Seasonal Models

Pure Seasonal Models

- simple pure seasonal AR model

$$y_t = \phi_s y_{t-s} + a_t$$

ACF: spike at each multiple of s
PACF: single spike at lag s

- simple pure seasonal MA model

$$y_t = a_t + \theta_s a_{t-s}$$

PACF: spike at each multiple of s
ACF: single spike at lag s

- in practice most time series contain a seasonal AR or MA component at the same time as regular AR or MA component

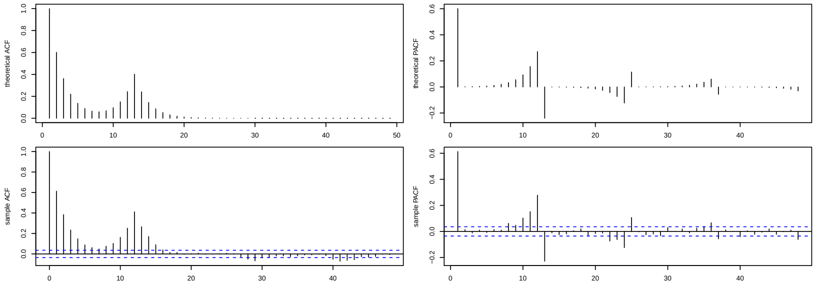
Additive Seasonal AR model

- AR model with an additive seasonal MA component

$$(1-\phi)x_t = (1+\Theta B^s)a_t$$

so that $x_t = \phi x_{t-1} + a_t + \Theta a_{t-s}$

- example: $\phi = 0.6$, $\Theta = 0.5$

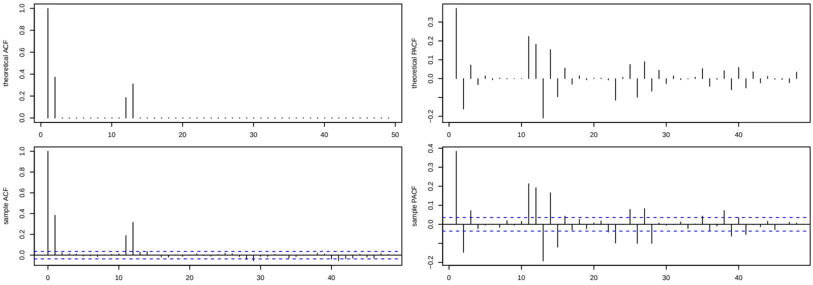


Additive Seasonal MA model

- MA model with an additive seasonal MA component

$$x_t = (1 + \theta B + \Theta B^s) a_t$$

  so that $x_t = a_t + \theta a_{t-1} + \Theta a_{t-s}$
- ACF: $\rho_1 \neq 0$, $\rho_{s-1} \neq 0$, $\rho_s \neq 0$
- example: $\theta = 0.6$, $\Theta = 0.5$

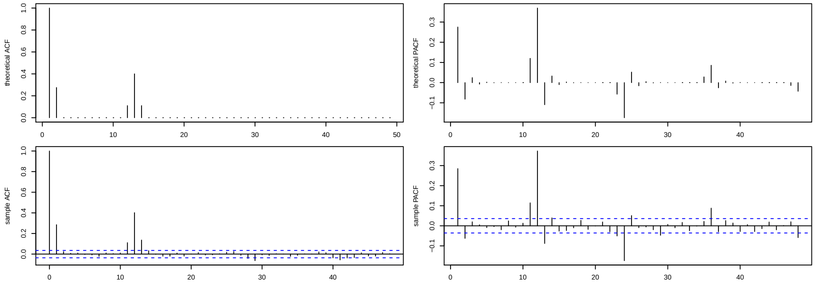


Multiplicative Seasonal MA model

- Multiplicative seasonal MA model

$$x_t = (1 + \theta B)(1 + \Theta B^s)a_t$$

  so that $x_t = a_t + \theta a_{t-1} + \Theta a_{t-s} + \theta\Theta a_{t-s-1}$
- ACF: $\rho_1 \neq 0$, $\rho_{s-1} \neq 0$, $\rho_s \neq 0$, $\rho_{s+1} \neq 0$
- compared to the additive model, multiplicative model allows for *interaction of regular and seasonal components*
- example: $\theta = 0.3$, $\Theta = 0.5$

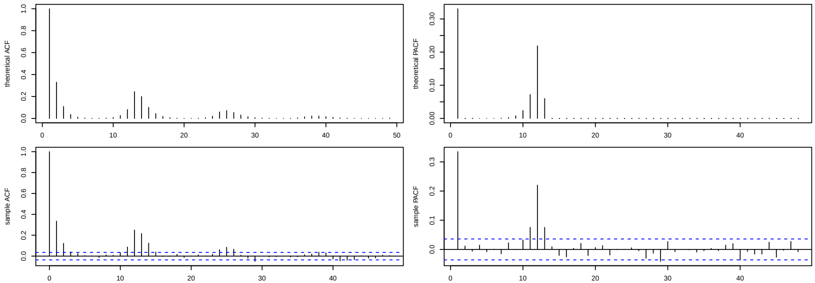


Multiplicative Seasonal AR model

- multiplicative AR model with seasonal component

$$(1 - \phi B)(1 - \Phi B^s)x_t = a_t$$

  so that $x_t = \phi x_{t-1} + \Phi x_{t-s} + \phi\Phi x_{t-s-1} + a_t$
- ACF: if $\phi > 0$, $\Phi > 0$ exponential decay interrupted by increasing autocorrelations around *each* multiple of s
  PACF: large spikes at lag 1 and lag s and multiple smaller spikes between lag 2 and lag $s + 1$
- example: $\phi_1 = 0.3$, $\Phi_1 = 0.5$.



Seasonal Differencing

- for economic data that is nonstationarity due to economic growth a common approach is to transform data using a logartihm and apply regular differencing
$$w_t = \Delta \log y_t$$
where $\Delta = 1 - B$, so that $w_t = (1 - B) \log y_t$
- for economic data that is both nonstationarity due to economic growth and shows seasonal pattern the approach is to transform data using a logartihm and apply both regular and seasonal differencing
$$w_t = \Delta_s \Delta \log y_t$$
where $\Delta = 1 - B$ and $\Delta_s = 1 - B^s$, so we have $w_t = (1 - B^s)(1 - B) \log y_t$
- ocassionally data has to be diffferenced more than once by applying $\Delta^d = (1 - B)^d$, or $\Delta_s^D = (1 - B^s)^D$
- multiplicative models are written in the form ARIMA$(p, d, q)(P, D, Q)_s$
- in practice ARIMA$(1, 1, 0)(0, 1, 1)_s$ and ARIMA$(0, 1, 1)(0, 1, 1)_s$ occur routinely

Example: Johnson & Johnson quarterly earnings per share

```r
str(y)
```

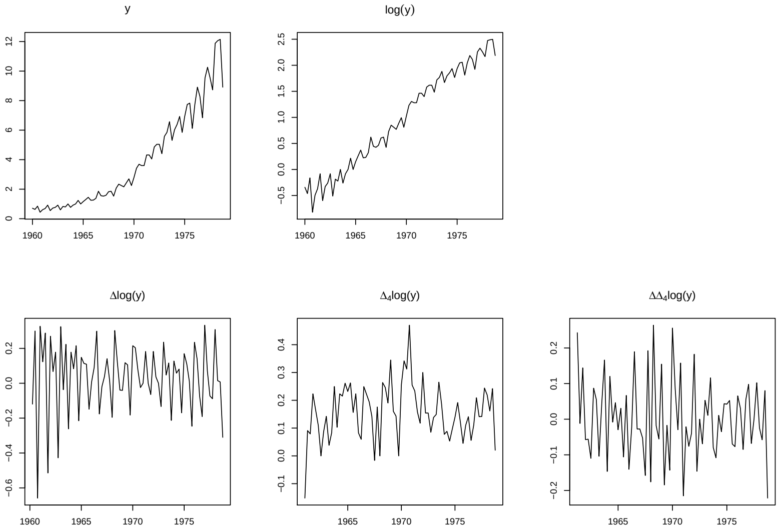
```
##  Time-Series [1:84] from 1960 to 1981: 0.71 0.63 0.85 0.44 0.61 0.69 0.92 0.55 0.72 0.77 ...
```

```r
# split sample into two parts - estimation sample and prediction sample
yall <- y
y1 <- window(yall, end=c(1978,4))
y2 <- window(yall, start=c(1979,1))

# first part used to identify and estimate the model
y <- y1

# log, log-change, seasonal log change
ly <- log(y)
dly1 <- diff(ly)
dly4 <- diff(ly,4)
dly4_1 <- diff(diff(ly),4)
```

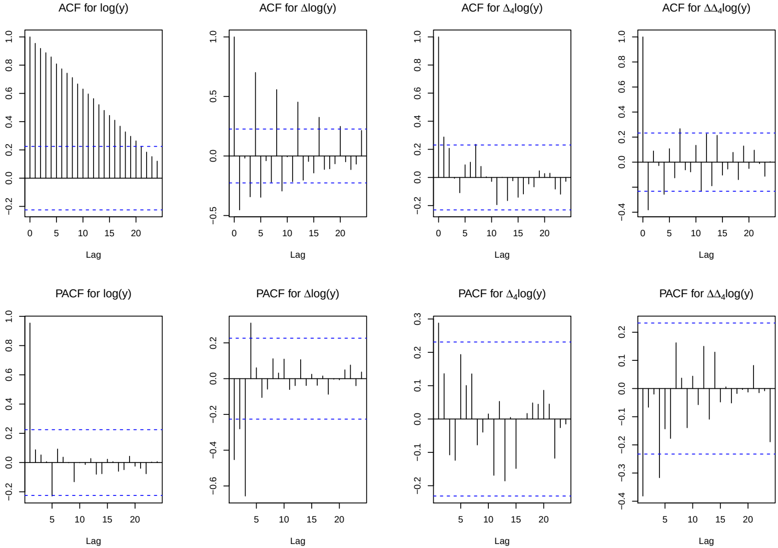
Original and transformed data

```r
par(mfrow=c(2,3))
plot(y, main=expression(y))
plot(ly, main=expression(log(y)))
plot.new()
plot(dly1, main=expression(paste(Delta, "log(y)")))
plot(dly4, main=expression(paste(Delta[4], "log(y)")))
plot(dly4_1, main=expression(paste(Delta, Delta[4], "log(y)")))
```

Original and transformed data



ACF and PACF


ACF for log(y)
ACF for Δlog(y)
ACF for $\Delta_4$log(y)
ACF for $\Delta\Delta_4$log(y)
PACF for log(y)
PACF for Δlog(y)
PACF for $\Delta_4$log(y)
PACF for $\Delta\Delta_4$log(y)
Lag

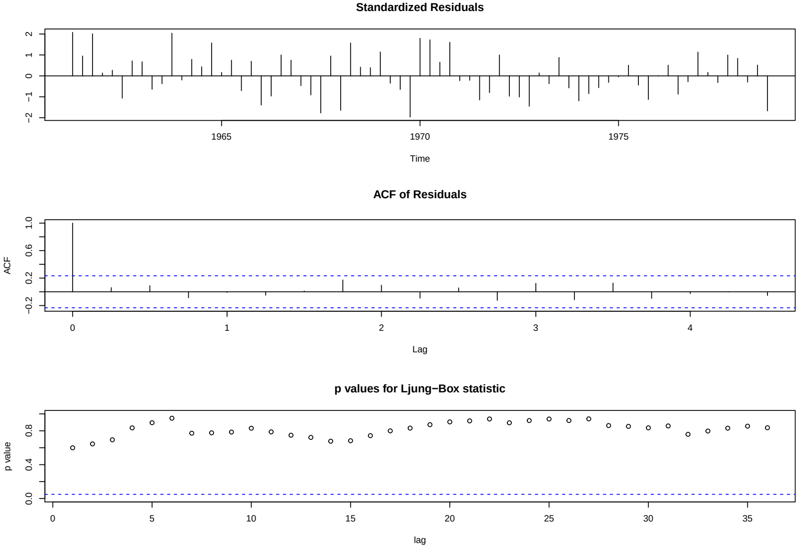
Estimate model m1 for $\Delta_s \Delta \log y_t$

```r
# estimate model - twice differenced data
m1 <- arima(dly4_1,order=c(0,0,1),seasonal=list(order=c(0,0,1),period=4))
m1
```

```
##
## Call:
## arima(x = dly4_1, order = c(0, 0, 1), seasonal = list(order = c(0, 0, 1), period = 4))
##
## Coefficients:
##           ma1     sma1  intercept
##       -0.6604  -0.3492     0.0013
## s.e.   0.1084   0.1101     0.0026
##
## sigma^2 estimated as 0.008374:  log likelihood = 68.42,  aic = -128.84
```

Check model m1 for adequacy

```
tsdiag(m1,gof.lag=36)
```



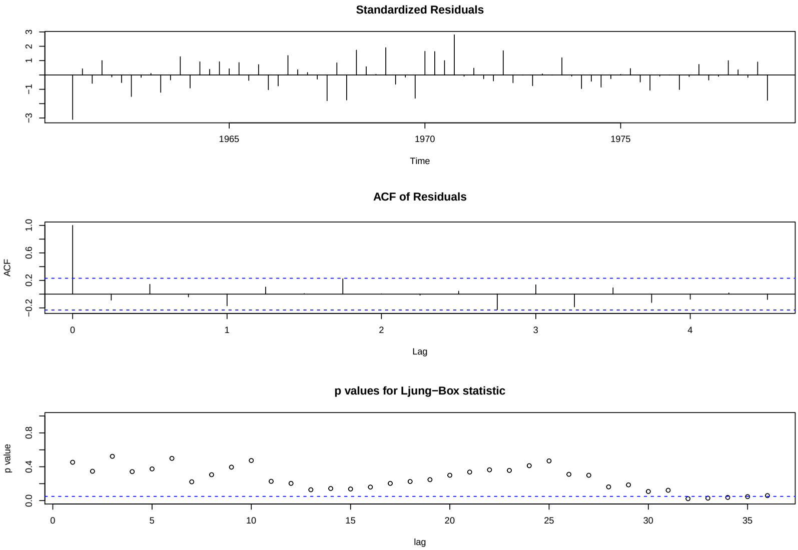
Estimate model `m2` for $\Delta_s \log y_t$

```r
# estimate model - seasonally differenced data
m2 <- arima(dly4,order=c(1,0,0))
m2
```

```
##
## Call:
## arima(x = dly4, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.3412     0.1557
## s.e.  0.1214     0.0166
##
## sigma^2 estimated as 0.008689:  log likelihood = 68.62,  aic = -131.24
```

Check model m2 for adequacy

```
tsdiag(m2,gof.lag=36)
```



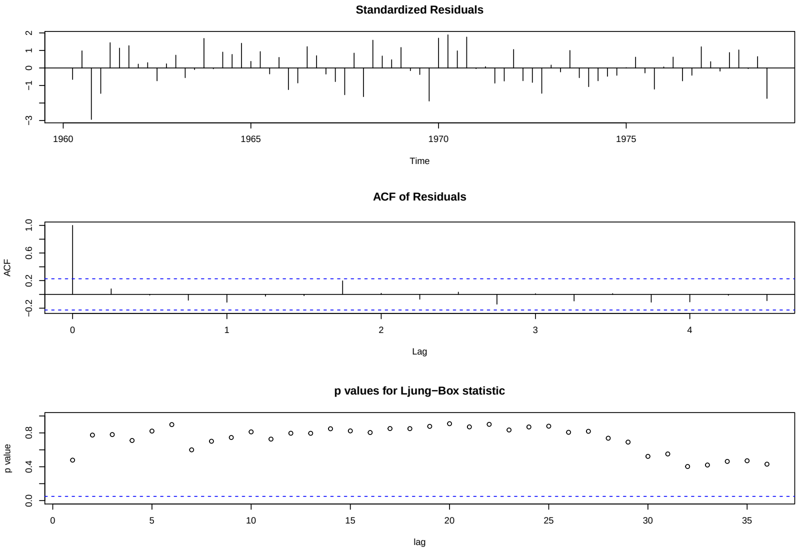
Estimate model m3 for $\Delta \log y_t$

```r
# estimate model - regularly differenced data
m3 <- arima(dly1,order=c(0,0,1),seasonal=list(order=c(1,0,1),period=4))
m3
```

```
##
## Call:
## arima(x = dly1, order = c(0, 0, 1), seasonal = list(order = c(1, 0, 1), period = 4))
##
## Coefficients:
##           ma1    sar1     sma1  intercept
##       -0.7047  0.9386  -0.3166     0.0309
## s.e.   0.1047  0.0443   0.1285     0.0220
##
## sigma^2 estimated as 0.008275:  log likelihood = 70.16,  aic = -130.32
```

Check model m3 for adequacy

```
tsdiag(m3,gof.lag=36)
```



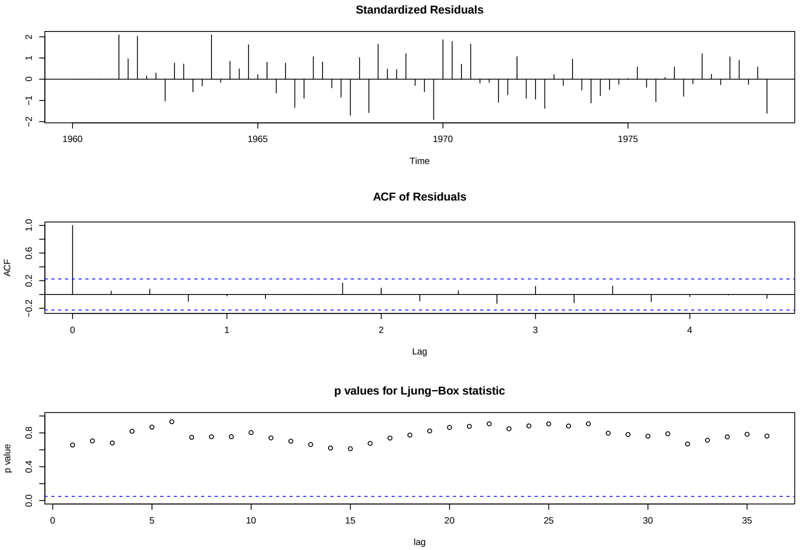
Estimate model m4 for log y_t

```r
# estimate model - data not differenced
m4 <- arima(ly,order=c(0,1,1),seasonal=list(order=c(0,1,1),period=4))
m4
```

```
##
## Call:
## arima(x = ly, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 4))
##
## Coefficients:
##           ma1     sma1
##       -0.6559  -0.3492
## s.e.   0.1094   0.1104
##
## sigma^2 estimated as 0.008409:  log likelihood = 68.28,  aic = -130.57
```

Check model m4 for adequacy

```
tsdiag(m4,gof.lag=36)
```



Forecasts

```
library("forecast")

# construct eight quarters ahead forecasts
m1.fcast <- forecast(m1, h=8)
m2.fcast <- forecast(m2, h=8)
m3.fcast <- forecast(m3, h=8)
m4.fcast <- forecast(m4, h=8)
```

Forecasts

```
par(mfrow=c(2,2), cex=0.7, mar=c(2,4,3,1))
plot(m1.fcast, xlim=c(1970,1981))
lines(diff(diff(log(yall),4)))
plot(m2.fcast, xlim=c(1970,1981))
lines(diff(log(yall),4))
plot(m3.fcast, xlim=c(1970,1981), ylim=c(-0.4,0.6))
lines(diff(log(yall)))
plot(m4.fcast, xlim=c(1970,1981), ylim=c(0.5,3.5))
lines(log(yall))
```