Lecture Notes of Bus 41202 (Spring 2006)
Analysis of Financial Time Series
Ruey S. Tsay

Simple AR models (regression ?)

AR(1) model:

1. Form: $r_t = \phi_0 + \phi_1 r_{t-1} + a_t$, where ϕ_0 and ϕ_1 are real numbers, which are referred to as "parameters" (to be estimated from the data in an application). For example,

$$r_t = 0.005 + 0.2 r_{t-1} + a_t$$

2. Stationarity: necessary and sufficient condition $|\phi_1| < 1$. Why?
3. Mean: $E(r_t) = \frac{\phi_0}{1-\phi_1}$
4. Variance: $\mathrm{Var}(r_t) = \frac{\sigma_a^2}{1-\phi_1^2}$.
5. Autocorrelations: $\rho_1 = \phi_1, \rho_2 = \phi_1^2$, etc. In general, $\rho_k = \phi_1^k$ and ACF ρ_k decays exponentially as k increases,
6. Forecast (minimum squared error):

   (a) 1-step ahead forecast at time n, the forecast origin:

$$\hat{r}_n(1) = \phi_0 + \phi_1 r_n$$

   (b) 1-step ahead forecast error:

$$e_n(1) = r_{n+1} - \hat{r}_n(1) = a_{n+1}$$

Thus, a_{n+1} is the *un-predictable* part of r_{n+1}. It is the shock at time $n+1$!

(c) Variance of 1-step ahead forecast error:

$$\text{Var}[e_n(1)] = \text{Var}(a_{n+1}) = \sigma_a^2.$$

(d) 2-step ahead forecast:

$$\hat{r}_n(2) = \phi_0 + \phi_1 \hat{r}_n(1)$$

(e) 2-step ahead forecast error:

$$e_n(2) = r_{n+2} - \hat{r}_n(2) = a_{n+2} + \phi_1 a_{n+1}$$

(f) Variance of 2-step ahead forecast error:

$$\text{Var}[e_n(2)] = (1 + \phi_1^2)\sigma_a^2$$

which is greater than or equal to $\text{Var}[e_n(1)]$, implying that uncertainty in forecasts increases as the number of steps increases.

(g) Behavior of multi-step ahead forecasts.

7. A compact form: $(1 - \phi_1 B) r_t = \phi_0 + a_t$.

AR(2) model:

1. Form: $r_t = \phi_0 + \phi_1 r_{t-1} + \phi_2 r_{t-2} + a_t$, or

$$(1 - \phi_1 B - \phi_2 B^2) r_t = \phi_0 + a_t.$$

2. Stationarity condition: (factor of polynomial)

3. Characteristic equation: $(1 - \phi_1 x - \phi_2 x^2) = 0$

4. Mean: $E(r_t) = \frac{\phi_0}{1-\phi_1-\phi_2}$

5. ACF: $\rho_0 = 1$, $\rho_1 = \frac{\phi_1}{1-\phi_2}$,

$$\rho_\ell = \phi_1 \rho_{\ell-1} + \phi_2 \rho_{\ell-1}, \quad \ell \geq 2.$$

6. Stochastic business cycle: if $\phi_1^2 + 4\phi_2 < 0$, then r_t shows characteristics of business cycles with average length

$$k = \frac{2\pi}{\cos^{-1}[\phi_1/(2\sqrt{-\phi_2})]},$$

where the cosine inverse is stated in radian. If we denote the solutions of the polynomial as $a \pm bi$, where $i = \sqrt{-1}$, then we have $\phi_1 = 2a$ and $\phi_2 = -(a^2 + b^2)$ so that

$$k = \frac{2\pi}{\cos^{-1}(a/\sqrt{a^2+b^2})}.$$

In R or S-Plus, one can obtain $\sqrt{a^2+b^2}$ using the command **Mod**.

7. Forecasts: Similar to AR(1) models

Building an AR model

- Order specification

  1. Partial ACF: (naive, but effective)

     – Use consecutive fittings

     – See Text (p. 36) for details

- Key feature: PACF cuts off at lag p for an AR(p) model.
- Illustration

2. Akaike information criterion

$$AIC(\ell) = \ln(\tilde{\sigma}_\ell^2) + \frac{2\ell}{T},$$

for an AR(ℓ) model, where $\tilde{\sigma}_\ell^2$ is the MLE of residual variance.

Find the AR order with *minimum* AIC for $\ell \in [0, \cdots, P]$.

3. BIC criterion:

$$BIC = \ln(\tilde{\sigma}_\ell^2) + \frac{\ell \ln(T)}{T}.$$

- Needs a constant term? Check the sample mean.
- Estimation: least squares method or maximum likelihood method
- Model checking:
  1. Residual: obs minus the fit, i.e. 1-step ahead forecast errors at each time point.
  2. Residual should be close to white noise if the model is adequate. Use Ljung-Box statistics of residuals, but degrees of freedom is $m - g$, where g is the number of AR coefficients used in the model.
- Many software packages available, e.g. SCA, Splus, Eviews, SAS, SPSS, etc.

Example: US GNP growth rate series revisited.
R demonstration:

```
> setwd("C:/teaching/bs41202")
> library(fSeries)
Loading required package: fBasics

> da=read.table("dgnp82.dat")
> x=da[,1]
> plot(x,type='l')  % Plot not shown in this handout.
> title(main='Growth rate of U.S. GNP: 1947-1991')  % title of plot.
> acf(x,lag.max=12) % Compute ACF (not shown in this handout)
> pacf(x,lag.max=12) % Compute PACF (not shown in this handout)
> Box.test(x,lag=10,type='Ljung')

        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

> plot(m2$residuals,type='l')  % 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

> k=2*pi/acos(1.590253/1.913308)
> k
[1] 10.65638

> 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
```

S-Plus demonstration

```
> module(finmetrics)
> gnp=scan(file='dgnp82.dat')

> plot(gnp,type='l')
> acf(gnp,lag.max=12)
Call: acf(x = gnp, lag.max = 12)  % Plot not shown in the handout.

Autocorrelation matrix:
   lag     gnp
 1   0  1.0000
 2   1  0.3769
 3   2  0.2539
 4   3  0.0125
 5   4 -0.0859
 6   5 -0.1071
 7   6 -0.0575
 8   7 -0.0182
 9   8 -0.0772
10   9 -0.0702
11  10  0.0104
12  11 -0.0230
13  12 -0.0967
> acf(gnp,lag.max=12,type='partial') % Compute PACF
Call: acf(x = gnp, lag.max = 12, type = "partial")

Partial Correlation matrix:
   lag     gnp
 1   1  0.3769
 2   2  0.1304
 3   3 -0.1421
 4   4 -0.0988
 5   5 -0.0199
 6   6  0.0325
 7   7  0.0120
```

```
 8   8 -0.1106
 9   9 -0.0415
10  10  0.0981
11  11 -0.0370
12  12 -0.1533
> ord=ar(gnp,order.max=10) % Perform order selection via AIC
> ord$aic
 [1] 27.5691310  2.6081086  1.5895550  0.0000000  0.2734771  2.2034466
 [7]  4.0171066  5.9916210  5.8264833  7.5230025  7.8223499
> ord$order
[1] 3

> m1=arima.mle(gnp,model=list(order=c(3,0,0))) %This fit misses the mean.
> summary(m1)
Call: arima.mle(x = gnp, model = list(order = c(3, 0, 0)))
Method:  Maximum Likelihood  with likelihood conditional on  3 observati
ons

ARIMA order:  3 0 0

         Value Std. Error t-value % No intercept because the program assumes it is zero.
ar(1)  0.45420    0.07597  5.9780
ar(2)  0.26680    0.08095  3.2960
ar(3) -0.03817    0.07597 -0.5024

Variance-Covariance Matrix:
             ar(1)        ar(2)        ar(3)
ar(1)  0.005771926 -0.002566306 -0.001441892
ar(2) -0.002566306  0.006552753 -0.002566306
ar(3) -0.001441892 -0.002566306  0.005771926

Estimated innovations variance:  0.0001

Optimizer has  converged
Convergence Type: relative function convergence
AIC: -1085.0397

> x=gnp-mean(gnp)             % Remove sample mean.
> m1=arima.mle(x,model=list(order=c(3,0,0)))
> summary(m1)
Call: arima.mle(x = x, model = list(order = c(3, 0, 0)))
Method:  Maximum Likelihood  with likelihood conditional on  3 observati
ons

ARIMA order:  3 0 0
```

```
        Value Std. Error t-value
ar(1)  0.3509    0.07523   4.664  % Fitted model is
ar(2)  0.1809    0.07863   2.301  % x(t)=0.351x(t-1)+0.181x(t-2)-0.144x(t-3)+a(t).
ar(3) -0.1443    0.07523  -1.919

Variance-Covariance Matrix:
              ar(1)        ar(2)         ar(3)
ar(1)  0.0056599161 -0.001877448 -0.0007529176
ar(2) -0.0018774480  0.006182526 -0.0018774480
ar(3) -0.0007529176 -0.001877448  0.0056599161

Estimated innovations variance:  0.0001

Optimizer has  converged
Convergence Type: relative function convergence
AIC: -1104.1574
> names(m1)
 [1] "model"     "var.coef"  "method"    "series"    "aic"
 [6] "loglik"    "sigma2"    "n.used"    "n.cond"    "converged"
[11] "conv.type" "call"
> names(m1$model)
[1] "order" "ar"    "ndiff"

> m1$model$ar
[1]  0.3509107  0.1809056 -0.1443412
>
> arima.diag(m1)  % Model checking, plots not shown.

> p1=c(1,-m1$model$ar) % Further analysis of the fitted model.
> roots=polyroot(p1)
> roots
[1]  1.582837+1.057071e+000i -1.912355-6.609277e-017i
[3]  1.582837-1.057071e+000i
> Mod(roots)
[1] 1.903359 1.912355 1.903359
> k=2*pi/acos(1.582837/1.903359)
> k
[1] 10.67098
>

> arima.forecast(x,m1$model,8)     % prediction
$mean:
[1] -0.00651901645 -0.00317061250 -0.00023632985  0.00028445018
[5]  0.00051471315  0.00026618912  0.00014546524  0.00002490612

$std.err:
```

```
[1] 0.009779314 0.010363943 0.010782026 0.010784985 0.010785783
[6] 0.010791060 0.010791857 0.010792592
```

Moving-average (MA) model

Model with finite time lags of memory!

Some daily stock returns have minor serial correlations. Can be modeled as MA or AR models.

<u>MA(1) model</u>

- Form: $r_t = \mu + a_t - \theta a_{t-1}$
- Stationarity: always stationary.
- Mean (or expectation): $E(r_t) = \mu$
- Variance: $\text{Var}(r_t) = (1 + \theta^2)\sigma_a^2$.
- Autocovariance:
  1. Lag 1: $\text{Cov}(r_t, r_{t-1}) = -\theta\sigma_a^2$
  2. Lag ℓ: $\text{Cov}(r_t, r_{t-\ell}) = 0$ for $\ell > 1$.

  Thus, r_t is not related to $r_{t-2}, r_{t-3}, \cdots$.
- ACF: $\rho_1 = \frac{-\theta}{1+\theta^2}$, $\rho_\ell = 0$ for $\ell > 1$.

  Finite memory! MA(1) models do not remember what happen two time periods ago.
- Forecast (at origin $t = n$):

1. 1-step ahead: $\hat{r}_n(1) = \mu - \theta a_n$. Why? Because at time n, a_n is known, but a_{n+1} is not.
2. 1-step ahead forecast error: $e_n(1) = a_{n+1}$ with variance σ_a^2.
3. Multi-step ahead: $\hat{r}_n(\ell) = \mu$ for $\ell > 1$.

   Thus, for an MA(1) model, the multi-step ahead forecasts are just the mean of the series. Why? Because the model has memory of 1 time period.
4. Multi-step ahead forecast error:

$$e_n(\ell) = a_{n+\ell} - \theta a_{n+\ell-1}$$

5. Variance of multi-step ahead forecast error:

   $(1 + \theta^2)\sigma_a^2 =$ variance of r_t.

- Invertibility:
  - Concept: r_t is a proper linear combination of a_t and the past observations $\{r_{t-1}, r_{t-2}, \cdots\}$.
  - Why is it important? It provides a simple way to obtain the shock a_t.

    For an invertible model, the dependence of r_t on $r_{t-\ell}$ converges to zero as ℓ increases.
  - Condition: $|\theta| < 1$.
  - Invertibility of MA models is the dual property of stationarity for AR models.

MA(2) model

- Form: $r_t = \mu + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2}$. or

$$r_t = \mu + (1 - \theta_1 B - \theta_2 B^2) a_t.$$

- Stationary with $E(r_t) = \mu$.
- Variance: $\mathrm{Var}(r_t) = (1 + \theta_1^2 + \theta_2^2)\sigma_a^2$.
- ACF: $\rho_2 \neq 0$,but $\rho_\ell = 0$ for $\ell > 2$.
- Forecasts go the the mean after 2 periods.

Building an MA model

- Specification: Use sample ACF

  Sample ACFs are all small after lag q for an MA(q) series. (See test of ACF.)

- Constant term? Check the sample mean.
- Estimation: use maximum likelihood method
  - Conditional: Assume $a_t = 0$ for $t \leq 0$
  - Exact: Treat a_t with $t \leq 0$ as parameters, estimate them to obtain the likelihood function.

  Exact method is preferred, but it is more computing intensive.

- Model checking: examine residuals (to be white noise)

- Forecast: use the residuals as $\{a_t\}$ (which can be obtained from the data and fitted parameters) to perform forecasts.

Example:Daily log return of the value-weighted index

R demonstration

```
> setwd("C:/teaching/bs41202")
> library(fSeries)
> da=read.table("d-ibmvwew6202.txt")
> dim(da)
[1] 10194     4

> vw=log(1+da[,3])*100  % Compute percentage log returns of the vw index.
> acf(vw,lag.max=10)  % ACF plot is not shon in this handout.
> m1=arima(vw,order=c(0,0,1))  % fits an MA(1) model
> m1

Call:
arima(x = vw, order = c(0, 0, 1))

Coefficients:
         ma1  intercept
      0.1465     0.0396 % The model is vw(t) = 0.0396+a(t)+0.1465a(t-1).
s.e.  0.0099     0.0100

sigma^2 estimated as 0.7785:  log likelihood = -13188.48,  aic = 26382.96
> tsdiag(m1)
> predict(m1,5)
$pred
Time Series:
Start = 10195
End = 10199
Frequency = 1
[1] 0.05036298 0.03960887 0.03960887 0.03960887 0.03960887

$se
Time Series:
Start = 10195
End = 10199
Frequency = 1
[1] 0.8823290 0.8917523 0.8917523 0.8917523 0.8917523
```

S-Plus demonstration

```
> vw=d6202[,3]  % Identify the vw-index returns.
> lnvw=log(1+vw) % compute log returns.
```

```
> acf(lnvw,lag.max=10) % ACF plot i snot shown in this handout.
Call: acf(x = lnvw, lag.max = 10)

Autocorrelation matrix:
   lag    lnvw
 1   0  1.0000
 2   1  0.1402
 3   2 -0.0120
 4   3 -0.0027
 5   4  0.0029
 6   5  0.0075
 7   6 -0.0149
 8   7 -0.0066
 9   8 -0.0034
10   9 -0.0085
11  10 -0.0074
> length(lnvw)
[1] 10194
> x1=rep(1,10194)  % Create a constant to handle non-zero mean
> m1=arima.mle(lnvw,xreg=x1,model=list(order=c(0,0,1)))
> summary(m1)
Call: arima.mle(x = lnvw, model = list(order = c(0, 0, 1)), xreg = x1)
Method:  Maximum Likelihood  with likelihood conditional on  0 observati
ons

ARIMA order:  0 0 1

           Value Std. Error t-value
ma(1) -0.1465000   0.009797  -14.96
   x1  0.0003962         NA      NA % Model is vw = .000396+a(t)+0.1465a(t-1)

Variance-Covariance Matrix:
              ma(1)
ma(1) 0.00009599039

Estimated innovations variance:  0.0001

Optimizer has  converged
Convergence Type: relative function convergence
AIC: -67509.2476
> arima.diag(m1)  % Plots not shown in this handout.

> arima.forecast(lnvw,model=m1$model,6)
$mean:
[1] 0.0001581654 0.0000000000 0.0000000000 0.0000000000 0.0000000000
[6] 0.0000000000   % Need to add the constant 0.000396 to the forecast.
```

```
$std.err:
[1] 0.008830056 0.008924361 0.008924361 0.008924361 0.008924361
[6] 0.008924361
```

Mixed ARMA model: A compact form for flexible models. Focus on the ARMA(1,1) model for

1. simplicity
2. useful for understanding GARCH models in Ch. 3 for volatility modeling.

ARMA(1,1) model

- Form: $(1-\phi_1 B)r_t = \phi_0 + (1-\theta B)a_t$ or

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

  A combination of an AR(1) on the LHS and an MA(1) on the RHS.

- Stationarity: same as AR(1)
- Invertibility: same as MA(1)
- Mean: as AR(1), i.e. $E(r_t) = \frac{\phi_0}{1-\phi_1}$
- Variance: given in the text
- ACF: Satisfies $\rho_k = \phi_1 \rho_{k-1}$ for $k > 1$, but

$$\rho_1 = \phi_1 - [\theta_1 \sigma_a^2 / \mathrm{Var}(r_t)] \neq \phi_1.$$

  This is the difference between AR(1) and ARMA(1,1) models.

- PACF: does not cut off at finite lags.

Building an ARMA(1,1) model

- Specification: use EACF or AIC
- What is EACF? How to use it? [See text].
- Estimation: cond. or exact likelihood method
- Model checking: as before
- Forecast: MA(1) affects the 1-step ahead forecast. Others are similar to those of AR(1) models.

Three model representations:

- ARMA form: compact, useful in estimation and forecasting
- AR representation: (by long division)

$$r_t = \phi_0 + a_t + \pi_1 r_{t-1} + \pi_2 r_{t-2} + \cdots$$

  It tells how r_t depends on its past values.

- MA representation: (by long division)

$$r_t = \mu + a_t + \psi_1 a_{t-1} + \psi_2 a_{t-2} + \cdots$$

  It tells how r_t depends on the past shocks.

For a stationary series, ψ_i converges to zero as $i \to \infty$. Thus, the effect of any shock is transitory.

The MA representation is particularly useful in computing variances of forecast errors.

For a ℓ-step ahead forecast, the forecast error is

$$e_n(\ell) = a_{n+\ell} + \psi_1 a_{n+\ell-1} + \cdots + \psi_{\ell-1} a_{n+1}.$$

The variance of forecast error is

$$\mathrm{Var}[e_n(\ell)] = (1 + \psi_1^2 + \cdots + \psi_{\ell-1}^2)\sigma_a^2.$$

Unit-root Nonstationarity

Random walk

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

  $$p_t = \sum_{i=0}^{\infty} a_{t-i} = \sum_{i=0}^{\infty} \psi_i a_{t-i}$$

  where $\psi_i = 1$ for all i. Thus, ψ_i does not converge to zero. The effect of any shock is permanent.

Random walk with drift

- Form: $p_t = \mu + p_{t-1} + a_t, \quad \mu \neq 0.$

- Has a unit root
- Nonstationary
- Strong memory
- Has a time trend with slope μ. Why?

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 quarterly series and $s = 12$ for monthly series.

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

Meaning of the constant term in a model

- MA model: mean
- AR model: related to mean
- 1st differenced: time slope, etc.

Practical implication in financial time series

Example: Monthly log returns of General Electrics (GE) from 1926 to 1999 (74 years)

Sample mean: 1.04%, std($\hat{\mu}$) = 0.26

Very significant!

is about 12.45% a year

$1 investment in the beginning of 1926 is worth

- annual compounded payment: $5907
- quarterly compounded payment: $8720
- monthly compounded payment: $9570
- Continuously compounded?

Unit-root test (for self study only)

Let p_t be the log price of an asset. To test that p_t is not predictable (i.e. has a unit root), two models are commonly employed:

$$\begin{aligned} p_t &= \phi_1 p_{t-1} + e_t \\ p_t &= \phi_0 + \phi_1 p_{t-1} + e_t. \end{aligned}$$

The hypothesis of interest is $H_o : \phi_1 = 1$ vs $H_a : \phi_1 < 1$.

Dickey-Fuller test is the usual t-ratio of the OLS estimate of ϕ_1 being 1. This is the DF unit-root test. The t-ratio, however, has a non-standard limiting distribution.

Let $\Delta p_t = p_t - p_{t-1}$. Then, the augmented DF unit-root test for an AR(p) model is based on

$$\Delta p_t = c_t + \beta p_{t-1} + \sum_{i=1}^{p-1} \phi_i \Delta p_{t-i} + e_t.$$

The t-ratio of the OLS estimate of β is the ADF unit-root test statistic. Again, the statistic has a non-standard limiting distribution.

Example: Consider the log series of U.S. quaterly real GDP series from 1947.I to 2005.IV. (Federal Reserve Bank of St. Louis).

R demonstration

```
> help(UnitrootTests)  % See the tests available

> adfTest(gdp,lags=4,type=c("c"))  % Assume an AR(4) model.

Title:
Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 4
  STATISTIC:
    Dickey-Fuller: -1.1199
  P VALUE:
    0.6397    % Cannot reject a unit root.

*** A more careful analysis
> x=diff(gdp) % Take the first difference
> ord=ar(x)  % Find AR models for x series.
> ord

Call:
ar(x = x)

Coefficients:
      1        2        3        4
 0.3021   0.1311  -0.0856  -0.1060

Order selected 4  sigma^2 estimated as  8.592e-05
> adfTest(gdp,lags=5,type=c("c"))

Title:
Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 5
  STATISTIC:
```

```
    Dickey-Fuller: -1.1339
  P VALUE:
    0.6345
```

S-Plus demonstration

```
> da=read.table("r-gdp05.txt")
> dim(da)
[1] 236   4
> plot(da[,4],type='l')
> module(finmetrics)

> gdp=log(da[,4])
> plot(gdp,type='l')

> x=diff(gdp)  % take the first difference

> ord=ar(x)
> ord
$order:
[1] 4

> adf=unitroot(gdp,trend='c',lags=5,method='adf')
> adf

Test for Unit Root: Augmented DF Test

Null Hypothesis: there is a unit root
   Type of Test: t-test
 Test Statistic: -1.12
        P-value: 0.7083

Coefficients:
    lag1    lag2    lag3    lag4    lag5 constant
 -0.0012  0.2954  0.1358 -0.0864 -0.1108  0.0168

Degrees of freedom: 231 total; 225 residual
Residual standard error: 0.009283
```