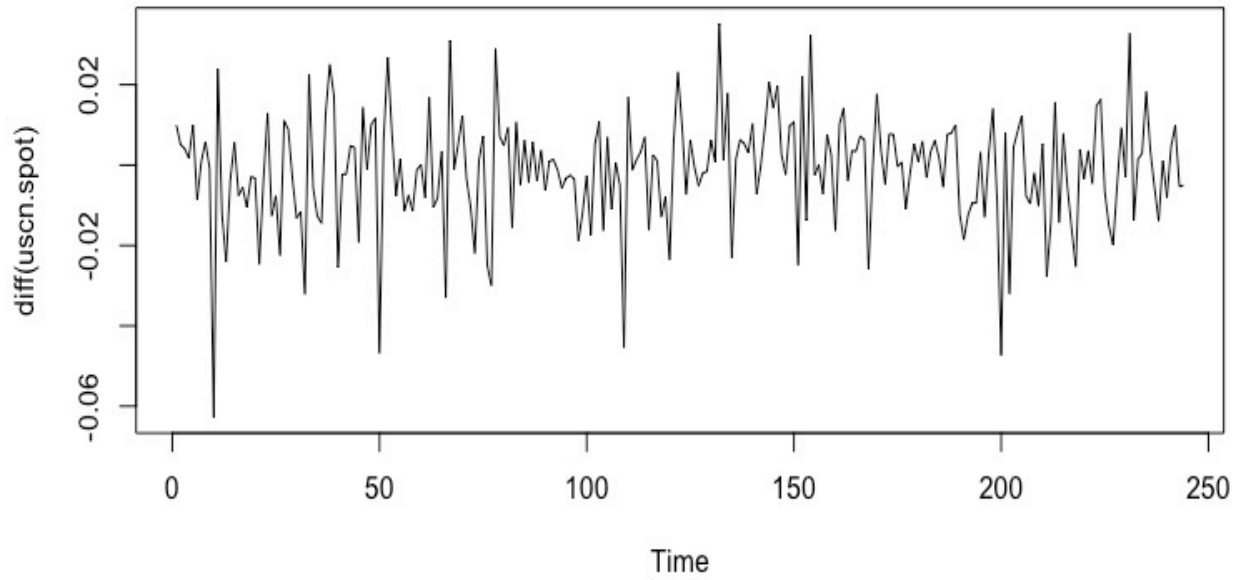
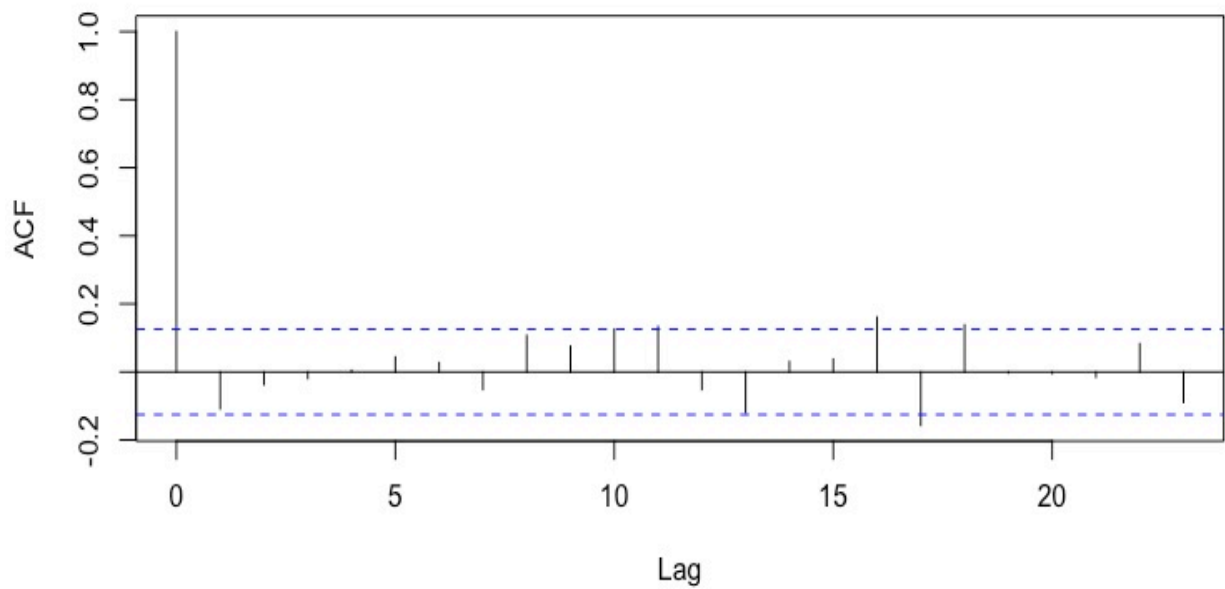
> y=diff(uscns.spot)
> m2=adfTest(y, lags = 6, type = c("ct"), title = NULL,
+           description = NULL)
Warning message:
In adfTest(y, lags = 6, type = c("ct"), title = NULL, description = NULL) :
  p-value smaller than printed p-value
> m3=auto.arima(uscns.spot)
> m3
Series: uscn.spot
ARIMA(2,1,2) with drift

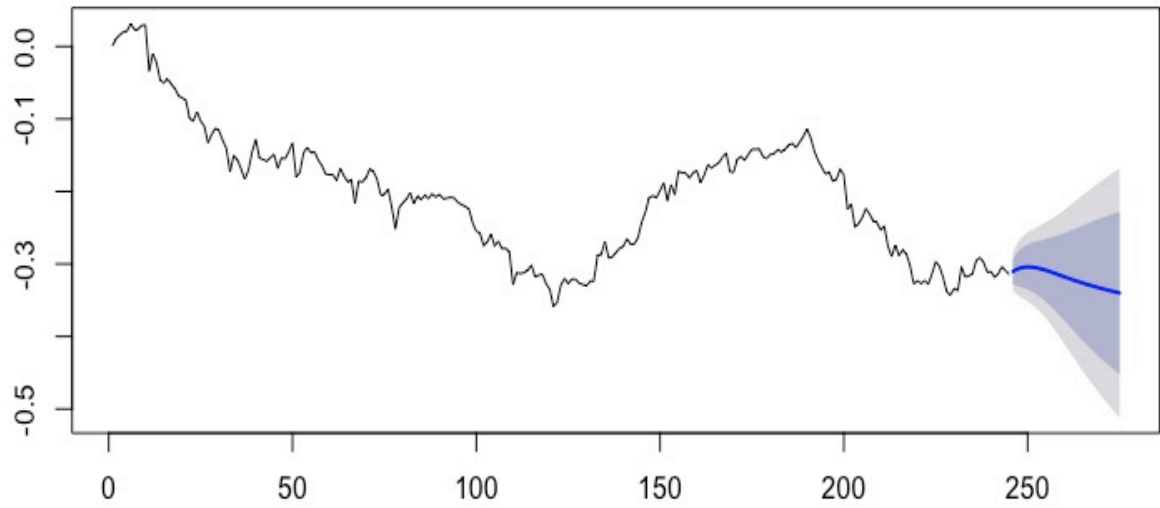
Coefficients:
      ar1      ar2      ma1      ma2      drift
  1.7250 -0.7574 -1.8812  0.9277 -0.0012
s.e.  0.0705  0.0703  0.0416  0.0418  0.0012

sigma^2 estimated as 0.0001834:  log likelihood=705.46
AIC=-1398.93  AICc=-1398.57  BIC=-1377.95
> par(mfrow=c(1,1))
> plot(forecast(m3,h=30))

```

Log of US/CN spot exchange rate**Series uscn.spot**

First difference of Log of US/CN spot exchange rate**Series diff(uscn.spot)**

Forecasts from ARIMA(2,1,2) with drift

2.21 Seasonal Time Series

Seasonal Time Series: TS with periodic patterns and useful in

- predicting quarterly earnings
- pricing weather-related derivatives
- analysis of transactions data (high-frequency data), e.g., U-shaped pattern in intraday trading intensity, volatility, etc.

2.21.1 Multiplicative Model

Let y_t be the monthly data. Denoting 1959 as year 0, we can write the time index as $t = \text{year} + \text{month}$, e.g., $y_1 = y_{0,1}$, $y_2 = y_{0,2}$ etc. The multiplicative model is based on the following consideration:

	Month						
Year	Jan	Feb	Mar	...	Oct	Nov	Dec
1959	$y_{0,1}$	$y_{0,2}$	$y_{0,3}$...	$y_{0,10}$	$y_{0,11}$	$y_{0,12}$
1960	$y_{1,1}$	$y_{1,2}$	$y_{1,3}$...	$y_{1,10}$	$y_{1,11}$	$y_{1,12}$
1961	$y_{2,1}$	$y_{2,2}$	$y_{2,3}$...	$y_{2,10}$	$y_{2,11}$	$y_{2,12}$
1962	$y_{3,1}$	$y_{3,2}$	$y_{3,3}$...	$y_{3,10}$	$y_{3,11}$	$y_{3,12}$
⋮	⋮	⋮	⋮		⋮	⋮	⋮

The Application of Program R

```
#EE435
setwd("/Users/wasinsiwasarit/Desktop/EE435")
cat(rep("\n",50)) #clear R Console

da=read.table("q-earn-jnj.txt")
jnj=da[,1]
ts.plot(jnj)
ljj=log(jnj)
ts.plot(ljj)
acf(ljj)
djj=diff(ljj)
acf(djj,lag=20)
dd <- diff(djj,4) ### seasonal difference
acf(dd)
m5 = arima(ljj,order=c(0,1,1),seasonal=list(order=c(0,1,1),period=4))
m5
tsdiag(m5,gof=12)
```

```
predict(m5,8)
```

Results from the Program R

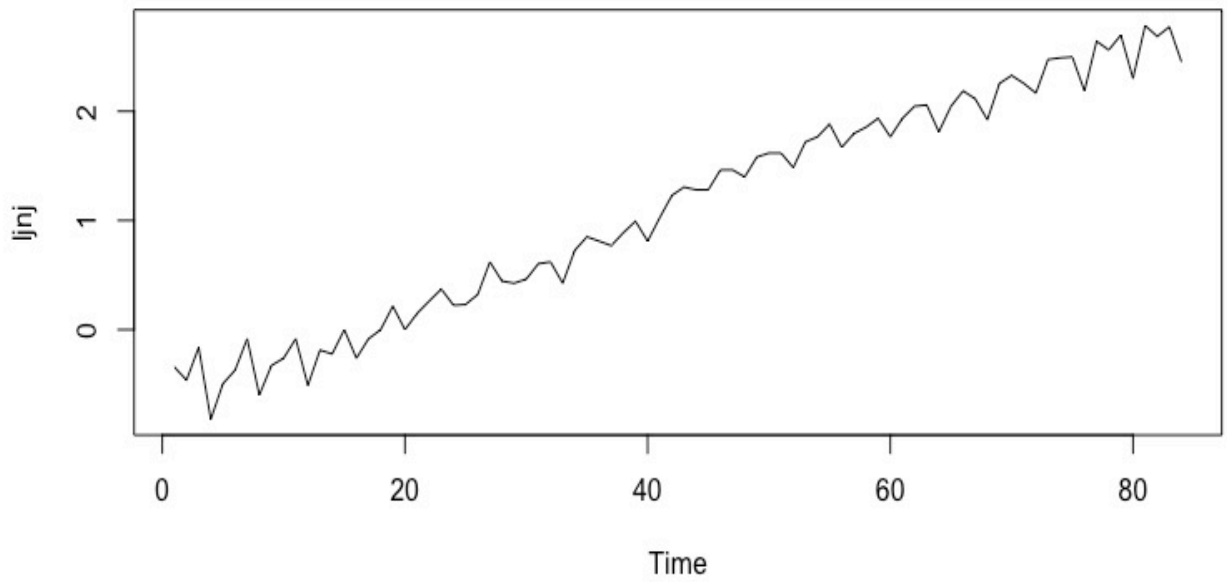
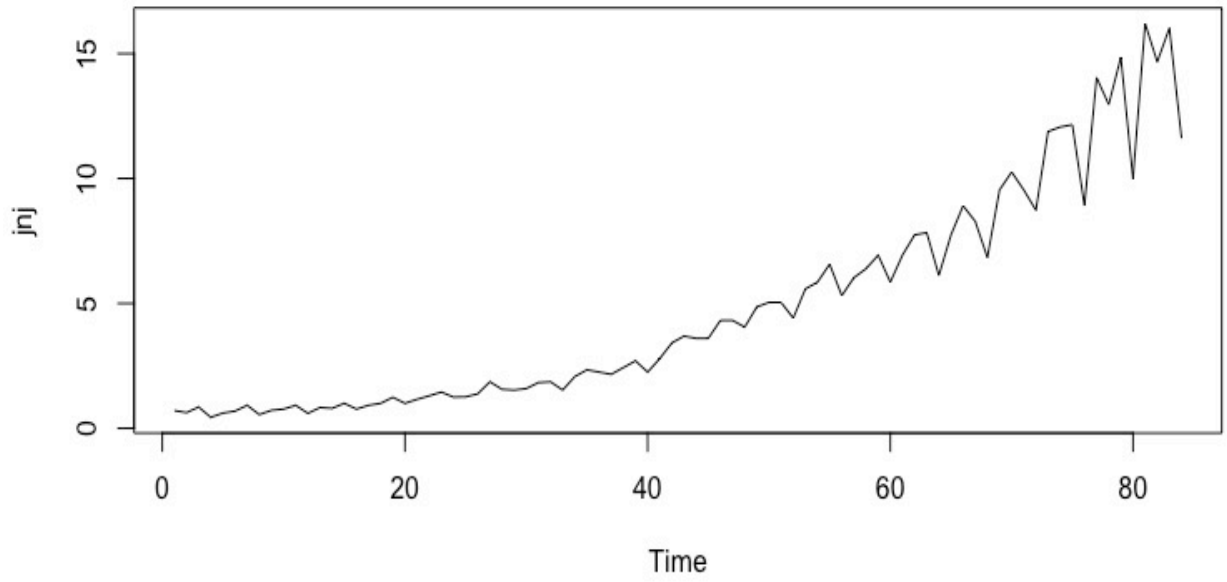
```
> da=read.table("q-earn-jnj.txt")
> jnj=da[,1]
> ts.plot(jnj)
> llnj=log(jnj)
> ts.plot(llnj)
> acf(llnj)
> dlnj=diff(llnj)
> acf(dlnj,lag=20)
> dd <- diff(dlnj,4) ### seasonal difference
> acf(dd)
> m5 = arima(llnj,order=c(0,1,1),seasonal=list(order=c(0,1,1),period=4))
> m5

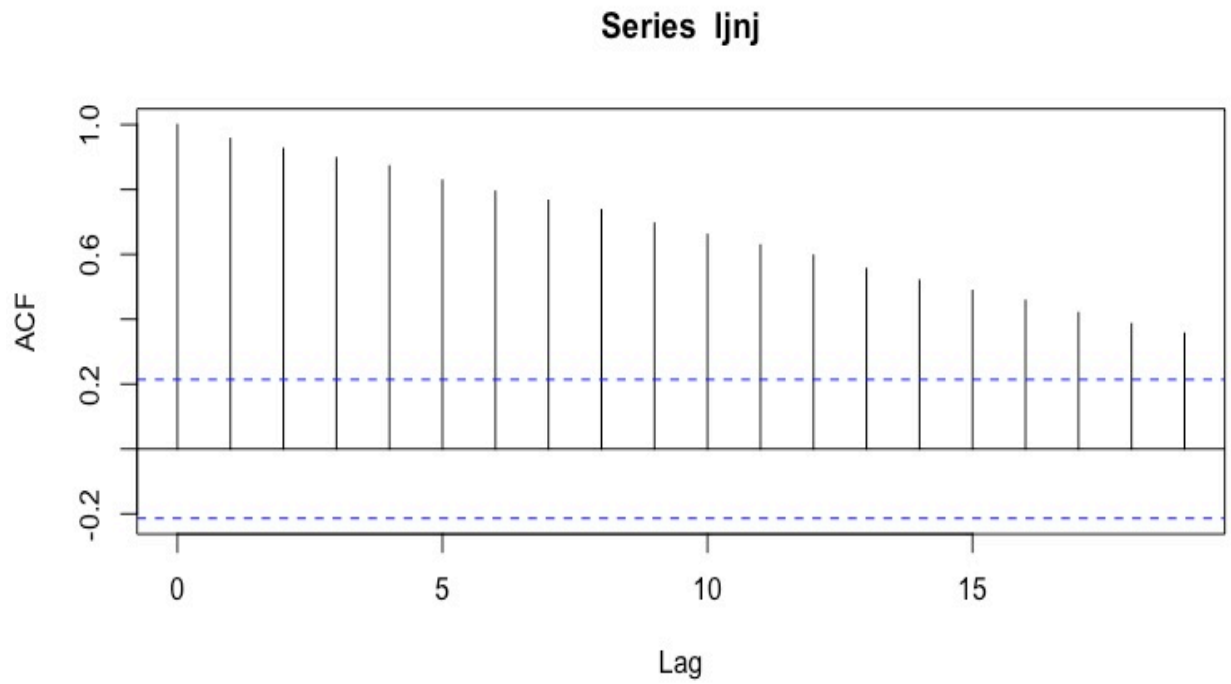
Call:
arima(x = llnj, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period
= 4))

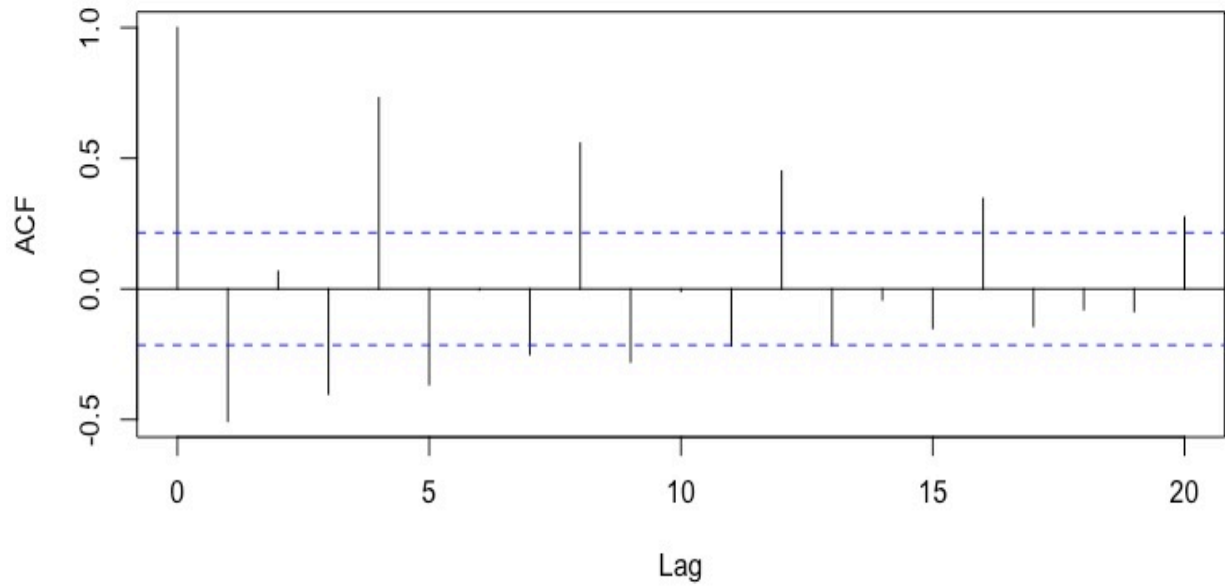
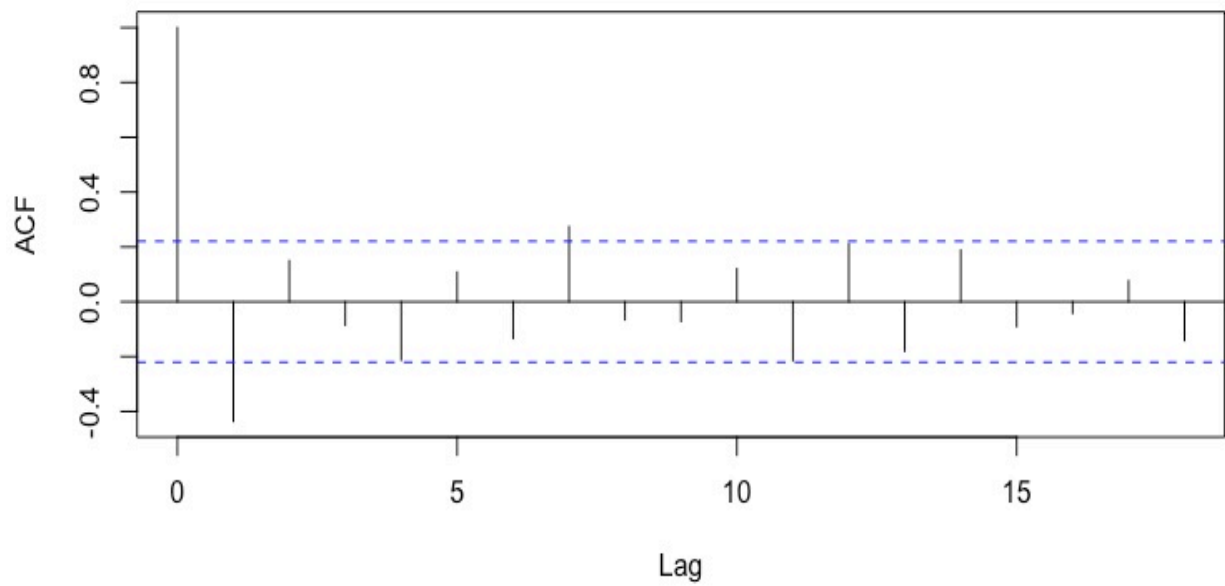
Coefficients:
          ma1      sma1
      -0.6809  -0.3146
s.e.    0.0982   0.1070

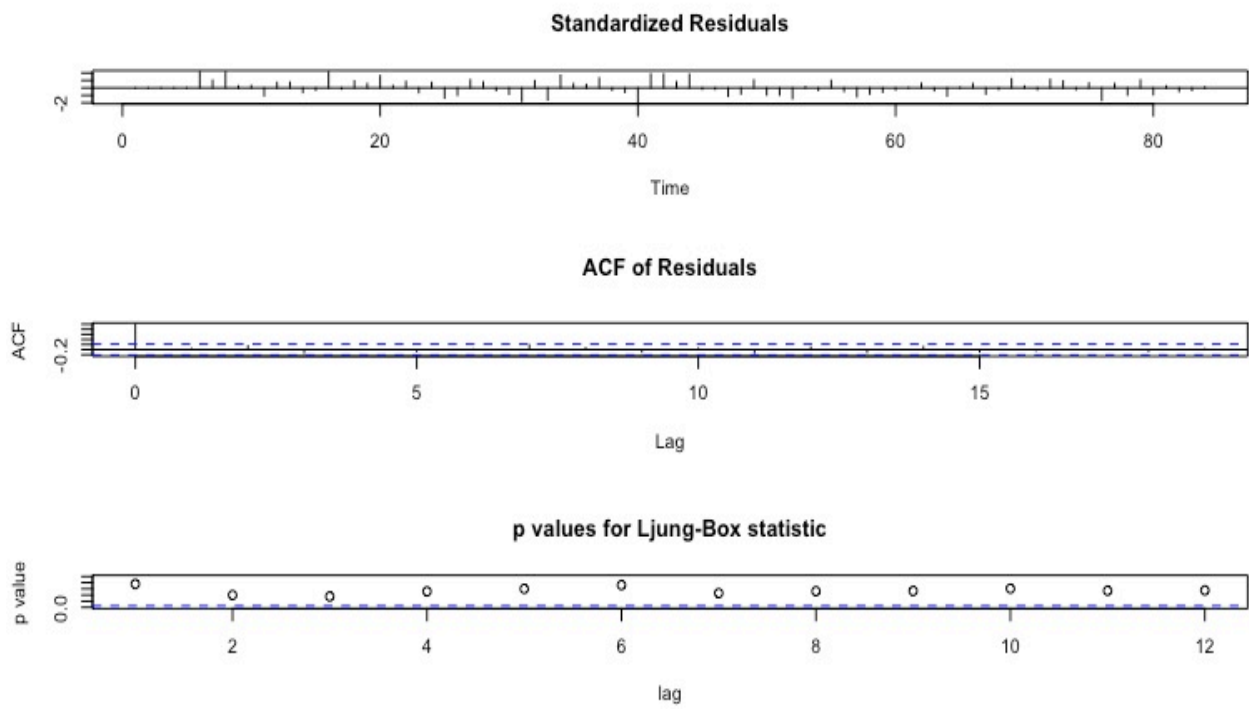
sigma^2 estimated as 0.007931:  log likelihood = 78.38,  aic = -150.75
> tsdiag(m5,gof=12)
> predict(m5,8)
$pred
Time Series:
Start = 85
End = 92
Frequency = 1
[1] 2.905343 2.823891 2.912148 2.581085 3.036450 2.954999 3.043255 2.712193

$se
Time Series:
Start = 85
End = 92
Frequency = 1
[1] 0.08905414 0.09347899 0.09770366 0.10175307 0.13548771 0.14370561
    0.15147833 0.15887122
```





Series djnj**Series dd**





3. Conditional Heteroskedastic Models

3.1 Characteristics of Volatility

What is the volatility of an asset?

Answer: Conditional standard deviation of the asset return (price).

Why is volatility important?

Has many important applications:

- Option (derivative) pricing, e.g., Black-Scholes formula
- Risk management, e.g. value at risk (VaR)
- Asset allocation, e.g., minimum-variance portfolio; see pages 184- 185 of Campbell, Lo and MacKinlay (1997).
- Interval forecasts

A key characteristic: Not directly observable!!

How to calculate volatility?

There are several versions of sample volatility, but the conditional standard

deviation is commonly used.

1. Use high-frequency data: French, Schwert Stambaugh (1987);
 - Realized volatility of daily log returns: use intraday high- frequency log returns.
 - Use daily high, low, and closing (log) prices, e.g. range = daily high - daily low.
2. Implied volatility of options data, e.g, VIX of CBOE.
3. Econometric modeling: use daily or monthly returns.

We focus on the econometric modeling first. Use of high frequency data to compute realized volatility will be discussed later.

Note: In most applications, volatility is annualized. This can easily be done by taking care of the data frequency. For instance, if we use daily returns in econometric modeling, then the annualized volatility is

$$\sigma_t^* = \sqrt{252}\sigma_t$$

where σ_t is the estimated volatility derived from an employed model.

If we use monthly returns, then the annualized volatility is

$$\sigma_t^* = \sqrt{12}\sigma_t$$

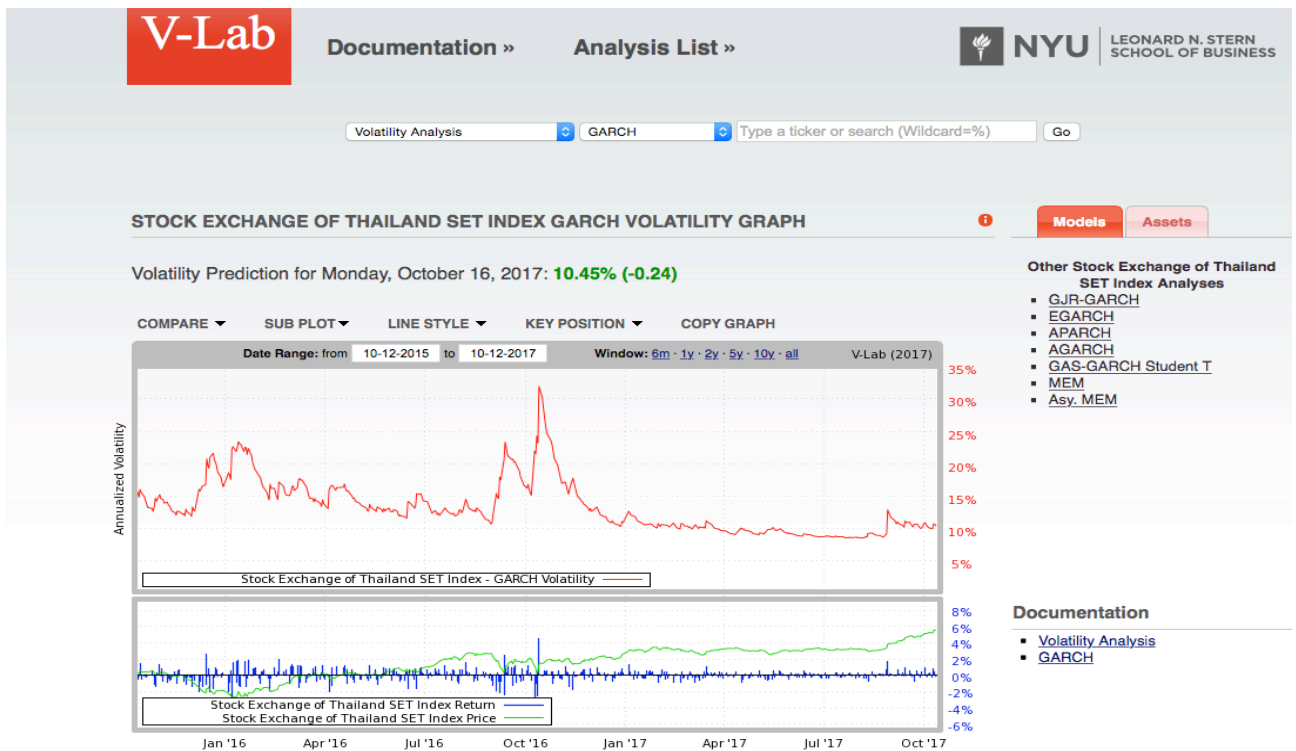
where σ_t is the estimated volatility derived from the employed model for the monthly returns.

3.2 Basic idea of econometric modeling:

Shocks of asset returns are NOT serially correlated, but dependent, implying that the serial dependence in asset returns is nonlinear. As shown by the ACF of returns and absolute returns of some assets we discussed so far.

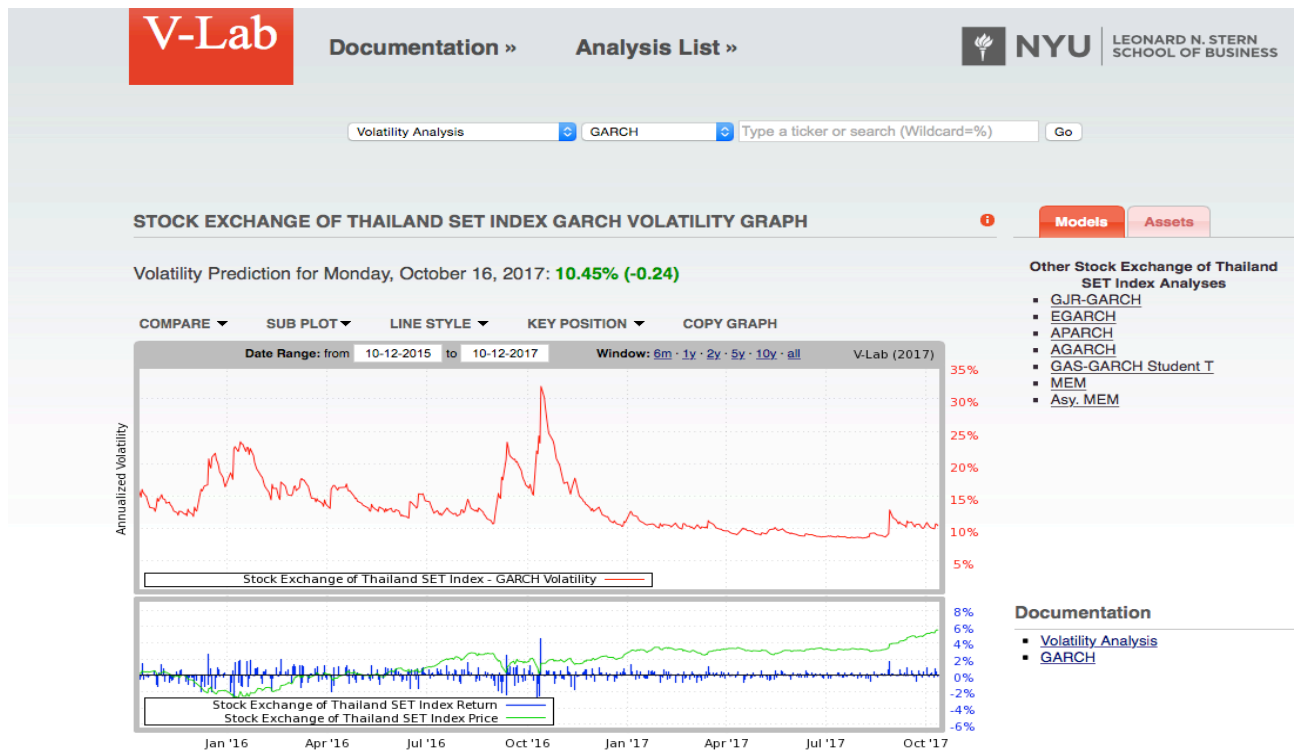
3.3 Basic Structure Volatility

$$r_t = \mu_t + a_t$$
$$\mu_t = \phi_0 + \sum_{i=1}^p \phi_i r_{t-i} - \sum_{i=1}^q \theta_i a_{t-i},$$
$$\sigma_t^2 = \text{Var}(r_t | F_{t-1}) = \text{Var}(a_t | F_{t-1})$$



The Application of Program R

```
#EE435
setwd("/Users/wasinsiwasarit/Desktop/EE435")
cat(rep("\n",50)) #clear R Console
require(quantmod)
getSymbols("^GSPC",from="2007-01-03",to="2017-10-06")
dim(GSPC)
head(GSPC)
spc <- log(as.numeric(GSPC[,6]))
rtn <- diff(spc)
acf(rtn)
m1 <- arima(rtn,order=c(0,0,2))
m1
plot(m1$residuals)
acf(m1$residuals)
acf(m1$residuals^2)
```



The Main Result of the Program R

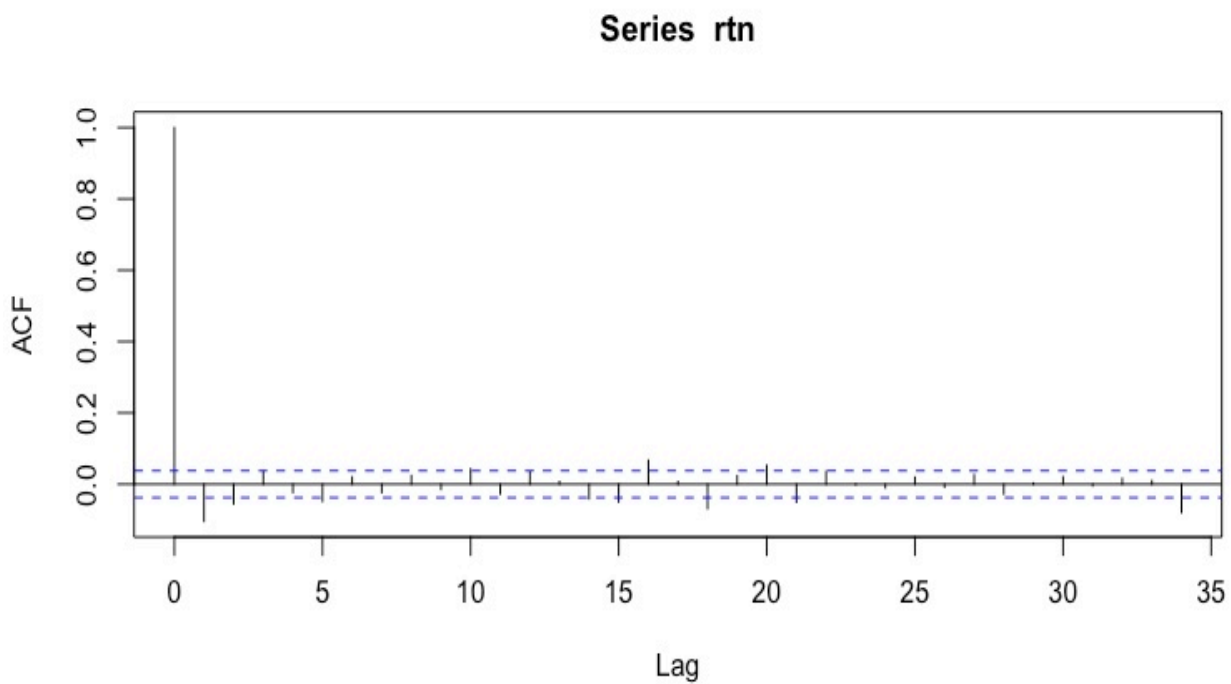
```
> require(quantmod)
> getSymbols("^GSPC", from="2007-01-03", to="2017-10-06")
[1] "GSPC"
> dim(GSPC)
[1] 2710    6
> head(GSPC)
      GSPC.Open GSPC.High GSPC.Low GSPC.Close GSPC.Volume GSPC.Adjusted
2007-01-03  1418.03   1429.42  1407.86   1416.60  3429160000    1416.60
2007-01-04  1416.60   1421.84  1408.43   1418.34  3004460000    1418.34
2007-01-05  1418.34   1418.34  1405.75   1409.71  2919400000    1409.71
2007-01-08  1409.26   1414.98  1403.97   1412.84  2763340000    1412.84
2007-01-09  1412.84   1415.61  1405.42   1412.11  3038380000    1412.11
2007-01-10  1408.70   1415.99  1405.32   1414.85  2764660000    1414.85
> spc <- log(as.numeric(GSPC[,6]))
> rtn <- diff(spc)
> acf(rtn)
> m1 <- arima(rtn, order=c(0,0,2))
> m1
```

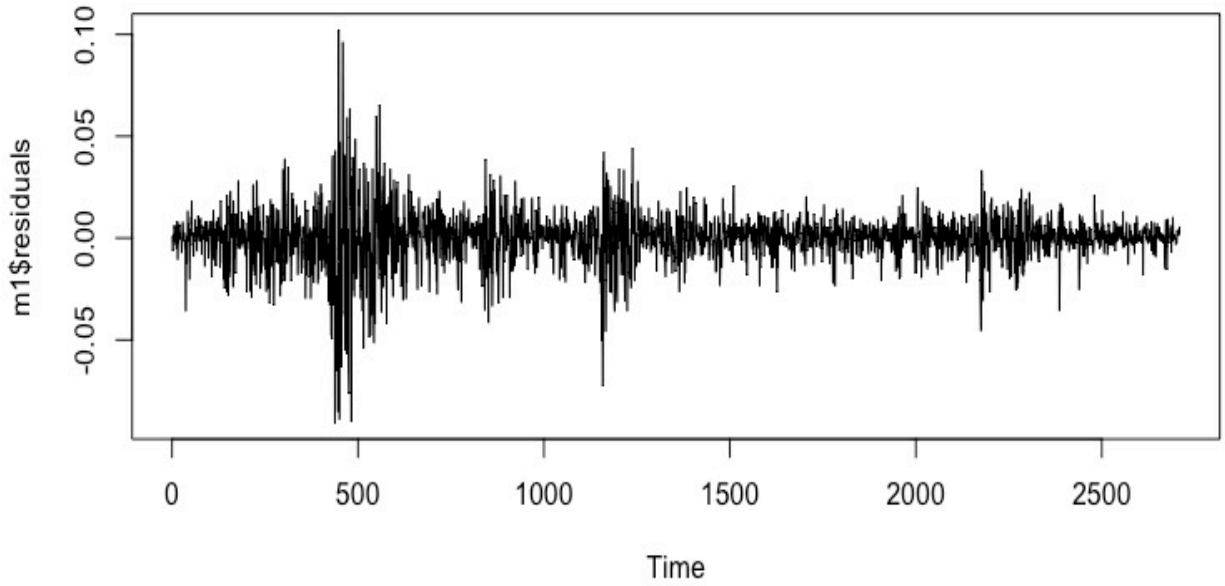
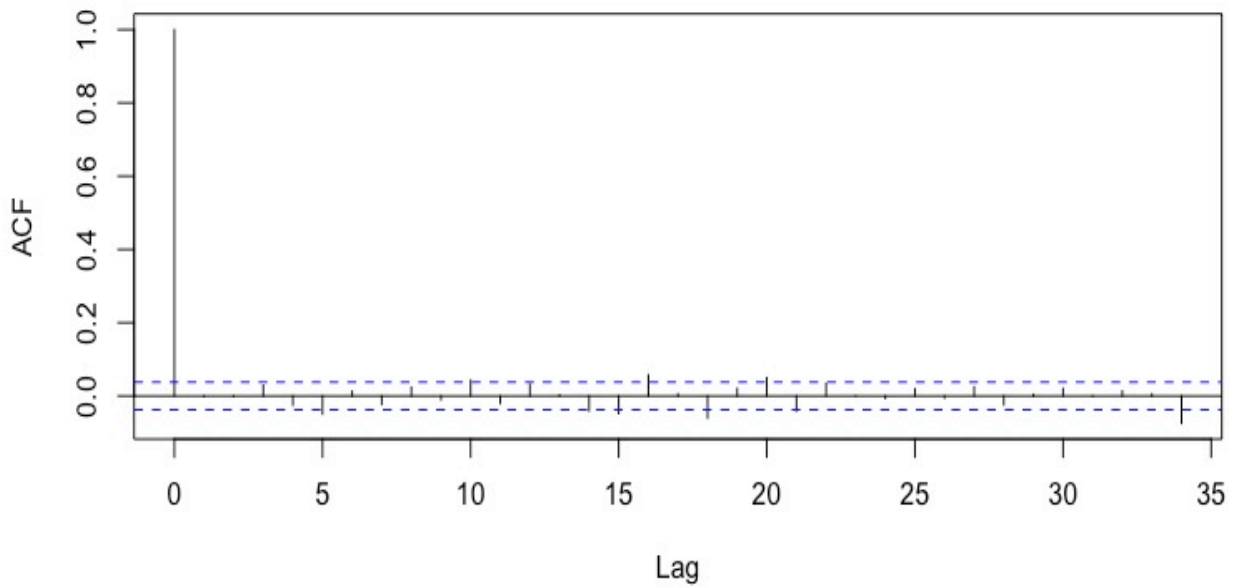
Call:

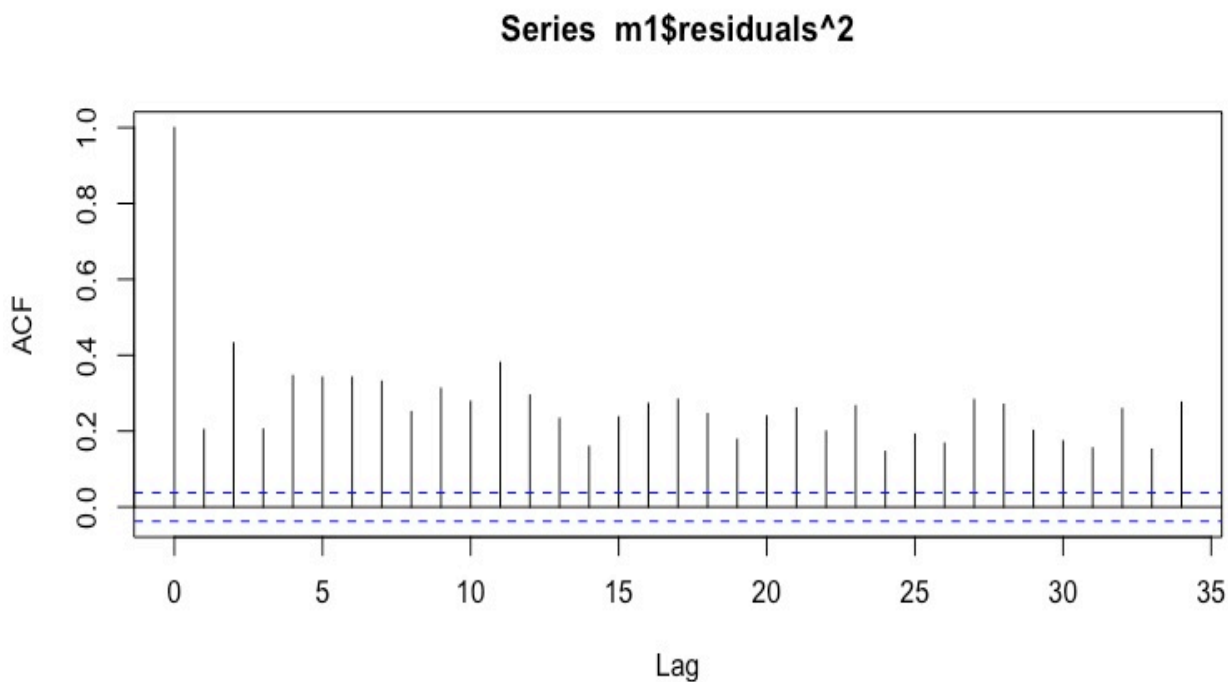
```
arima(x = rtn, order = c(0, 0, 2))

Coefficients:
      ma1      ma2  intercept
-0.1089 -0.0535      2e-04
s.e.    0.0193   0.0200      2e-04

sigma^2 estimated as 0.0001608:  log likelihood = 7988.28,  aic = -15968.56
> plot(m1$residuals)
> acf(m1$residuals)
> acf(m1$residuals^2)
```



**Series m1\$residuals**



3.4 Univariate Volatility Models Discussed:

1. Autoregressive Conditional Heteroskedastic (ARCH) model of Engle (1982)
 2. Generalized ARCH (GARCH) model of Bollerslev (1986)
 3. Exponential GARCH (EGARCH) model of Nelson (1991)
 4. IGARCH models (used by RiskMetrics)
 5. Exponential GARCH (EGARCH) model of Nelson (1991)
 6. Threshold GARCH model of Zakoian (1994) or GJR model of Glosten, Jagannathan, and Runkle (1993)
 7. Asymmetric power ARCH (APARCH) models of Ding, Granger and Engle (1994), [TGARCH and GJR models are special cases of APARCH models.]
 8. Stochastic volatility (SV) models of Melino and Turnbull (1990), Harvey, Ruiz and Shephard (1994), and Jacquier, Polson and Rossi (1994).

The first model in this class is ARCH Model.

3.4.1 ARCH model

$$r_t = \mu_t + a_t$$

$$\mu_t = \phi_0 + \sum_{i=1}^p \phi_i r_{t-i} - \sum_{i=1}^q \theta_i a_{t-i},$$

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

$$a_t = \sigma_t \epsilon_t$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 a_{t-1}^2 + \dots + \alpha_m a_{t-m}^2$$

where ϵ_t is a sequence of iid r.v. with mean 0 and variance 1, $\alpha_0 > 0$ and $\alpha_i \geq 0$ for $i > 0$

Consider an ARCH(1) model

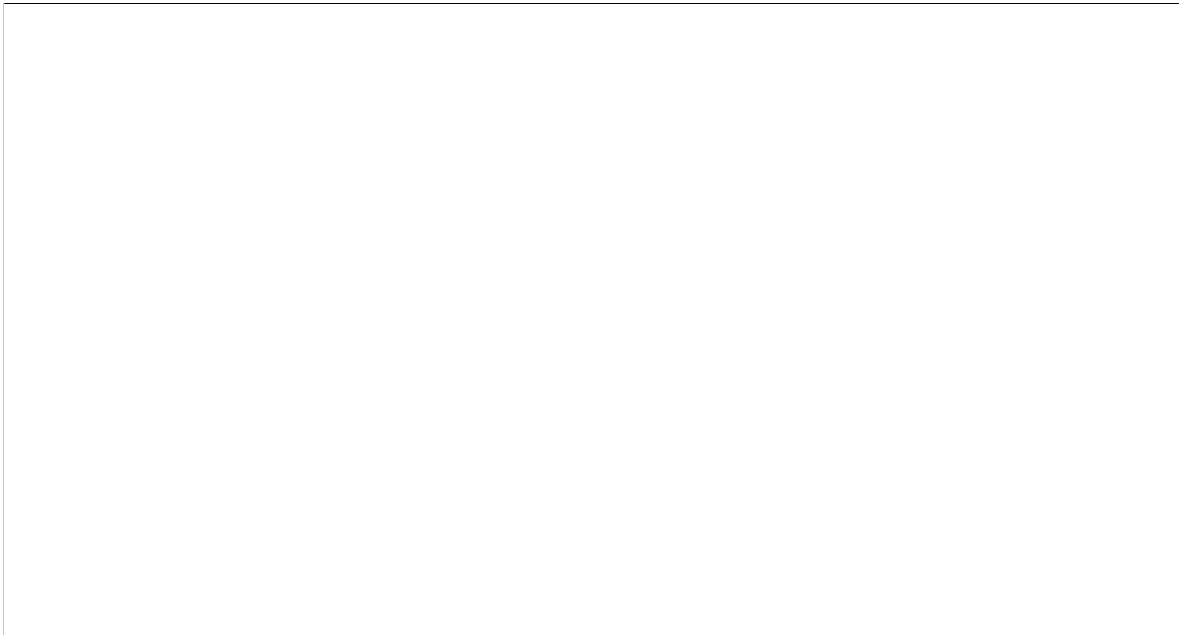
$$a_t = \sigma_t \epsilon_t$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 a_{t-1}^2$$

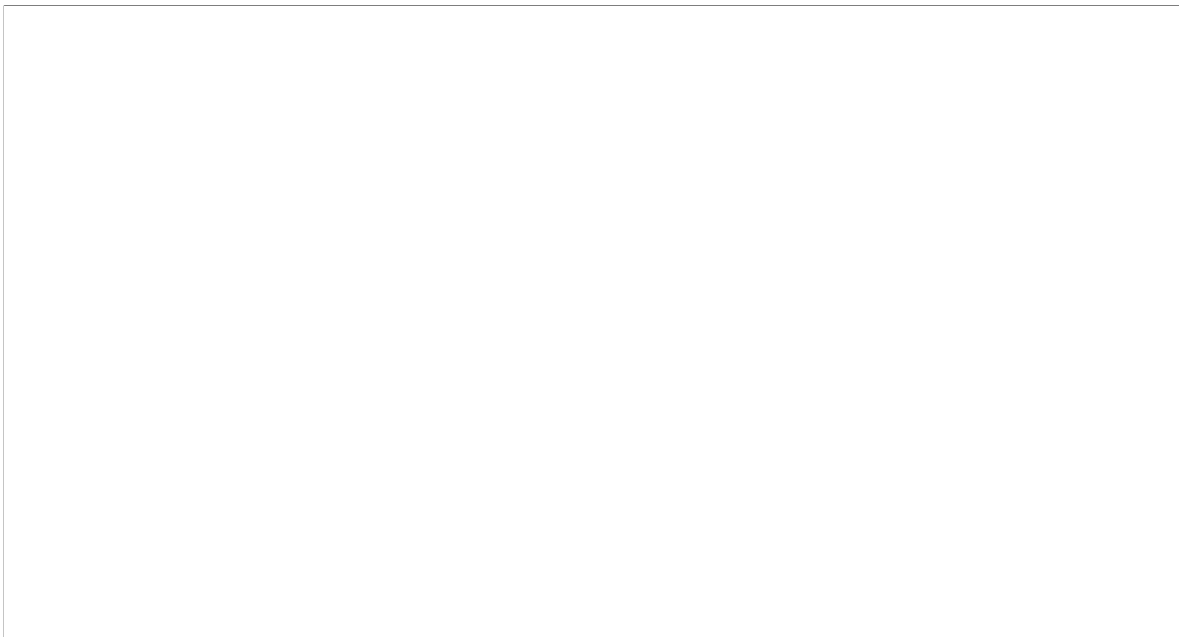
Properties of ARCH models)

1. The Uncondition Mean of a_t

2. The Unconditiona Variance of a_t



3. Under normality (m_4)



The 3rd property implies heavy tails.

Advantages

- Simplicity
- Generates volatility clustering • Heavy tails (high kurtosis)

Weaknesses

- Symmetric between positive negative prior returns • Restrictive
- Provides no explanation
- Not sufficiently adaptive in prediction

Building an ARCH Model

1. Modeling the mean effect and testing for ARCH effects

H_0 : no ARCH effects

H_a : ARCH effects

Use Q-statistics of squared residuals; McLeod and Li (1983) Engle (1982)

2. Order determination

Use PACF of the squared residuals. (In practice, simply try some reasonable order).

3. Estimation: Conditional MLE

4. Model checking: Q-stat of standardized residuals and squared standardized residuals. Skewness & Kurtosis of standardized residuals.

R provides many plots for model checking and for presenting the results.

5. Software: We use R with the package fGarch. (Other software available).

Estimation: Conditional MLE

Special Note: In this course, we estimate volatility models using the R package `fGarch` with `garchFit` command. The program is easy to use and allows for several types of innovational distributions: The default is Gaussian (norm), standardized Student-t distribution (std), generalized error distribution (ged), skew normal distribution (snorm), skew Student-t (sstd), skew generalized error distribution (sged), and standardized inverse normal distribution (snig). Except for the inverse normal distribution, other distribution functions are discussed in the textbook. Readers should check the book for details about the density functions and their parameters.

The Application of Program R: ARCH model

```
setwd("/Users/wasin_siwasarit/documents/EE435")
cat(rep("\n",50)) #clear R Console
install.packages("fGarch")
library(fGarch)
da=read.table("m-intc7303.txt",header=T)
head(da)
par(mfrow=c(1,1))
intc=log(da$rtn+1)
acf(intc)
acf(intc^2)
pacf(intc^2)
Box.test(intc^2,lag=10,type='Ljung')
m1=garchFit(~garch(3,0),data=intc,trace=F)
summary(m1)
m1=garchFit(~garch(1,0),data=intc,trace=F)
summary(m1)
plot(m1)
m2=garchFit(~garch(1,0),data=intc,cond.dist="std",trace=F)
summary(m2)
#plot(m2)
predict(m2,5)
```

The Main Results:

```
> install.packages("fGarch")
trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.4/fGarch_
  3010.82.1.tgz'
Content type 'application/x-gzip' length 425617 bytes (415 KB)
=====
downloaded 415 KB
```

The downloaded binary packages are in
 /var/folders/zv/z36cxqhj3jzfl135k41c8vg80000gn/T//RtmpiEqVwB/downloaded_
 packages

```
> library(fGarch)
Loading required package: timeDate
Loading required package: timeSeries
Loading required package: fBasics
```

```
Rmetrics Package fBasics
Analysing Markets and calculating Basic Statistics
Copyright (C) 2005-2014 Rmetrics Association Zurich
Educational Software for Financial Engineering and Computational Science
Rmetrics is free software and comes with ABSOLUTELY NO WARRANTY.
https://www.rmetrics.org --- Mail to: info@rmetrics.org
```

```
> da=read.table("m-intc7303.txt",header=T)
> head(da)
date      rtn
1 19730131  0.01005
2 19730228 -0.13930
3 19730330  0.06936
4 19730430  0.08649
5 19730531 -0.10448
6 19730629  0.13333
> par(mfrow=c(1,1))
> intc=log(da$rtn+1)
> acf(intc)
> acf(intc^2)
> pacf(intc^2)
> Box.test(intc^2,lag=10,type='Ljung')
```

Box-Ljung test

```
data: intc^2
X-squared = 59.722, df = 10, p-value = 4.091e-09
```

```

> m1=garchFit(~garch(3,0),data=intc,trace=F)
> summary(m1)

Title:
GARCH Modelling

Call:
garchFit(formula = ~garch(3, 0), data = intc, trace = F)

Mean and Variance Equation:
data ~ garch(3, 0)
<environment: 0x10446f068>
[data = intc]

Conditional Distribution:
norm

Coefficient(s):
mu      omega    alpha1    alpha2    alpha3
0.016572 0.012043 0.208649 0.071837 0.049045

Std. Errors:
based on Hessian

Error Analysis:
Estimate Std. Error t value Pr(>|t|)
mu      0.016572    0.006423    2.580 0.00988 **
omega   0.012043    0.001579    7.627 2.4e-14 ***
alpha1  0.208649    0.129177    1.615 0.10626
alpha2  0.071837    0.048551    1.480 0.13897
alpha3  0.049045    0.048847    1.004 0.31536
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log Likelihood:
233.4286    normalized: 0.6274962

Description:
Tue Oct 24 23:33:08 2017 by user:

Standardised Residuals Tests:
Statistic p-Value
Jarque-Bera Test R Chi^2 169.773 0
Shapiro-Wilk Test R W 0.960696 1.970626e-08
Ljung-Box Test R Q(10) 10.97025 0.3598405

```

```

Ljung-Box Test      R      Q(15)  19.59024  0.1882211
Ljung-Box Test      R      Q(20)  20.82192  0.40768
Ljung-Box Test      R^2    Q(10)   5.376602  0.8646439
Ljung-Box Test      R^2    Q(15)  22.7346   0.08993974
Ljung-Box Test      R^2    Q(20)  23.70577  0.255481
LM Arch Test        R      TR^2    20.48506  0.05844884

```

Information Criterion Statistics:

```

AIC      BIC      SIC      HQIC
-1.228111 -1.175437 -1.228466 -1.207193

```

```

> m1=garchFit(~garch(1,0),data=intc,trace=F)
> summary(m1)

```

```

Title:
GARCH Modelling

```

```

Call:
garchFit(formula = ~garch(1, 0), data = intc, trace = F)

```

Mean and Variance Equation:

```

data ~ garch(1, 0)
<environment: 0x10f737400>
[data = intc]

```

Conditional Distribution:

```
norm
```

Coefficient(s):

```

mu      omega    alpha1
0.01657 0.01249 0.36345

```

Std. Errors:

```
based on Hessian
```

Error Analysis:

```

Estimate Std. Error t value Pr(>|t|)
mu      0.016570    0.006161    2.689 0.00716 **
omega   0.012490    0.001549    8.061 6.66e-16 ***
alpha1  0.363447     0.131598    2.762 0.00575 **

```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Log Likelihood:

```
230.2423      normalized: 0.6189309
```

Description:

Tue Oct 24 23:33:14 2017 by user:

Standardised Residuals Tests:

Statistic p-Value

Jarque-Bera Test	R	Chi ²	122.404	0
Shapiro-Wilk Test	R	W	0.9647625	8.273101e-08
Ljung-Box Test	R	Q(10)	13.72604	0.1858587
Ljung-Box Test	R	Q(15)	22.31714	0.09975386
Ljung-Box Test	R	Q(20)	23.88257	0.2475594
Ljung-Box Test	R ²	Q(10)	12.50025	0.25297
Ljung-Box Test	R ²	Q(15)	30.11276	0.01152131
Ljung-Box Test	R ²	Q(20)	31.46404	0.04935483
LM Arch Test	R	TR ²	22.036	0.0371183

Information Criterion Statistics:

AIC	BIC	SIC	HQIC
-1.221733	-1.190129	-1.221861	-1.209182

> plot(m1)

Make a plot selection (or 0 to exit):

1: Time Series	2: Conditional SD
3: Series with 2 Conditional SD Superimposed	4: ACF of Observations
5: ACF of Squared Observations	6: Cross Correlation
7: Residuals	8: Conditional SDs
9: Standardized Residuals	10: ACF of Standardized Residuals
11: ACF of Squared Standardized Residuals	12: Cross Correlation between r ² and r
13: QQ-Plot of Standardized Residuals	

Selection: 13

Make a plot selection (or 0 to exit):

1: Time Series	2: Conditional SD
3: Series with 2 Conditional SD Superimposed	4: ACF of Observations
5: ACF of Squared Observations	6: Cross Correlation
7: Residuals	8: Conditional SDs
9: Standardized Residuals	10: ACF of Standardized Residuals
11: ACF of Squared Standardized Residuals	12: Cross Correlation between r ² and r

```

13:  QQ-Plot of Standardized Residuals

Selection: 2

Make a plot selection (or 0 to exit):

1:  Time Series                2:  Conditional SD
3:  Series with 2 Conditional SD Superimposed  4:  ACF of Observations
5:  ACF of Squared Observations  6:  Cross Correlation
7:  Residuals                  8:  Conditional SDs
9:  Standardized Residuals      10: ACF of Standardized
    Residuals
11: ACF of Squared Standardized Residuals  12: Cross Correlation
    between r^2 and r
13:  QQ-Plot of Standardized Residuals

Selection: 0
> m2=garchFit(~garch(1,0),data=intc,cond.dist="std",trace=F)
> summary(m2)

Title:
GARCH Modelling

Call:
garchFit(formula = ~garch(1, 0), data = intc, cond.dist = "std",
trace = F)

Mean and Variance Equation:
data ~ garch(1, 0)
<environment: 0x10f6e4630>
[data = intc]

Conditional Distribution:
std

Coefficient(s):
mu      omega    alpha1    shape
0.021571 0.013424 0.259867 5.985979

Std. Errors:
based on Hessian

Error Analysis:
Estimate Std. Error t value Pr(>|t|)
mu      0.021571  0.006054   3.563 0.000366 ***
omega   0.013424  0.001968   6.820 9.09e-12 ***

```

```

alpha1  0.259867    0.119901    2.167 0.030209 *
shape   5.985979    1.660030    3.606 0.000311 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log Likelihood:
242.9678    normalized:  0.6531391

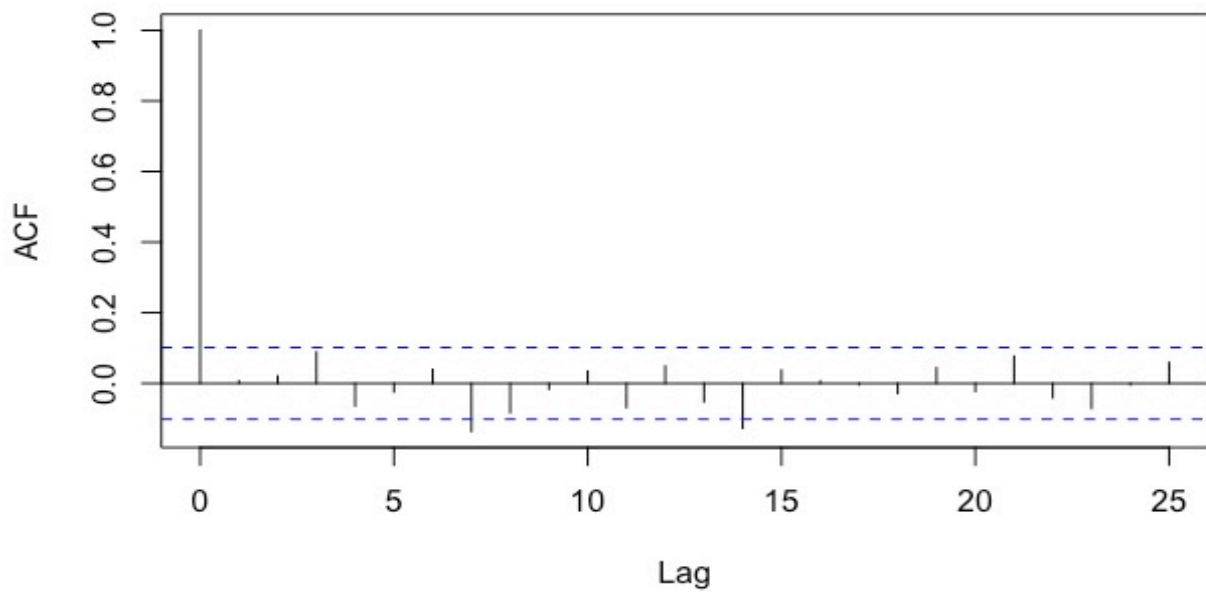
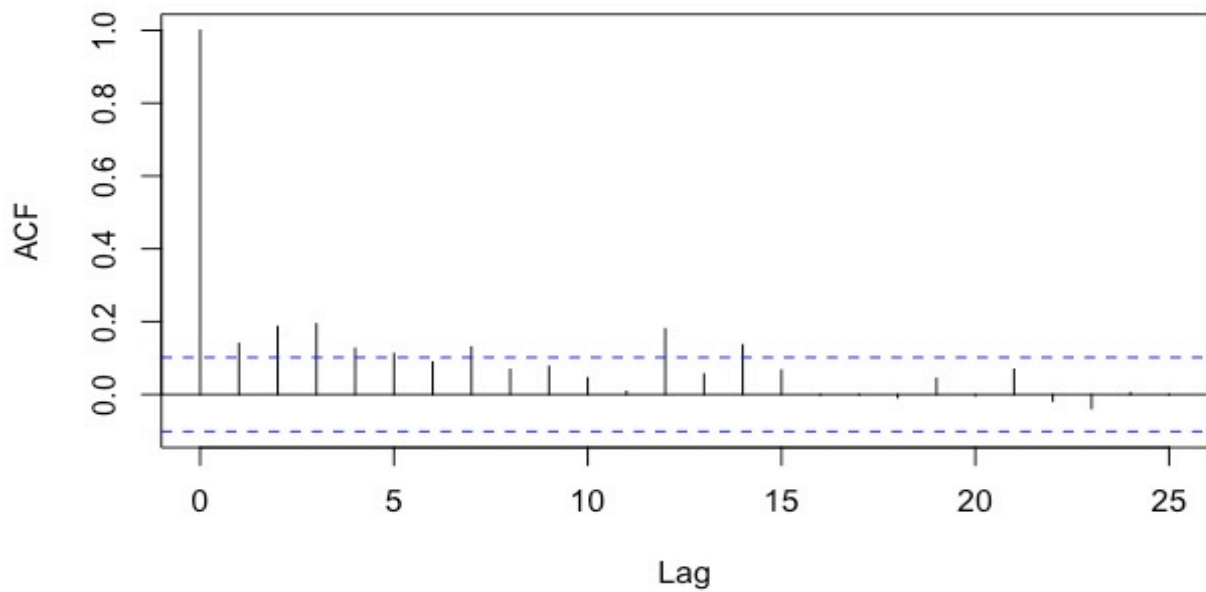
Description:
Tue Oct 24 23:35:37 2017 by user:

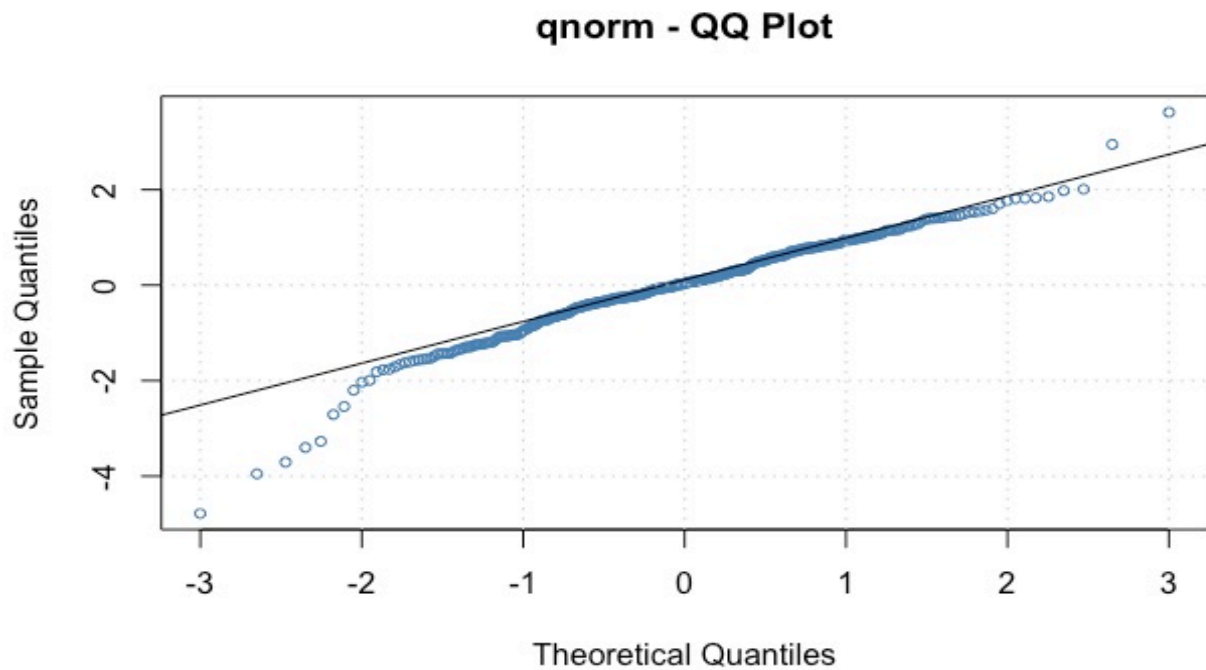
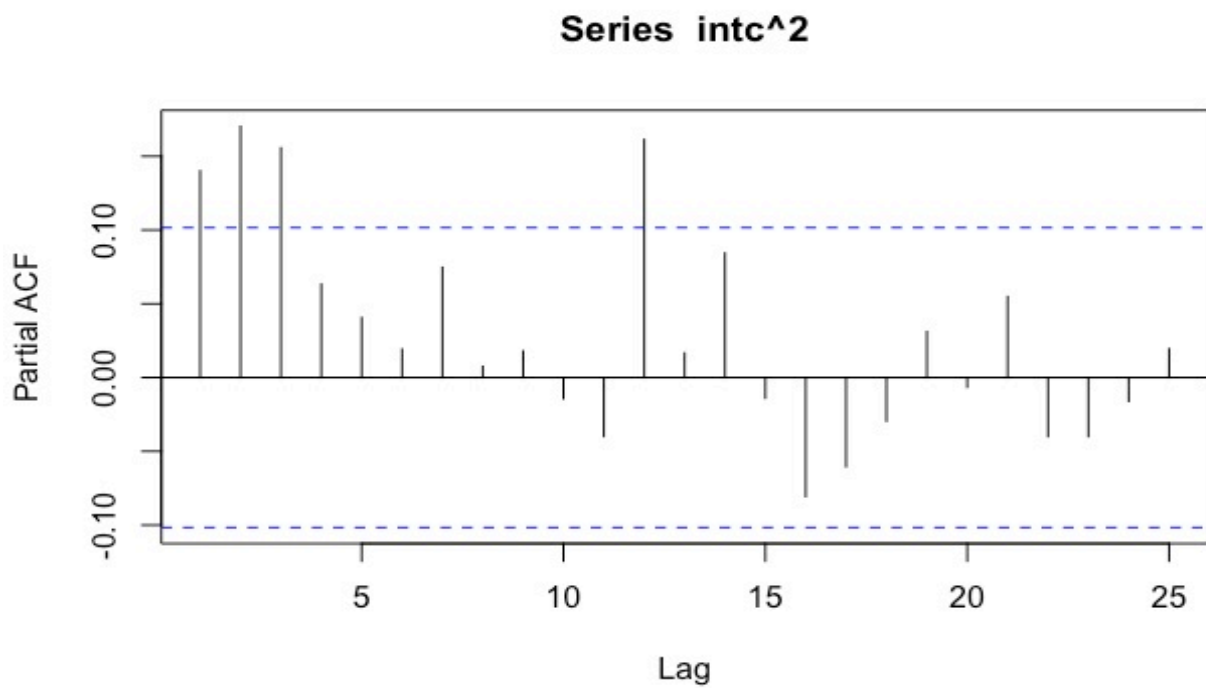
Standardised Residuals Tests:
Statistic p-Value
Jarque-Bera Test  R    Chi^2  130.8931  0
Shapiro-Wilk Test R    W      0.9637533 5.744995e-08
Ljung-Box Test   R    Q(10)  14.31288  0.1591926
Ljung-Box Test   R    Q(15)  23.34043  0.07717449
Ljung-Box Test   R    Q(20)  24.87286  0.2063387
Ljung-Box Test   R^2  Q(10)  15.35917  0.1195054
Ljung-Box Test   R^2  Q(15)  33.96318  0.003446127
Ljung-Box Test   R^2  Q(20)  35.46828  0.01774746
LM Arch Test     R    TR^2   24.11517  0.01961957

Information Criterion Statistics:
AIC      BIC      SIC      HQIC
-1.284773 -1.242634 -1.285001 -1.268039

> #plot(m2)
> predict(m2,5)
meanForecast meanError standardDeviation
1      0.021571 0.1207911          0.1207911
2      0.021571 0.1312069          0.1312069
3      0.021571 0.1337810          0.1337810
4      0.021571 0.1344418          0.1344418
5      0.021571 0.1346130          0.1346130

```

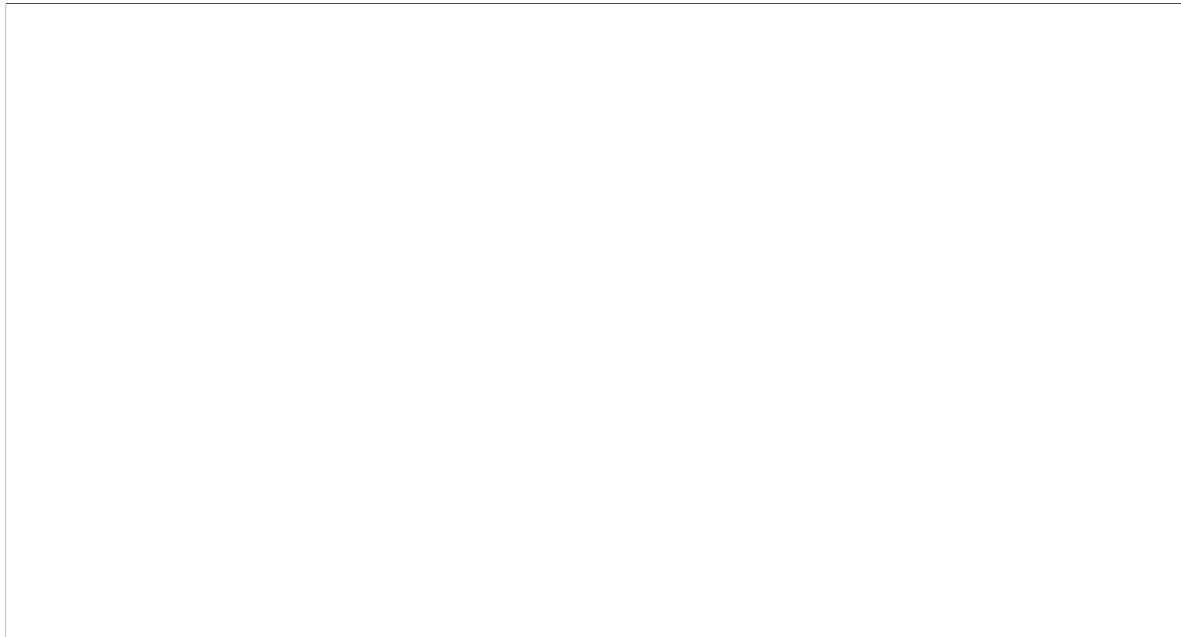
Series intc**Series intc^2**



3.4.2 GARCH Model

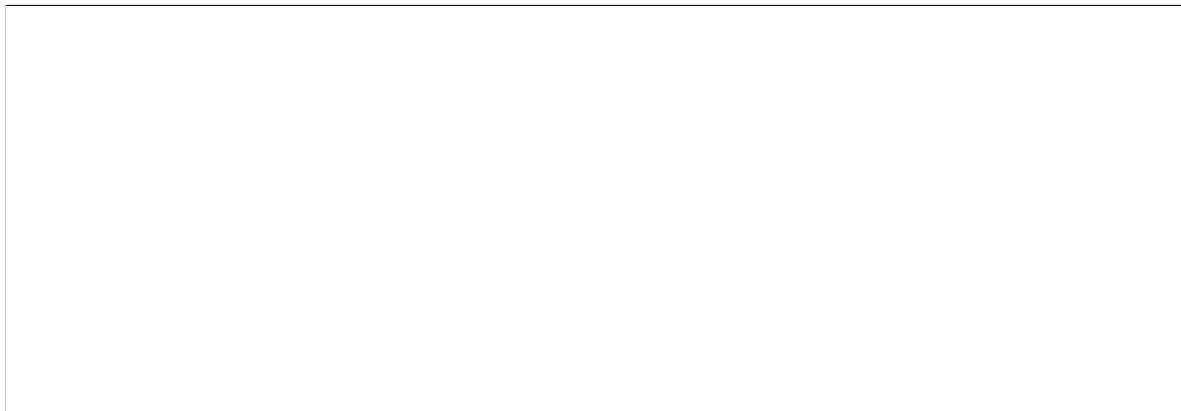
$$a_t = \sigma_t \epsilon_t$$
$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^m \alpha_i a_{t-i}^2 + \sum_{j=1}^s \beta_j \sigma_{t-j}^2$$

where ϵ_t is a sequence of iid r.v. with mean 0 and variance 1. with $\alpha_0 > 0$, $\alpha_i \geq 0$, $\beta_j \geq 0$ and $\sum_{i=1}^{\max(m,s)} (\alpha_i + \beta_i) < 1$.

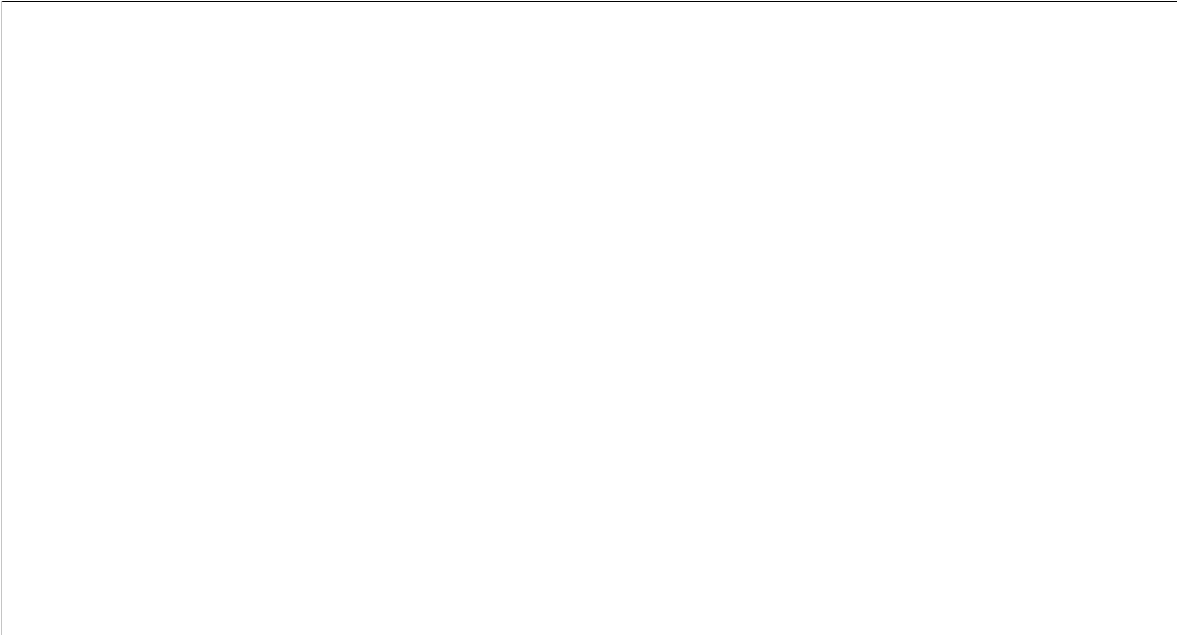


Properties of GARCH models

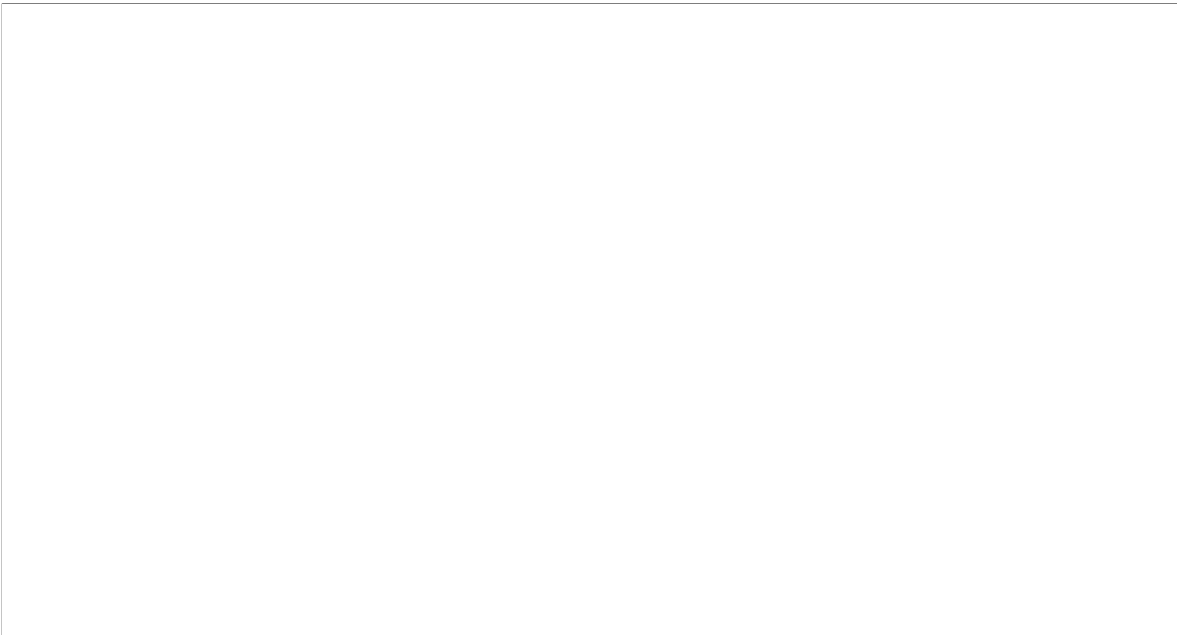
1. The Uncondition Mean of a_t



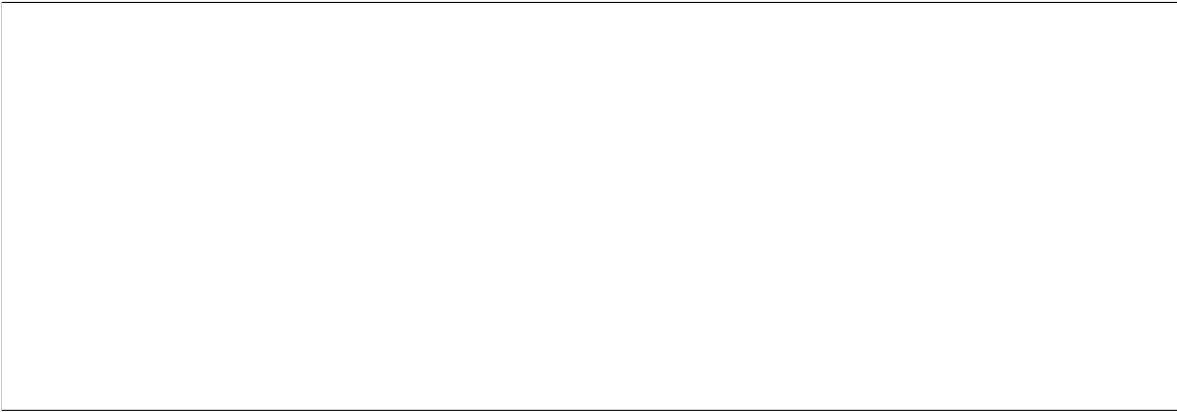
2. The Unconditiona Variance of a_t



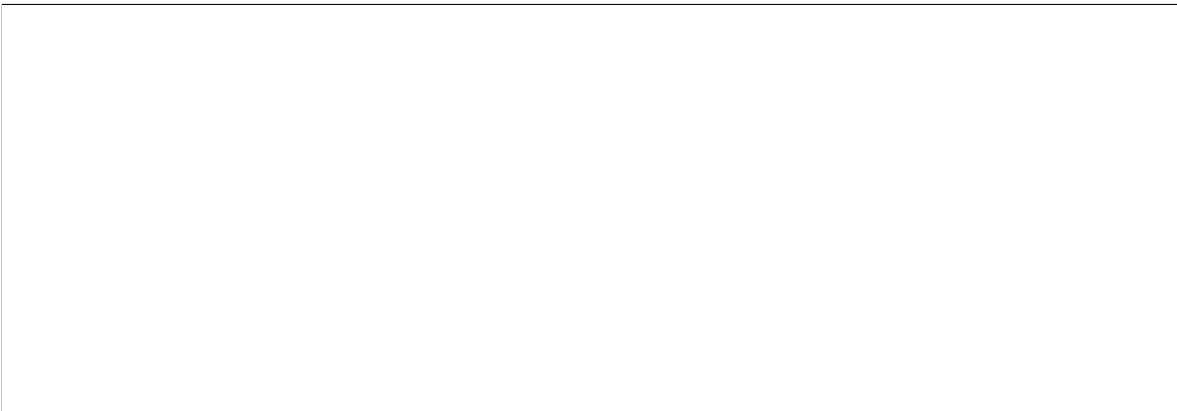
3. Under normality (m_4)



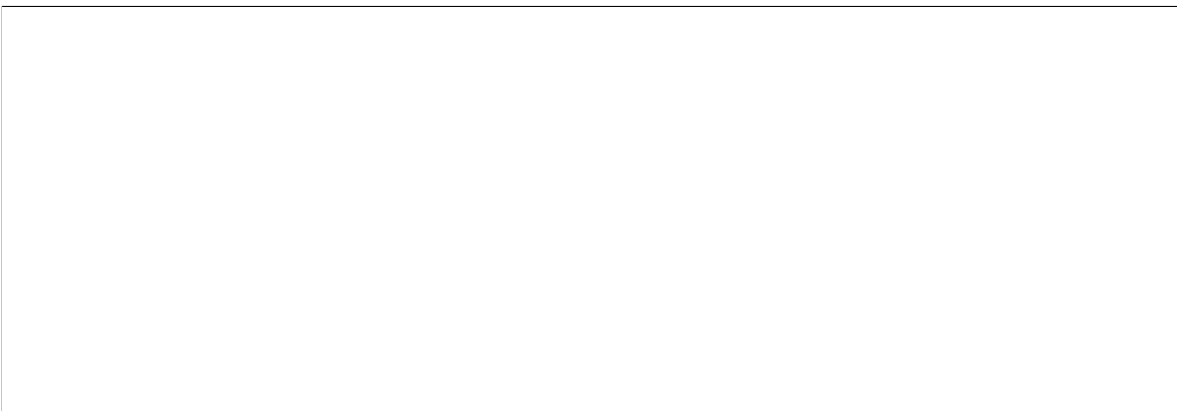
1-step ahead forecasting



2-step ahead forecasting



l-step ahead forecasting



The Application of Program R :GARCH model

```

setwd("/Users/wasin_siwasarit/documents/EE435")
cat(rep("\n",50)) #clear R Console
library(fGarch)
da <- read.table("sp500.txt")
sp <- da[,1]
ts.plot(sp)
pacf(sp)
m1 <- arima(sp,order=c(3,0,0))
acf(m1$residuals^2)
Box.test(m1$residuals^2,lag=10,type='Ljung')
m2 <- garchFit(~arma(3,0)+garch(1,1),data=sp,trace=F)
summary(m2)
plot(m2)
m3=garchFit(~garch(1,1),data=sp5,trace=F)
summary(m3)
predict(m3,6)

```

The Main Results of Program R :GARCH model

```

> library(fGarch)
> da <- read.table("sp500.txt")
> sp <- da[,1]
> ts.plot(sp)
> pacf(sp)
> m1 <- arima(sp,order=c(3,0,0))
> acf(m1$residuals^2)
> Box.test(m1$residuals^2,lag=10,type='Ljung')

Box-Ljung test

data:  m1$residuals^2
X-squared = 353.77, df = 10, p-value < 2.2e-16

> m2 <- garchFit(~arma(3,0)+garch(1,1),data=sp,trace=F)
> summary(m2)

Title:
GARCH Modelling

Call:
garchFit(formula = ~arma(3, 0) + garch(1, 1), data = sp, trace = F)

Mean and Variance Equation:

```

```

data ~ arma(3, 0) + garch(1, 1)
<environment: 0x110bdb950>
[data = sp]

Conditional Distribution:
norm

Coefficient(s):
mu          ar1          ar2          ar3          omega          alpha1
          beta1
7.7077e-03  3.1968e-02  -3.0261e-02  -1.0649e-02  7.9746e-05
          1.2425e-01  8.5302e-01

Std. Errors:
based on Hessian

Error Analysis:
Estimate Std. Error t value Pr(>|t|)
mu       7.708e-03  1.607e-03  4.798 1.61e-06 ***
ar1      3.197e-02  3.837e-02  0.833 0.40473
ar2     -3.026e-02  3.841e-02  -0.788 0.43076
ar3     -1.065e-02  3.756e-02  -0.284 0.77677
omega    7.975e-05  2.810e-05  2.838 0.00454 **
alpha1   1.242e-01  2.247e-02  5.529 3.22e-08 ***
beta1    8.530e-01  2.183e-02  39.075 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log Likelihood:
1272.179    normalized:  1.606287

Description:
Tue Oct 31 23:26:26 2017 by user:

Standardised Residuals Tests:
Statistic p-Value
Jarque-Bera Test  R    Chi^2  73.04843  1.110223e-16
Shapiro-Wilk Test R    W      0.9857968  5.961501e-07
Ljung-Box Test   R    Q(10)  11.56744  0.315048
Ljung-Box Test   R    Q(15)  17.78747  0.2740039
Ljung-Box Test   R    Q(20)  24.11916  0.2372257
Ljung-Box Test   R^2  Q(10)  10.31614  0.4132086
Ljung-Box Test   R^2  Q(15)  14.22819  0.5082976
Ljung-Box Test   R^2  Q(20)  16.79405  0.6663036
LM Arch Test     R    TR^2   13.34305  0.3446072

```

```
Information Criterion Statistics:
AIC      BIC      SIC      HQIC
-3.194897 -3.153581 -3.195051 -3.179018

> plot(m2)

Make a plot selection (or 0 to exit):

1:  Time Series                                2:  Conditional SD
3:  Series with 2 Conditional SD Superimposed  4:  ACF of
    Observations
5:  ACF of Squared Observations              6:  Cross
    Correlation
7:  Residuals                                8:  Conditional SDs
9:  Standardized Residuals                  10:  ACF of
    Standardized Residuals
11:  ACF of Squared Standardized Residuals  12:  Cross
    Correlation between r^2 and r
13:  QQ-Plot of Standardized Residuals

Selection: 3

Make a plot selection (or 0 to exit):

1:  Time Series                                2:  Conditional SD
3:  Series with 2 Conditional SD Superimposed  4:  ACF of
    Observations
5:  ACF of Squared Observations              6:  Cross
    Correlation
7:  Residuals                                8:  Conditional SDs
9:  Standardized Residuals                  10:  ACF of
    Standardized Residuals
11:  ACF of Squared Standardized Residuals  12:  Cross
    Correlation between r^2 and r
13:  QQ-Plot of Standardized Residuals

Selection: 0
> m3=garchFit(~garch(1,1),data=sp,trace=F)
> summary(m3)

Title:
GARCH Modelling

Call:
garchFit(formula = ~garch(1, 1), data = sp, trace = F)
```

```

Mean and Variance Equation:
data ~ garch(1, 1)
<environment: 0x11128cb80>
[data = sp]

Conditional Distribution:
norm

Coefficient(s):
mu      omega      alpha1      beta1
7.4497e-03  8.0615e-05  1.2198e-01  8.5436e-01

Std. Errors:
based on Hessian

Error Analysis:
Estimate Std. Error t value Pr(>|t|)
mu      7.450e-03  1.538e-03  4.845 1.27e-06 ***
omega   8.061e-05  2.833e-05  2.845 0.00444 **
alpha1  1.220e-01  2.202e-02  5.540 3.02e-08 ***
beta1   8.544e-01  2.175e-02  39.276 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log Likelihood:
1269.455    normalized:  1.602848

Description:
Tue Oct 31 23:26:36 2017 by user:

Standardised Residuals Tests:
Statistic p-Value
Jarque-Bera Test  R    Chi^2  80.32111  0
Shapiro-Wilk Test R    W      0.98505  3.136885e-07
Ljung-Box Test   R    Q(10)  11.2205  0.340599
Ljung-Box Test   R    Q(15)  17.99703 0.262822
Ljung-Box Test   R    Q(20)  24.29896 0.2295768
Ljung-Box Test   R^2  Q(10)  9.920157 0.4475259
Ljung-Box Test   R^2  Q(15)  14.21124 0.509572
Ljung-Box Test   R^2  Q(20)  16.75081 0.6690903
LM Arch Test     R    TR^2   13.04872 0.3655092

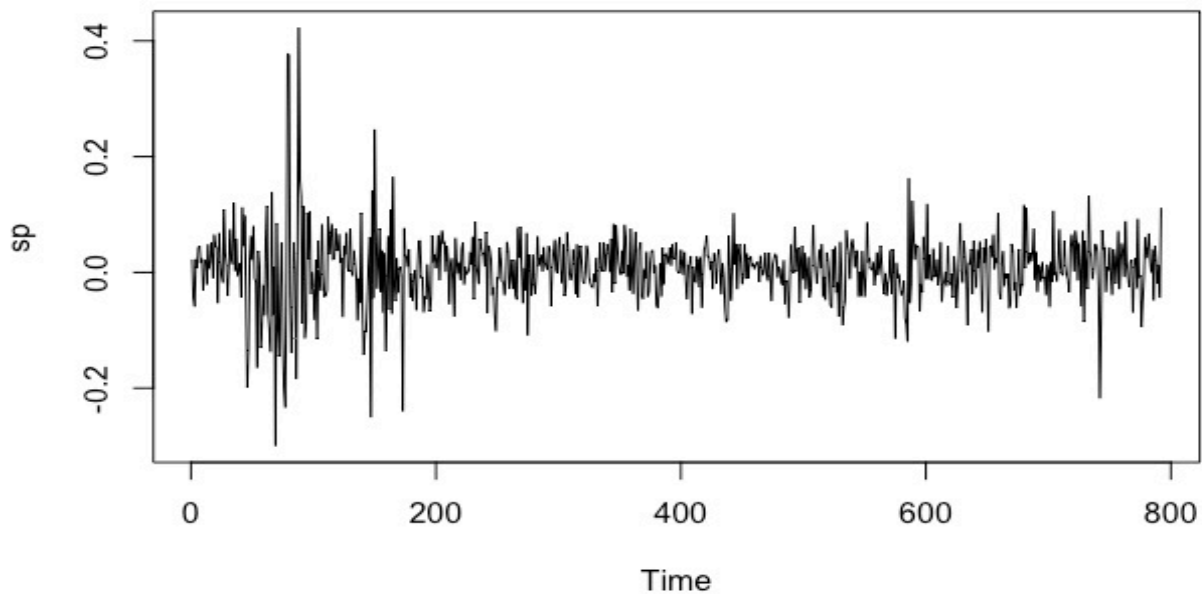
Information Criterion Statistics:
AIC      BIC      SIC      HQIC

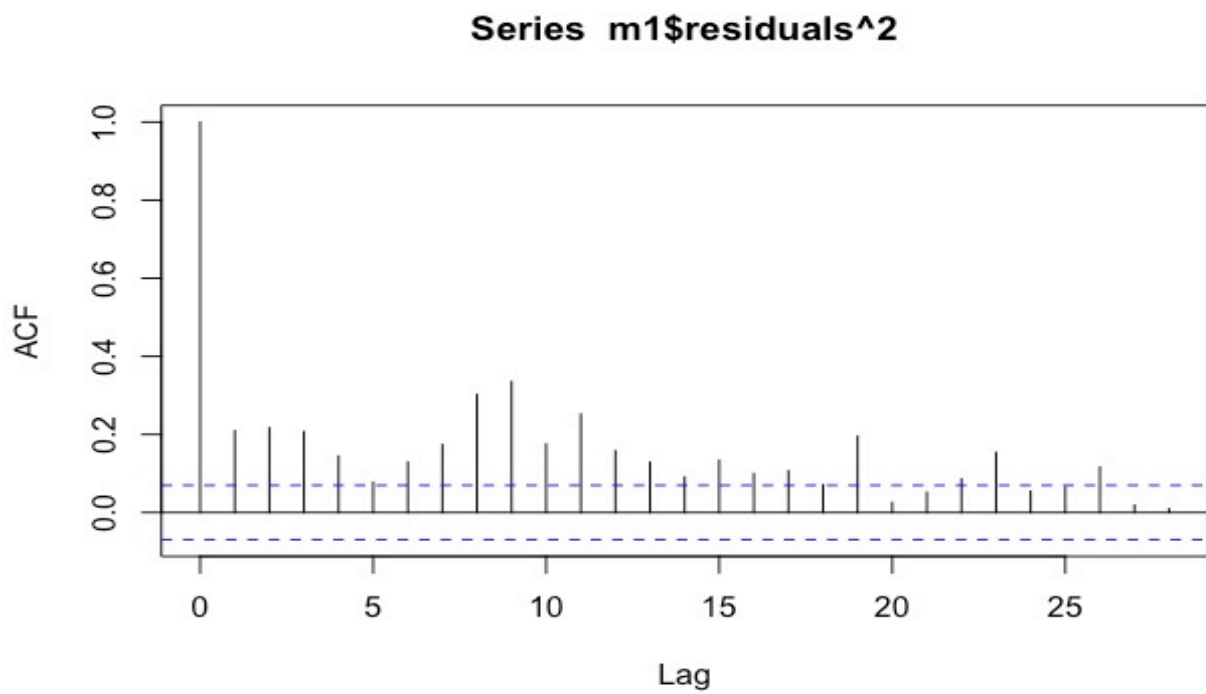
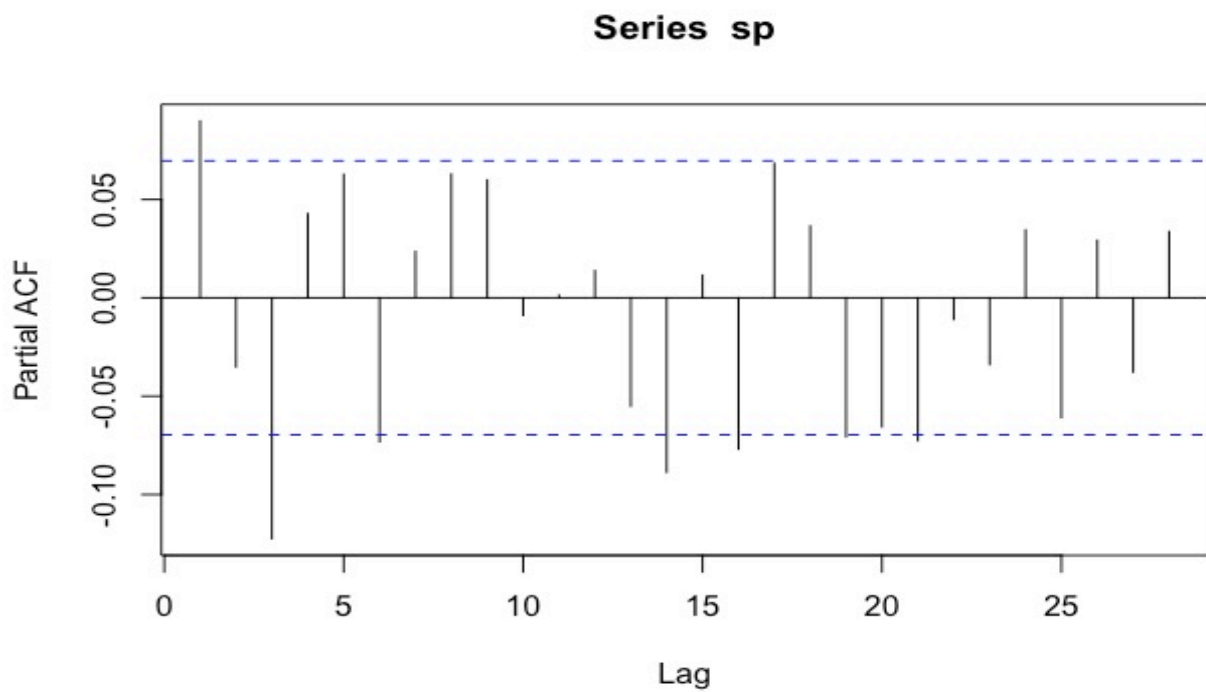
```

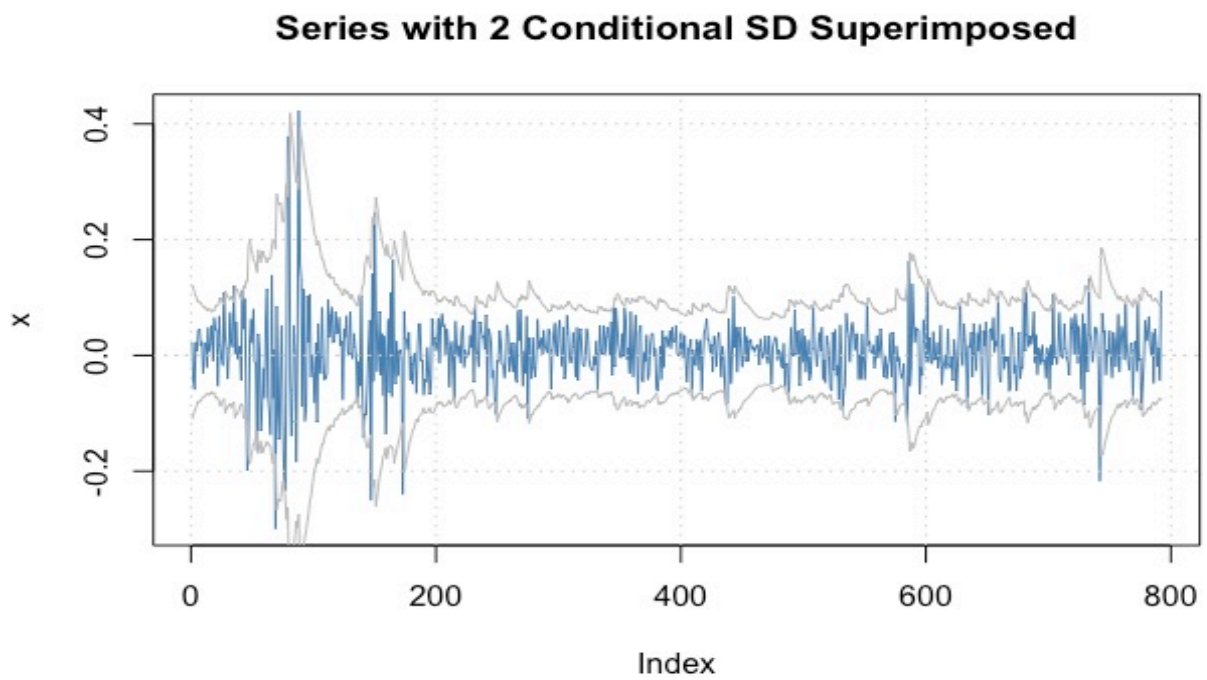
```
-3.195594 -3.171985 -3.195645 -3.186520
```

```
> predict(m3,6)
```

	meanForecast	meanError	standardDeviation
1	0.007449721	0.05377242	0.05377242
2	0.007449721	0.05388567	0.05388567
3	0.007449721	0.05399601	0.05399601
4	0.007449721	0.05410353	0.05410353
5	0.007449721	0.05420829	0.05420829
6	0.007449721	0.05431039	0.05431039









4. Multiple Financial Time Series with Applications

We shall focus on two series.

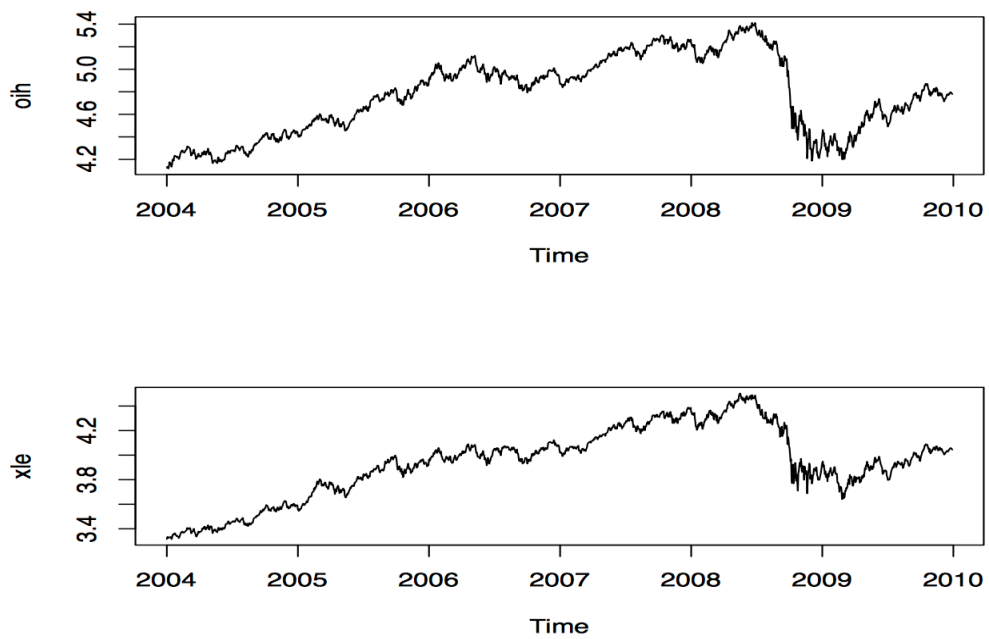
$$\mathbf{X}_t = \begin{bmatrix} x_{1t} \\ x_{2t} \end{bmatrix}$$

Some examples: (a) U.S. quarterly GDP and unemployment rate series; (b) The daily closing prices of oil related ETFs, e.g. oil services holdings (OIH) and energy select sector SPDR (XLE); and, for more than 2 series, (c) quarterly GDP growth rates of Canada, United Kingdom, and United States.

Why consider two series jointly?

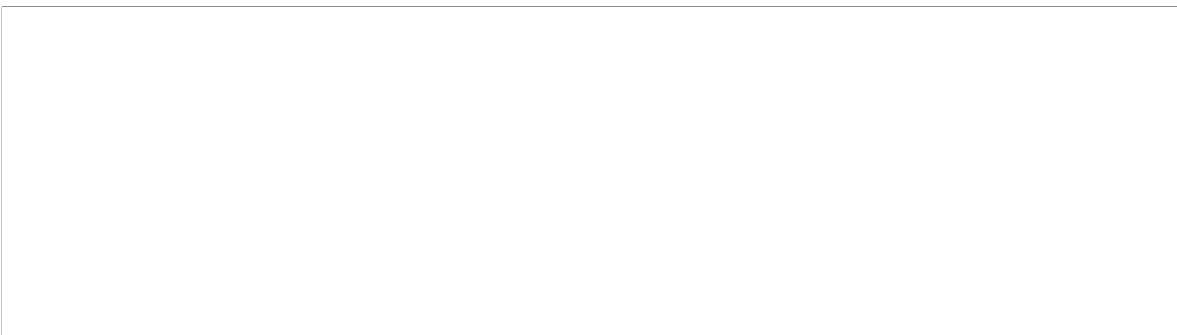
(a) Obtain the relationship between the series and (b) improve the accuracy of forecasts (use more information). See Figure 1 for the log prices of the two energy funds. The prices seem to move in unison.

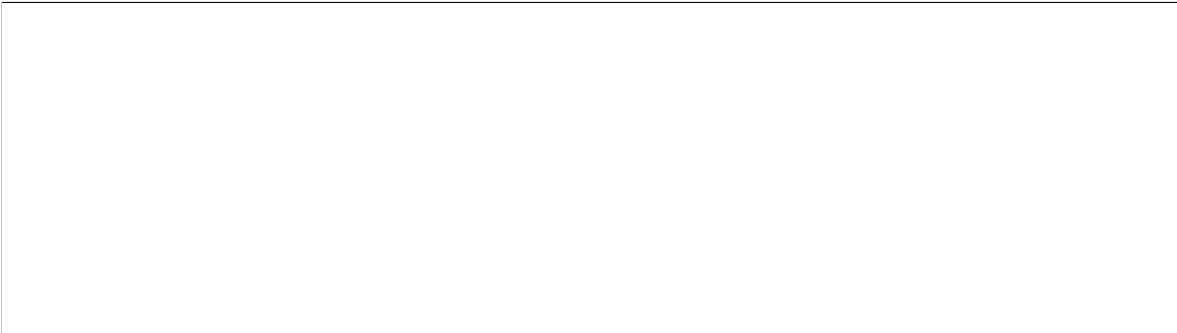
Figure1: Daily log prices of OIH and XLE funds from January 2004 to December 2009



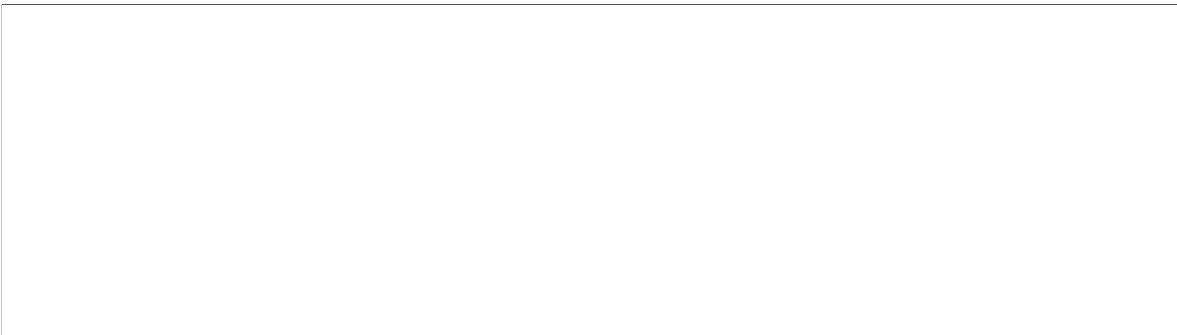
Some background:
Weak stationarity: Both

$E(X_t)$



$\text{Cov}(X_t, X_{t-j})$ 

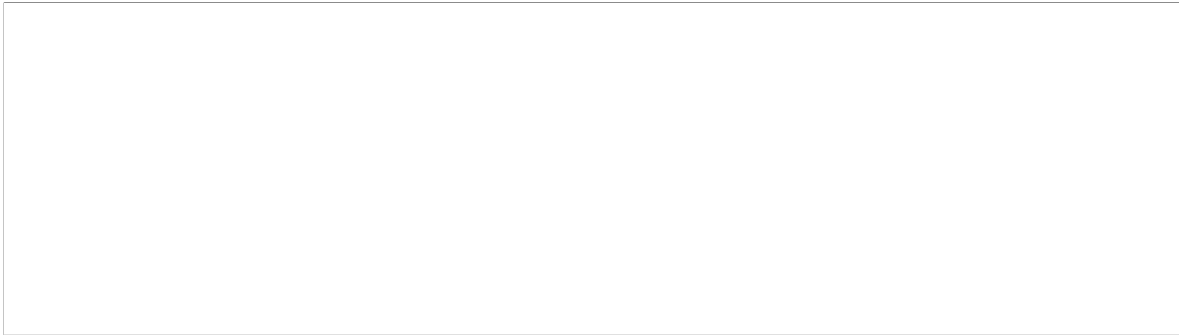
Auto-covariance matrix: Lag-l



Not symmetric if $l \neq 0$



Let the diagonal matrix D be



Cross-Correlation Matrix:

$$\rho_l = D^{-1} \Gamma D^{-1}$$

Thus, $\rho_{ij}(l)$ is the cross-correlation between x_{it} and $x_{j,t-l}$

For instance, $\text{cor}(x_{1t}, x_{2,t-1}) = \text{cor}(x_{2t}, x_{1,t+1})$

Testing for serial dependence

Multivariate version of Ljung-Box $Q(m)$ statistics

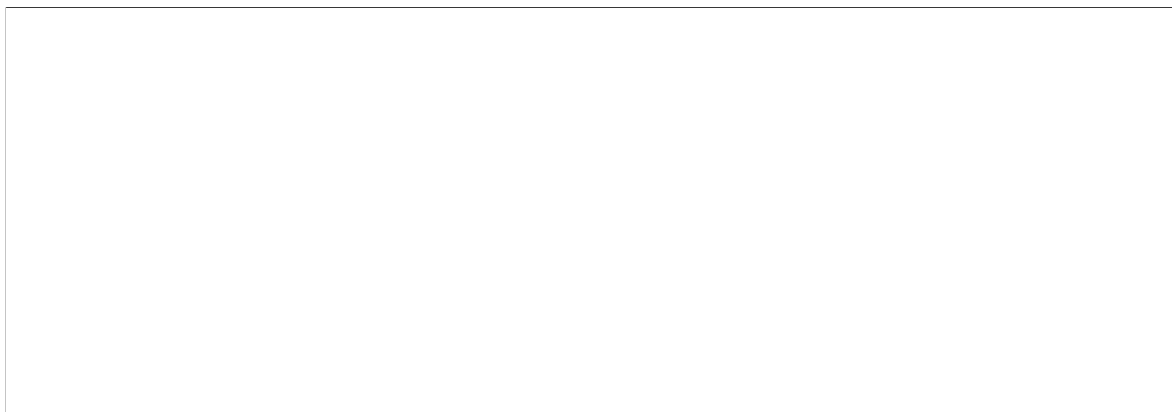
$$H_0 : \rho_1 = \dots = \rho_m = 0$$

vs.

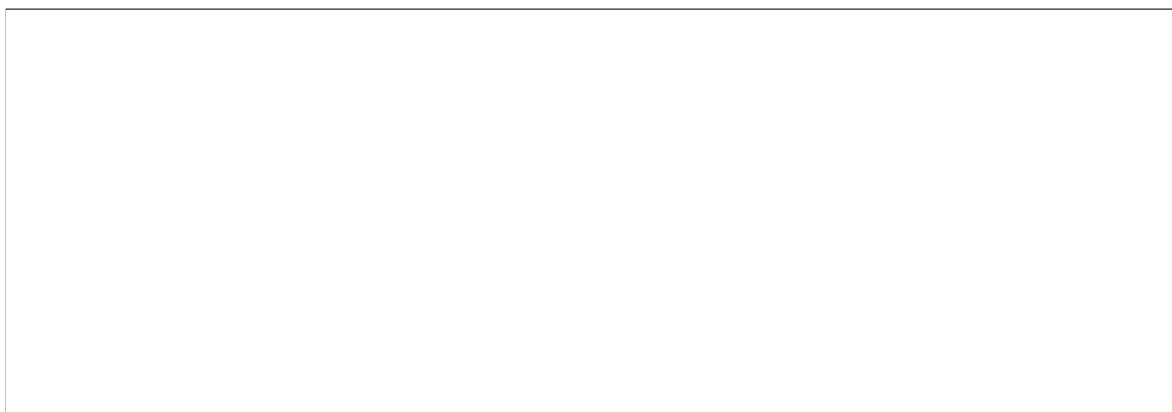
$$H_a : \rho_i \neq 0$$

4.1 Vector Autoregressive Model (VAR)

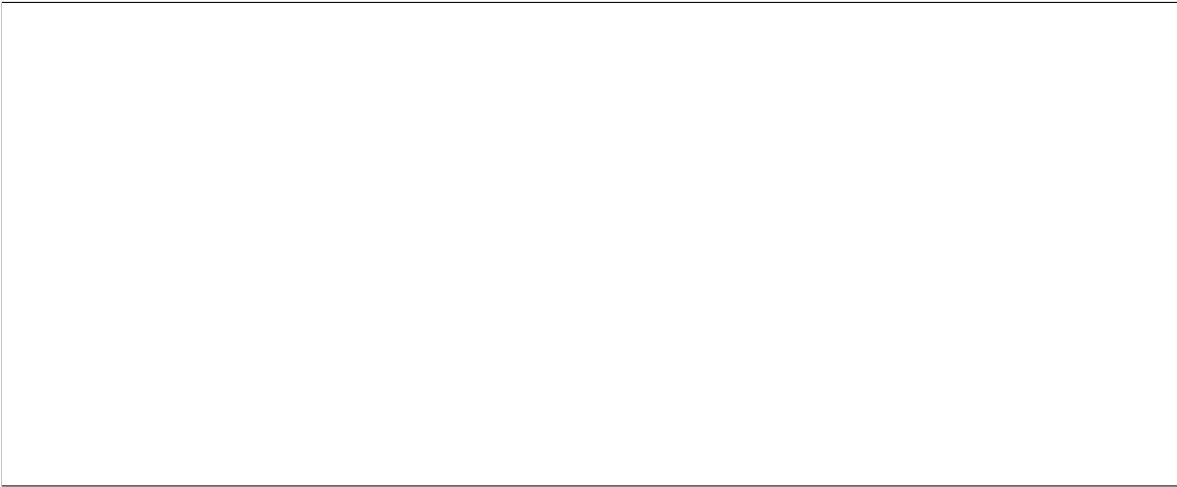
VAR(1) model for two return series:



where $\mathbf{a}_t = (a_{1t}, a_{2t})'$ is a sequence of iid bivariate normal random vectors with mean zero and covariance matrix

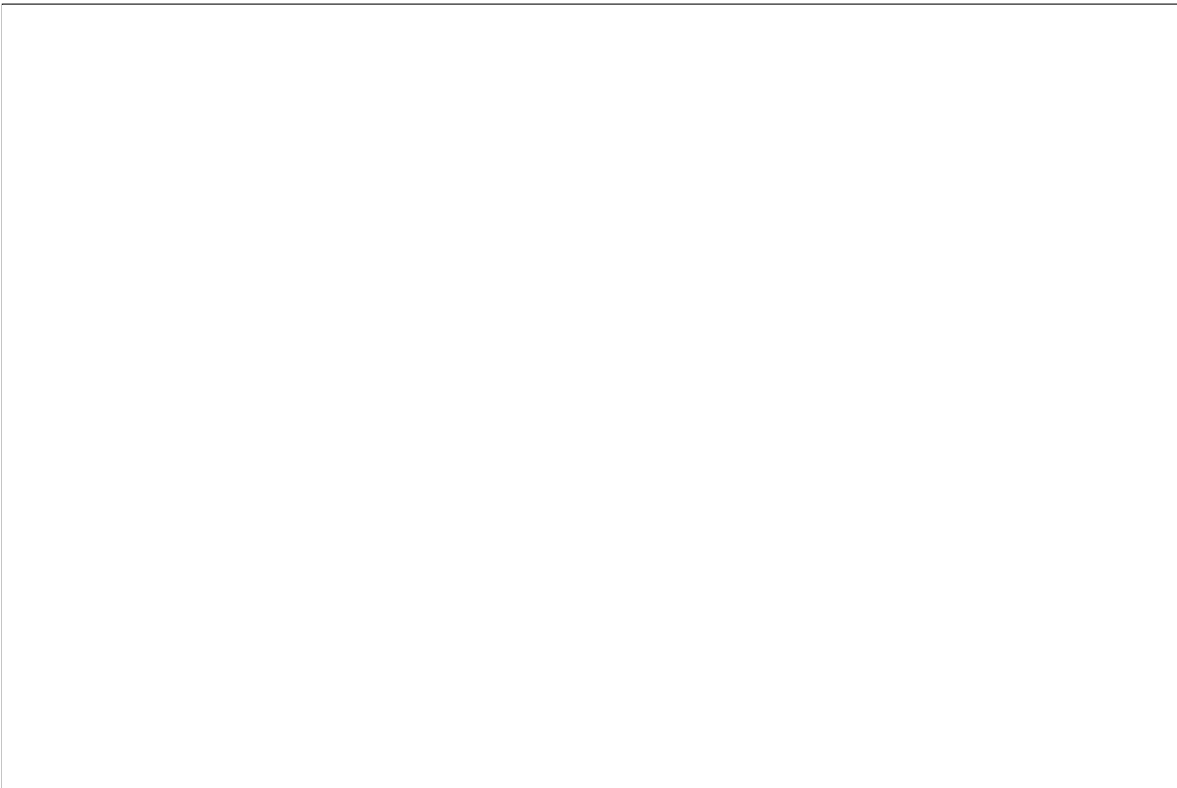


Rewrite the model as:



The above equations are called the Standard form or Reduced-form

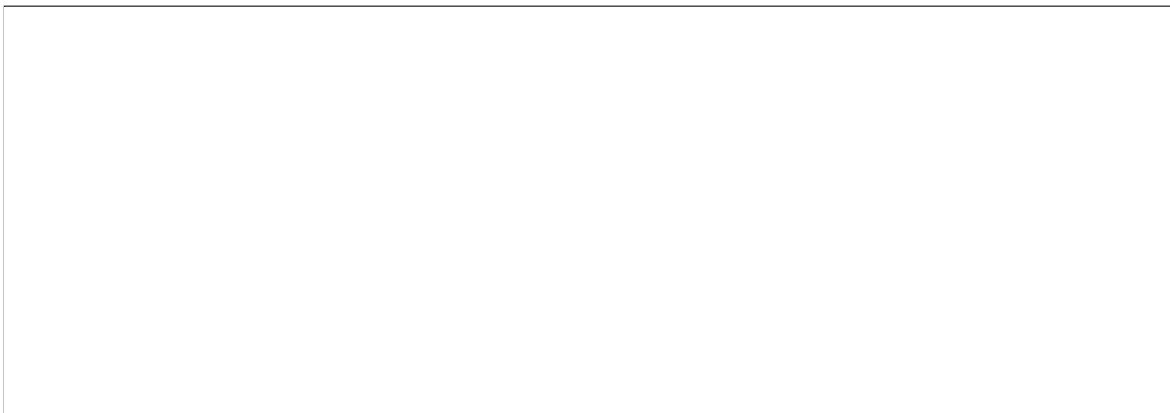
Based on the Standard form, we can re-write it to be the new form which is called the Structural equations Model.



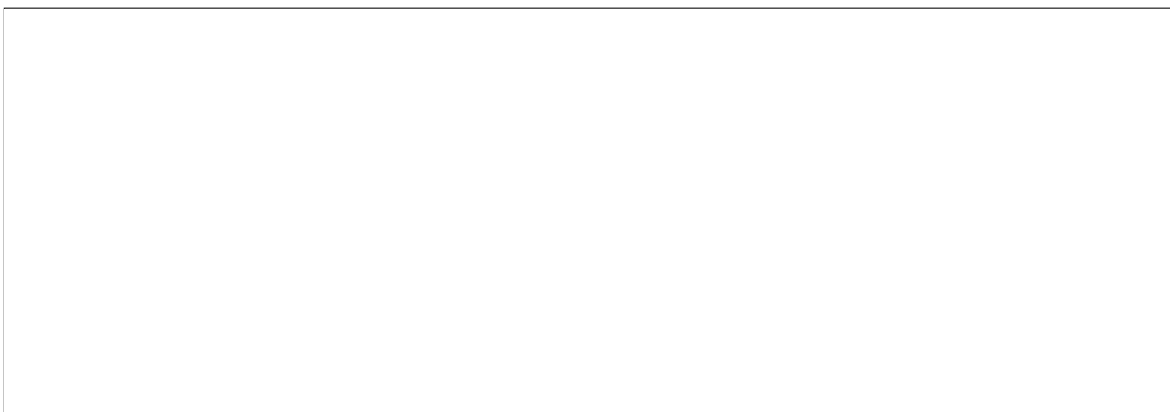
It is clearly seen that the term $\mathbf{a}_t = (a_{1t}, a_{2t})$ are depended on the shocks $\epsilon_{1t}, \epsilon_{2t}$.

Properties of a_{1t}, a_{2t}

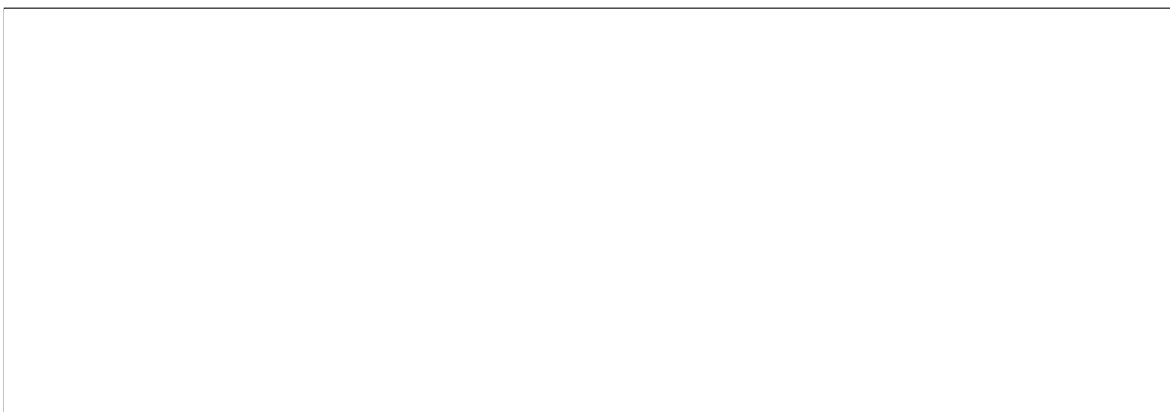
Expected Value a_{1t} :



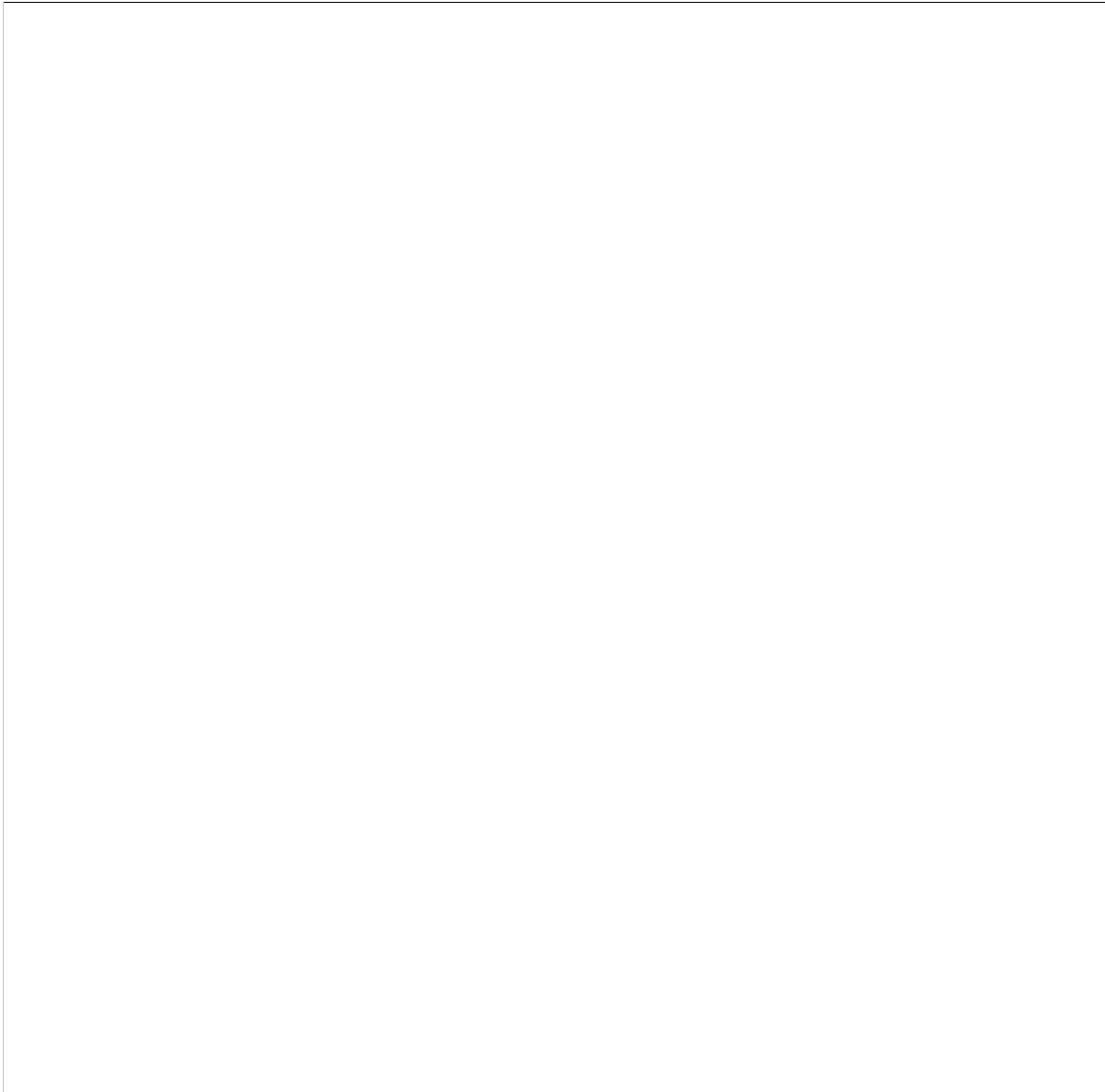
Variance a_{1t} :



Covariance ระหว่าง a_{1t} และ a_{2t} :



4.2 Stationarity of The VAR Model



4.3 Estimation and Model Checking

For a specified VAR model, one can estimate the parameters using either the ordinary least squares method or the maximum likelihood method.

A fitted model should then be checked carefully for any possible inadequacy. The $Q_k(m)$ statistic can be applied to the residual series to check the assumption that there are no serial or cross-correlations in the residuals. For a fitted VAR(p) model, the $Q_k(m)$ statistic of the residuals is asymptotically a chi-squared distribution with $k^2m - g$ degrees of freedom, where g is the number of estimated parameters in the AR coefficient matrixes

4.4 Forecasting



4.5 Identification



4.6 Identificationn (continued)



4.7 The Impulse Response Function



4.8 The Impulse Response Function (Cont.)



4.9 Variance Decomposition



4.10 Variance Decomposition (Cont.)



Consider the U.S monthly personal consumption expenditures (PCE) and disposable personal income (DSPi) from January 1959 to March 2012. The data are from FRED of the Federal Reserve Bank of St. Louis, in billions of dollars and seasonally adjusted.

Program R for the VAR model

```
setwd("/Users/wasin_siwasarit/documents/EE435")
cat(rep("\n",50)) #clear R Console
install.packages('MTS')
require(MTS)
##Example1
da1=read.table("m-pce.txt",header=T)
da2=read.table("m-dspi.txt",header=T)
head(da1)
head(da2)
x=cbind(da1$pce,da2$dspi)
x=log(x)
zt=diffM(x)*100
colnames(zt) <- c("pceg","dspig")
tdx=da1[,1]+da1[,2]/12
dim(da1)
MTSplot(zt,tdx[2:639])
VARorder(zt)
m1=VAR(zt,3)
MTSdiag(m1)
VARpred(m1,4)
```

Program R for the VAR model

```
> require(MTS)
> ##Example1
> da1=read.table("m-pce.txt",header=T)
> da2=read.table("m-dspi.txt",header=T)
> head(da1)
year mon day    pce
1 1959  1  1 306.7
2 1959  2  1 310.2
3 1959  3  1 313.3
4 1959  4  1 312.8
5 1959  5  1 316.7
6 1959  6  1 318.8
> head(da2)
year mon day  dspi
1 1959  1  1 340.7
```

```

2 1959  2  1 342.4
3 1959  3  1 344.8
4 1959  4  1 347.7
5 1959  5  1 350.0
6 1959  6  1 352.2
> x=cbind(da1$pce,da2$dspi)
> x=log(x)
> zt=diffM(x)*100
> colnames(zt) <- c("pceg","dspig")
> tdx=da1[,1]+da1[,2]/12
> dim(da1)
[1] 639  4
> MTSplot(zt,tdx[2:639])
> VARorder(zt)
selected order: aic = 8
selected order: bic = 3
selected order: hq = 3
Summary table:
p      AIC      BIC      HQ      M(p) p-value
[1,]  0 -1.8494 -1.8494 -1.8494  0.0000  0.0000
[2,]  1 -1.9011 -1.8731 -1.8902 39.9049  0.0000
[3,]  2 -1.9306 -1.8747 -1.9089 26.0294  0.0000
[4,]  3 -2.0291 -1.9452 -1.9965 68.5648  0.0000
[5,]  4 -2.0386 -1.9268 -1.9952 13.5685  0.0088
[6,]  5 -2.0485 -1.9087 -1.9942 13.7797  0.0080
[7,]  6 -2.0544 -1.8867 -1.9893 11.2718  0.0237
[8,]  7 -2.0544 -1.8588 -1.9785  7.6679  0.1045
[9,]  8 -2.0546 -1.8310 -1.9678  7.7194  0.1024
[10,] 9 -2.0506 -1.7991 -1.9530  5.1918  0.2682
[11,]10 -2.0398 -1.7603 -1.9313  1.0229  0.9063
[12,]11 -2.0430 -1.7355 -1.9236  9.4609  0.0506
[13,]12 -2.0418 -1.7064 -1.9116  6.8356  0.1448
[14,]13 -2.0423 -1.6789 -1.9013  7.7812  0.0999
> m1=VAR(zt,3)
Constant term:
Estimates:  0.4003634 0.3960725
Std.Error:  0.04660595 0.05873444
AR coefficient matrix
AR( 1 )-matrix
[,1]  [,2]
[1,] -0.153  0.130
[2,]  0.151 -0.194
standard error
[,1]  [,2]
[1,]  0.0409 0.0319
[2,]  0.0515 0.0402

```

```

AR( 2 )-matrix
[,1] [,2]
[1,] 0.00526 0.126
[2,] 0.19778 -0.124
standard error
[,1] [,2]
[1,] 0.0411 0.0326
[2,] 0.0518 0.0411
AR( 3 )-matrix
[,1] [,2]
[1,] 0.0531 0.128
[2,] 0.3486 -0.100
standard error
[,1] [,2]
[1,] 0.0403 0.0322
[2,] 0.0508 0.0406

Residuals cov-mtx:
[,1] [,2]
[1,] 0.2947970 0.1088635
[2,] 0.1088635 0.4681942

det(SSE) = 0.126171
AIC = -2.0325
BIC = -1.948644
HQ = -1.999946
> MTSdiag(m1)
[1] "Covariance matrix:"
pceg dspig
pceg 0.295 0.109
dspig 0.109 0.469
CCM at lag: 0
[,1] [,2]
[1,] 1.000 0.293
[2,] 0.293 1.000
Simplified matrix:
CCM at lag: 1
. .
. .
CCM at lag: 2
. .
. .
CCM at lag: 3
. .
. .
CCM at lag: 4

```

```
. +  
. .  
CCM at lag: 5  
. .  
+ .  
CCM at lag: 6  
+ .  
. .  
CCM at lag: 7  
+ .  
. .  
CCM at lag: 8  
. .  
+ .  
CCM at lag: 9  
. .  
. .  
CCM at lag: 10  
. .  
. .  
CCM at lag: 11  
+ .  
+ .  
CCM at lag: 12  
. .  
+ +  
CCM at lag: 13  
. .  
. .  
CCM at lag: 14  
. .  
. .  
CCM at lag: 15  
. .  
. .  
CCM at lag: 16  
. +  
. .  
CCM at lag: 17  
. .  
. .  
CCM at lag: 18  
. .  
+ .  
CCM at lag: 19  
. +
```

```

. .
CCM at lag: 20
+ .
. .
CCM at lag: 21
. +
. .
CCM at lag: 22
. .
. .
CCM at lag: 23
. .
. .
CCM at lag: 24
. .
. .
Hit Enter for p-value plot of individual ccm:

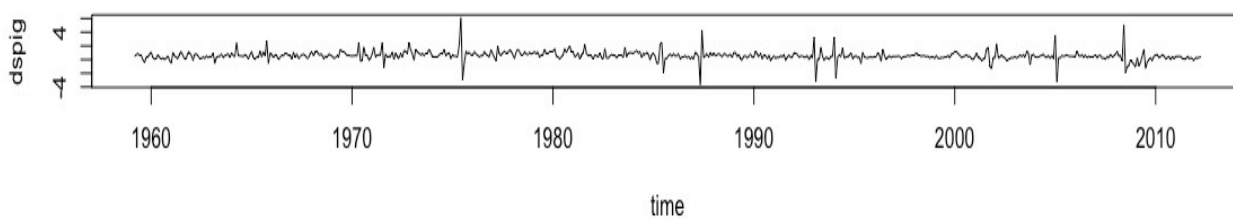
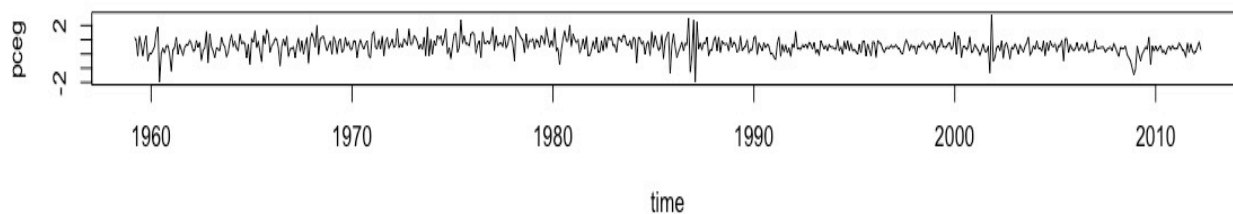
```

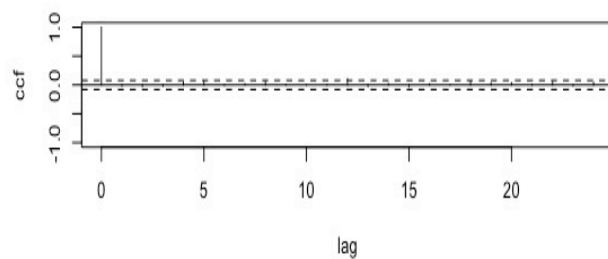
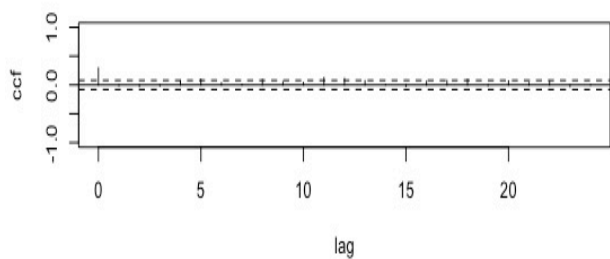
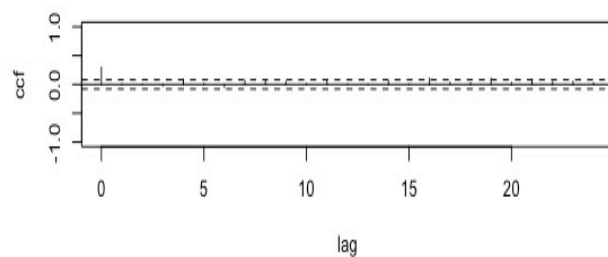
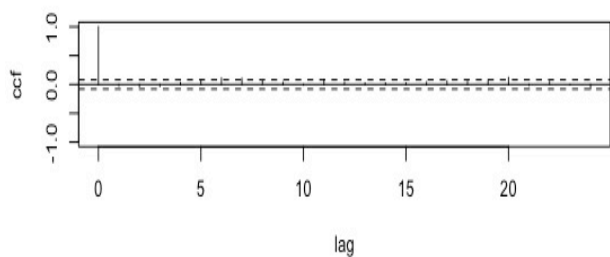
```
Hit Enter to compute MQ-statistics:
```

```
Ljung-Box Statistics:
```

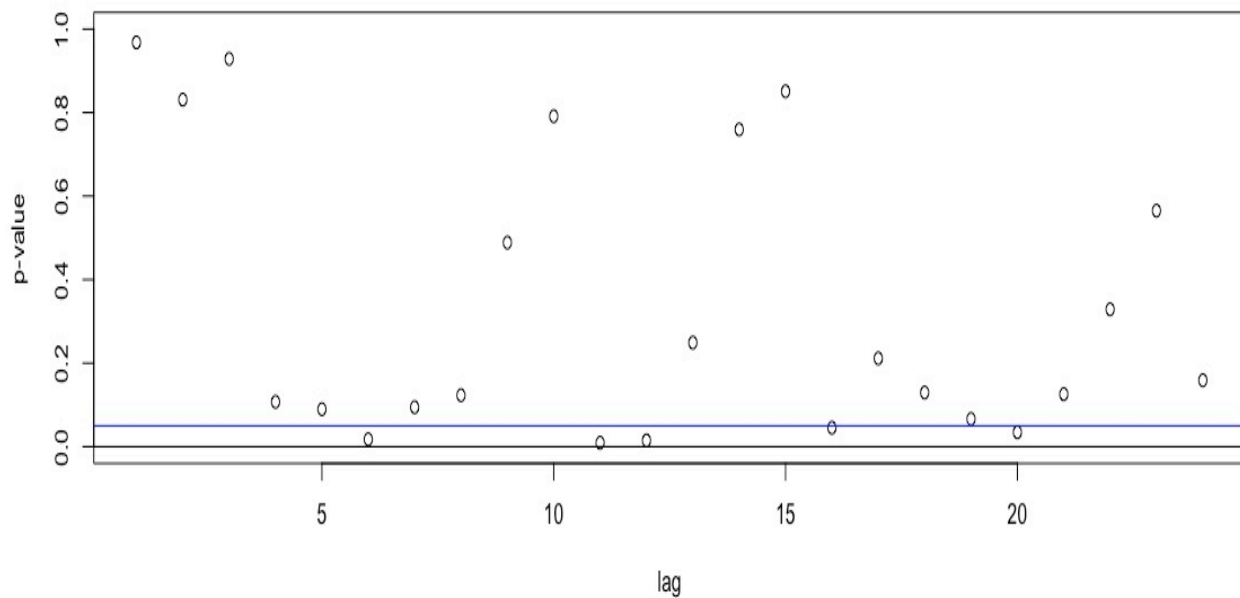
m	Q(m)	df	p-value	
[1,]	1.000	0.555	4.000	0.97
[2,]	2.000	2.035	8.000	0.98
[3,]	3.000	2.909	12.000	1.00
[4,]	4.000	10.533	16.000	0.84
[5,]	5.000	18.608	20.000	0.55
[6,]	6.000	30.601	24.000	0.17
[7,]	7.000	38.546	28.000	0.09
[8,]	8.000	45.818	32.000	0.05
[9,]	9.000	49.257	36.000	0.07
[10,]	10.000	50.958	40.000	0.11
[11,]	11.000	64.433	44.000	0.02
[12,]	12.000	76.827	48.000	0.01
[13,]	13.000	82.240	52.000	0.00
[14,]	14.000	84.112	56.000	0.01
[15,]	15.000	85.480	60.000	0.02
[16,]	16.000	95.225	64.000	0.01
[17,]	17.000	101.080	68.000	0.01
[18,]	18.000	108.216	72.000	0.00
[19,]	19.000	117.029	76.000	0.00
[20,]	20.000	127.423	80.000	0.00
[21,]	21.000	134.630	84.000	0.00
[22,]	22.000	139.242	88.000	0.00
[23,]	23.000	142.203	92.000	0.00
[24,]	24.000	148.814	96.000	0.00

```
Hit Enter to obtain residual plots:  
> VARpred(m1,4)  
orig 638  
Forecasts at origin: 638  
pceg dspig  
[1,] 0.5077 0.6494  
[2,] 0.5425 0.6182  
[3,] 0.5468 0.4325  
[4,] 0.5644 0.5374  
Standard Errors of predictions:  
[,1] [,2]  
[1,] 0.5430 0.6842  
[2,] 0.5525 0.6972  
[3,] 0.5567 0.7017  
[4,] 0.5628 0.7169  
Root mean square errors of predictions:  
[,1] [,2]  
[1,] 0.5459 0.6880  
[2,] 0.5536 0.6987  
[3,] 0.5572 0.7022  
[4,] 0.5634 0.7186
```





Significance plot of CCM



Program R for the VAR model

```
##Example 2
setwd("/Users/wasin_sivasarit/documents/EE435")
cat(rep("\n",50)) #clear R Console
install.packages('vars')
require(vars)
da1=read.table("m-pce.txt",header=T)
da2=read.table("m-dspi.txt",header=T)
head(da1)
head(da2)
x=cbind(da1$pce,da2$dspi)
x=log(x)
rt=diff(x)*100
colnames(rt) <- c("pceg","dspig")
varfit=VAR(rt,p=3)
summary(varfit)
varfit.prd <- predict(varfit, n.ahead = 15, ci = 0.95)
fanchart(varfit.prd)
impresp=irf(varfit)
plot(impresp)
fevd(varfit, n.ahead = 20)
```

Program R for the VAR model

```
> install.packages('vars')
> require(vars)
> da1=read.table("m-pce.txt",header=T)
> da2=read.table("m-dspi.txt",header=T)
> head(da1)
year mon day   pce
1 1959   1   1 306.7
2 1959   2   1 310.2
3 1959   3   1 313.3
4 1959   4   1 312.8
5 1959   5   1 316.7
6 1959   6   1 318.8
> head(da2)
year mon day  dspi
1 1959   1   1 340.7
2 1959   2   1 342.4
3 1959   3   1 344.8
4 1959   4   1 347.7
5 1959   5   1 350.0
6 1959   6   1 352.2
```

```

> x=cbind(da1$pce,da2$dspi)
> x=log(x)
> rt=diff(x)*100
> colnames(rt) <- c("pceg","dspig")
> varfit=VAR(rt,p=3)
> summary(varfit)

VAR Estimation Results:
=====
Endogenous variables: pceg, dspig
Deterministic variables: const
Sample size: 635
Log Likelihood: -1144.79
Roots of the characteristic polynomial:
0.6421 0.6421 0.6413 0.6089 0.5576 0.5576
Call:
VAR(y = rt, p = 3)

Estimation results for equation pceg:
=====
pceg = pceg.l1 + dspig.l1 + pceg.l2 + dspig.l2 + pceg.l3 + dspig.l3 + const

Estimate Std. Error t value Pr(>|t|)
pceg.l1 -0.152761 0.040884 -3.736 0.000204 ***
dspig.l1 0.129994 0.031868 4.079 5.10e-05 ***
pceg.l2 0.005258 0.041079 0.128 0.898193
dspig.l2 0.126390 0.032586 3.879 0.000116 ***
pceg.l3 0.053129 0.040273 1.319 0.187576
dspig.l3 0.128344 0.032242 3.981 7.68e-05 ***
const 0.400363 0.046606 8.590 < 2e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.546 on 628 degrees of freedom
Multiple R-Squared: 0.07572, Adjusted R-squared: 0.06689
F-statistic: 8.574 on 6 and 628 DF, p-value: 5.643e-09

Estimation results for equation dspig:
=====
dspig = pceg.l1 + dspig.l1 + pceg.l2 + dspig.l2 + pceg.l3 + dspig.l3 + const

Estimate Std. Error t value Pr(>|t|)
pceg.l1 0.15102 0.05152 2.931 0.003500 **

```

```

dspig.l1 -0.19374    0.04016   -4.824  1.77e-06 ***
pceg.l2  0.19778    0.05177    3.820  0.000146 ***
dspig.l2 -0.12396    0.04107   -3.019  0.002643 **
pceg.l3  0.34860    0.05075    6.869  1.56e-11 ***
dspig.l3 -0.10024    0.04063   -2.467  0.013894 *
const    0.39607    0.05873    6.743  3.51e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

Residual standard error: 0.6881 on 628 degrees of freedom
Multiple R-Squared: 0.1034,    Adjusted R-squared: 0.09482
F-statistic: 12.07 on 6 and 628 DF,  p-value: 7.381e-13

```

Covariance matrix of residuals:

```

pceg dspig
pceg  0.2981 0.1101
dspig 0.1101 0.4734

```

Correlation matrix of residuals:

```

pceg dspig
pceg  1.000 0.293
dspig 0.293 1.000

```

```

> varfit.prd <- predict(varfit, n.ahead = 15, ci = 0.95)
> fanchart(varfit.prd)
> impresp=irf(varfit)
> plot(impresp)
Hit <Return> to see next plot:
> fevd(varfit, n.ahead = 20)
$pceg
pceg      dspig
[1,] 1.0000000 0.0000000
[2,] 0.9763077 0.02369232
[3,] 0.9675234 0.03247656
[4,] 0.9588132 0.04118678
[5,] 0.9565904 0.04340957
[6,] 0.9564229 0.04357713
[7,] 0.9561258 0.04387419
[8,] 0.9558947 0.04410532
[9,] 0.9559018 0.04409823
[10,] 0.9559040 0.04409603
[11,] 0.9559044 0.04409564

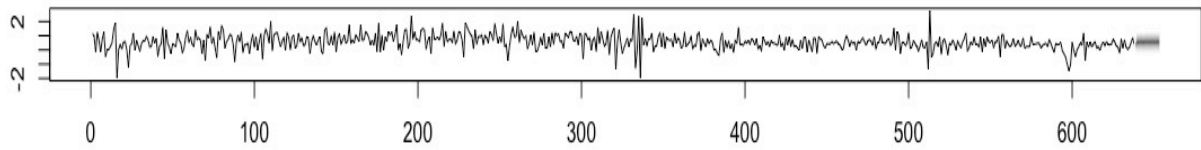
```

```
[12,] 0.9558926 0.04410736
[13,] 0.9558919 0.04410806
[14,] 0.9558920 0.04410804
[15,] 0.9558920 0.04410803
[16,] 0.9558919 0.04410815
[17,] 0.9558918 0.04410820
[18,] 0.9558918 0.04410822
[19,] 0.9558918 0.04410822
[20,] 0.9558918 0.04410822
```

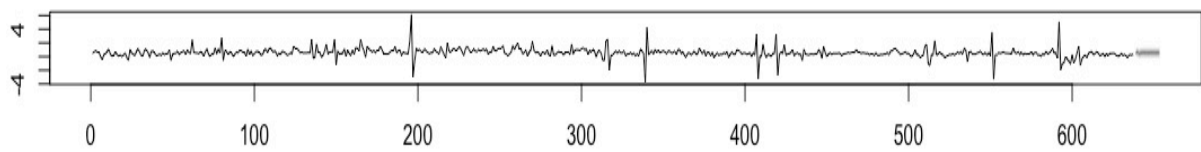
```
$dspig
```

```
pceg      dspig
[1,] 0.08586488 0.9141351
[2,] 0.08652867 0.9134713
[3,] 0.09417276 0.9058272
[4,] 0.13183574 0.8681643
[5,] 0.13367158 0.8663284
[6,] 0.13390144 0.8660986
[7,] 0.13436153 0.8656385
[8,] 0.13457126 0.8654287
[9,] 0.13458297 0.8654170
[10,] 0.13458069 0.8654193
[11,] 0.13458069 0.8654193
[12,] 0.13459594 0.8654041
[13,] 0.13459823 0.8654018
[14,] 0.13459834 0.8654017
[15,] 0.13459813 0.8654019
[16,] 0.13459823 0.8654018
[17,] 0.13459829 0.8654017
[18,] 0.13459834 0.8654017
[19,] 0.13459834 0.8654017
[20,] 0.13459834 0.8654017
```

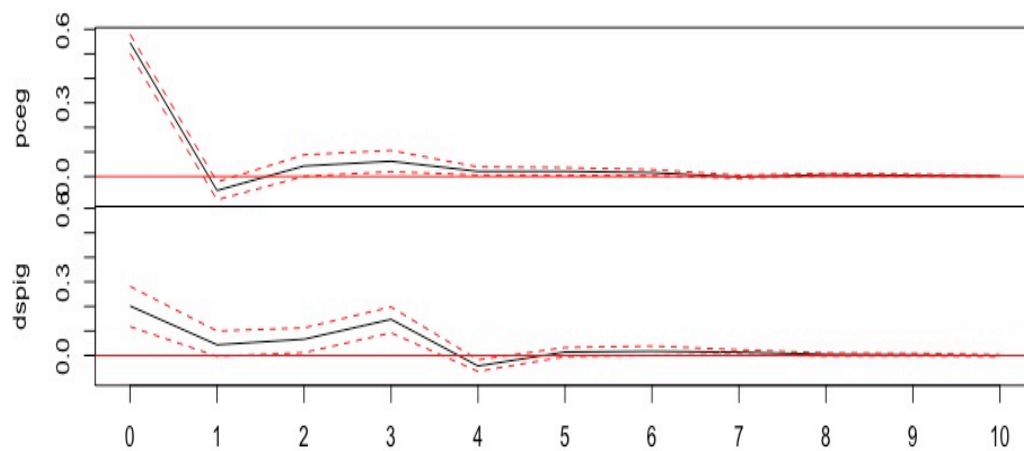
Fanchart for variable pceg



Fanchart for variable dspig

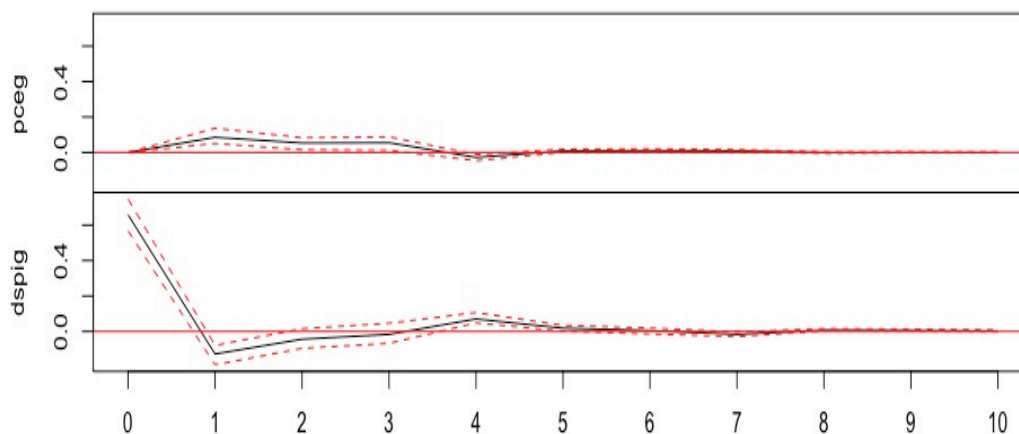


Orthogonal Impulse Response from pceg



95 % Bootstrap CI, 100 runs

Orthogonal Impulse Response from dspig



95 % Bootstrap CI, 100 runs

Program R for the VAR model

```
> require(MTS)
Loading required package: MTS

Attaching package: 'MTS'

The following object is masked from 'package:vars':

VAR

> da=read.table("q-gdpun.txt",header=T)
> head(da)
year mon day    gdp  unemp
1 1948  1  1  7.3878  3.7333
2 1948  4  1  7.4053  3.6667
3 1948  7  1  7.4110  3.7667
4 1948 10  1  7.4134  3.8333
5 1949  1  1  7.3983  4.6667
6 1949  4  1  7.3954  5.8667
> xt=da[,4:5]
> MTSplot(xt)
> tdx=da[,1]+da[,2]/12
```

```
> MTSplot(xt,tdx)
> ccm(xt)
[1] "Covariance matrix:"
gdp unemp
gdp 0.302 0.24
unemp 0.240 2.32
CCM at lag: 0
[,1] [,2]
[1,] 1.000 0.286
[2,] 0.286 1.000
Simplified matrix:
CCM at lag: 1
+ +
+ +
CCM at lag: 2
+ +
+ +
CCM at lag: 3
+ +
+ +
CCM at lag: 4
+ +
+ +
CCM at lag: 5
+ +
+ +
CCM at lag: 6
+ +
+ +
CCM at lag: 7
+ +
+ +
CCM at lag: 8
+ +
+ +
CCM at lag: 9
+ +
+ +
CCM at lag: 10
+ +
+ +
CCM at lag: 11
+ +
+ +
CCM at lag: 12
+ +
```

```
+ +
Hit Enter for p-value plot of individual ccm:

> xt=log(xt)
> rt=diffM(xt) ### differencing all series
> ccm(rt)
[1] "Covariance matrix:"
gdp      unemp
gdp      1.57e-06 -6.34e-05
unemp    -6.34e-05  5.21e-03
CCM at lag:  0
[,1]  [,2]
[1,]  1.000 -0.701
[2,] -0.701  1.000
Simplified matrix:
CCM at lag:  1
+ -
- +
CCM at lag:  2
+ .
- +
CCM at lag:  3
. .
- .
CCM at lag:  4
. +
. -
CCM at lag:  5
- +
+ -
CCM at lag:  6
. +
. -
CCM at lag:  7
. .
. .
CCM at lag:  8
. .
. -
CCM at lag:  9
. .
. .
CCM at lag: 10
. .
. .
CCM at lag: 11
```

```

. .
. .
CCM at lag: 12
. .
. -
Hit Enter for p-value plot of individual ccm:
mq(rt,lag=10)
>
> VARorder(rt)
selected order: aic = 13
selected order: bic = 1
selected order: hq = 1
Summary table:
p      AIC      BIC      HQ      M(p) p-value
[1,]  0 -19.6004 -19.6004 -19.6004  0.0000  0.0000
[2,]  1 -20.0592 -19.9989 -20.0349 104.0058  0.0000
[3,]  2 -20.0818 -19.9611 -20.0331 12.0598  0.0169
[4,]  3 -20.0644 -19.8833 -19.9913  3.6796  0.4511
[5,]  4 -20.1028 -19.8614 -20.0053 15.0534  0.0046
[6,]  5 -20.0916 -19.7899 -19.9699  4.8802  0.2998
[7,]  6 -20.0933 -19.7312 -19.9472  7.4052  0.1160
[8,]  7 -20.0683 -19.6459 -19.8979  2.0412  0.7282
[9,]  8 -20.0654 -19.5826 -19.8706  6.3427  0.1750
[10,] 9 -20.0558 -19.5126 -19.8366  4.9911  0.2882
[11,]10 -20.0505 -19.4470 -19.8070  5.7647  0.2174
[12,]11 -20.0619 -19.3980 -19.7940  8.8768  0.0643
[13,]12 -20.1079 -19.3837 -19.8157 15.3240  0.0041
[14,]13 -20.1109 -19.3263 -19.7943  7.1245  0.1295
> m1=VAR(rt,p = 1,output = T)
Constant term:
Estimates:  0.0008020814 0.01873175
Std.Error:  0.0001191335 0.005648777
AR coefficient matrix
AR( 1 )-matrix
[,1]      [,2]
[1,]  0.21 -0.00326
[2,] -17.37  0.39903
standard error
[,1]      [,2]
[1,] 0.0871 0.00151
[2,] 4.1280 0.07158

Residuals cov-mtx:
[,1]      [,2]
[1,] 1.347916e-06 -4.183392e-05
[2,] -4.183392e-05  3.030425e-03

```

```
det(SSE) = 2.334682e-09
AIC = -19.84015
BIC = -19.7798
HQ = -19.8158
> MTSdiag(m1)
[1] "Covariance matrix:"
gdp      unemp
gdp      1.35e-06 -0.000042
unemp    -4.20e-05 0.003044
CCM at lag: 0
[,1]  [,2]
[1,]  1.000 -0.655
[2,] -0.655 1.000
Simplified matrix:
CCM at lag: 1
. .
. .
CCM at lag: 2
. .
- .
CCM at lag: 3
. .
. .
CCM at lag: 4
. +
. -
CCM at lag: 5
. +
+ -
CCM at lag: 6
. .
. .
CCM at lag: 7
. .
. .
CCM at lag: 8
. .
. -
CCM at lag: 9
. .
. .
CCM at lag: 10
. .
. .
CCM at lag: 11
```

```
. .  
. .  
CCM at lag: 12  
. .  
+ -  
CCM at lag: 13  
. .  
. .  
CCM at lag: 14  
. .  
. .  
CCM at lag: 15  
. .  
. .  
CCM at lag: 16  
. -  
. .  
CCM at lag: 17  
. .  
. .  
CCM at lag: 18  
. .  
. .  
CCM at lag: 19  
. .  
. .  
CCM at lag: 20  
. .  
. .  
CCM at lag: 21  
. .  
. .  
CCM at lag: 22  
. .  
. .  
CCM at lag: 23  
. .  
. .  
CCM at lag: 24  
. +  
. .  
Hit Enter for p-value plot of individual ccm:  
m1$coef  
Hit Enter to compute MQ-statistics:  
  
Ljung-Box Statistics:
```

m	Q(m)	df	p-value	
[1,]	1.00	1.49	4.00	0.83
[2,]	2.00	10.59	8.00	0.23
[3,]	3.00	11.89	12.00	0.45
[4,]	4.00	33.32	16.00	0.01
[5,]	5.00	43.94	20.00	0.00
[6,]	6.00	48.80	24.00	0.00
[7,]	7.00	52.69	28.00	0.00
[8,]	8.00	58.67	32.00	0.00
[9,]	9.00	65.70	36.00	0.00
[10,]	10.00	69.24	40.00	0.00
[11,]	11.00	71.87	44.00	0.01
[12,]	12.00	86.76	48.00	0.00
[13,]	13.00	91.83	52.00	0.00
[14,]	14.00	95.02	56.00	0.00
[15,]	15.00	102.00	60.00	0.00
[16,]	16.00	111.66	64.00	0.00
[17,]	17.00	114.31	68.00	0.00
[18,]	18.00	123.62	72.00	0.00
[19,]	19.00	125.68	76.00	0.00
[20,]	20.00	131.12	80.00	0.00
[21,]	21.00	134.42	84.00	0.00
[22,]	22.00	135.03	88.00	0.00
[23,]	23.00	136.47	92.00	0.00
[24,]	24.00	156.92	96.00	0.00

Hit Enter to obtain residual plots:

```
VARpred(m1,10)
```

```
> m2=refVAR(m1)
```

Constant term:

```
Estimates: 0.0008020814 0.01873175
```

```
Std.Error: 0.0001191335 0.005648777
```

AR coefficient matrix

```
AR( 1 )-matrix
```

```
[,1]      [,2]
```

```
[1,] 0.21 -0.00326
```

```
[2,] -17.37 0.39903
```

standard error

```
[,1]      [,2]
```

```
[1,] 0.0871 0.00151
```

```
[2,] 4.1280 0.07158
```

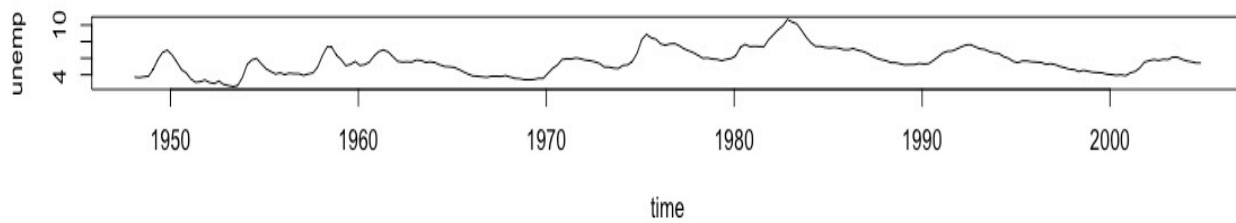
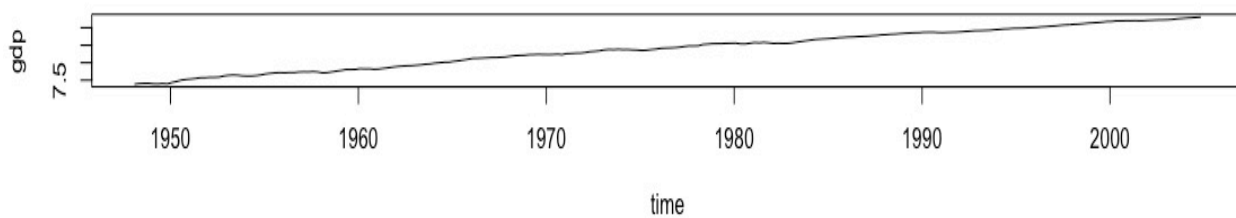
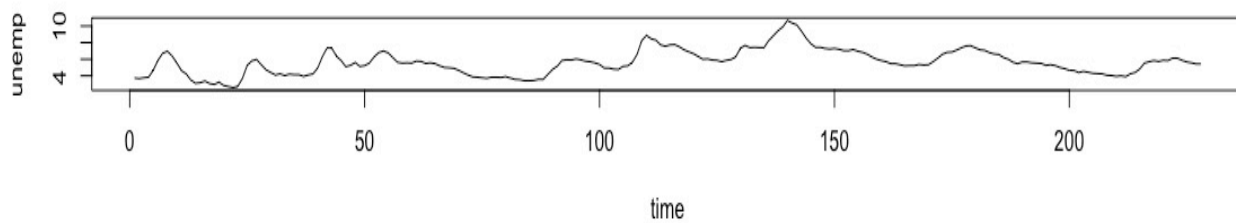
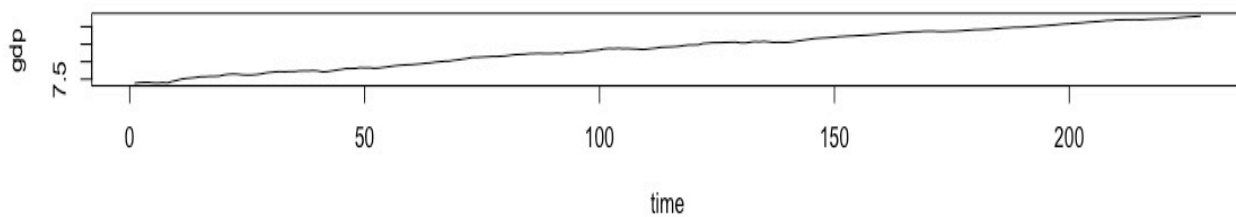
Residuals cov-mtx:

```
[,1]      [,2]
```

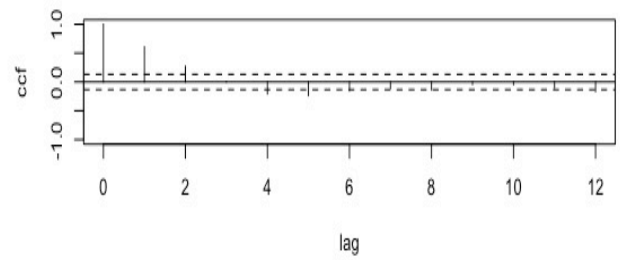
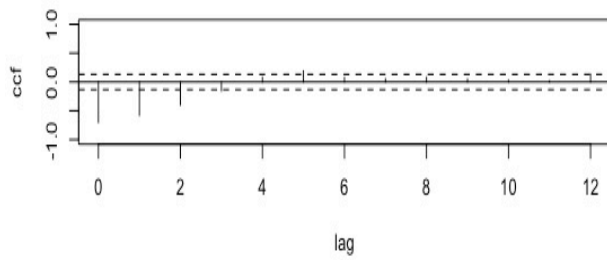
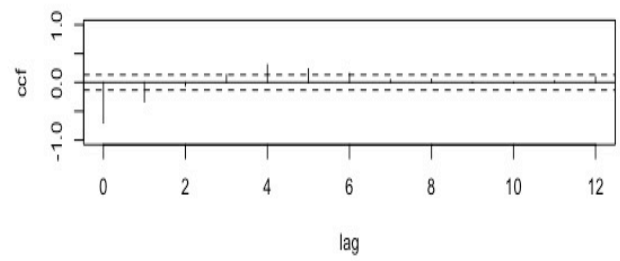
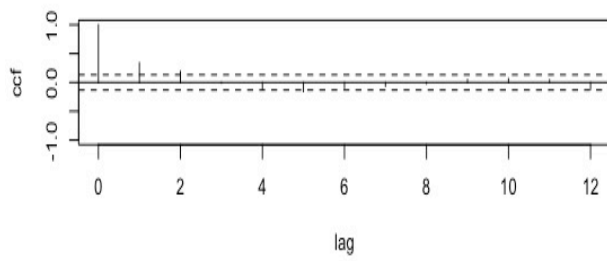
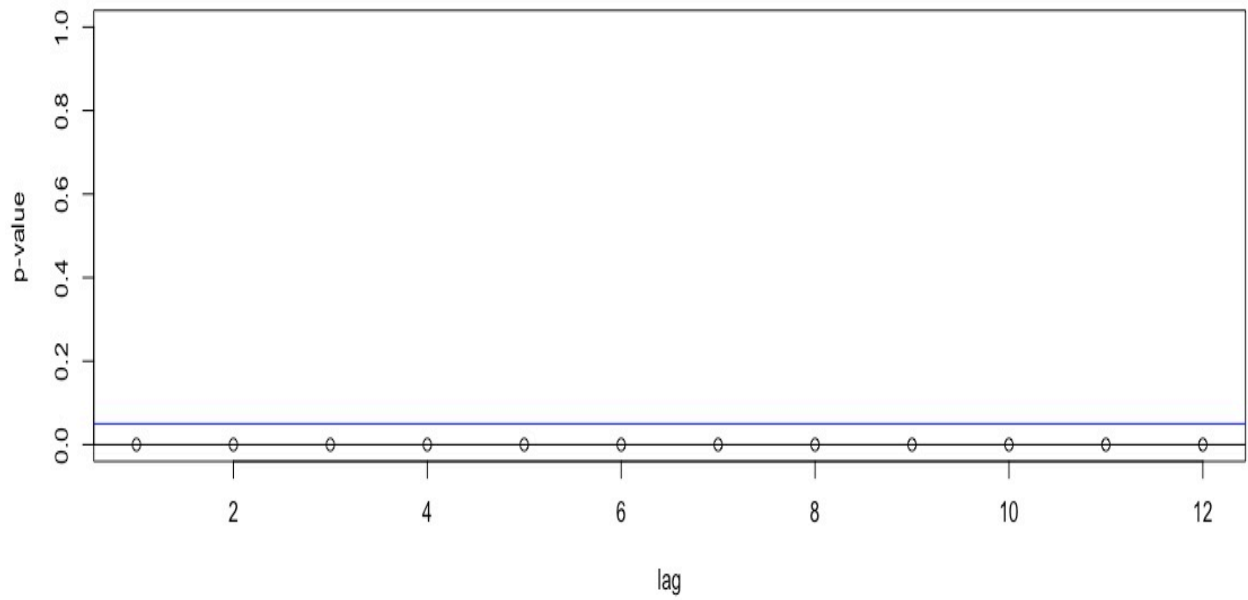
```
[1,] 1.347916e-06 -4.183392e-05
```

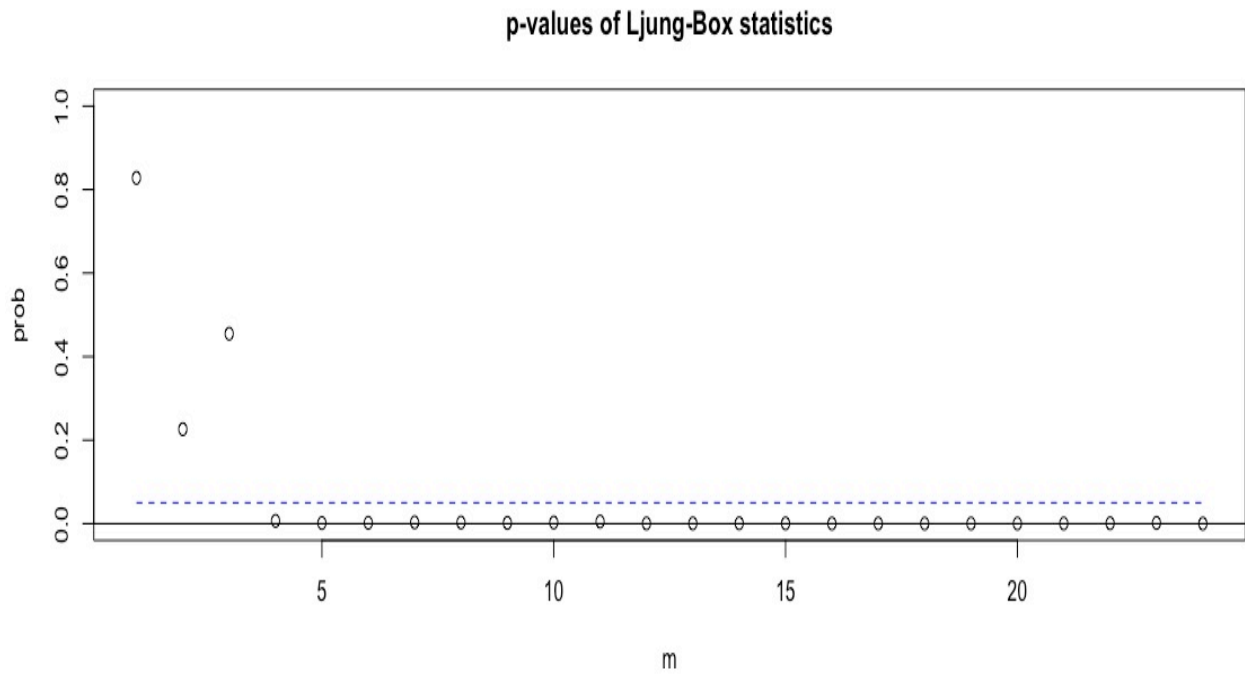
```
[2,] -4.183392e-05 3.030425e-03
```

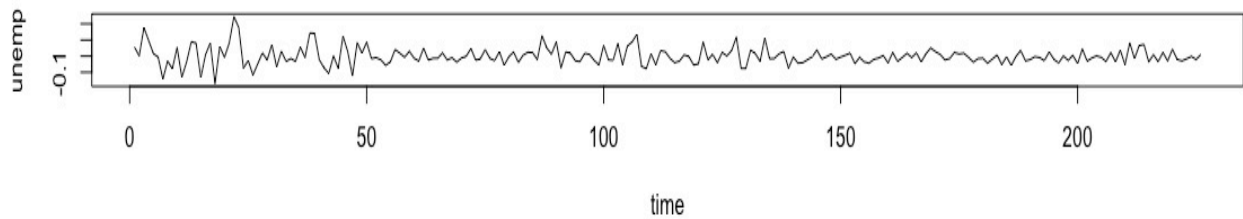
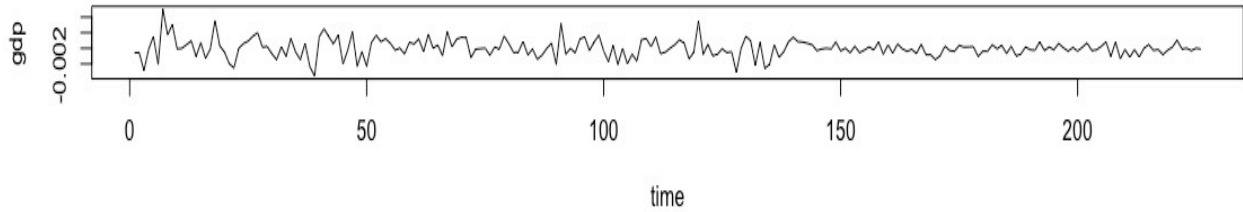
```
det(SSE) = 2.334682e-09
AIC = -19.84015
BIC = -19.7798
HQ = -19.8158
> VARpred(m2,10)
orig 227
Forecasts at origin: 227
gdp    unemp
[1,] 0.001015 0.001178
[2,] 0.001012 0.001579
[3,] 0.001010 0.001791
[4,] 0.001009 0.001909
[5,] 0.001008 0.001975
[6,] 0.001008 0.002013
[7,] 0.001007 0.002033
[8,] 0.001007 0.002045
[9,] 0.001007 0.002052
[10,] 0.001007 0.002055
Standard Errors of predictions:
[,1]    [,2]
[1,] 0.001161 0.05505
[2,] 0.001224 0.06708
[3,] 0.001241 0.07059
[4,] 0.001246 0.07166
[5,] 0.001248 0.07199
[6,] 0.001248 0.07210
[7,] 0.001248 0.07213
[8,] 0.001249 0.07214
[9,] 0.001249 0.07215
[10,] 0.001249 0.07215
Root mean square errors of predictions:
[,1]    [,2]
[1,] 0.001169 0.05541
[2,] 0.029206 2.89629
[3,] 0.015611 1.66231
[4,] 0.008776 0.93573
[5,] 0.005026 0.52809
[6,] 0.003001 0.30204
[7,] 0.001975 0.17957
[8,] 0.001515 0.11706
[9,] 0.001338 0.08875
[10,] 0.001277 0.07775
```



Significance plot of CCM







4.11 Cointegration and Error-Correction Models

4.11.1 Linear Combinations of Integrated Variables

$$m_t = \beta_0 + \beta_1 p_t + \beta_2 y_t + \beta_3 r_t + e_t$$

m_t is the demand for money

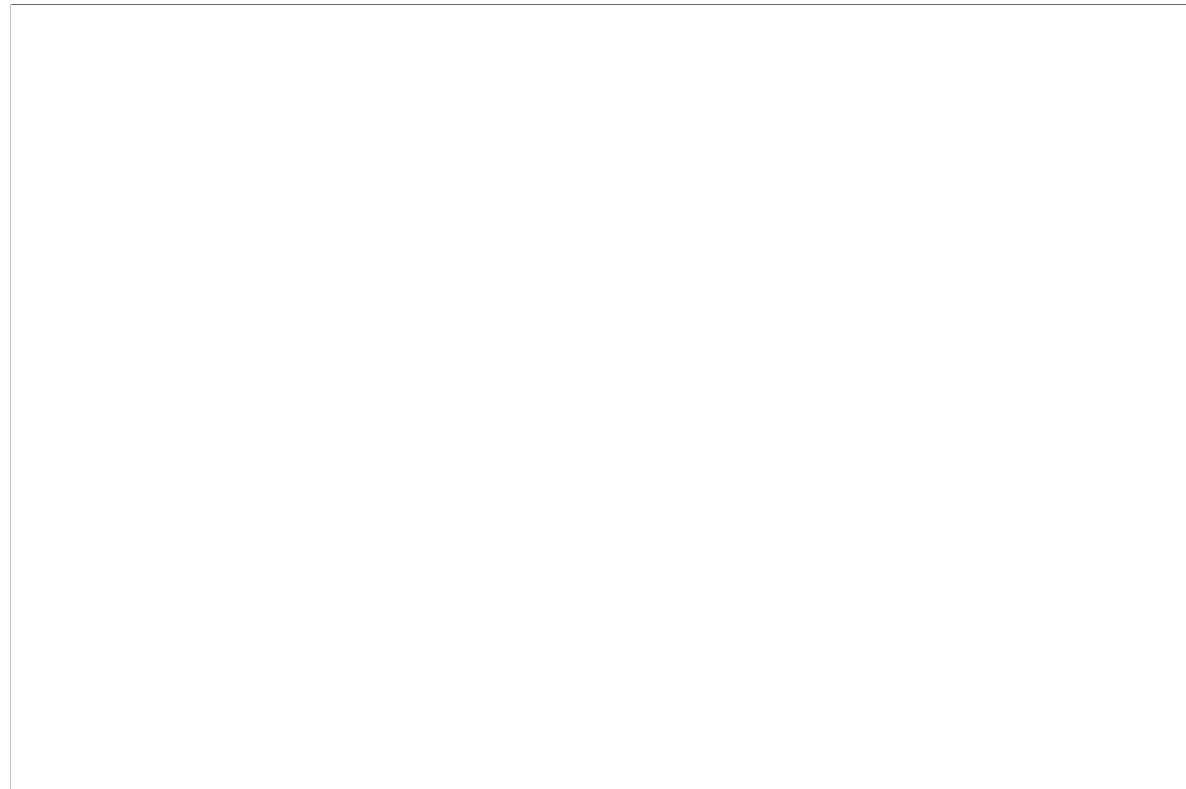
p_t is the price level

y_t is the income level r_t is the interest rate

e_t is the disturbance term.



All of these examples illustrate the concept of cointegration as introduced by Engle and Granger (1987). Their formal analysis begins by considering a set of economic variables in long-run equilibrium when



The components of the vector $x_t = (x_{1t}, x_{2t}, \dots, x_{nt})$ are said to be cointegrated of order d, b , denoted by $x_t \text{ CI}(d, b)$ if

1. All components of x_t are integrated of order d .
2. There exists a vector $\beta = (\beta_1, \beta_2, \dots, \beta_n)$ such that the linear combination $\beta x_t = \beta_1 x_{1t} + \beta_2 x_{2t} + \dots + \beta_n x_{nt}$ is integrated of order $(d-b)$ where $b > 0$. Note that the vector β is called the cointegrating vector.

4.11.2 Cointegration and Error Correction

In the error-correction model, the short-term dynamics of the variables in the system are influenced by the deviation from equilibrium.

$$\Delta r_{St} = \alpha_S(r_{Lt-1} - \beta r_{St-1}) + \epsilon_{St}$$

$$\Delta r_{Lt} = -\alpha_L(r_{Lt-1} - \beta r_{St-1}) + \epsilon_{Lt}$$



All data are contained in the file " stock-dividend.txt" which are the stock price index (S&P500) and the dividends from January 1871 to February 2017, totally 1754 observations

Let s_t is the the log of stock prices and d_t is the log of dividends.

Figure 4.1

```
> library(fUnitRoots)
> library(quantmod)
> stock_dividend=read.table("stock_dividend.txt",header=T)
> head(stock_dividend)
      Date      P      D
1 1871.01  4.44  0.26
2 1871.02  4.50  0.26
3 1871.03  4.61  0.26
4 1871.04  4.74  0.26
5 1871.05  4.86  0.26
6 1871.06  4.82  0.26
> st =log(stock_dividend$P)
> plot.ts(st,main="S&P 500 Stock Price Index")
> dt = log(stock_dividend$D)
> plot.ts(dt,main="Dividend")
> m1=adfTest(st, lags = 3, type = c("ct"), title = NULL,description = NULL)
> m1@test$p.value

0.6583257
> m1@test$parameter
Lag Order
      3
> m1@test$lm

Call:
lm(formula = y.diff ~ y.lag.1 + 1 + tt + y.diff.lag)

Coefficients:
(Intercept)      y.lag.1          tt  y.diff.lag1  y.diff.lag2
 1.030e-03  -2.731e-03   1.322e-05   3.044e-01  -7.510e-02
y.diff.lag3
-2.102e-02
```

Figure 4.1 (Cont.)

```

> m2=adfTest(dt, lags = 3, type = c("ct"), title = NULL,description = NULL)
> m2@test$p.value

0.5448925
> m2@test$parameter
Lag Order
      3
> m2@test$lm

Call:
lm(formula = y.diff ~ y.lag.1 + 1 + tt + y.diff.lag)

Coefficients:
(Intercept)      y.lag.1          tt  y.diff.lag1  y.diff.lag2  y.diff.lag3
-2.890e-03  -1.266e-03   4.814e-06   2.187e-01   3.377e-01   1.594e-01

> m3=adfTest(diff(st), lags = 3, type = c("ct"), title = NULL,description = NULL)
Warning message:
In adfTest(diff(st), lags = 3, type = c("ct"), title = NULL, description = NULL) :
  p-value smaller than printed p-value
> m3@test$p.value

0.01
> m3@test$parameter
Lag Order
      3
> m3@test$lm

Call:
lm(formula = y.diff ~ y.lag.1 + 1 + tt + y.diff.lag)

Coefficients:
(Intercept)      y.lag.1          tt  y.diff.lag1  y.diff.lag2  y.diff.lag3
-1.109e-04  -7.605e-01   3.184e-06   6.496e-02  -8.138e-03  -4.429e-02

> m4=adfTest(diff(dt), lags = 3, type = c("ct"), title = NULL,description = NULL)
Warning message:
In adfTest(diff(dt), lags = 3, type = c("ct"), title = NULL, description = NULL) :
  p-value smaller than printed p-value
> m4@test$p.value

0.01
> m4@test$parameter
Lag Order
      3
> m4@test$lm

Call:
lm(formula = y.diff ~ y.lag.1 + 1 + tt + y.diff.lag)

Coefficients:
(Intercept)      y.lag.1          tt  y.diff.lag1  y.diff.lag2  y.diff.lag3
 1.627e-05  -2.427e-01   8.055e-07  -5.612e-01  -2.741e-01  -1.489e-01

```

Figure 4.2

```

> fit <- lm(st ~dt)
> summary(fit)

Call:
lm(formula = st ~ dt)

Residuals:
    Min       1Q   Median       3Q      Max
-1.07695 -0.19837  0.01338  0.21131  0.81566

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.136369   0.007250   432.6  <2e-16 ***
dt          1.196288   0.004387   272.7  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2951 on 1752 degrees of freedom
Multiple R-squared:  0.977,    Adjusted R-squared:  0.977
F-statistic: 7.435e+04 on 1 and 1752 DF,  p-value: < 2.2e-16

> error=residuals(fit)
> m1=adfTest(error, lags = 3, type = c("ct"), title = NULL,description = NULL)
Warning message:
In adfTest(error, lags = 3, type = c("ct"), title = NULL, description = NULL) :
  p-value smaller than printed p-value
> m1@test$p.value

0.01
> m1@test$parameter
Lag Order
      3
> m1@test$lm

Call:
lm(formula = y.diff ~ y.lag.1 + 1 + tt + y.diff.lag)

Coefficients:
(Intercept)   y.lag.1         tt  y.diff.lag1  y.diff.lag2  y.diff.lag3
 4.397e-05  -1.377e-02  -5.008e-08   3.229e-01  -5.263e-02   4.615e-03

```