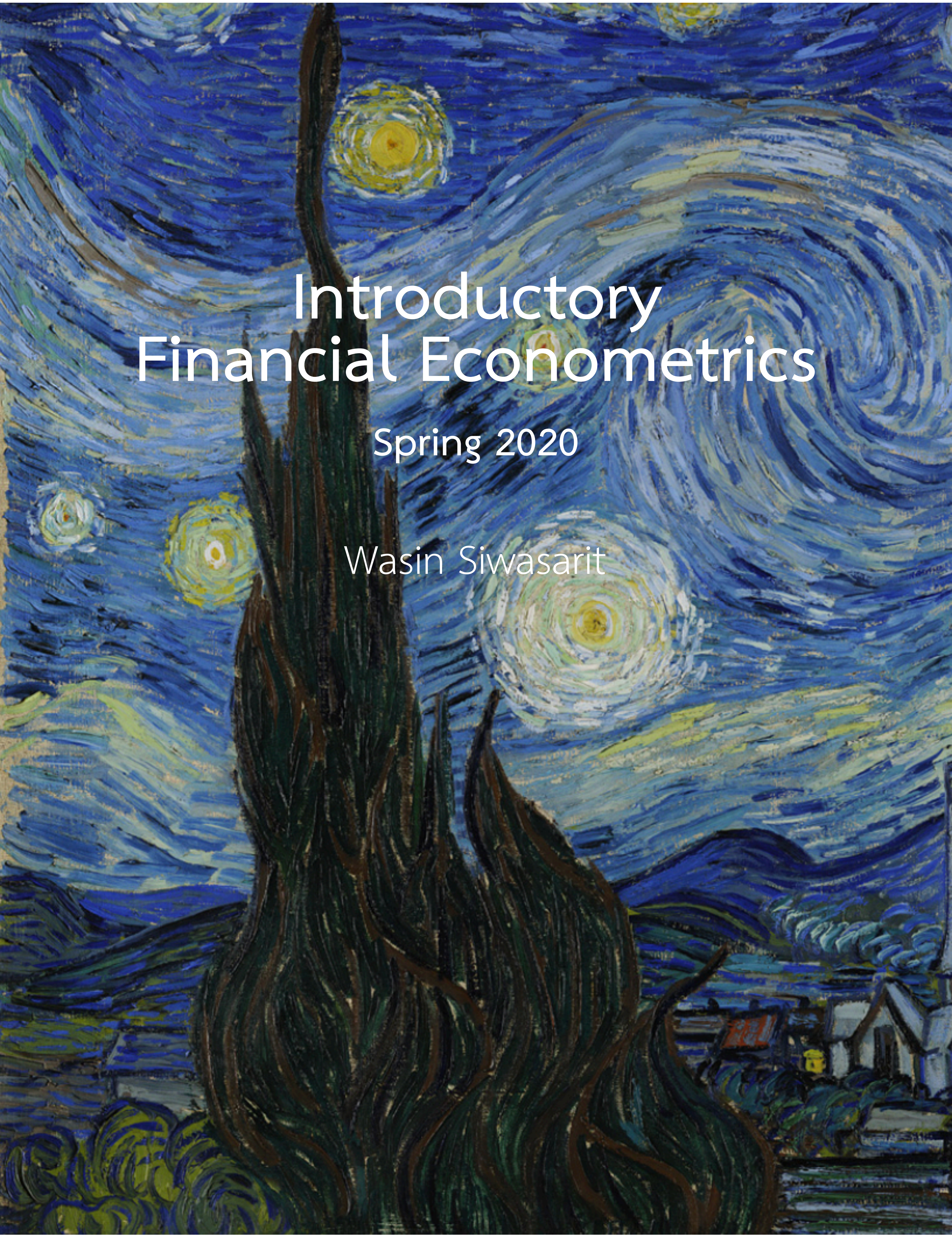
Introductory Financial Econometrics

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

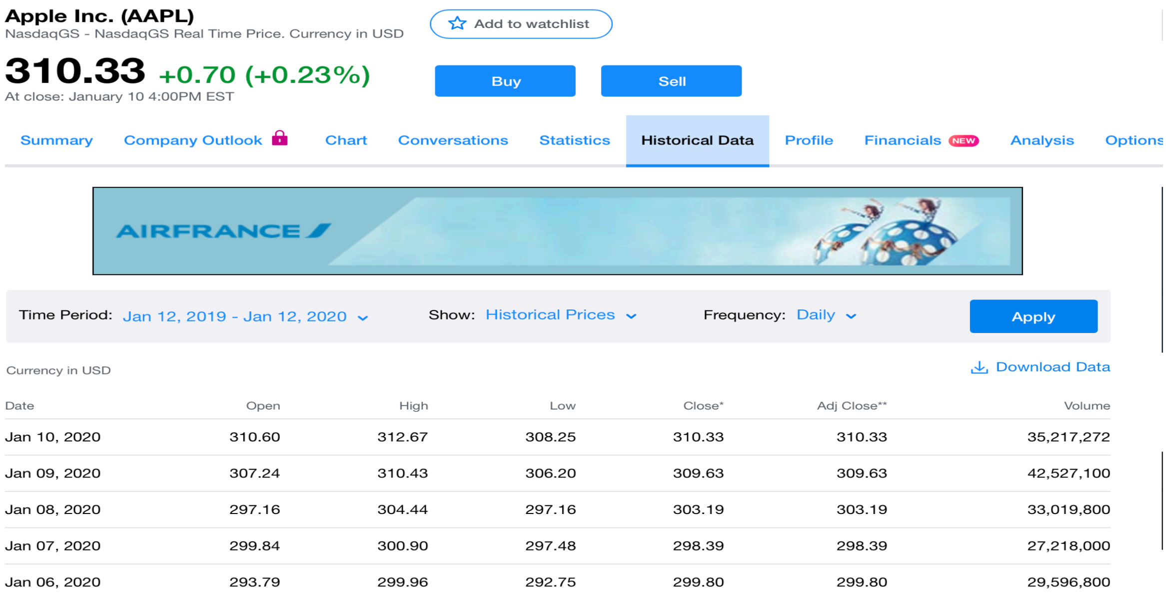
Let P_t be the price of an asset at time t, and assume no dividend. One-period simple return: Gross return

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

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 NEW 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 $i = 1, ..., N$; and $t = 1, ..., T$

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 (Symmestry)

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

Kurtosis (Fat-taills)

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

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

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

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

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
```

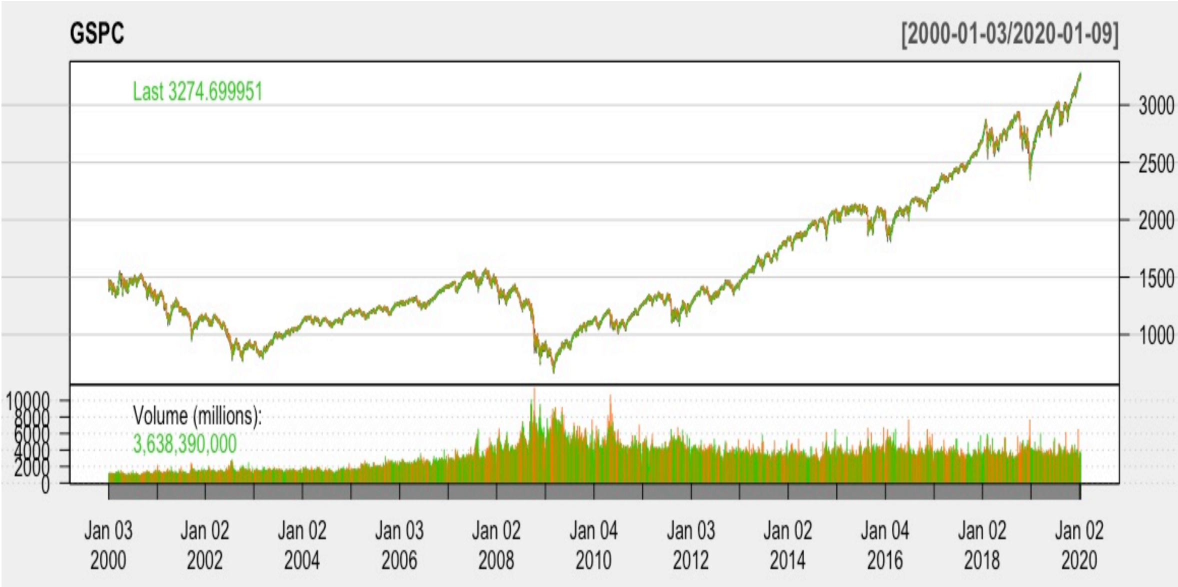
```
Variance                   0.0002
Stdev                      0.0126
Skewness                  -0.3865
Kurtosis                  11.0457
> #3 QQ-plots and tests for normality
> #
> # use qqnorm function
> par(mfrow=c(1,1))
> qqnorm(newlogreturn)
> qqline(newlogreturn, col = 2)
> jarque.bera.test(newlogreturn)

        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
```

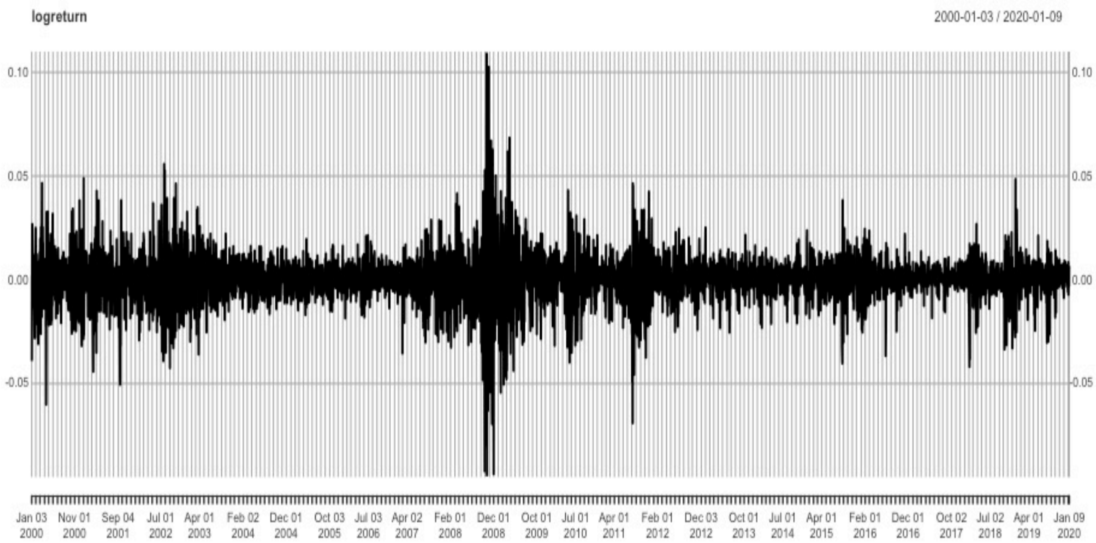
FE Print out (Cont.)

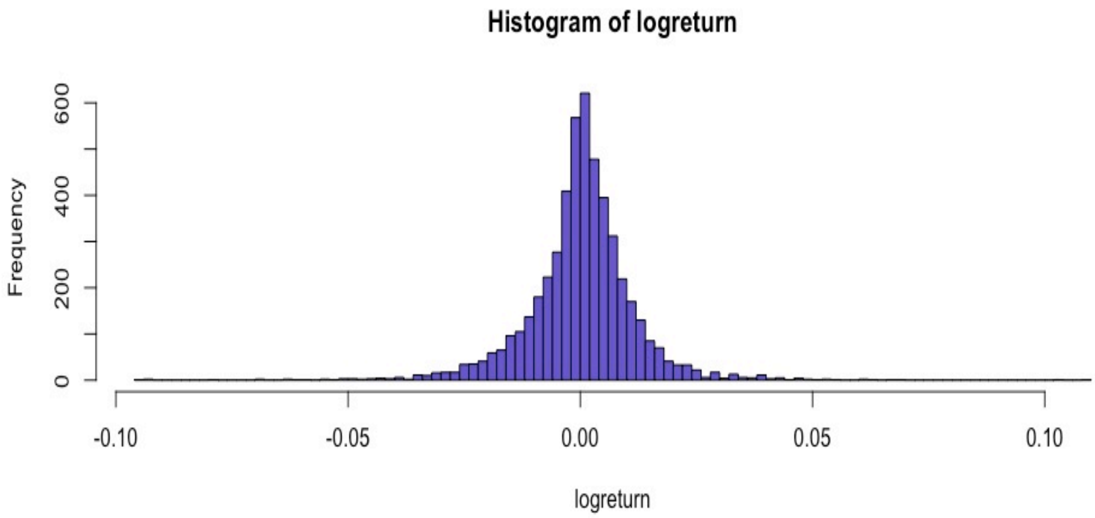
```
> t.test(newlogreturn)

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
```

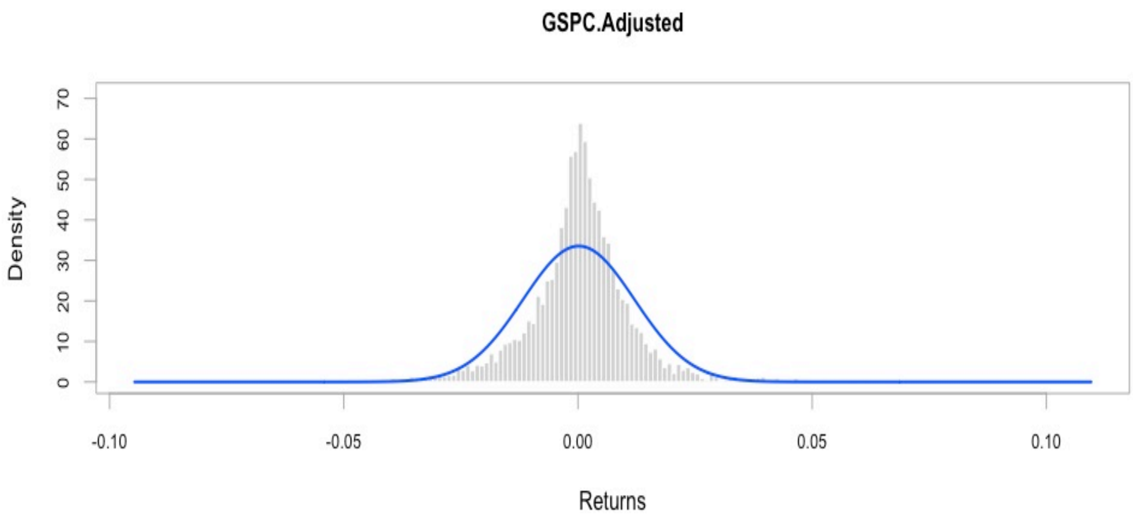

GSPC
[2000-01-03/2020-01-09]
Last 3274.699951
3000
2500
2000
1500
1000
Volume (millions):
3,638,390,000
10000
8000
6000
4000
2000
0
Jan 03 2000
Jan 02 2002
Jan 02 2004
Jan 03 2006
Jan 02 2008
Jan 04 2010
Jan 03 2012
Jan 02 2014
Jan 04 2016
Jan 02 2018
Jan 02 2020

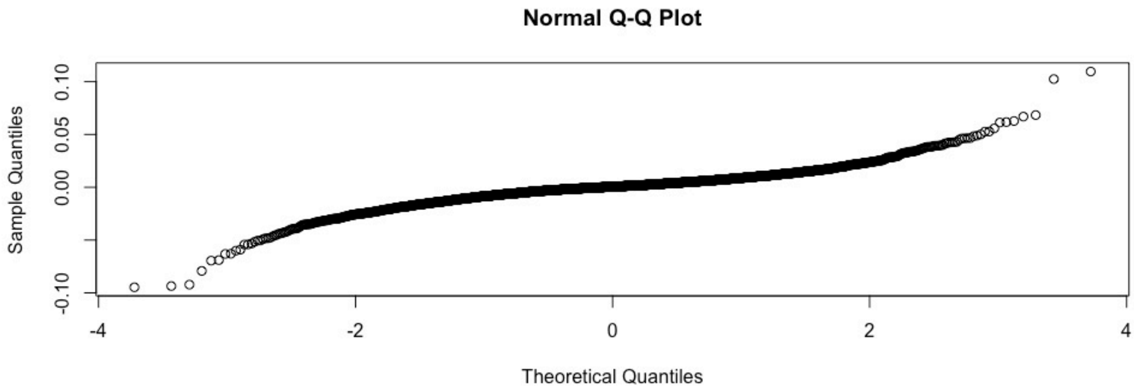

price
2000-01-03 / 2020-01-09
3000
2500
2000
1500
1000
Jan 03 2000
Nov 01 2000
Sep 04 2001
Jul 01 2002
Apr 01 2003
Feb 02 2004
Dec 01 2004
Oct 03 2005
Jul 03 2006
Apr 02 2007
Feb 01 2008
Dec 01 2008
Oct 01 2009
Jul 01 2010
Apr 01 2011
Feb 01 2012
Dec 03 2012
Oct 01 2013
Jul 01 2014
Apr 01 2015
Feb 01 2016
Dec 01 2016
Oct 02 2017
Jul 02 2018
Apr 01 2019
Jan 09 2020

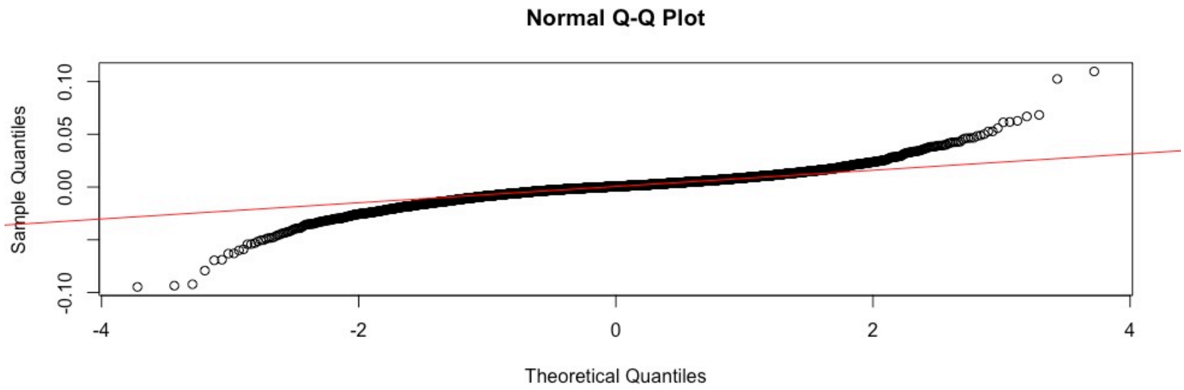

logprice
2000-01-03 / 2020-01-09
8.0
7.5
7.0
Jan 03 2000
Nov 01 2000
Sep 04 2001
Jul 01 2002
Apr 01 2003
Feb 02 2004
Dec 01 2004
Oct 03 2005
Jul 03 2006
Apr 02 2007
Feb 01 2008
Dec 01 2008
Oct 01 2009
Jul 01 2010
Apr 01 2011
Feb 01 2012
Dec 03 2012
Oct 01 2013
Jul 01 2014
Apr 01 2015
Feb 01 2016
Dec 01 2016
Oct 02 2017
Jul 02 2018
Apr 01 2019
Jan 09 2020


logreturn
2000-01-03 / 2020-01-09
0.10
0.05
0.00
-0.05
Jan 03 2000
Nov 01 2000
Sep 04 2001
Jul 01 2002
Apr 01 2003
Feb 02 2004
Dec 01 2004
Oct 03 2005
Jul 03 2006
Apr 02 2007
Feb 01 2008
Dec 01 2008
Oct 01 2009
Jul 01 2010
Apr 01 2011
Feb 01 2012
Dec 03 2012
Oct 01 2013
Jul 01 2014
Apr 01 2015
Feb 01 2016
Dec 01 2016
Oct 02 2017
Jul 02 2018
Apr 01 2019
Jan 09 2020


Histogram of logreturn
Frequency
0
200
400
600
-0.10
-0.05
0.00
0.05
0.10
logreturn


GSPC.Adjusted
Density
0
10
20
30
40
50
60
70
-0.10
-0.05
0.00
0.05
0.10
Returns


Normal Q-Q Plot
Sample Quantiles
-0.10
0.00
0.05
0.10
-4
-2
0
2
4
Theoretical Quantiles


Normal Q-Q Plot
Sample Quantiles
-0.10
0.00
0.05
0.10
-4
-2
0
2
4
Theoretical Quantiles

2.1 Basic Concepts

Financial TS: collection of a financial measurement over time.

Example: log return of apple r_t

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

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

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

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

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

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

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:

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

Serial (or-auto) correlations:

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

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):

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

$H_a : \rho_i \neq 0$

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

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
```

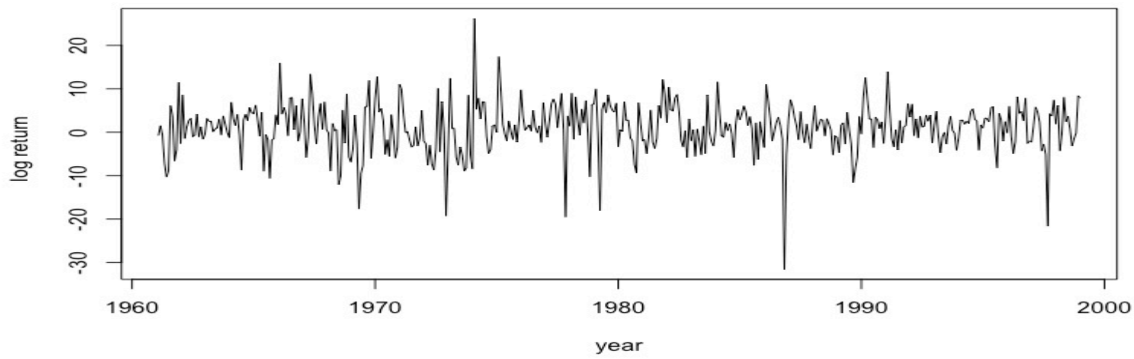
```
pv = 2*(1-pnorm(tt2))
pv
m1=acf(log_return)
names(m1)
m1$acf
m2=pacf(log_return)
names(m2)
m2$acf
Box.test(log_return,lag=12,type='Ljung')
```

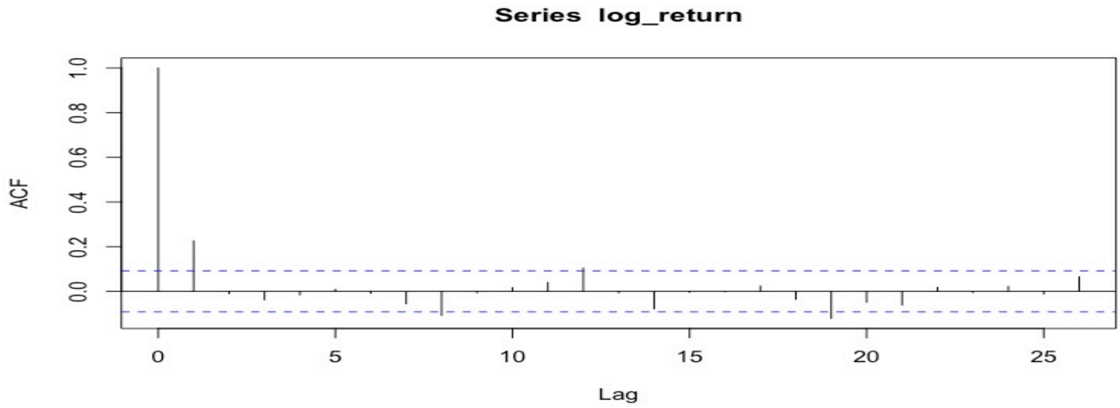
FE Analysis 2

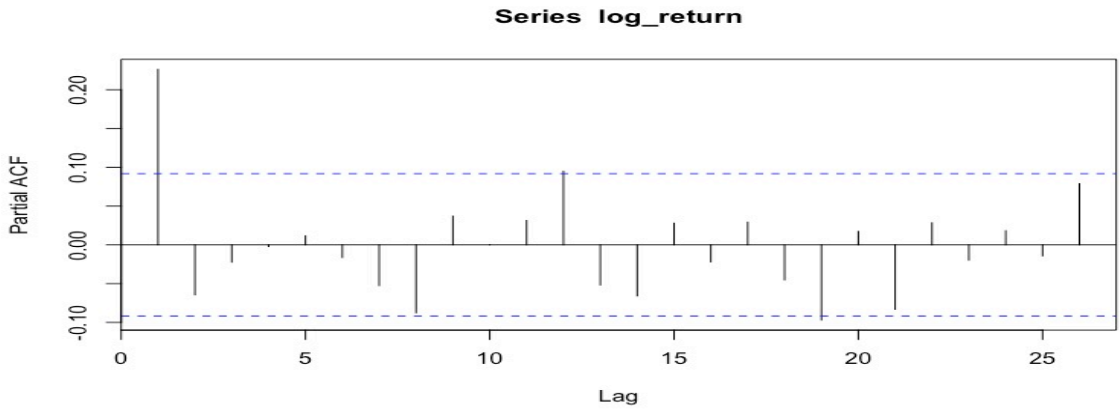
```
> da <- read.table("CRSP.txt")
> log_return =da[,1]
> par(mfcol=c(1,1))
> length(log_return)
[1] 456
> tdx = c(1:456)/12+1961
> plot(tdx,log_return,xlab='year',ylab='log_return',type='l')
> basicStats(log_return)
            log_return
nobs        456.000000
NAs           0.000000
Minimum     -31.588000
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
```


log return
20
10
0
-10
-20
-30
1960
1970
1980
1990
2000
year


Series log_return
ACF
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
Lag


Series log_return
Partial ACF
0.20
0.10
0.00
-0.10
0
5
10
15
20
25
Lag

2.2 Back-Shift (lag) operator

Definition $Br_t = r_{t-1}$ or $Lr_t = r_{t-1}$
$B^2 r_t = B(Br_t) = Br_{t-1} = r_{t-1}$
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 =
$B^2 r_5$ =

A proper perspective: at a time point t

Available data: $\{r_1, r_2, ...r_{t-1}\}$ ≡F_{t-1}

The return is decomposed into two parts:

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

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

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

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 itterature 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 build- ing 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 forecasing r_t. Thus the linear model can be:

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

wherea_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

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

2.4.1 Properties of AR models

AR (1) model

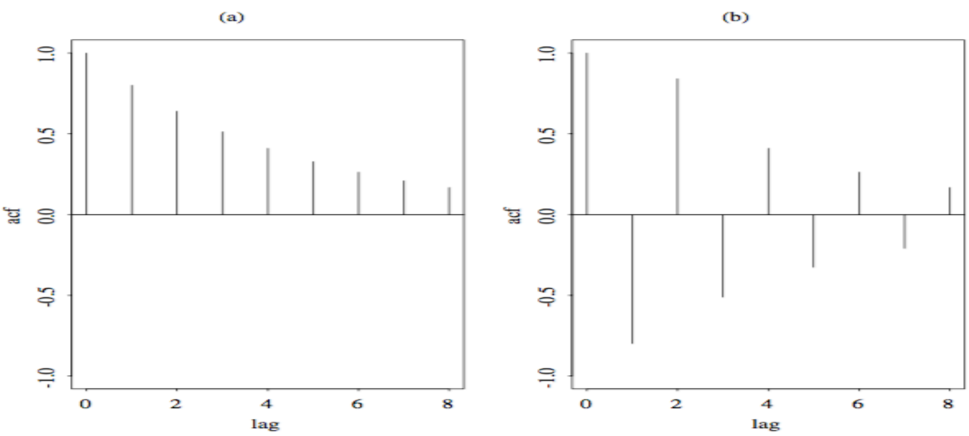
the necessary condition for AR model is that series r_t have to be weak stationarity.

Unconditional mean

Unconditional variance

Unconditional autocorrelations

Figure The autocorrelation function of an AR(1) model: (a) for ϕ_1 = 0.8, and (b) for ϕ_1 = −0.8.



AR (2) model

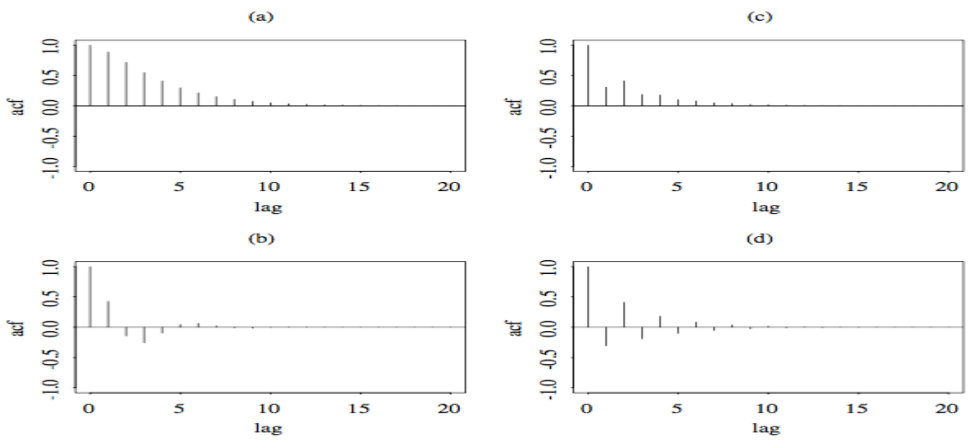
$$r_t = \phi_0 + \phi_1 r_{t-1} + \phi_2 r_{t-2} + a_t$$

Unconditional mean

Unconditional variance

Unconditional autocorrelations

Figure The autocorrelation function of an AR(2) model: (a) ϕ_1 = 1.2 and ϕ_2 = −0.35, (b) ϕ_1 =0.6 and ϕ_2 =−0.4,(c) ϕ_1 =0.2and ϕ_2 =0.35,(d) ϕ_1 =−0.2 and ϕ_2 =0.35.



2.4.2 Stationarity of AR(p) Model

In case of AR(p)

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

We can write down the Unconditional Mean as :

$$E(r_t) = \frac{\phi_o}{1 - \phi_1 - ... - \phi_p},$$

which the Polynomial equation can be expressed as

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

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

$$(1 - \phi_1 B - \phi_2 B^2 - ... - \phi_p B^P)\rho_l = 0$$

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

Theefore, 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 $l \in [0, ..., P]$

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 $l \in [0, ..., P]$

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

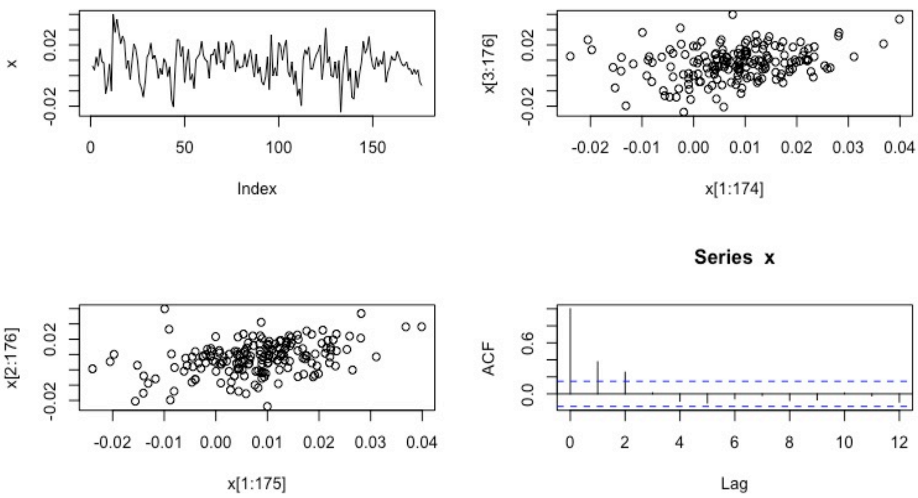
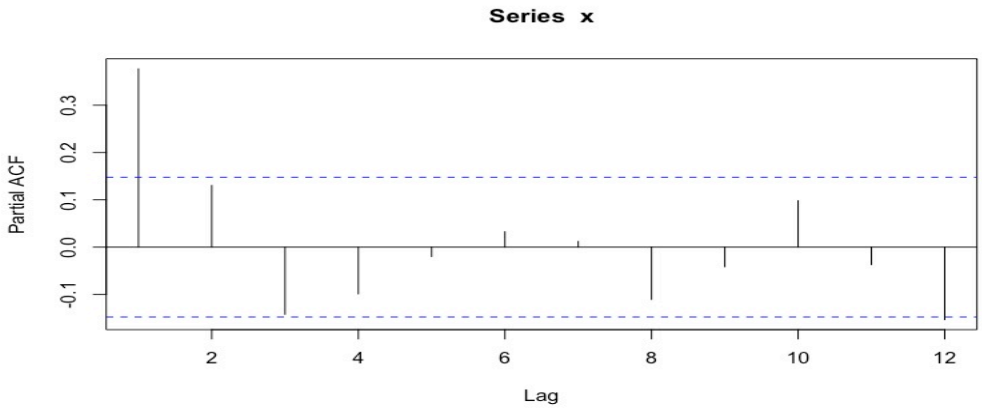


Figure: GDP growth, ACF and PACF (cont.)



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:

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

Stability and Stationarity Condition

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

we can write it as

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

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 $\mathbf{F}^{\mathbf{j}}$ provided all of the eigenvalues of $\mathbf{F}$ have modulus less than 1.

2.6 Finding Eigenvalue

λ is an eigenvalue of $\mathbf{F}$ and $\mathbf{x}$ is the eigenvector if

$$\mathbf{F}\mathbf{x} = \boldsymbol{\lambda}\mathbf{x}$$

Example: AR(2)

The eigenvalues of $\mathbf{F}$ 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 root may be real or complex. Complex roots induce periodic behavior in y_t. If λ_i is complex then

$$\lambda_i = a + bi$$

$$a = Rcos(\theta), b = Rsin(\theta)$$

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

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

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

t=p+1,...,T

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

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

where the residual term is

$$\widehat{a_t} = r_t - \widehat{r_t}$$

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

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

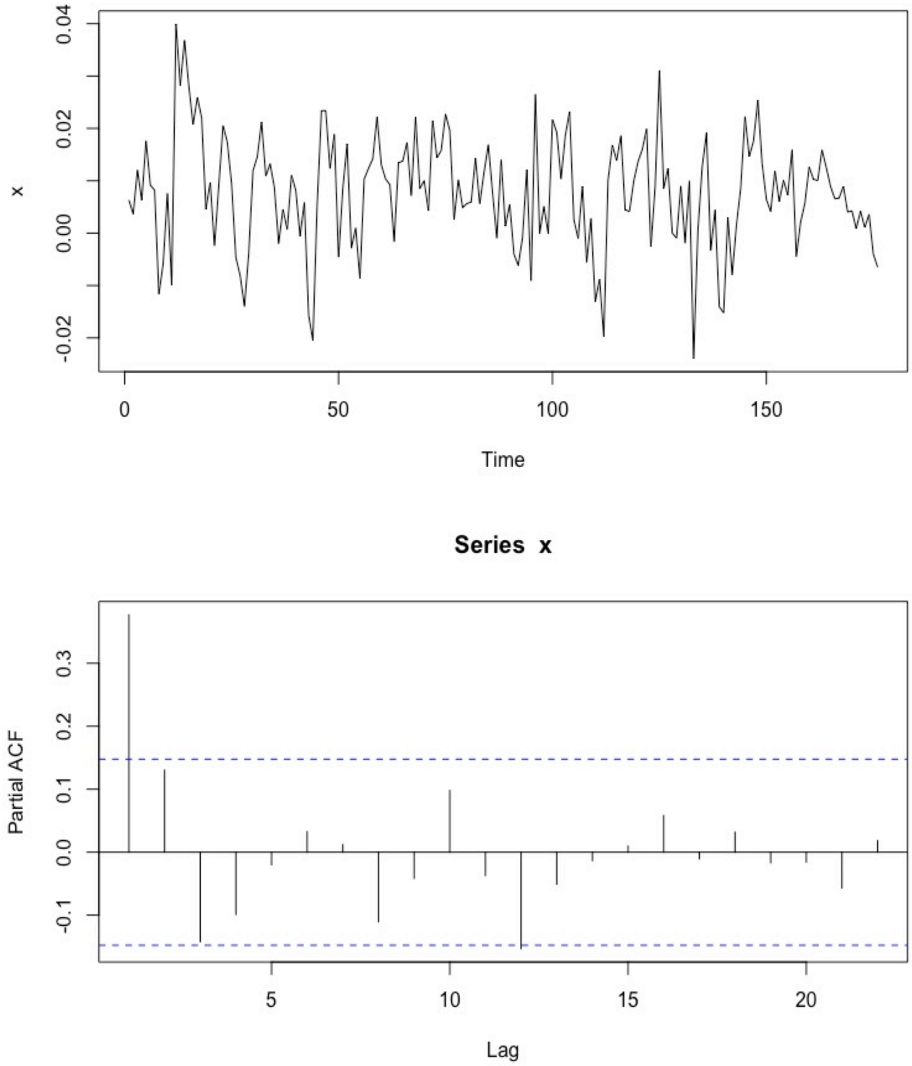
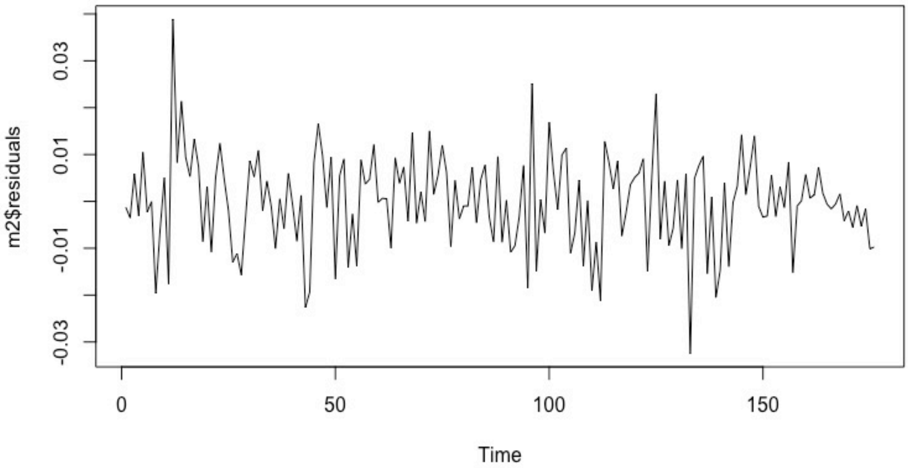
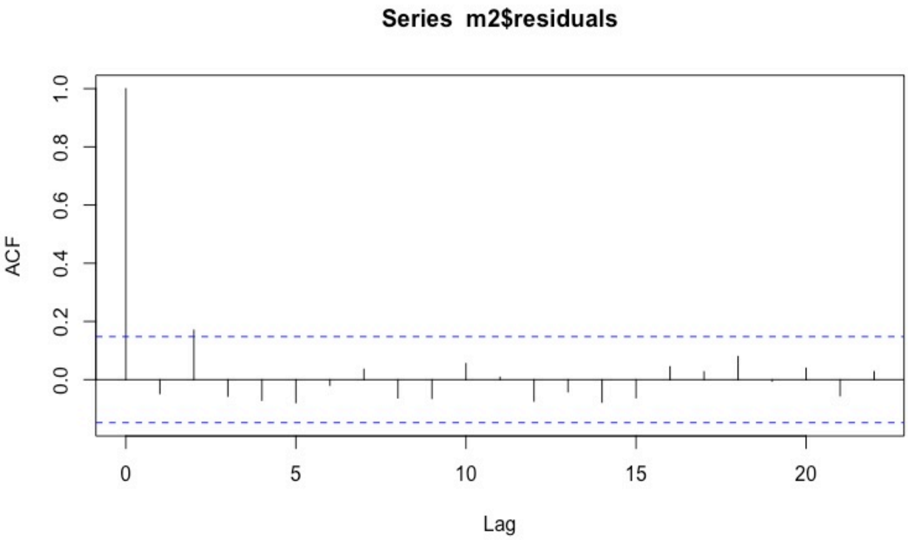


Figure: Residual Term from the estimated model and ACF of Residual Term





2.10 Moving-Average Models (MAs)

An AR model with infinite order can be written as:

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

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

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

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.

$E(r_t)$

$Var(r_t)$

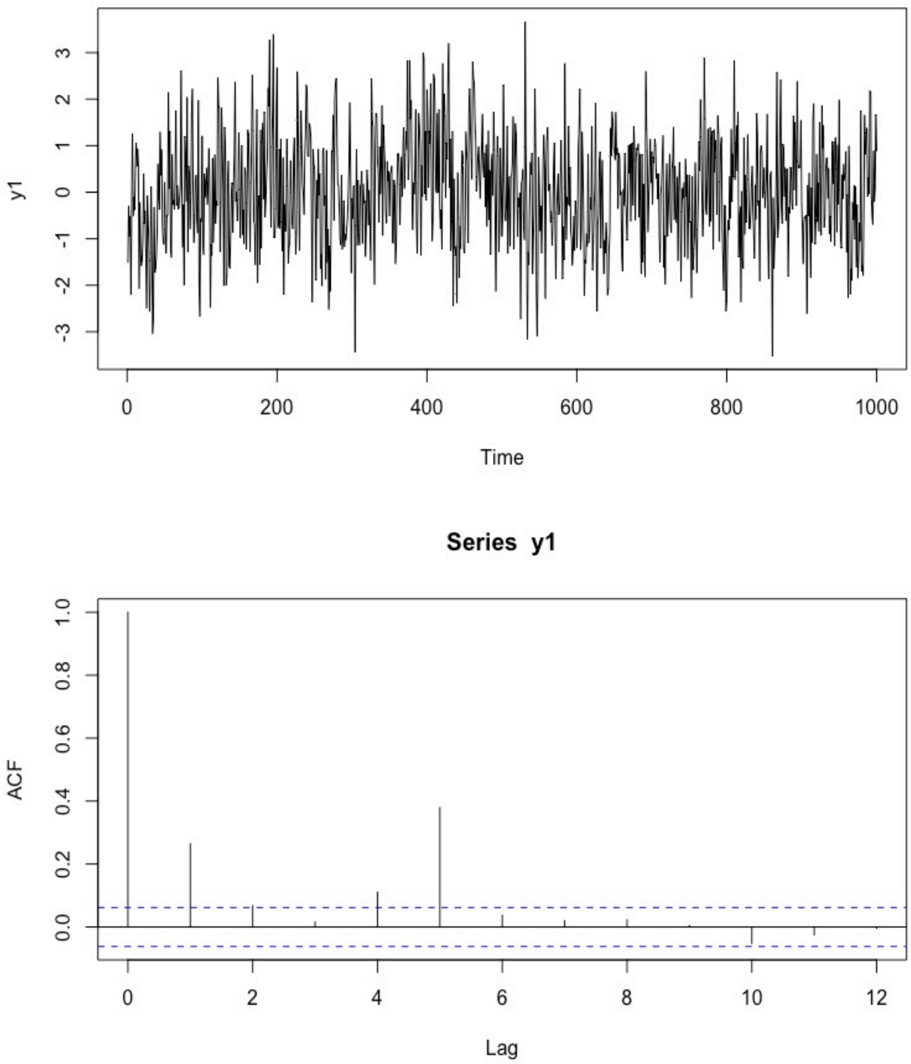
Autocorrelation Function

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 follwing:

Figure of the MA(5) model and its ACF



2.11 Parameter Estimation

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

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

t=q+1,...,T

The fitted model can be written as:

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

We can define the residual terms from the below equation:

$$\widehat{a_t} = r_t - \widehat{r_t}$$

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

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)
```

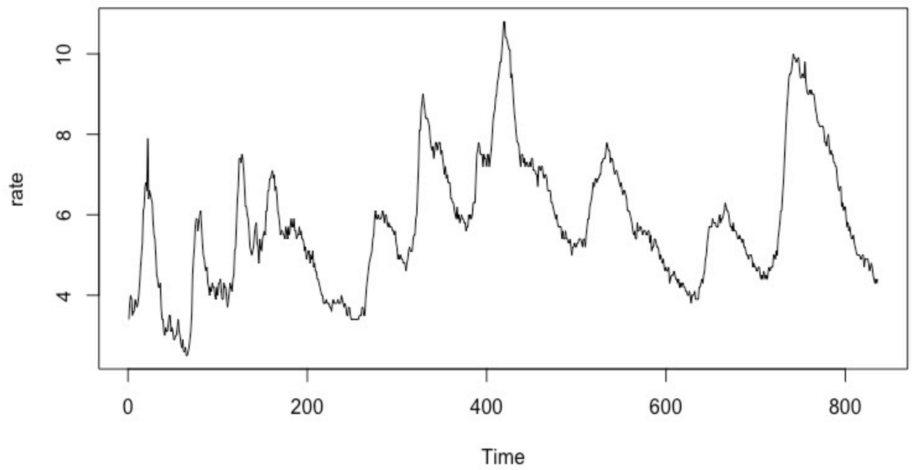
The main results:

```
> 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

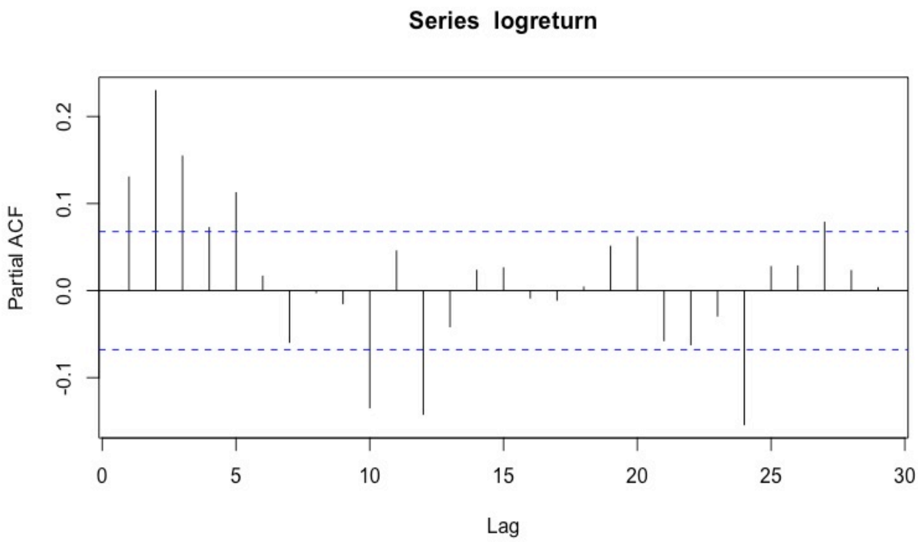


Figure: Residual Term Civilian Unemployment Rate from the AR(13) model

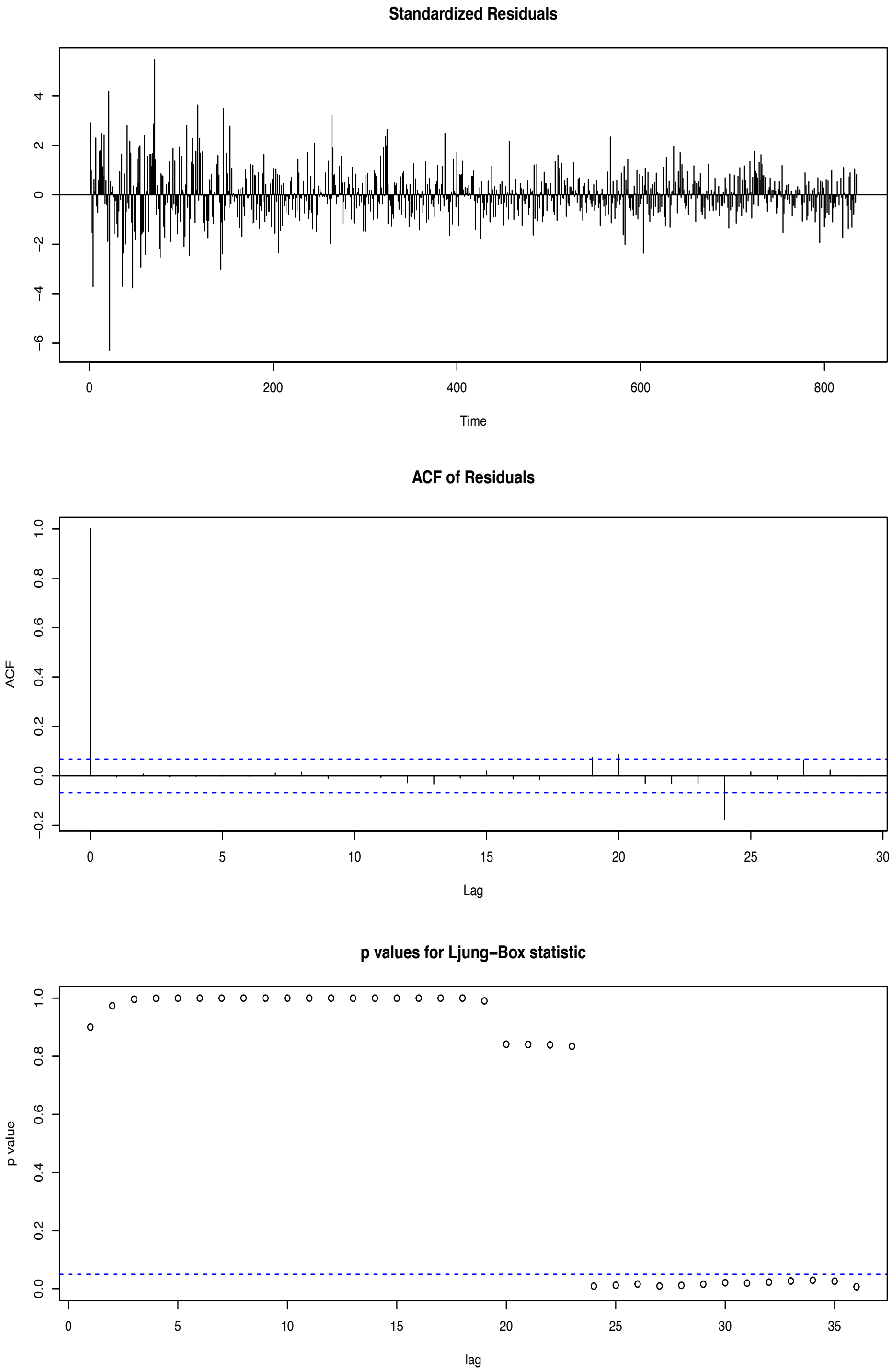
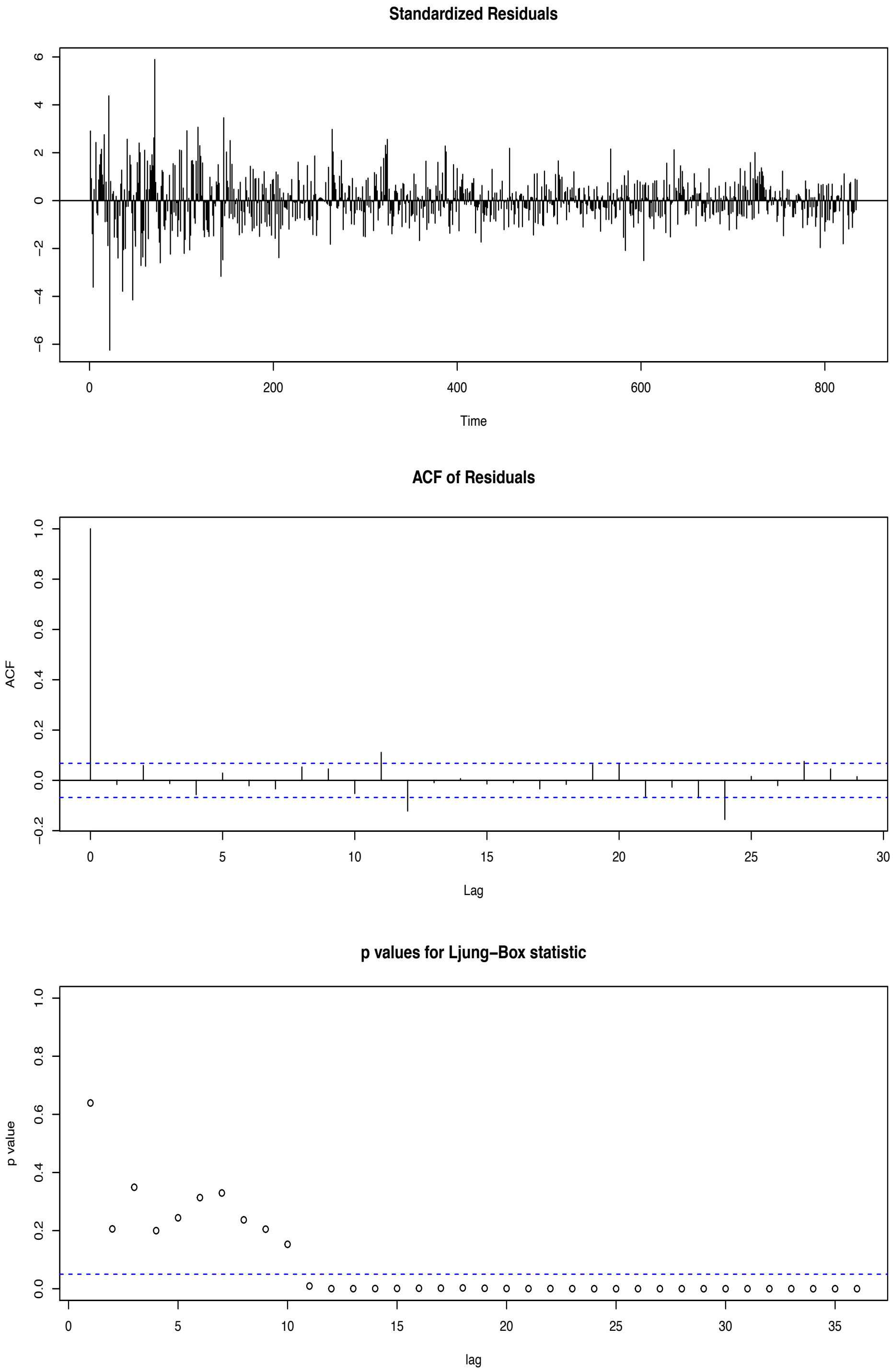


Figure: Residual Term Civilian Unemployment Rate from the ARMA(2,0,2)



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 varinace of r_t diverges to infinity as t increases.

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

Repeated Substitution shows:

2.19.2 Random Walk with Drift

Form:

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

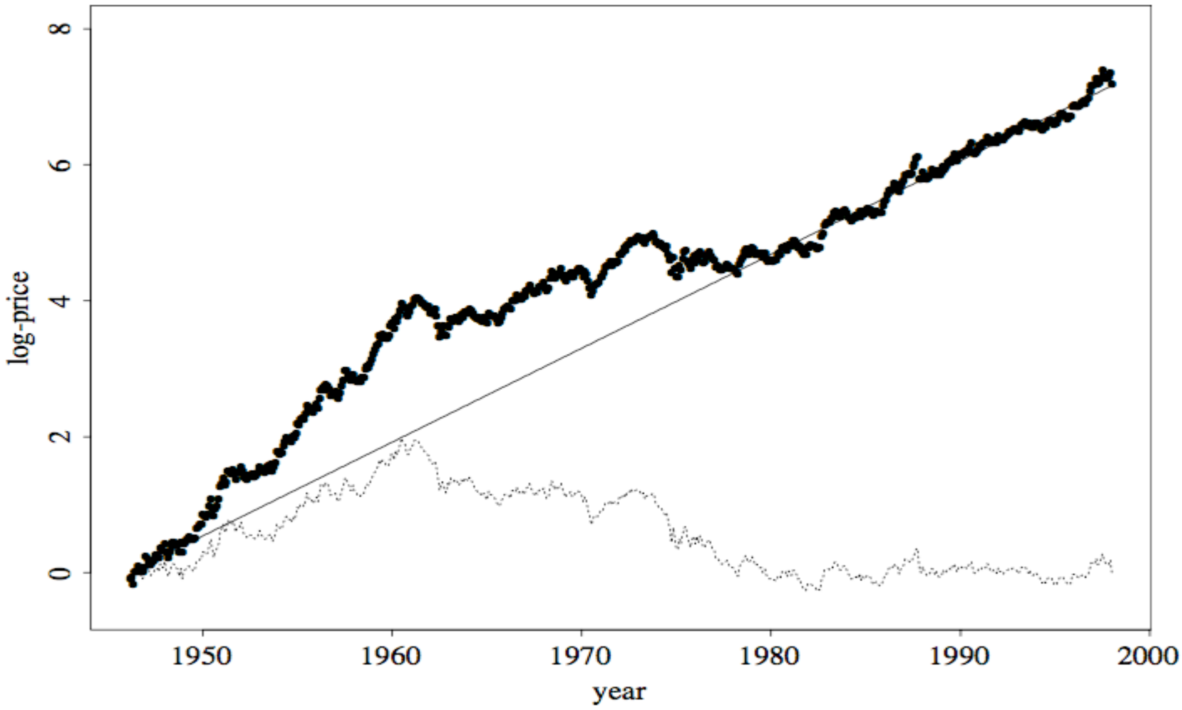
Has a unit root

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

1^{st} difference:

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

If p_t is the log-price, then the 1^{st} difference is simply the log return. Typically, 1^{st} 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 quaterly 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

♡ 1^{st} 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:

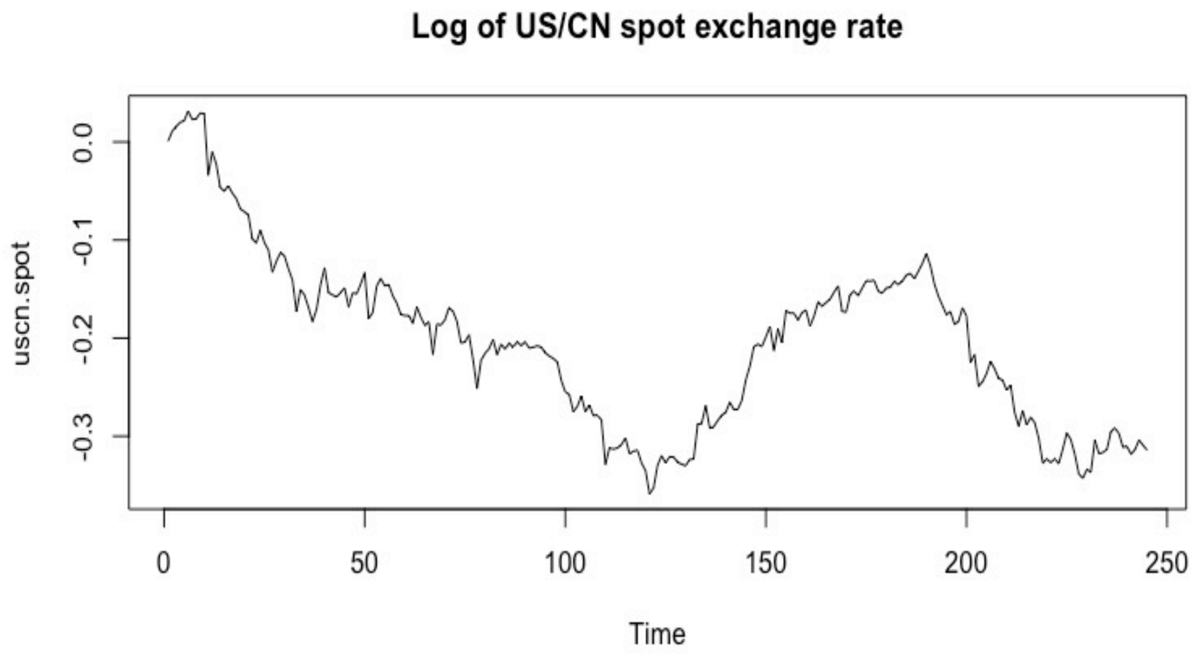
$$p_t = \phi_0 + \phi_1 p_{t-1} + e_t$$

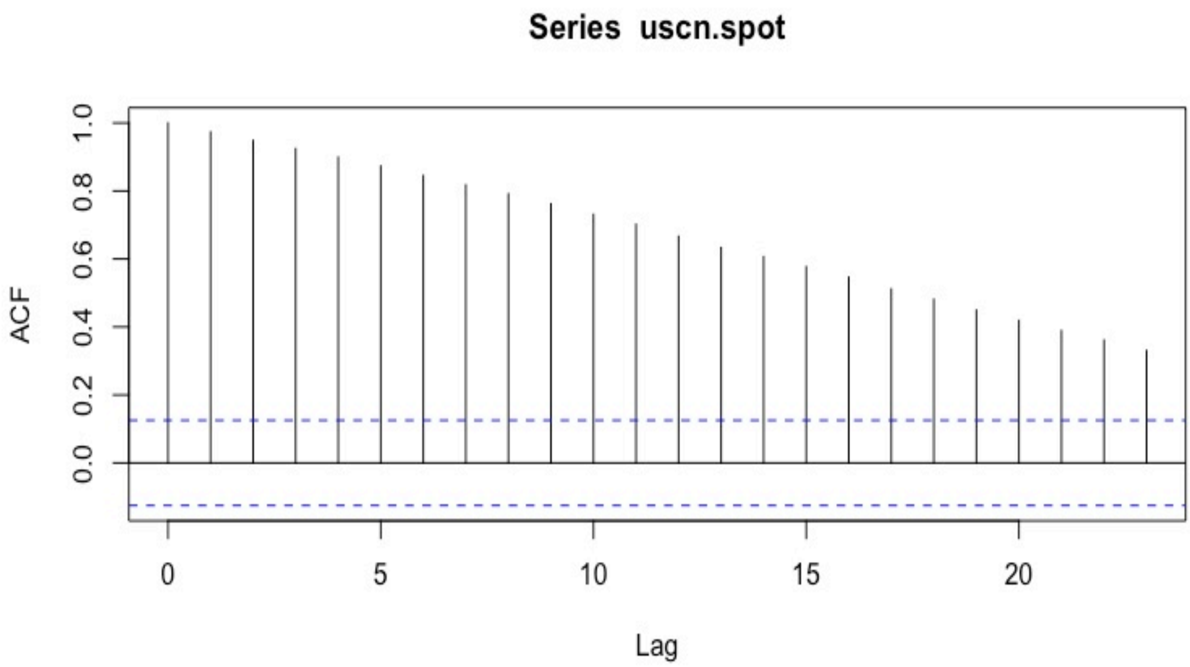
We can apply the following process:

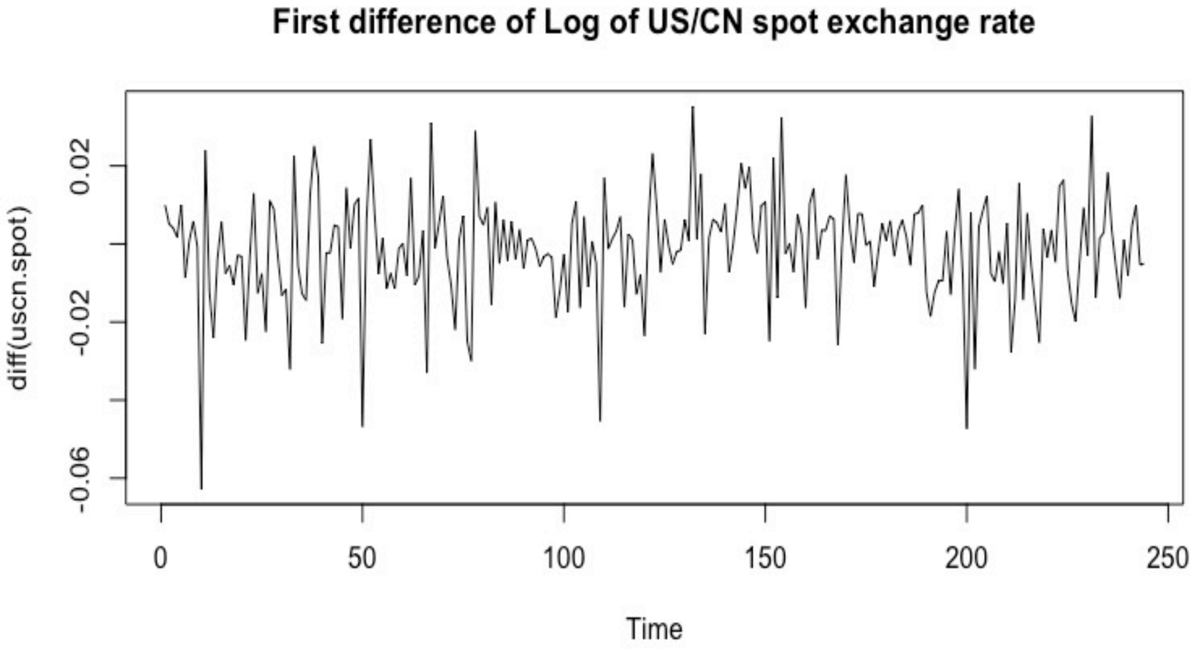
The Result from Unit-root Test

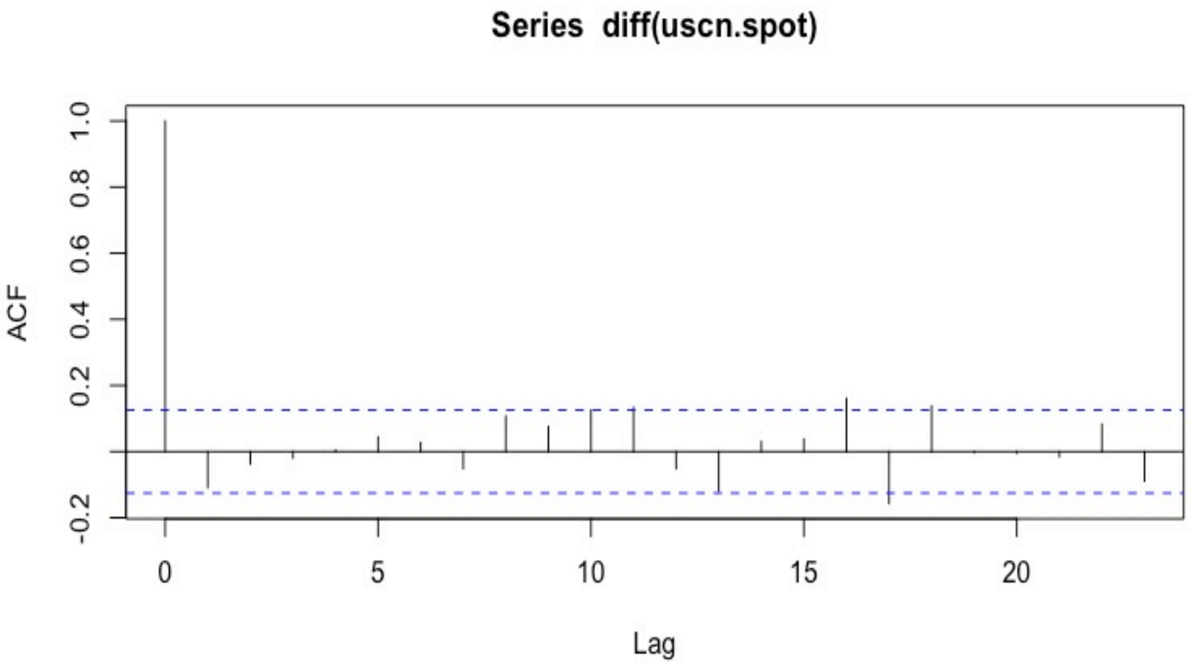
```
> setwd("/Users/wasinsiwasarit/Desktop/EE435")
> cat(rep("\n",50))  #clear R Console
> library(quantmod)
Loading required package: xts
Loading required package: zoo
Attaching package: ‘’zoo
The following objects are masked from ‘package:’base:
    as.Date, as.Date.numeric
Loading required package: 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=";")
```

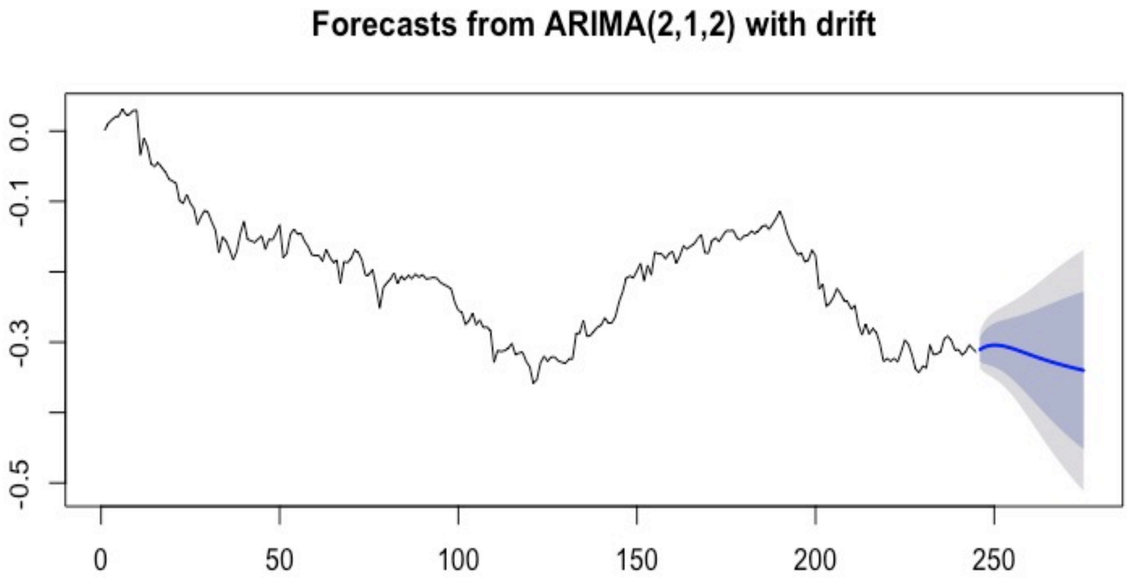
```
> uscn.spot = lexrates[,"USCNS"]
> plot.ts(uscn.spot,main="Log of US/CN spot exchange rate")
> xx = acf(uscn.spot)
> plot.ts(diff(uscn.spot),main="First difference of Log of US/CN spot exchange
    rate")
> xx = acf(diff(uscn.spot))
> m1=adfTest(uscn.spot, lags = 2, type = c("c"), title = NULL,
+              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(uscn.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(uscn.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
uscn.spot
0.0
-0.1
-0.2
-0.3
0
50
100
150
200
250
Time


Series uscn.spot
ACF
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
Lag


First difference of Log of US/CN spot exchange rate
diff(uscn.spot)
0.02
-0.02
-0.06
0
50
100
150
200
250
Time


Series diff(uscn.spot)
ACF
1.0
0.8
0.6
0.4
0.2
-0.2
0
5
10
15
20
Lag


Forecasts from ARIMA(2,1,2) with drift
0.0
-0.1
-0.3
-0.5
0
50
100
150
200
250

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 = year + 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	$\cdots$	Oct	Nov	Dec
1959	$y_{0,1}$	$y_{0,2}$	$y_{0,3}$	$\cdots$	$y_{0,10}$	$y_{0,11}$	$y_{0,12}$
1960	$y_{1,1}$	$y_{1,2}$	$y_{1,3}$	$\cdots$	$y_{1,10}$	$y_{1,11}$	$y_{1,12}$
1961	$y_{2,1}$	$y_{2,2}$	$y_{2,3}$	$\cdots$	$y_{2,10}$	$y_{2,11}$	$y_{2,12}$
1962	$y_{3,1}$	$y_{3,2}$	$y_{3,3}$	$\cdots$	$y_{3,10}$	$y_{3,11}$	$y_{3,12}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$

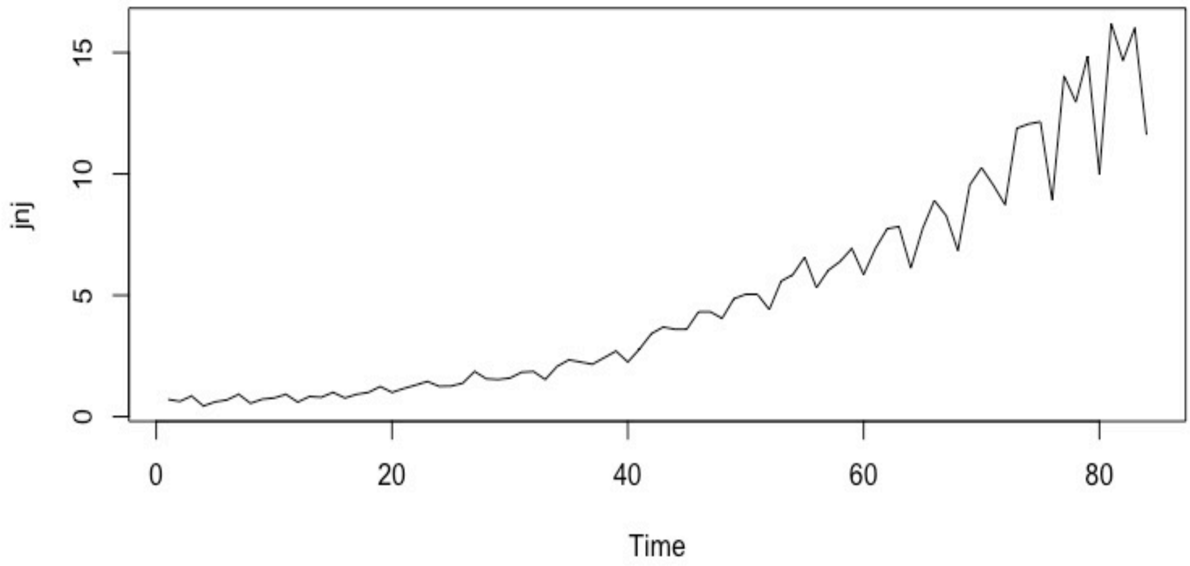
The Application of Program R

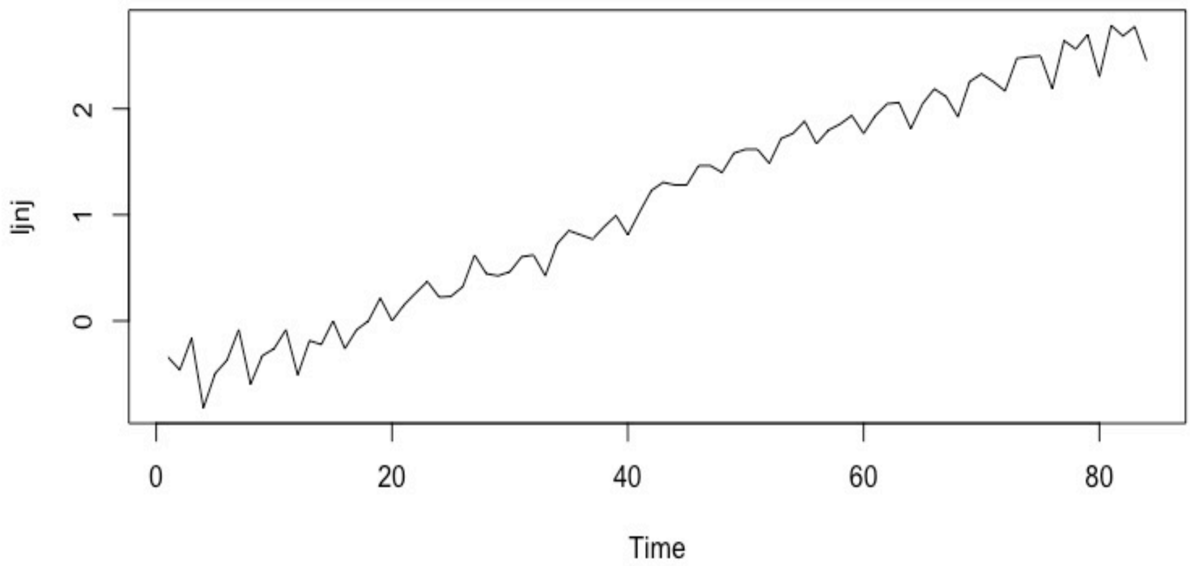
```
#EE435
setwd("/Users/wasinsiwasarit/Desktop/EE435")
cat(rep("\n",50))  #clear R Console

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

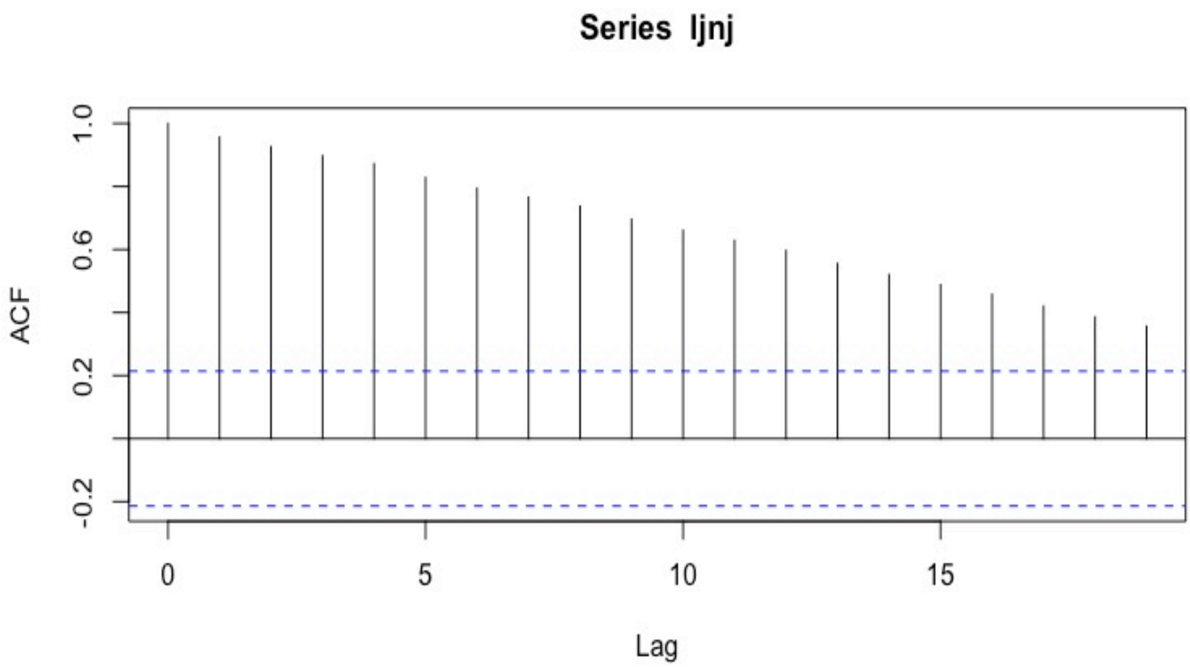
```
predict(m5,8)
```

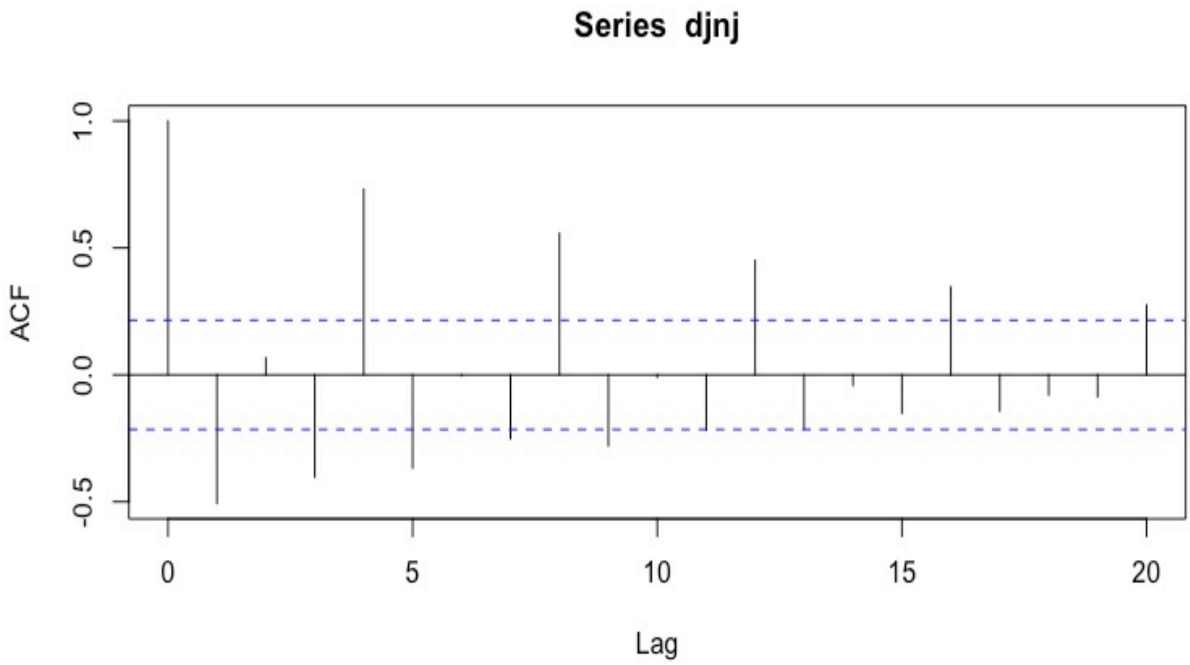
Results from the Program R

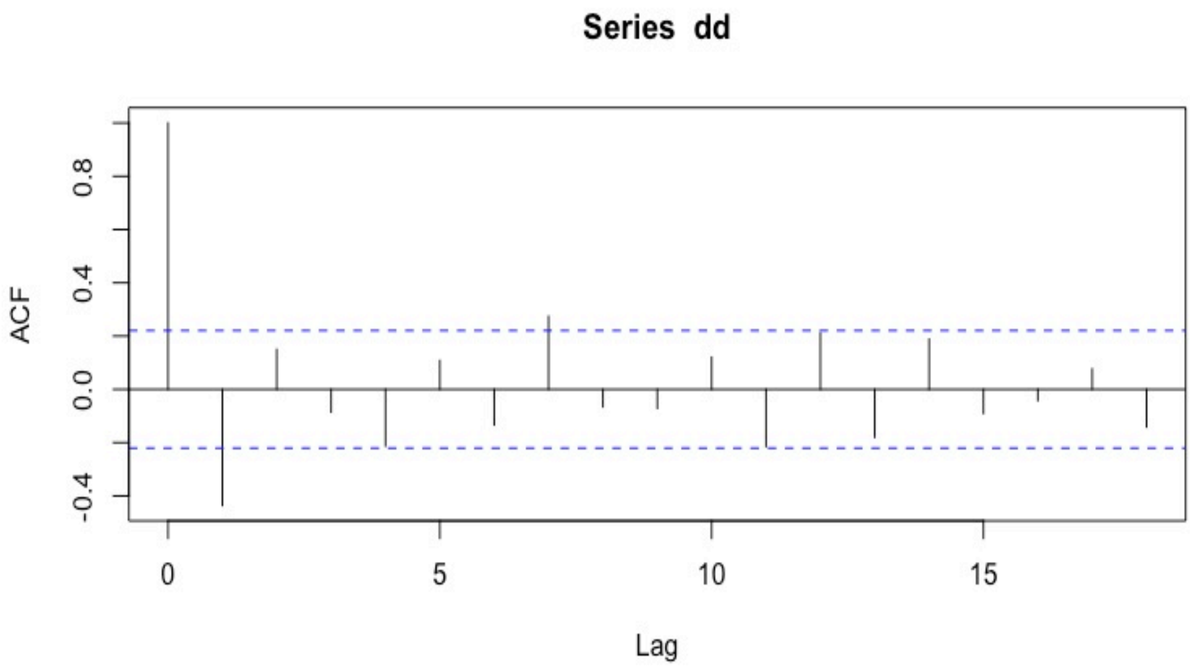
```
> da=read.table("q-earn-jnj.txt")
> jnj=da[,1]
> ts.plot(jnj)
> ljnj=log(jnj)
> ts.plot(ljnj)
> acf(ljnj)
> djnj=diff(ljnj)
> acf(djnj,lag=20)
> dd <- diff(djnj,4) ### seasonal difference
> acf(dd)
> m5 = arima(ljnj,order=c(0,1,1),seasonal=list(order=c(0,1,1),period=4))
> m5

Call:
arima(x = ljnj, 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
```

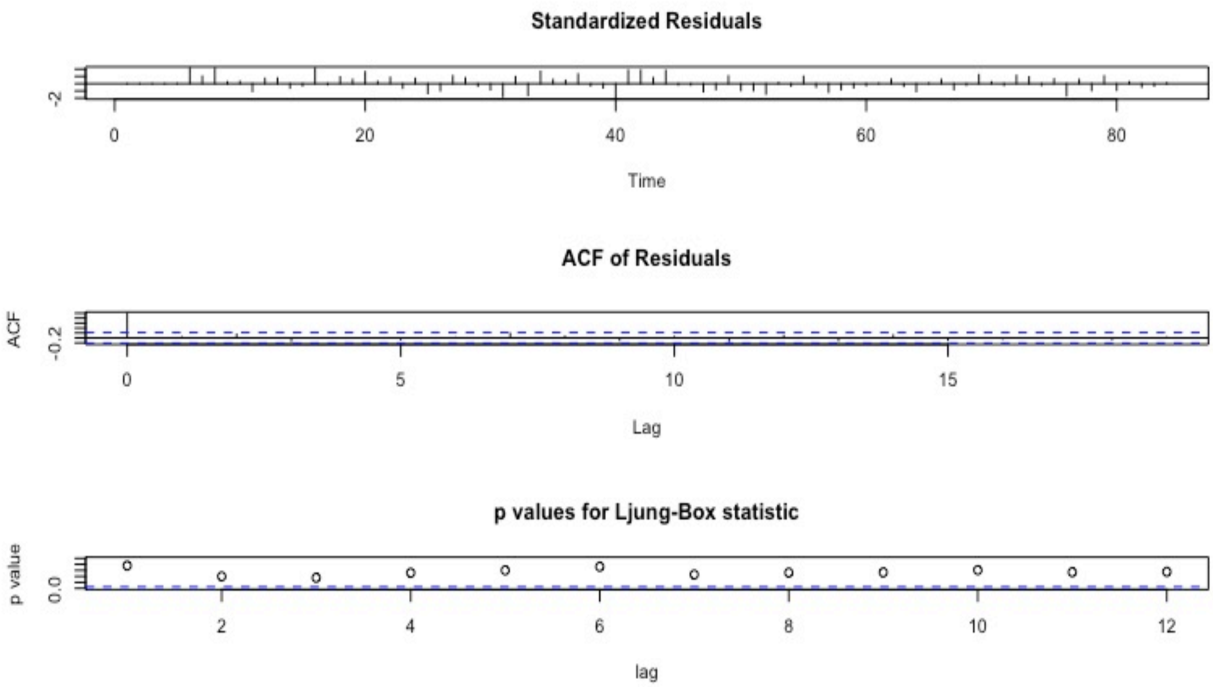

15
10
5
0
jnj
0
20
40
60
80
Time


2
1
0
ljnj
0
20
40
60
80
Time


Series ljnj
ACF
1.0
0.6
0.2
-0.2
0
5
10
15
Lag


Series djnj
ACF
1.0
0.5
0.0
-0.5
0
5
10
15
20
Lag


Series dd
ACF
0.8
0.4
0.0
-0.4
0
5
10
15
Lag



Standardized Residuals
-2
0
20
40
60
80
Time
ACF of Residuals
ACF
-0.2
0
5
10
15
Lag
p values for Ljung-Box statistic
p value
0.0
2
4
6
8
10
12
lag