

Lecture Note: Analysis of Financial Time Series

Spring 2006, Ruey S. Tsay

Seasonal Time Series

- TS with periodic pattern, e.g. quarterly earnings
- Useful in weather-related derivative pricing
- Useful in analysis of transactions data (high-frequency data), e.g. U-shaped pattern in intraday data

Example Demand of electricity of a manufacturing sector of U.S. from 1972 to 1993. The data are logged usage on the 15th day of each month. See Figure 1.

Example Quarterly earnings of Johnson & Johnson
See the time plot, Figures 2 and 3, and sample ACFs

Multiplicative model

Airline model (for quarterly series)

- Form:

$$r_t - r_{t-1} - r_{t-4} + r_{t-5} = a_t - \theta_1 a_{t-1} - \theta_4 a_{t-4} + \theta_1 \theta_4 a_{t-5}$$

or

$$(1 - B)(1 - B^4)r_t = (1 - \theta_1 B)(1 - \theta_4 B^4)a_t$$

- Define the differenced series w_t as

$$w_t = r_t - r_{t-1} - r_{t-4} + r_{t-5} = (r_t - r_{t-1}) - (r_{t-4} - r_{t-5}).$$

Demand of Electricity: 15th day of every month, 1972–1993

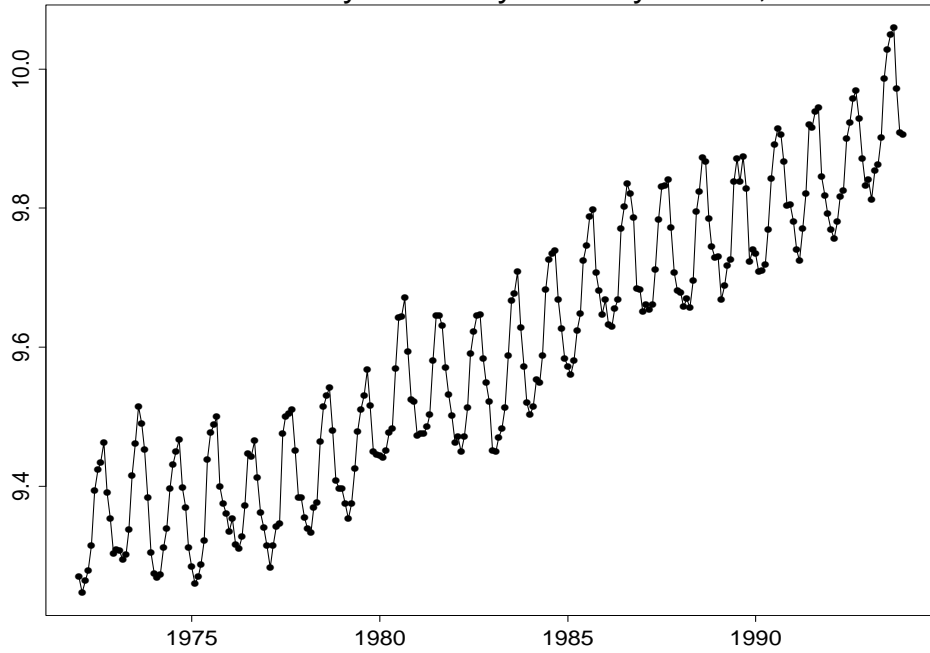


Figure 1: Time plot of electricity demand of an industrial sector: 15th day of each month from 1972 to 1993.

Quarterly earnings of JNJ: 1960–1980

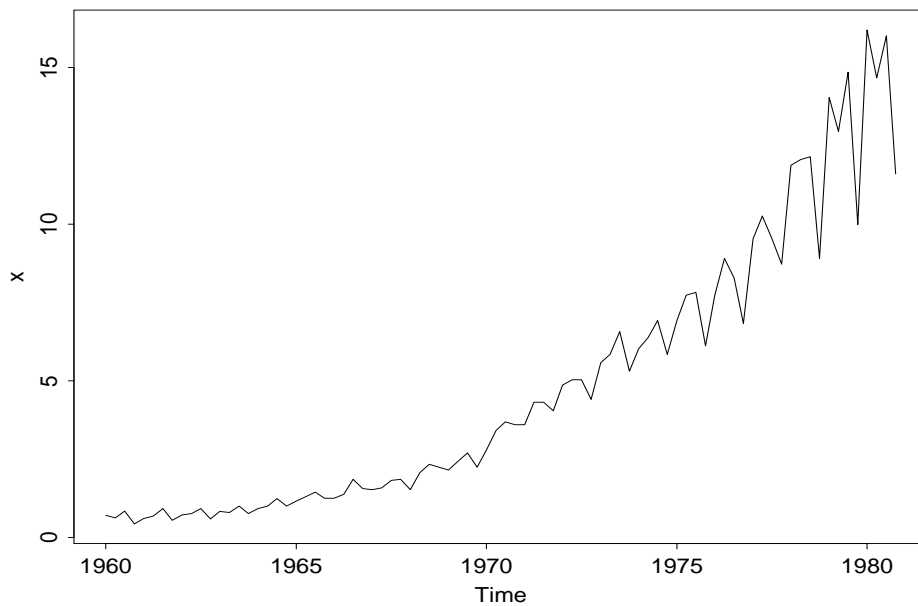


Figure 2: Time plot of quarterly earnings of Johnson and Johnson: 1960-1980

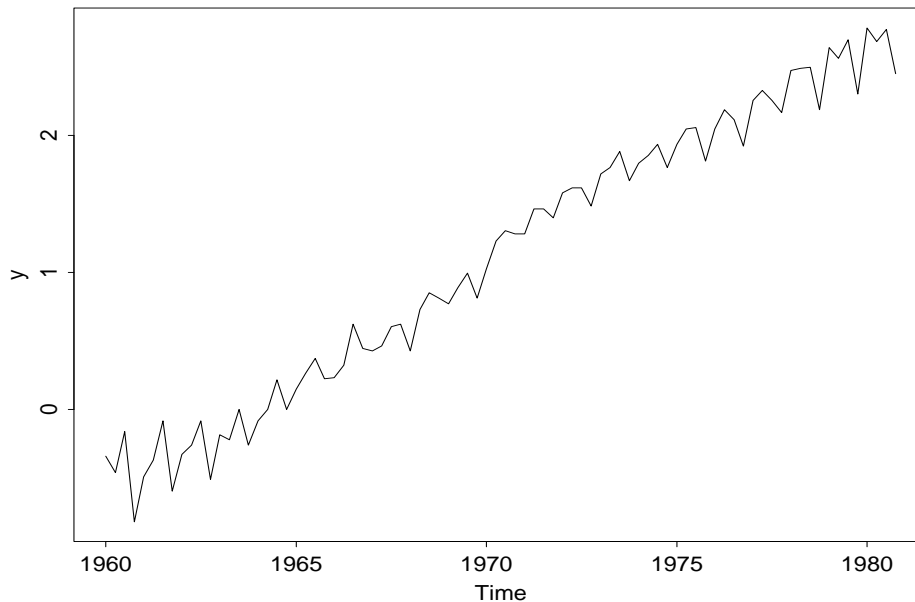


Figure 3: Time plot of quarterly logged earnings of Johnson and Johnson: 1960-1980

It is called *regular* and *seasonal* differenced series.

- ACF of w_t has a nice symmetric structure (see the text), i.e. $\rho_{s-1} = \rho_{s+1} = \rho_1 \rho_s$. Also, $\rho_\ell = 0$ for $\ell > s + 1$.
- This model is widely applicable to many many seasonal time series.
- Multiplicative model means that the regular and seasonal dependences are roughly orthogonal to each other.
- Forecasts: exhibit the same pattern as the observed series. See Figure 4.

Example Detailed analysis of J&J earnings.

R Demonstration: output edited.

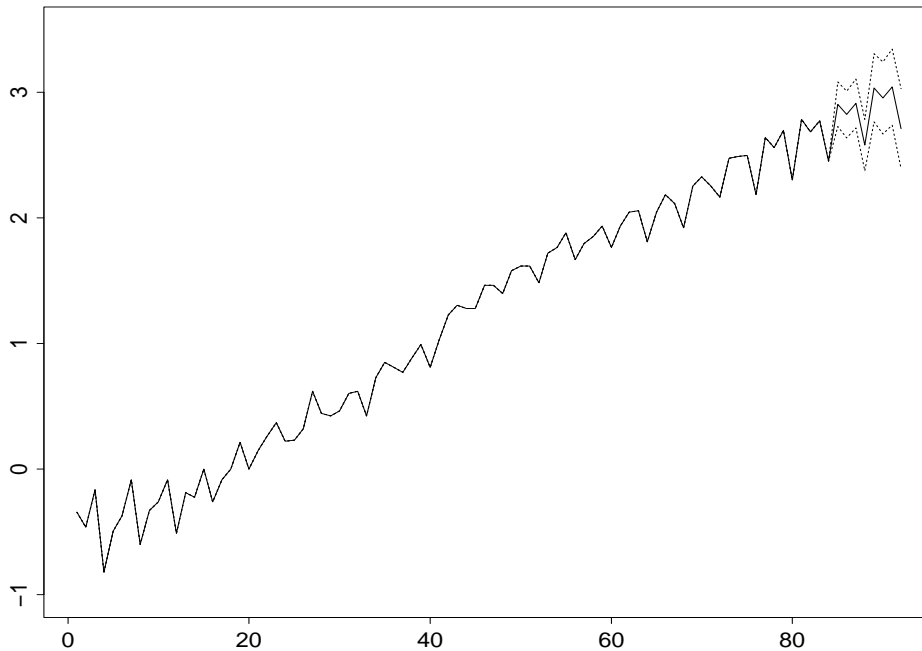


Figure 4: Forecast plot for the quarterly earnings of Johnson and Johnson. Data: 1960-1980, Forecasts: 1981-82.

```

> library(fSeries)
> setwd("C:/teaching/bs41202")
> x=ts(scan("jnj.dat"),frequency=4,start=c(1960,1)) % Load data into a time series object.
> plot(x,type='l') % Plot data with calendar time
> y=log(x) % Natural log transformation
> plot(y,type='l') % plot data
> points(y) % put circles on data points.
> par(mfcol=c(2,1)) % two plots per page
> acf(y,lag.max=16)
> y1=as.vector(y) % Creates a sequence of data in R
> acf(y1,lag.max=16)
> dy1=diff(y1) % regular difference
> acf(dy1,lag.max=16)
> sdy1=diff(dy1,4) % seasonal difference
> acf(sdy1,lag.max=12)
> m1=arima(y1,order=c(0,1,1),seasonal=list(order=c(0,1,1),period=4)) % Airline
      % model in R.

> m1

```

Call:

```
arima(x = y1, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 4))
```

```

Coefficients:
      ma1      sma1
      -0.6809 -0.3146 % The fitted model is (1-B^4)(1-B)R(t) =
s.e.    0.0982  0.1070 % (1-0.68B)(1-0.31B^4)a(t), var[a(t)] = 0.00793.

sigma^2 estimated as 0.00793: log likelihood = 78.38, aic = -150.75
> par(mfcol=c(1,1)) % One plot per page
> tsdiag(m1) % Model checking
> f1=predict(m1,8) % prediction
> names(f1)
[1] "pred" "se"
> f1
$pred % Point forecasts
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 % standard errors of point forecasts
Time Series:
Start = 85
End = 92
Frequency = 1
[1] 0.08905414 0.09347895 0.09770358 0.10175295 0.13548765 0.14370550
[7] 0.15147817 0.15887102

> s1=c(y1,f1$pred) % Join data with forecasts
> lcl=c(y1,f1$pred-2*f1$se) % Lower limit for 95% interval
> ucl=c(y1,f1$pred+2*f1$se) % Upper limit for 95% interval
> max(ucl)
[1] 3.346211
> min(y1)
[1] -0.8209806
> plot(s1,type='l',ylim=c(-1,3.5)) % Forecast plot
> lines(1:92,ucl,lty=2)
> lines(1:92,lcl,lty=2)

```

S-Plus Demonstration: output edited.

```

> x=ts(scan('jnj.dat'),frequency=4,start=c(1960,1)) % Load data into Splus
> plot(x,type='l') % Plot the data
> title(main='Quarterly earnings of JNJ: 1960-1980') % title of the plot
> y=log(x) % natural log transformation
> plot(y,type='l')
> par(mfcol=c(2,1)) % put two plots on a page
> acf(y,lag.max=16) % 16 lags of ACF

```

```

> acf(diff(y),lag.max=16)

> y1=as.vector(y)      % creates a sequence in Splus, not a time-series object.
> acf(y1,lag.max=16)

Autocorrelation matrix:
  lag    y1
1   0 1.0000
2   1 0.9566
3   2 0.9260    % Indicates 1st difference is needed.
4   3 0.8978
5   4 0.8723
6   5 0.8285
    ....
17  16 0.4578
> acf(diff(y1),lag.max=16)

> dy1=diff(y1)
> sdy1=diff(dy1,4)
> acf(sdy1,lag.max=12)
> tra=mean(sdy1)/sqrt(var(sdy1)/length(sdy1)) % Compute t-ratio of the mean.
> tra
[1] 0.3101582

> air=list(list(order=c(0,1,1)),list(order=c(0,1,1),period=4)) % Define the
                                     % airline model.
> m1=arima.mle(y1,model=air) % estimation

> summary(m1)
Call: arima.mle(x = y1, model = air)
Method: Maximum Likelihood with likelihood conditional on 5 observations

Multiplicative ARIMA model --
Model component 1
ARIMA order: 0 1 1

Model component 2
ARIMA order: 0 1 1
Period: 4

      Value Std. Error t-value    % Fitted model (1-B^4)(1-B)R(t) =
ma(1) 0.6809    0.08582   7.934    % (1-0.68B)(1-0.31B^4)a(t),
ma(4) 0.3146    0.11120   2.828    % with var[a(t)] = 0.0079.

Variance-Covariance Matrix:
      ma(1)      ma(4)

```

```
ma(1) 0.007364341 -0.002665339
ma(4) -0.002665339 0.012370336
```

```
Estimated innovations variance: 0.0079
```

```
Optimizer has converged
Convergence Type: relative function convergence
AIC: -152.7529
```

```
> arima.diag(m1) % Model checking
```

```
> f1=arima.forecast(y1,model=m1$model,8) % Forecasts of the next two years
```

```
> names(f1)
```

```
[1] "mean" "std.err"
```

```
> f1
```

```
$mean:
```

```
[1] 2.905343 2.823891 2.912148 2.581085 3.036450 2.954999 3.043255 2.712193
```

```
$std.err:
```

```
[1] 0.08905406 0.09347891 0.09770357 0.10175299 0.13548748 0.14370536
```

```
[7] 0.15147807 0.15887096
```

```
% The following commands create a forecast plot.
```

```
> s1=c(y1,f1$mean) % Join forecasts with data
```

```
> lcl=c(y1,f1$mean-2*f1$std.err) % Join data with lower limit
```

```
> ucl=c(y1,f1$mean+2*f1$std.err) % Join data with upper limit
```

```
> par(mfcol=c(1,1)) % One plot per page
```

```
> length(s1)
```

```
[1] 92
```

```
> max(ucl) % Maximum and minimum are used to set limits on plot.
```

```
[1] 3.346211
```

```
> min(y1)
```

```
[1] -0.8209806
```

```
> plot(s1,type='l',ylim=c(-1,3.5))
```

```
> lines(1:92,ucl,lty=2)
```

```
> lines(1:92,lcl,lty=2)
```

Consider monthly series with period 12. Airline model becomes

$$(1 - B)(1 - B^{12})r_t = (1 - \theta_1 B)(1 - \theta_{12} B^{12})a_t.$$

What is the pattern of ACF?

Regression Models with Time Series Errors

- Has many applications
- Impact of serial correlations in regression is often overlooked.

It may introduce biases in estimates and in standard errors, resulting in unreliable t-ratios.

- Detecting residual serial correlation: Use Q-stat instead of DW-statistic, which is not sufficient!
- Joint estimation of all parameters is preferred.
- Proper analysis: see the illustration below.

Example. U.S. weekly interest rate data: 1-year and 3-year constant maturity rates. Data are shown in Figure 5.

R Demonstration: output edited.

```
> library(fSeries)
> setwd("C:/teaching/bs41202")
> da=read.table("w-gs1n36299.txt") % load the data
> r1=da[,1] % 1-year rate
> r3=da[,2] % 3-year rate
> plot(r1,type='l') % Plot the data
> lines(1:1967,r3,lty=2)
> plot(r1,r3) % scatter plot of the two series

> m1=lm(r3~r1) % Fit a regression model with likelihood method.
> summary(m1)
Call:
lm(formula = r3 ~ r1)

Residuals:
    Min       1Q   Median       3Q      Max
-1.812147 -0.402280  0.003097  0.402588  1.338746

Coefficients:
```

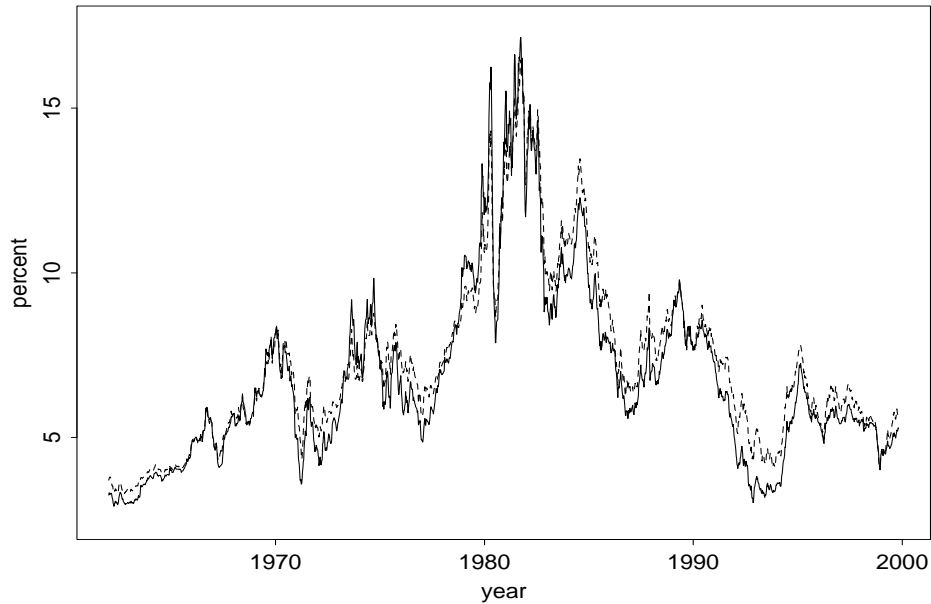


Figure 5: Time plots of U.S. weekly interest rates: 1-year constant maturity rate (solid line) and 3-year rate (dashed line).

```

                Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.910687    0.032250   28.24  <2e-16 ***
r1          0.923854    0.004389  210.51  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

Residual standard error: 0.538 on 1965 degrees of freedom
Multiple R-Squared: 0.9575,    Adjusted R-squared: 0.9575
F-statistic: 4.431e+04 on 1 and 1965 DF,  p-value: < 2.2e-16

```

```

> acf(m1$residuals)
> c3=diff(r3)
> c1=diff(r1)
> plot(c1,c3)

> m2=lm(c3~c1)    % Fit a regression with likelihood method.
> summary(m2)
Call:
lm(formula = c3 ~ c1)

```

```

Residuals:
      Min       1Q   Median       3Q      Max

```

```
-0.3806040 -0.0333840 -0.0005428 0.0343681 0.4741822
```

Coefficients:

```
          Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.0002475 0.0015380 0.161 0.872
c1           0.7810590 0.0074651 104.628 <2e-16 ***
```

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

Residual standard error: 0.06819 on 1964 degrees of freedom

Multiple R-Squared: 0.8479, Adjusted R-squared: 0.8478

F-statistic: 1.095e+04 on 1 and 1964 DF, p-value: < 2.2e-16

```
> acf(m2$residuals)
```

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

```
> m3=arima(c3,xreg=c1,order=c(0,0,1)) % Residuals follow an MA(1) model
```

```
> m3
```

Call:

```
arima(x = c3, order = c(0, 0, 1), xreg = c1)
```

Coefficients:

```
          ma1 intercept          c1 % Fitted model is
          0.2115      0.0002 0.7824 % c3 = 0.0002+0.782c1 + a(t)+0.212a(t-1)
s.e.      0.0224      0.0018 0.0077 % with var[a(t)] = 0.00446.
```

sigma² estimated as 0.004456: log likelihood = 2531.84, aic = -5055.69

```
> acf(m3$residuals)
```

```
> tsdiag(m3)
```

```
> m4=arima(c3,xreg=c1,order=c(1,0,0)) % Residuals follow an AR(1) model.
```

```
> m4
```

Call:

```
arima(x = c3, order = c(1, 0, 0), xreg = c1)
```

Coefficients:

```
          ar1 intercept          c1 % Fitted model is
          0.1922      0.0003 0.7829 % c3 = 0.0003 + 0.783c1 + a(t),
s.e.      0.0221      0.0019 0.0077 % a(t) = 0.192a(t-1)+e(t).
```

sigma² estimated as 0.004474: log likelihood = 2527.86, aic = -5047.72

S-Plus Demonstration

```
> module(finmetrics)
```

```

> da=read.table("w-gs1n3.dat")
> dim(da)
[1] 1967    3
> r3=da[,2]
> r1=da[,1]
> plot(r1,type='l')    % plot the data
> lines(1:1967,r3,lty=2)
> plot(r1,r3)
>
> m1=OLS(r3~r1)    % Least-square regression
> summary(m1)

Call:
OLS(formula = r3 ~ r1)

Residuals:
    Min       1Q   Median       3Q      Max
-1.8121 -0.4023  0.0031  0.4026  1.3387

Coefficients:
                Value Std. Error  t value Pr(>|t|)
(Intercept)   0.9107   0.0323   28.2380  0.0000 % Fitted model is
              r1   0.9239   0.0044  210.5084  0.0000 % r3=0.911+0.924r1 + e

Regression Diagnostics:

                R-Squared 0.9575 % R-square is 96%!!! Any good?
Adjusted R-Squared 0.9575
Durbin-Watson Stat 0.0190 % What is the ‘‘ideal’’ value of DW?

Residual Diagnostics:
                Stat      P-Value
Jarque-Bera     9.0032     0.0111
Ljung-Box    42303.0824     0.0000

Residual standard error: 0.538 on 1965 degrees of freedom
F-statistic: 44310 on 1 and 1965 degrees of freedom, the p-value is 0
> names(m1)
[1] "R"          "coef"       "df.resid"   "fitted"     "residuals" "assign"
[7] "contrasts" "ar.order"   "terms"      "call"

> acf(m1$residuals)    % ACF of residuals

> c3=diff(r3)    % Take the first difference
> c1=diff(r1)
> m2=OLS(c3~c1)    % LS regression of the differenced series

```

```

> summary(m2)

Call:
OLS(formula = c3 ~ c1)

Residuals:
    Min       1Q   Median       3Q      Max
-0.3806 -0.0334 -0.0005  0.0344  0.4742

Coefficients:
                Value Std. Error  t value Pr(>|t|)
(Intercept)  0.0002   0.0015    0.1609  0.8722  % c3 = 0.002+0.781c1+e
             c1  0.7811   0.0075  104.6283  0.0000

Regression Diagnostics:

                R-Squared 0.8479
Adjusted R-Squared 0.8478
Durbin-Watson Stat 1.6158

Residual Diagnostics:
                Stat    P-Value
Jarque-Bera 1508.0683  0.0000
Ljung-Box   230.5767  0.0000

Residual standard error: 0.06819 on 1964 degrees of freedom
F-statistic: 10950 on 1 and 1964 degrees of freedom, the p-value is 0
> acf(m2$residuals) % Plot not shown

> m3=arima.mle(c3,xreg=c1,model=list(order=c(0,0,1))) % Regression model with
                                                    % time-series errors
> summary(m3)
Call: arima.mle(x = c3, model = list(order = c(0, 0, 1)), xreg = c1)
Method: Maximum Likelihood with likelihood conditional on 0 observations

ARIMA order: 0 0 1

                Value Std. Error t-value
ma(1) -0.2115    0.02204  -9.594  % Fitted model
      c1  0.7824         NA      NA  % c3 = 0.782c1 + a(t)+0.212a(t-1).
                                           % Because arima.mle assumes mean of c3
Variance-Covariance Matrix: % is zero, there is no intercept.
      ma(1)
ma(1) 0.0004858961

Estimated innovations variance: 0.0045

```

```
Optimizer has converged
Convergence Type: relative function convergence
AIC: -5059.6702
> arima.diag(m3) & model checking
```

Long-memory models

- Meaning? ACF decays to zero very slowly!
- Example: ACF of squared or absolute log returns
ACFs are small, but decay very slowly.
- How to model long memory? Use “fractional” difference: namely, $(1 - B)^d r_t$, where $-0.5 < d < 0.5$.
- Importance? In theory, Yes. In practice, yet to be determined.

Summary of the chapter

- Sample ACF \Rightarrow MA order
- Sample PACF \Rightarrow AR order
- Some packages have “automatic” procedure to select a simple model for “conditional mean” of a FTS, e.g., SCA uses “iarima” for ARIMA models and R uses “ar” for AR models.
- Check a fitted model before forecasting, e.g. residual ACF and heteroscedasticity (chapter 3)
- Interpretation of a model, e.g. constant term &

For an AR(1) with coefficient ϕ_1 , the speed of mean reverting as measured by half-life is

$$k = \frac{\ln(0.5)}{\ln(\phi_1)}.$$

For an MA(q) model, forecasts revert to the mean in q steps.

- Make proper use of regression models with time series errors, e.g. regression with AR(1) residuals

Perform a joint estimation instead of using any two-step procedure, e.g. Cochrane-Orcutt (1949).

Example: Is there a Friday effect on asset returns?

If a daily market index is used, serial correlation may exist.

See Exercise 8 of Chapter 2.

- Basic properties of a random-walk model
- Multiplicative seasonal models, especially the so-called airline model.