Chpater 3 Selected Computer Exercises

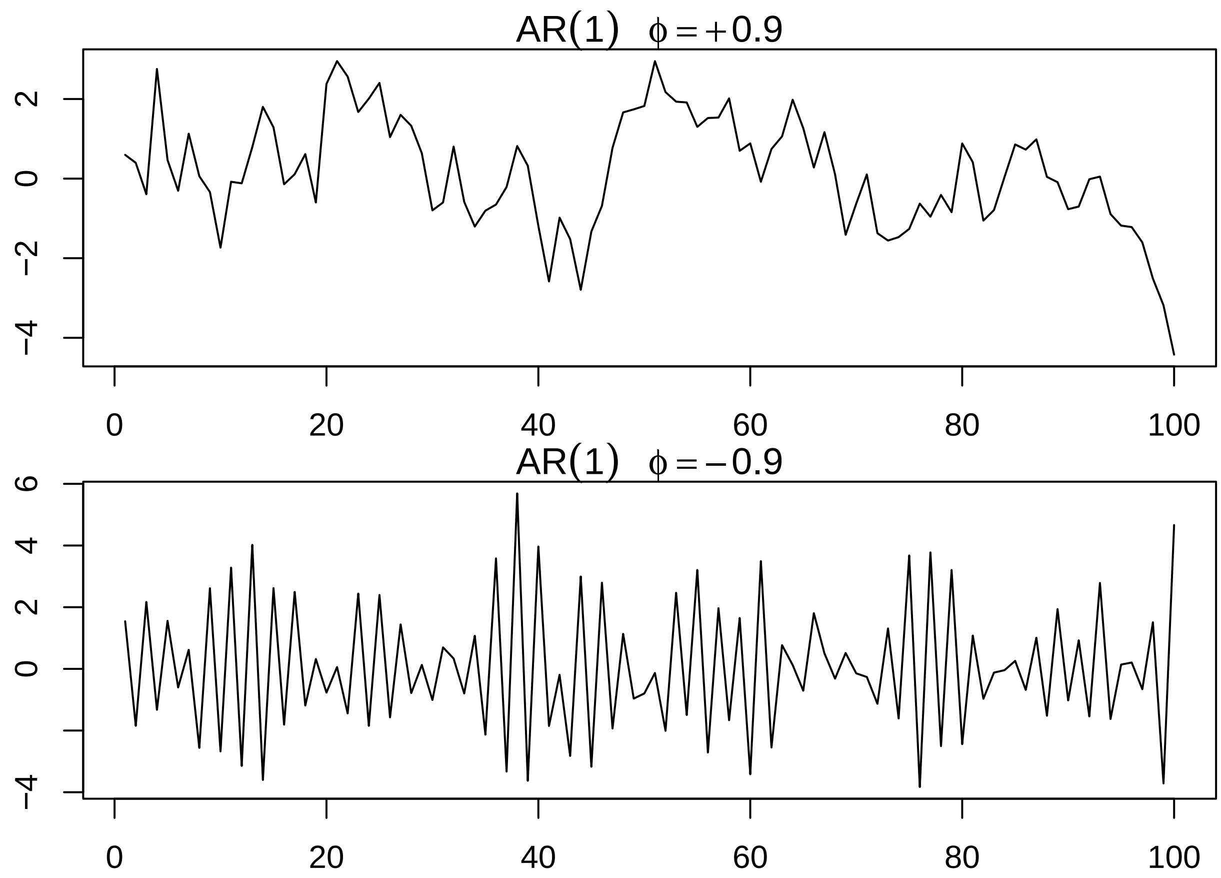
魏上傑

2023-05-02

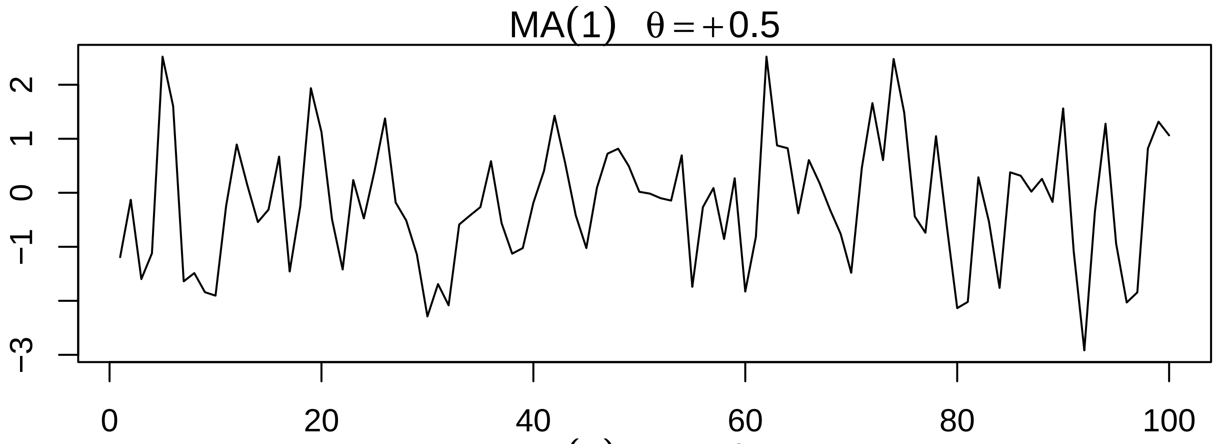
目錄

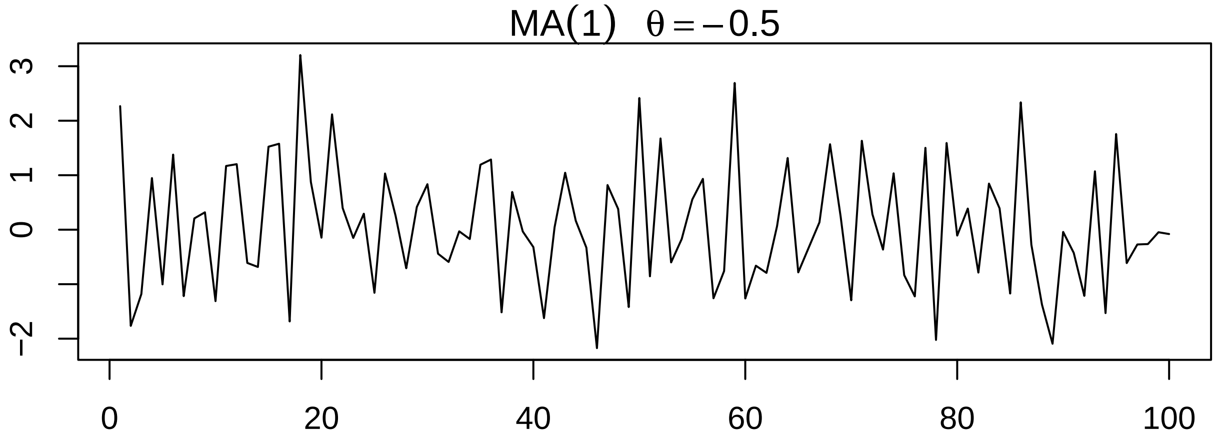
1 Section 3.2

```r
# Example 3.1
# simulated AR(1)
par(mfrow=c(2,1))
par(mar=c(2,2,1,1))
plot(arima.sim(list(order=c(1,0,0), ar=.9), n=100), ylab="x",
     main=expression(AR(1)~~~phi==+.9))
plot(arima.sim(list(order=c(1,0,0), ar=-.9), n=100), ylab="x",
     main=expression(AR(1)~~~phi==-.9))
```




```r
# example 3.4
# simulated MA(1)
par(mfrow = c(2,1))
par(mar=c(2,2,1,1))
plot(arima.sim(list(order=c(0,0,1), ma=.5), n=100), ylab="x",
      main=(expression(MA(1)~~~theta==+.5)))
plot(arima.sim(list(order=c(0,0,1), ma=-.5), n=100), ylab="x",
      main=(expression(MA(1)~~~theta==-.5)))
```





2 Section 3.3

2.1 Problem 3.6

For the AR(2) model given by $x_t = -.9x_{t-2} + w_t$, find the roots of the autoregressive polynomial, and then sketch the ACF, ρ(h).

The polynomial is $x_t + 0.9x_{t-2} = w_t$

```
z <- c(1, 0, .9) # coefficients of the polynomial
polyroot(z) # print out the roots
```

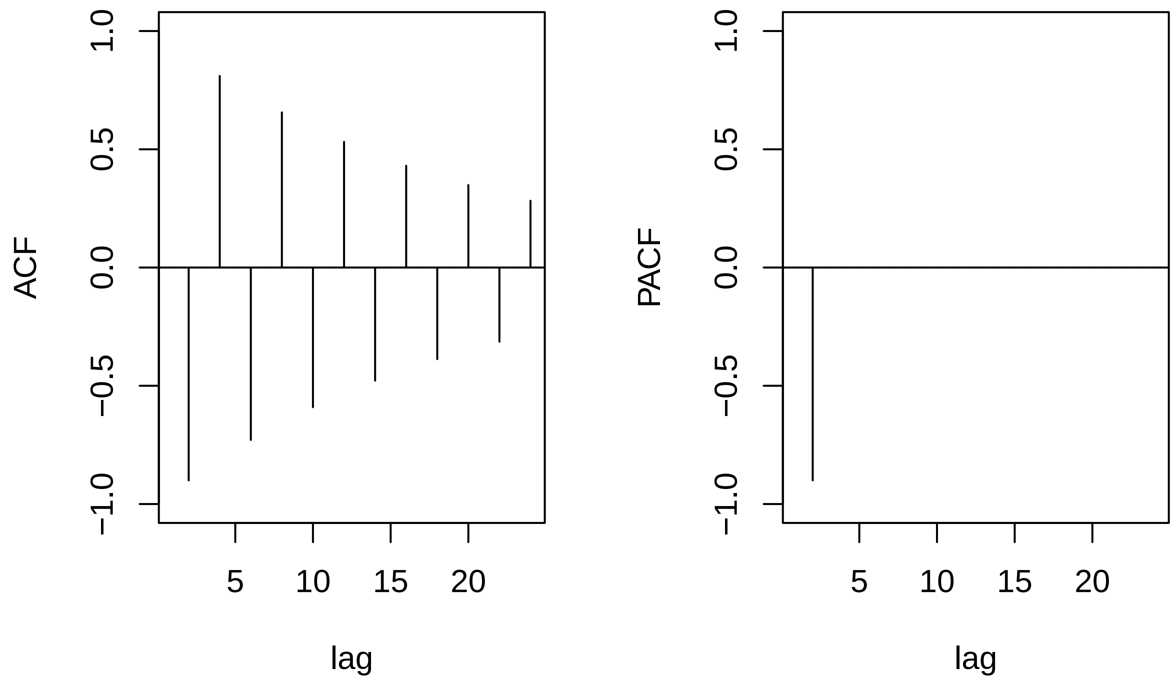
```
## [1] 0+1.054093i 0-1.054093i
```

```
abs(polyroot(z))
```

```
## [1] 1.054093 1.054093
```

All the roots are larger than 1 meaning that the model is causal.

```
ACF <- ARMAacf(ar=c(0, -.9), ma=0, 24)[-1] # we exclude the lag 0
PACF <- ARMAacf(ar=c(0, -.9), ma=0, 24, pacf=TRUE)
par(mfrow=c(1,2))
plot(ACF, type="h", xlab="lag", ylim=c(-1,1)); abline(h=0)
plot(PACF, type="h", xlab="lag", ylim=c(-1,1)); abline(h=0)
```



2.2 Example 3.11

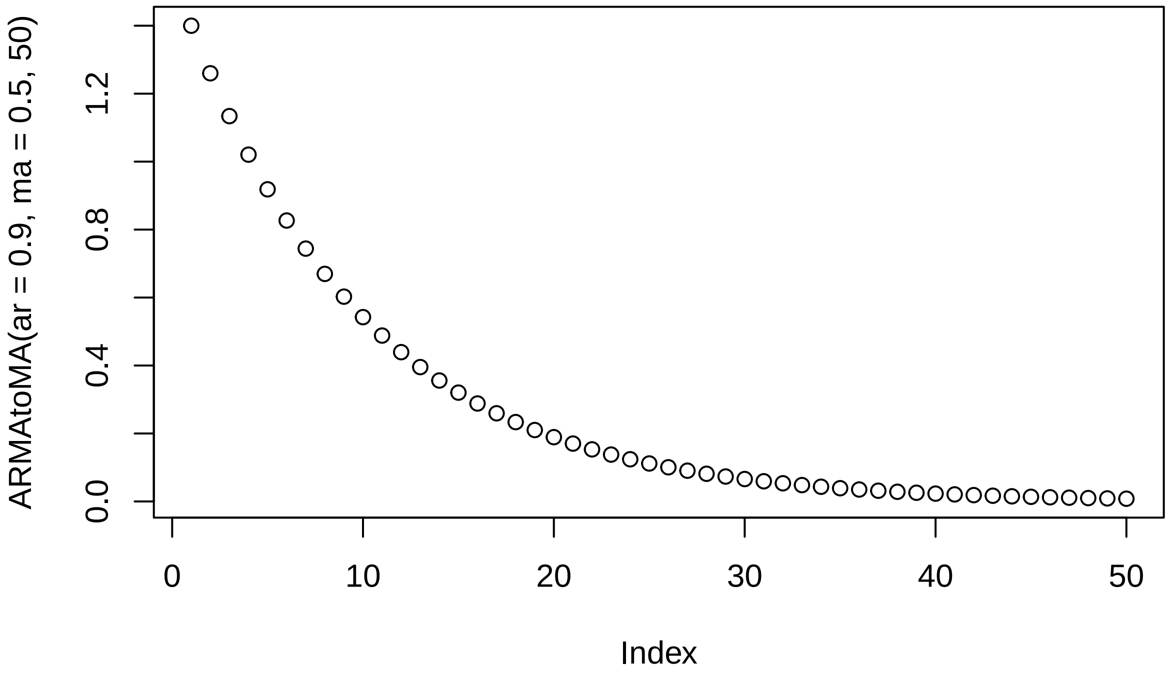
The ψ-weights for an ARMA Model.

Consider $x_t = .9x_{t-1} + .5w_{t-1} + w_t$

```
ARMAtoMA(ar=.9, ma=.5, 50) # for a list
```

```
##  [1] 1.400000000 1.260000000 1.134000000 1.020600000 0.918540000 0.826686000
##  [7] 0.744017400 0.669615660 0.602654094 0.542388685 0.488149816 0.439334835
## [13] 0.395401351 0.355861216 0.320275094 0.288247585 0.259422826 0.233480544
## [19] 0.210132489 0.189119240 0.170207316 0.153186585 0.137867926 0.124081134
## [25] 0.111673020 0.100505718 0.090455146 0.081409632 0.073268669 0.065941802
## [31] 0.059347622 0.053412859 0.048071573 0.043264416 0.038937975 0.035044177
## [37] 0.031539759 0.028385783 0.025547205 0.022992485 0.020693236 0.018623913
## [43] 0.016761521 0.015085369 0.013576832 0.012219149 0.010997234 0.009897511
## [49] 0.008907760 0.008016984
```

```
plot(ARMAtoMA(ar=.9, ma=.5, 50)) # for a graph
```



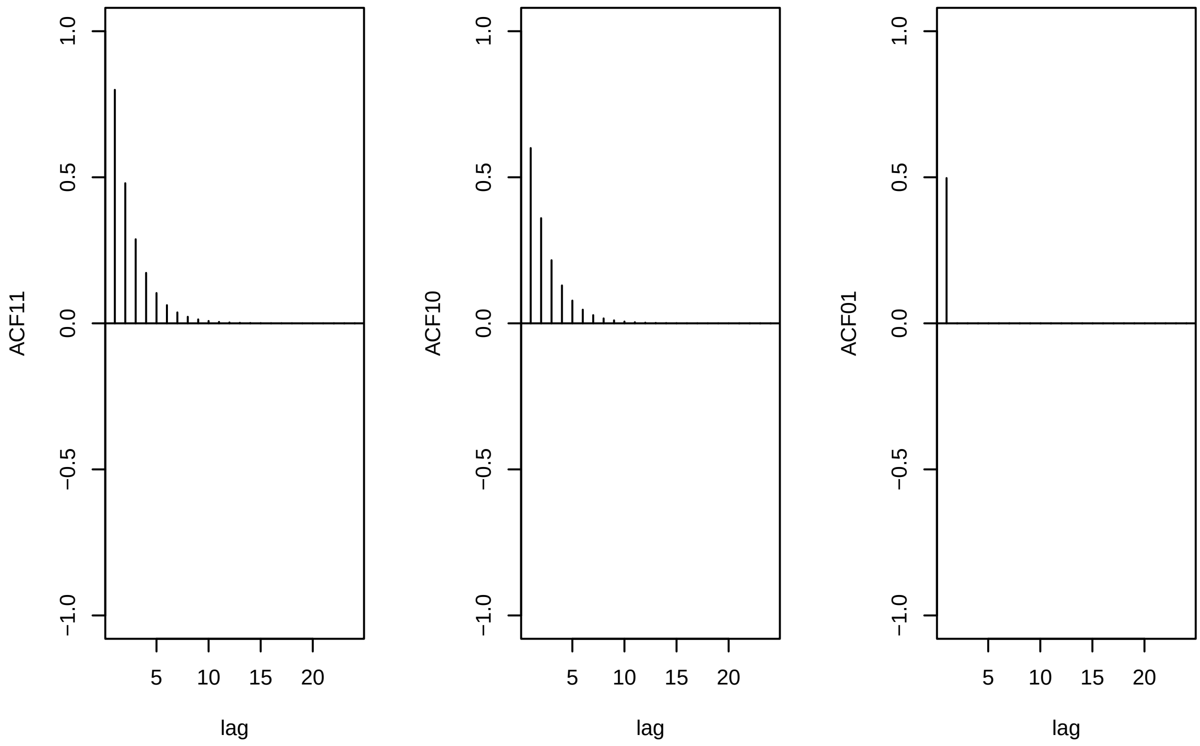
3 Section 3.4

3.1 Problem 3.8

Verify the calculations for the autocorrelation function of an ARMA(1, 1) process given in Example 3.13. Compare the form with that of the ACF for the ARMA(1,0) and the ARMA(0,1) series. Plot (or sketch) the ACFs of the three series on the same graph for $\phi = .6, \theta = .9$, and comment on the diagnostic capabilities of the ACF in this case.

$x_t = \phi x_{t-1} + \theta w_{t-1} + w_t$

```
phi=.6
theta=.9
ACF11 <- ARMAacf(ar=c(phi), ma=c(theta), 24)[-1]
ACF10 <- ARMAacf(ar=c(phi), ma=0, 24)[-1]
ACF01 <- ARMAacf(ar=0, ma=theta,24)[-1]
par(mfrow=c(1,3))
plot(ACF11, type="h", xlab="lag", ylim=c(-1,1)); abline(h=0)
plot(ACF10, type="h", xlab="lag", ylim=c(-1,1)); abline(h=0)
plot(ACF01, type="h", xlab="lag", ylim=c(-1,1)); abline(h=0)
```



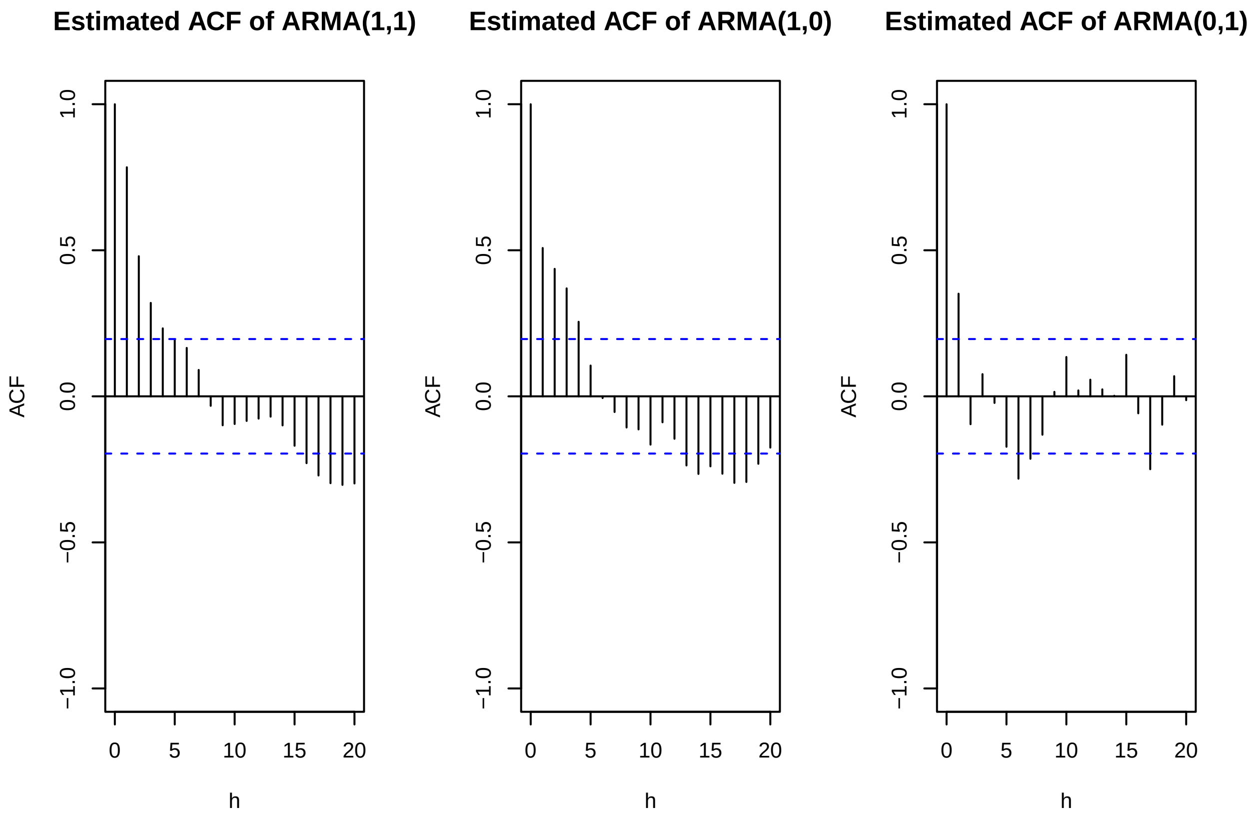
3.2 Problem 3.9

Generate n = 100 observations from each of the three models discussed in Problem 3.8. Compute the sample ACF for each model and compare it to the theoretical values. Compute the sample PACF for each of the generated series and compare the sample ACFs and PACFs with the general results given in Table 3.1.

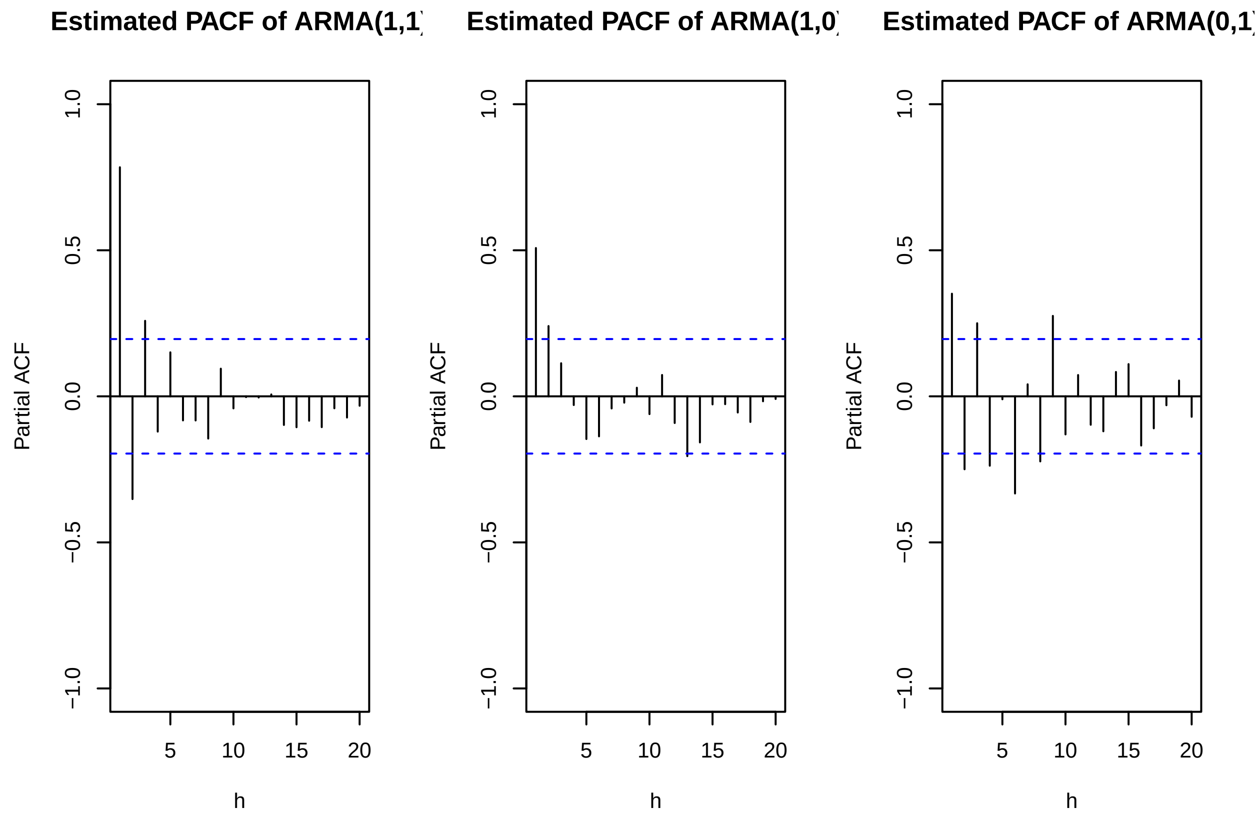
```
ARMA11 <- arima.sim(list(ar=c(phi), ma=c(theta)), n=100, rand.gen = rnorm, sd = 1)
ARMA10 <- arima.sim(list(ar=c(phi), ma=0), n=100, rand.gen = rnorm, sd=1)
ARMA01 <- arima.sim(list(ar=0, ma=c(theta)), n=100, rand.gen = rnorm, sd=1)
```

```
## Warning in min(Mod(polyroot(c(1, -model$ar)))): no non-missing arguments to
## min; returning Inf
```

```
# estimated acf
par(mfrow=c(1,3))
acf(ARMA11, type="correlation", lag.max = 20, xlab="h",
    ylim=c(-1,1), main= "Estimated ACF of ARMA(1,1)")
acf(ARMA10, type="correlation", lag.max = 20, xlab="h",
    ylim=c(-1,1), main= "Estimated ACF of ARMA(1,0)")
acf(ARMA01, type="correlation", lag.max = 20, xlab="h",
    ylim=c(-1,1), main= "Estimated ACF of ARMA(0,1)")
```




```
# estimated pacf
#Note: acf(x = x, type = "partial") also gives the PACF plot
par(mfrow=c(1,3))
pacf(ARMA11, lag.max = 20, xlab="h",
    ylim=c(-1,1), main= "Estimated PACF of ARMA(1,1)")
pacf(ARMA10, lag.max = 20, xlab="h",
    ylim=c(-1,1), main= "Estimated PACF of ARMA(1,0)")
pacf(ARMA01, lag.max = 20, xlab="h",
    ylim=c(-1,1), main= "Estimated PACF of ARMA(0,1)")
```



4 Section 3.5

4.1 Problem 3.10

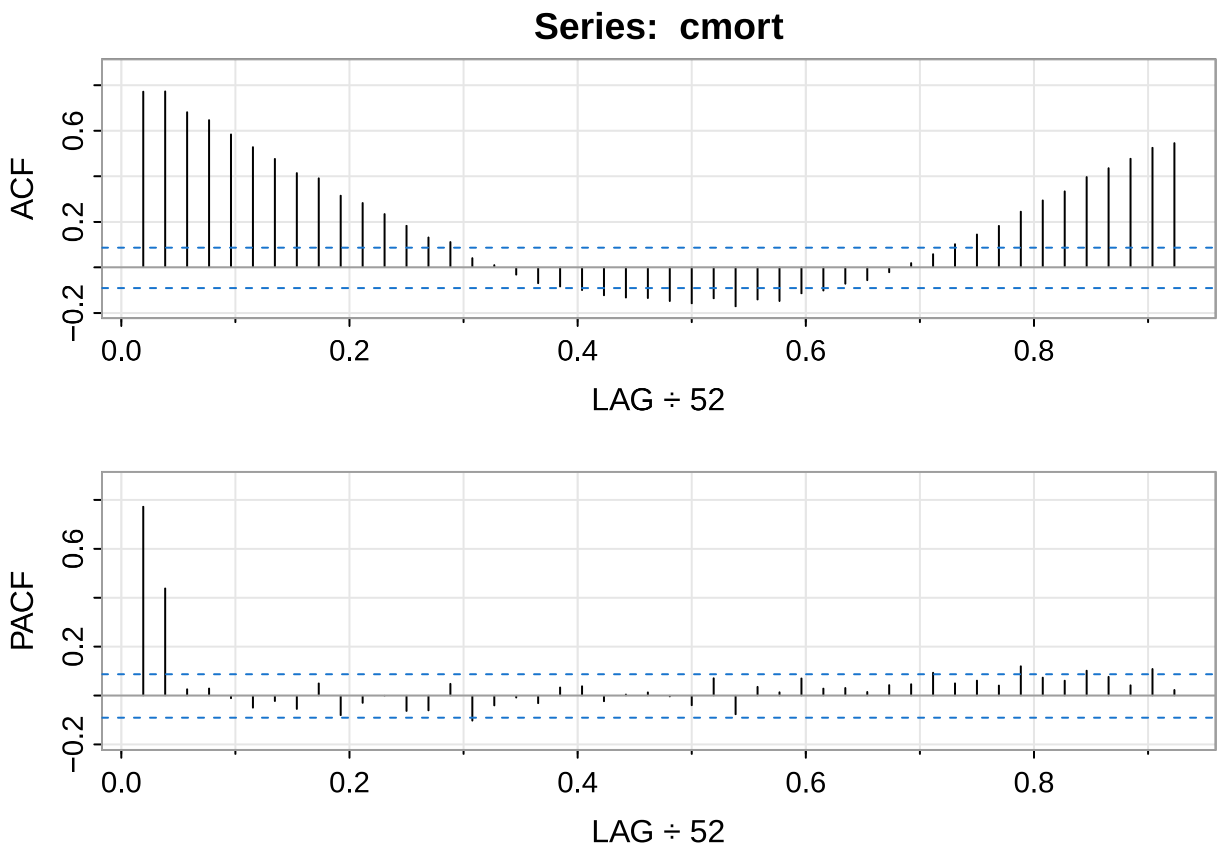
Let x_t represent the cardiovascular mortality series (cmort) discussed in Chapter 2, Example 2.2.

(a) Fit an AR(2) to x_t using linear regression as in Example 3.17.

AR(2): $x_t = \phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + w_t$

We use `acf2` to print and plot the ACF and PACF.

```r
acf2(cmort, 48) # will produce values and a graphic
```




```
##       [,1] [,2] [,3] [,4]  [,5]  [,6]  [,7]  [,8] [,9] [,10] [,11] [,12] [,13]
## ACF  0.77 0.77 0.68 0.65  0.58  0.53  0.48  0.41 0.39  0.32  0.28  0.23  0.18
## PACF 0.77 0.44 0.03 0.03 -0.01 -0.05 -0.02 -0.05 0.05 -0.08 -0.03  0.00 -0.06
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF   0.13  0.11  0.04  0.01 -0.03 -0.07 -0.08 -0.10 -0.12 -0.13 -0.13 -0.15
## PACF -0.06  0.05 -0.10 -0.04 -0.01 -0.03  0.03  0.04 -0.02  0.00  0.01  0.00
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
## ACF  -0.16 -0.14 -0.17 -0.14 -0.15 -0.11 -0.10 -0.07 -0.06 -0.02  0.02  0.06
## PACF -0.04  0.07 -0.08  0.03  0.01  0.07  0.03  0.03  0.01  0.04  0.05  0.09
##      [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF   0.10  0.14  0.18  0.24  0.29  0.33   0.4  0.44  0.48  0.53  0.55
## PACF  0.05  0.06  0.04  0.12  0.07  0.06   0.1  0.08  0.04  0.11  0.02
```

```r
regr <- ar.ols(cmort, order=2, demean = FALSE, intercept = TRUE)
regr
```

```
##
## Call:
## ar.ols(x = cmort, order.max = 2, demean = FALSE, intercept = TRUE)
##
```

```
## Coefficients:
##       1       2
## 0.4286  0.4418
##
## Intercept: 11.45 (2.394)
##
## Order selected 2  sigma^2 estimated as  32.32
```

```
regr$asy.se.coef # standard errors of the estimates
```

```
## $x.mean
## [1] 2.393673
##
## $ar
## [1] 0.03979433 0.03976163
```

The estimated model is

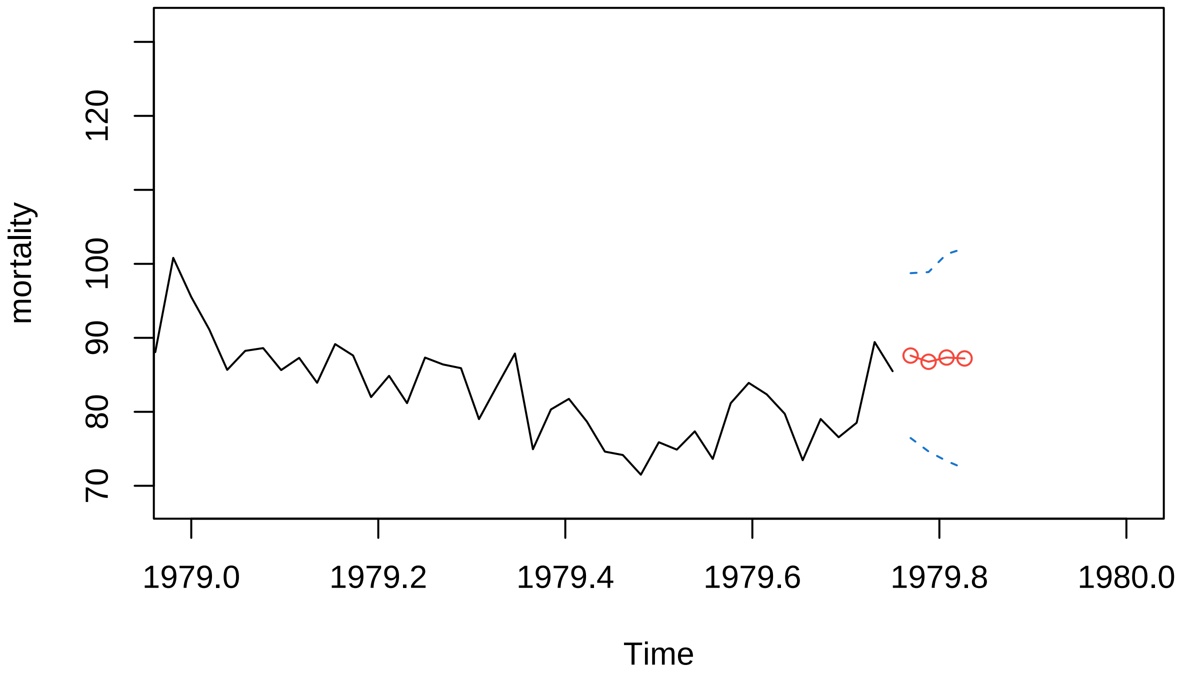
$$\hat{x}_t = \hat{\phi}_0 + \hat{\phi}_1 x_{t-1} + \hat{\phi}_2 x_{t-2} \tag{1}$$

$$\hat{x}_t = 11.45_{(2.394)} + 0.4286_{(0.03979433)} x_{t-1} + 0.4418_{(0.03976163)} x_{t-2} \tag{2}$$

with $\hat{\sigma}_w^2 = 32.32$

(b) Assuming the fitted model in (a) is the true model, find the forecasts over a four-week horizon, x_{n+m}^n, for m = 1, 2, 3, 4, and the corresponding 95% prediction intervals.

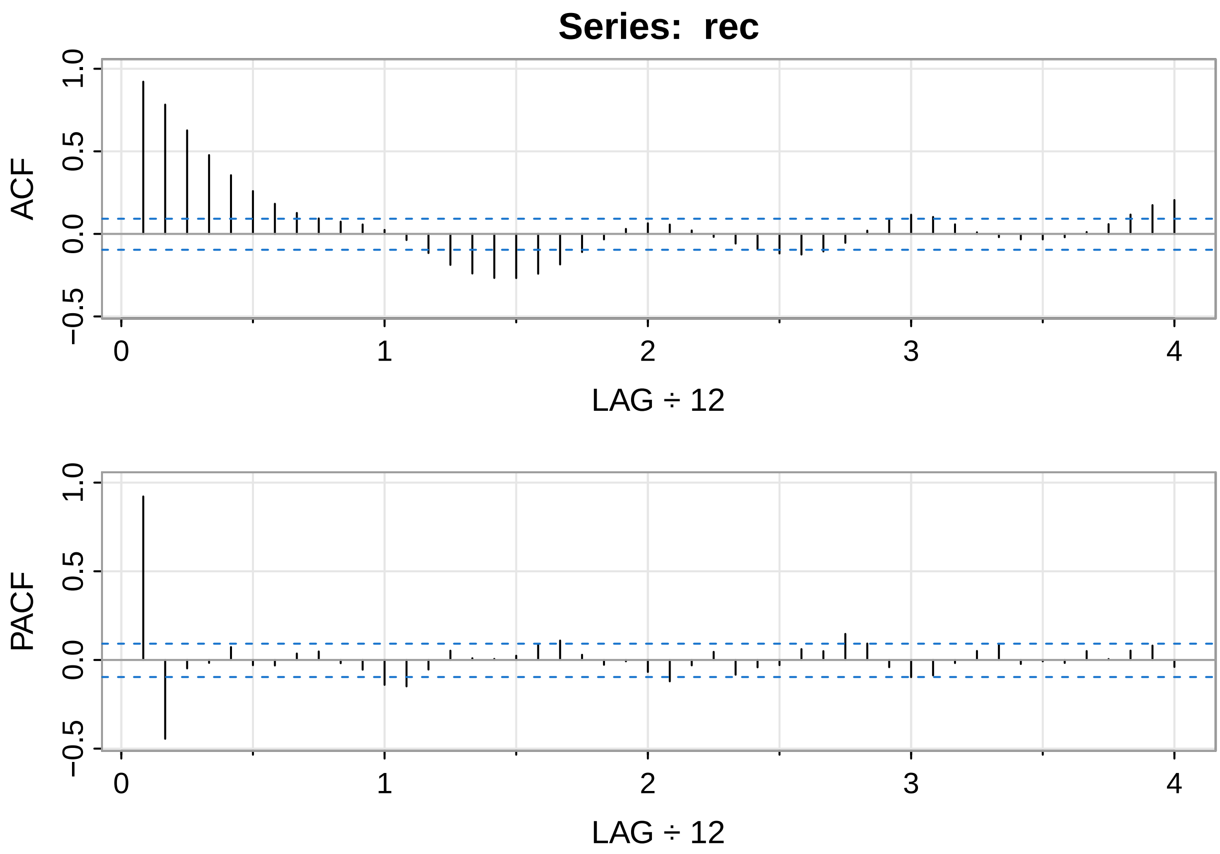
```
fore <- predict(regr, n.ahead = 4)
ts.plot(cmort, fore$pred, col=1:2, ylab="mortality", xlim=c(1979, 1980))
lines(fore$pred, type="p", col=2)
lines(fore$pred+qnorm(p = 0.975, mean = 0, sd = 1)*fore$se, lty="dashed", col=4)
lines(fore$pred-qnorm(p = 0.975, mean = 0, sd = 1)*fore$se, lty="dashed", col=4)
```



5 Section 3.6

5.1 Recruitment example

```
# example 3.17
acf2(rec, 48) # will produce values and a graphic
```




```
##       [,1]  [,2]  [,3]  [,4] [,5]  [,6]  [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## ACF  0.92  0.78  0.63  0.48 0.36  0.26  0.18 0.13 0.09  0.07  0.06  0.02 -0.04
## PACF 0.92 -0.44 -0.05 -0.02 0.07 -0.03 -0.03 0.04 0.05 -0.02 -0.05 -0.14 -0.15
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF  -0.12 -0.19 -0.24 -0.27 -0.27 -0.24 -0.19 -0.11 -0.03  0.03  0.06  0.06
## PACF -0.05  0.05  0.01  0.01  0.02  0.09  0.11  0.03 -0.03 -0.01 -0.07 -0.12
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
## ACF   0.02 -0.02 -0.06 -0.09 -0.12 -0.13 -0.11 -0.05  0.02  0.08  0.12  0.10
## PACF -0.03  0.05 -0.08 -0.04 -0.03  0.06  0.05  0.15  0.09 -0.04 -0.10 -0.09
##      [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF   0.06  0.01 -0.02 -0.03 -0.03 -0.02  0.01  0.06  0.12  0.17  0.20
## PACF -0.02  0.05  0.08 -0.02 -0.01 -0.02  0.05  0.01  0.05  0.08 -0.04
```

```r
regr <- ar.ols(rec, order=2, demean=FALSE, intercept=TRUE)
regr
```

```
##
## Call:
## ar.ols(x = rec, order.max = 2, demean = FALSE, intercept = TRUE)
##
## Coefficients:
##        1        2
```

```
##  1.3541  -0.4632
##
## Intercept: 6.737 (1.111)
##
## Order selected 2  sigma^2 estimated as  89.72
```

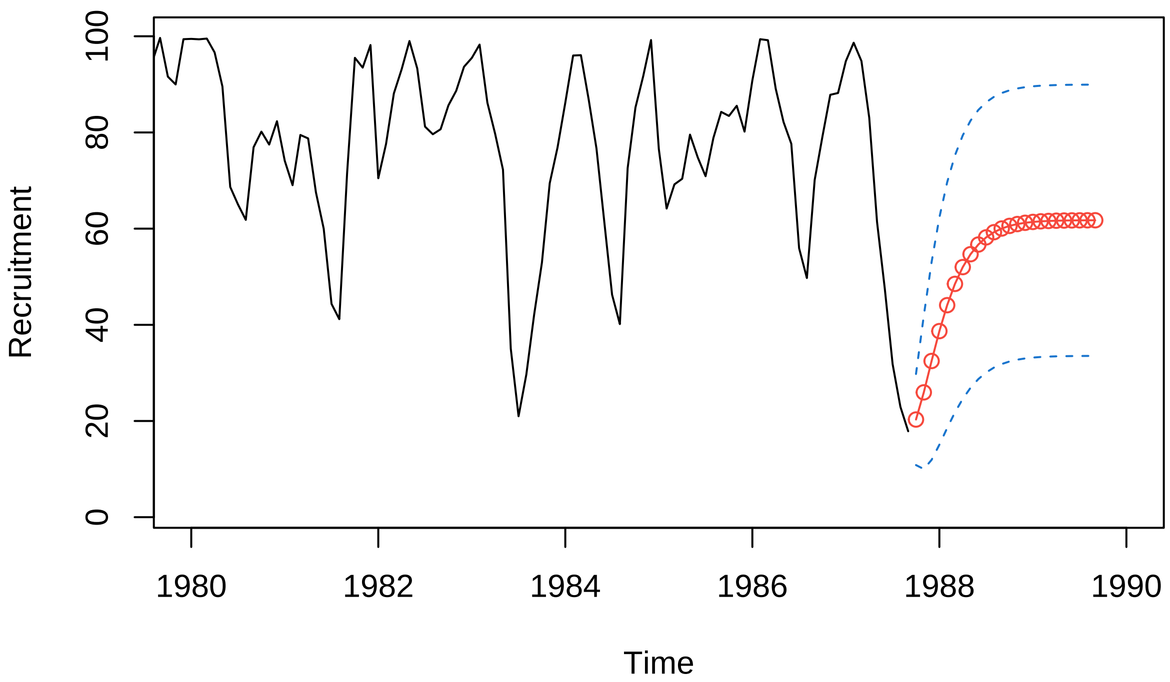
```
regr$asy.se.coef # std. errors of the estimates
```

```
## $x.mean
## [1] 1.110599
##
## $ar
## [1] 0.04178901 0.04187942
```

$$x_t = \phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + w_t \tag{3}$$
$$\hat{x}_t = \hat{\phi}_0 + \hat{\phi}_1 x_{t-1} + \hat{\phi}_2 x_{t-2} \tag{4}$$
$$\hat{x}_t = 6.74_{(1.11)} + 1.35_{(.04)} x_{t-1} - .46_{(.04)} x_{t-2} \tag{5}$$

with $\hat{\sigma}_w^2 = 89.72$

```
# example 3.24
# Forecasting the Recruitment Series
regr <- ar.ols(rec, order=2, demean=FALSE, intercept = TRUE)
fore <- predict(regr, n.ahead = 24)
ts.plot(rec, fore$pred, col=1:2, xlim=c(1980,1990), ylab="Recruitment")
lines(fore$pred, type="p", col=2)
lines(fore$pred+fore$se, lty="dashed", col=4)
lines(fore$pred-fore$se, lty="dashed", col=4)
```




```
# example 3.27 yule-walker estimation of Recruitment
# which are nearly identical to that of example 3.17

rec.yw <- ar.yw(rec, order=2)
rec.yw$x.mean # =62.26 (mean estimate)
```

```
## [1] 62.26278
```

```
rec.yw$ar # =1.33, -.44 (paramter estimates)
```

```
## [1]  1.3315874 -0.4445447
```

```
sqrt(diag(rec.yw$asy.var.coef)) # =.04, .04 (std. errors)
```

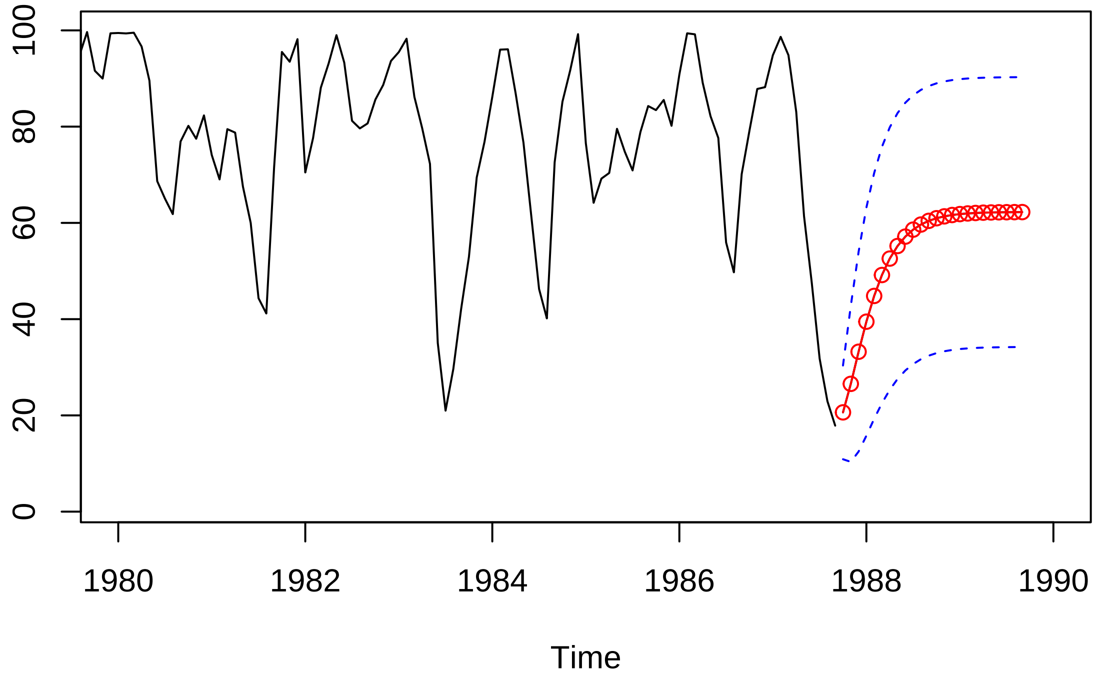
```
## [1] 0.04222637 0.04222637
```

```
rec.yw$var.pred #= 94.80 (error variance estimate)
```

```
## [1] 94.79912
```

To obtain the 24 month ahead predictions and their std. errors.

```
rec.pr <- predict(rec.yw, n.ahead = 24)
U <- rec.pr$pred + rec.pr$se
L <- rec.pr$pred - rec.pr$se
minx = min(rec,L); maxx = max(rec,U)
ts.plot(rec, rec.pr$pred, xlim=c(1980,1990), ylim=c(minx,maxx))
lines(rec.pr$pred, col="red", type="o")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
```




```
# example 3.30 MLE for Recruitment

rec.mle <- ar.mle(rec, order=2)
rec.mle$x.mean # 62.26
```

```
## [1] 62.26153
```

```
rec.mle$ar # 1.35, -.46
```

```
## [1]  1.3512809 -0.4612736
```

```
sqrt(diag(rec.mle$asy.var.coef)) # .04, .04
```

```
## [1] 0.04099159 0.04099159
```

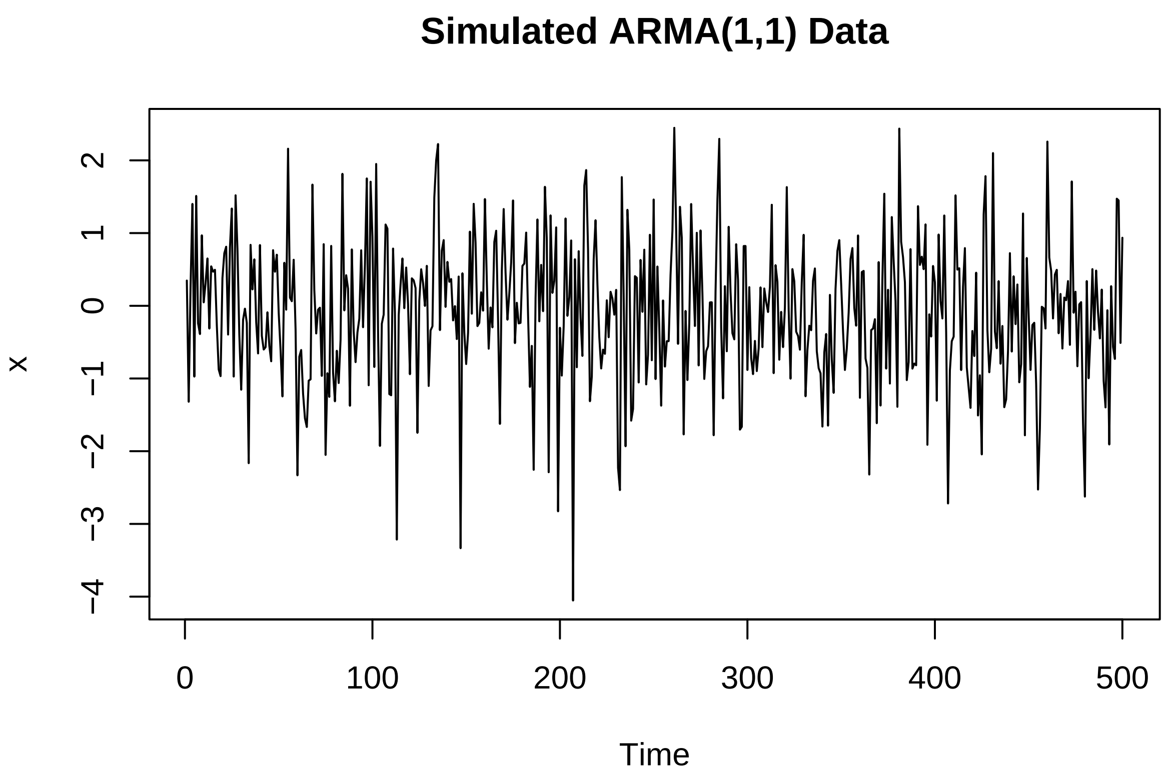
```
rec.mle$var.pred # 89.34
```

```
## [1] 89.33597
```

5.2 Problem 3.20

Repeat the following numerical exercise three times. Generate n = 500 observations from the ARMA model given by $x_t = .9x_{t-1} + w_t - .9w_{t-1}$, with $w_t \sim iidN(0, 1)$. Plot the simulated data, compute the sample ACF and PACF of the simulated data, and fit an ARMA(1, 1) model to the data. What happened and how do you explain the results?

```
x <- arima.sim(list(order=c(1,0,1), ar=.9, ma=-.9), n=500, rand.gen = rnorm, sd=1)
plot(x, main="Simulated ARMA(1,1) Data")
```




```
acf(x, plot = FALSE, main="Sample ACF of Simulated ARMA(1,1) Data")
```

```
##
```

```
## Autocorrelations of series 'x', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.035 -0.049  0.047 -0.053  0.038 -0.033 -0.121  0.030 -0.038  0.030
##     11     12     13     14     15     16     17     18     19     20     21
##  0.080 -0.064  0.017 -0.016  0.069  0.004 -0.042  0.002 -0.062 -0.034  0.045
##     22     23     24     25     26
##  0.013  0.089 -0.063 -0.003 -0.012
```

```
pacf(x, plot=FALSE, main="Sample PACF of Simulated ARMA(1,1) Data")
```

```
##
## Partial autocorrelations of series 'x', by lag
##
##      1      2      3      4      5      6      7      8      9     10     11
##  0.035 -0.050  0.050 -0.060  0.048 -0.045 -0.109  0.029 -0.046  0.044  0.062
##     12     13     14     15     16     17     18     19     20     21     22
## -0.053  0.013 -0.037  0.091 -0.028 -0.009  0.001 -0.074 -0.020  0.027  0.036
##     23     24     25     26
##  0.089 -0.081  0.017 -0.070
```

```
arima(x, order=c(1,0,1))
```

```
## Warning in arima(x, order = c(1, 0, 1)): possible convergence problem: optim
## gave code = 1
```

```
##
## Call:
## arima(x = x, order = c(1, 0, 1))
##
## Coefficients:
```

```
## Warning in sqrt(diag(x$var.coef)): NaNs produced
```

```
##           ar1     ma1  intercept
##       -0.7415  0.7996    -0.0893
## s.e.      NaN     NaN     0.0449
##
## sigma^2 estimated as 0.9454:  log likelihood = -695.45,  aic = 1398.9
```

5.3 Problem 3.21

Generate 10 realizations of length n = 200 each of an ARMA(1,1) process with $\phi = .9, \theta = .5, \sigma^2 = 1$. Find the MLEs of the three parameters in each case and compare the estimators to the true values.

```
x <- arima.sim(list(order=c(1,0,1), ar=.9, ma=.5), n=200, rand.gen = rnorm, sd=1)
```

```
MLE <- arima(x, order=c(1,0,1), method = "ML", include.mean = TRUE)
MLE
```

```
##
## Call:
## arima(x = x, order = c(1, 0, 1), include.mean = TRUE, method = "ML")
##
## Coefficients:
##          ar1     ma1  intercept
##       0.8814  0.5413     0.7786
## s.e.  0.0335  0.0572     0.8235
##
## sigma^2 estimated as 0.8645:  log likelihood = -270.54,  aic = 549.07
```

```
CSSML <- arima(x, order=c(1,0,1), method = "CSS-ML", include.mean = TRUE)
CSSML
```

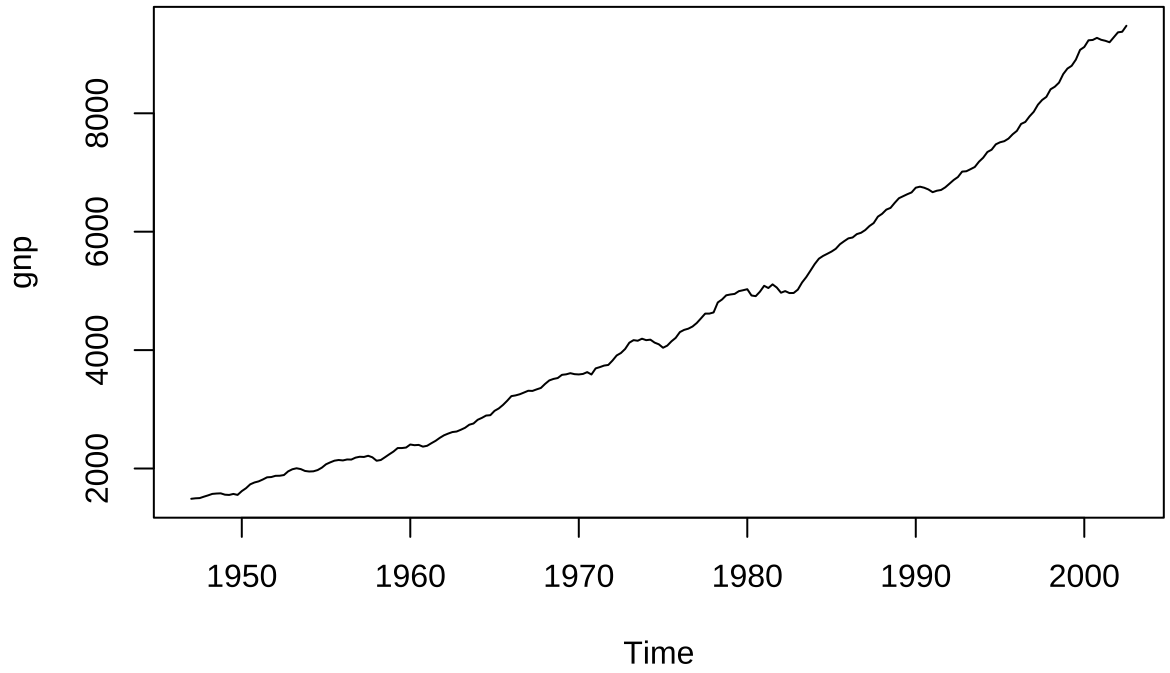
```
##
## Call:
## arima(x = x, order = c(1, 0, 1), include.mean = TRUE, method = "CSS-ML")
##
## Coefficients:
##          ar1     ma1  intercept
##       0.8815  0.5413     0.7783
## s.e.  0.0335  0.0572     0.8235
##
## sigma^2 estimated as 0.8645:  log likelihood = -270.54,  aic = 549.07
```

6 Section 3.8

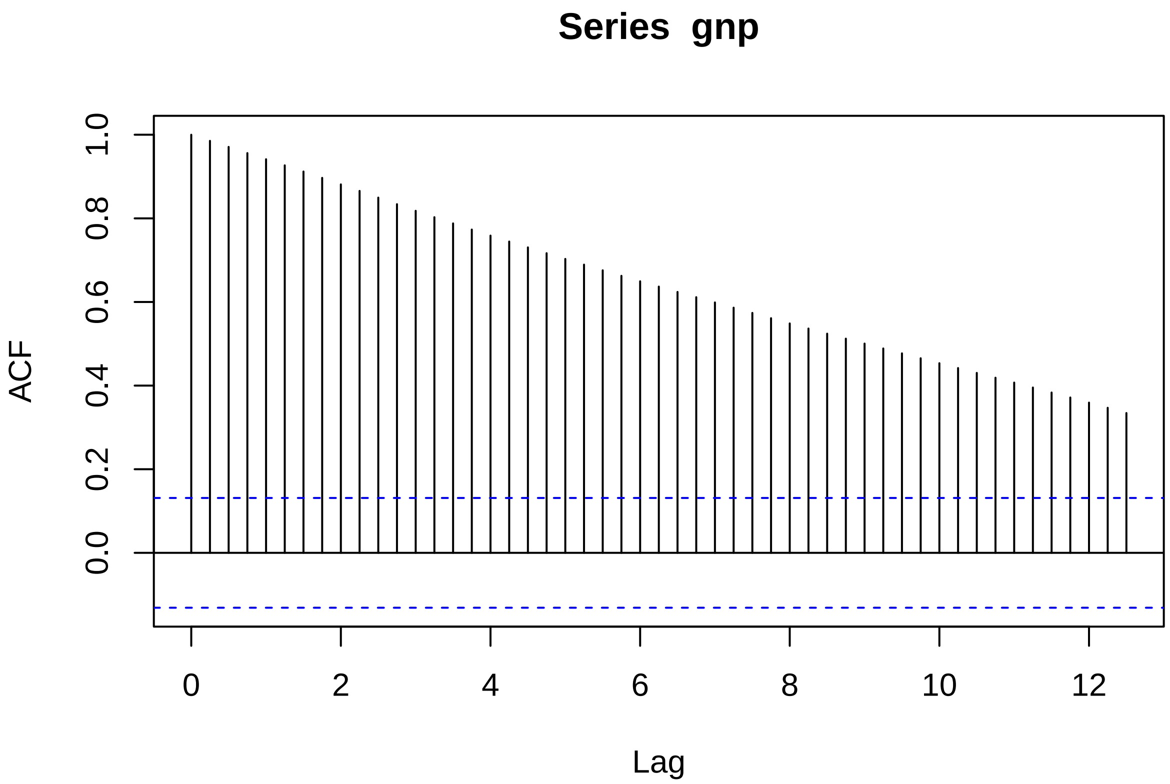
6.1 Example 3.38

Analysis of GNP Data

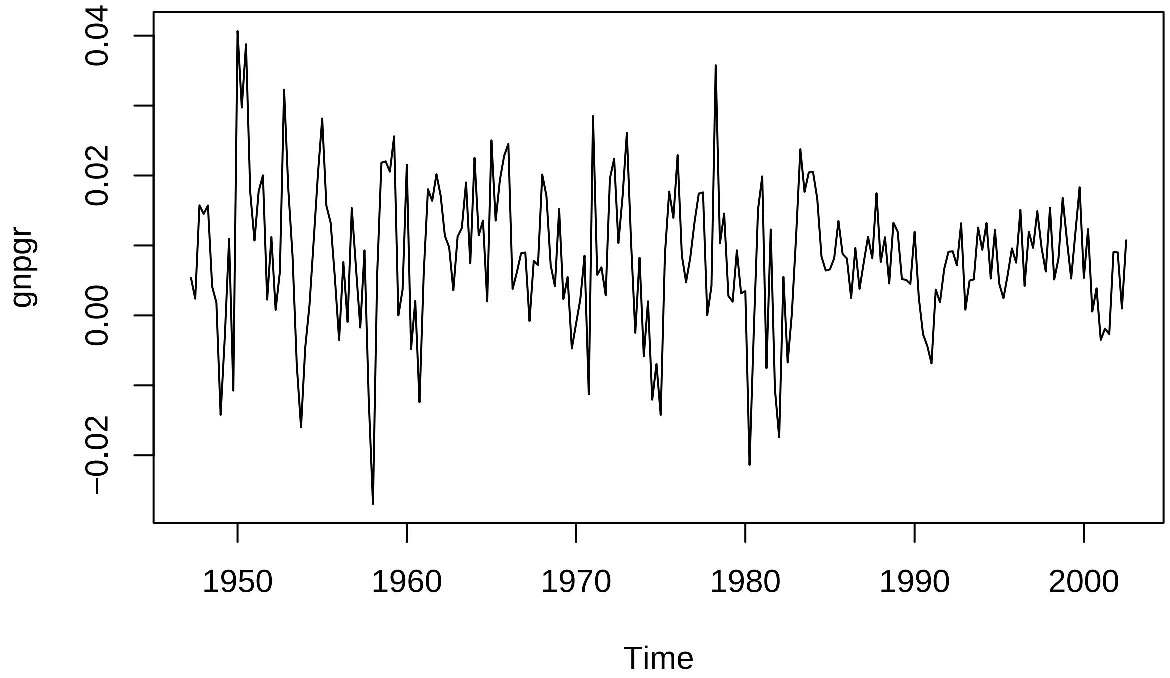
```
plot(gnp)
```



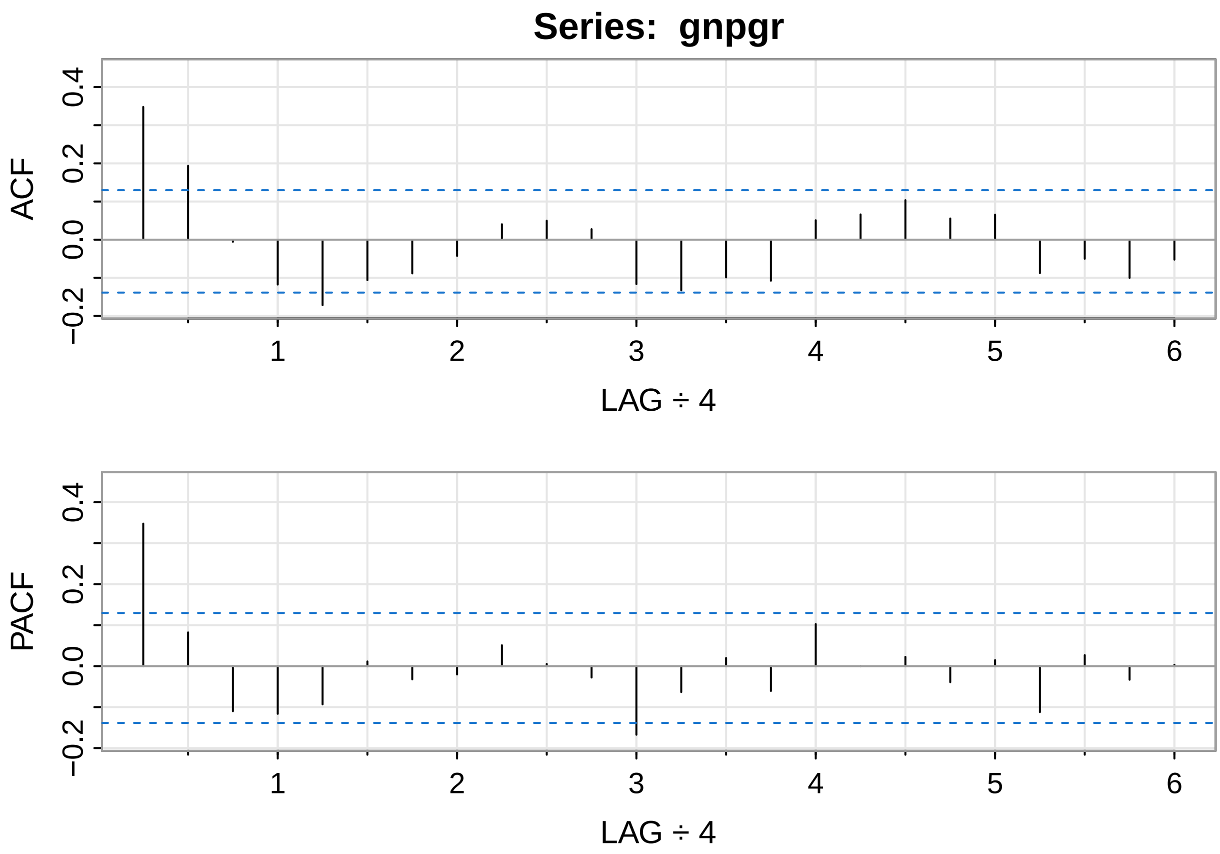

```
acf(gnp, 50)
```



```
# U.S. GNP quarterly growth rate plot
gnpgr=diff(log(gnp)) # growth rate
plot(gnpgr)
```




```
acf2(gnpgr, 24)
```

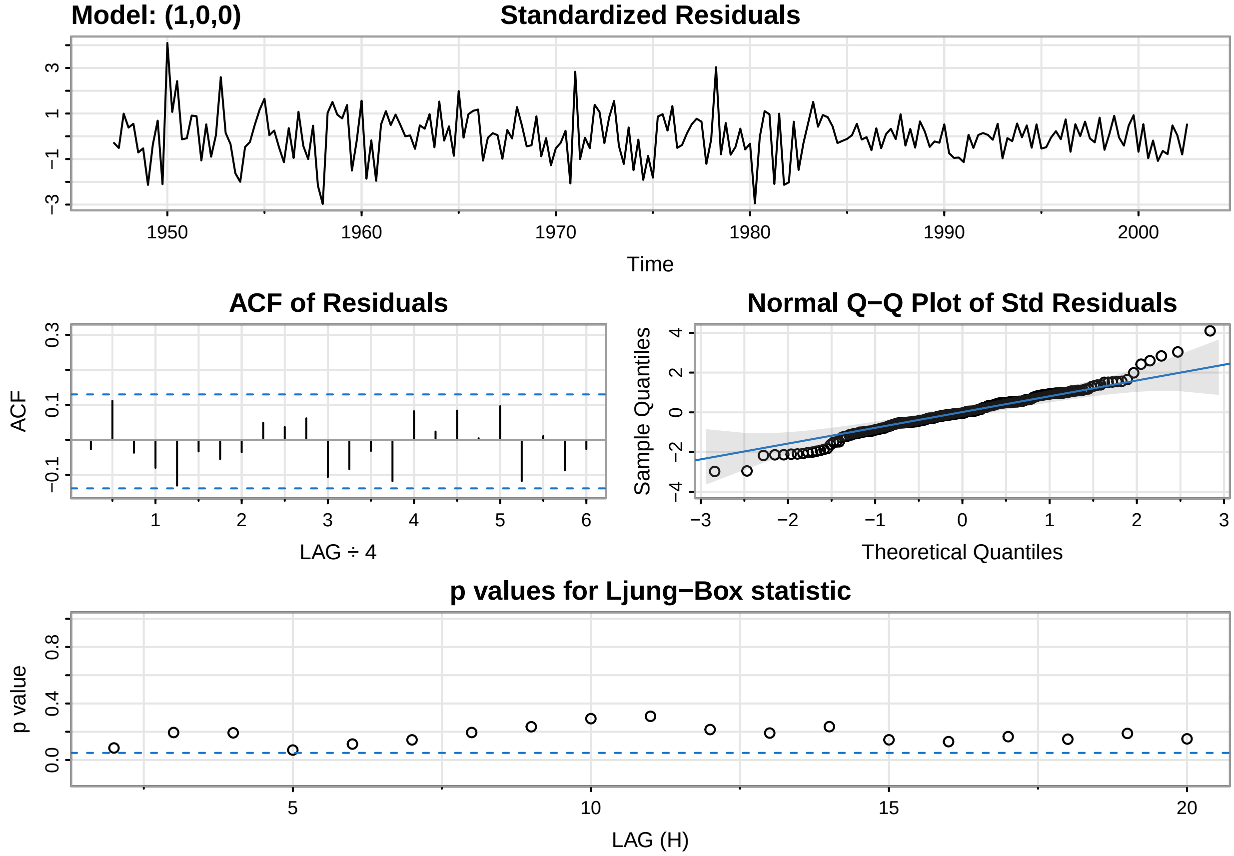



```
##       [,1] [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8] [,9] [,10] [,11] [,12] [,13]
## ACF  0.35 0.19 -0.01 -0.12 -0.17 -0.11 -0.09 -0.04 0.04  0.05  0.03 -0.12 -0.13
## PACF 0.35 0.08 -0.11 -0.12 -0.09  0.01 -0.03 -0.02 0.05  0.01 -0.03 -0.17 -0.06
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## ACF  -0.10 -0.11  0.05  0.07  0.10  0.06  0.07 -0.09 -0.05 -0.10 -0.05
## PACF  0.02 -0.06  0.10  0.00  0.02 -0.04  0.01 -0.11  0.03 -0.03  0.00
```

```
# you can also use arima()
# arima(gnpgr, order=c(1,0,0))
sarima(gnpgr, 1, 0, 0) # AR(1)
```

```
## initial  value -4.589567
## iter   2 value -4.654150
## iter   3 value -4.654150
## iter   4 value -4.654151
## iter   4 value -4.654151
## iter   4 value -4.654151
## final  value -4.654151
## converged
## initial  value -4.655919
## iter   2 value -4.655921
## iter   3 value -4.655922
```

```
## iter   4 value -4.655922
## iter   5 value -4.655922
## iter   5 value -4.655922
## iter   5 value -4.655922
## final  value -4.655922
## converged
```




```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1   xmean
##       0.3467  0.0083
## s.e.  0.0627  0.0010
##
## sigma^2 estimated as 9.03e-05:  log likelihood = 718.61,  aic = -1431.22
##
## $degrees_of_freedom
```

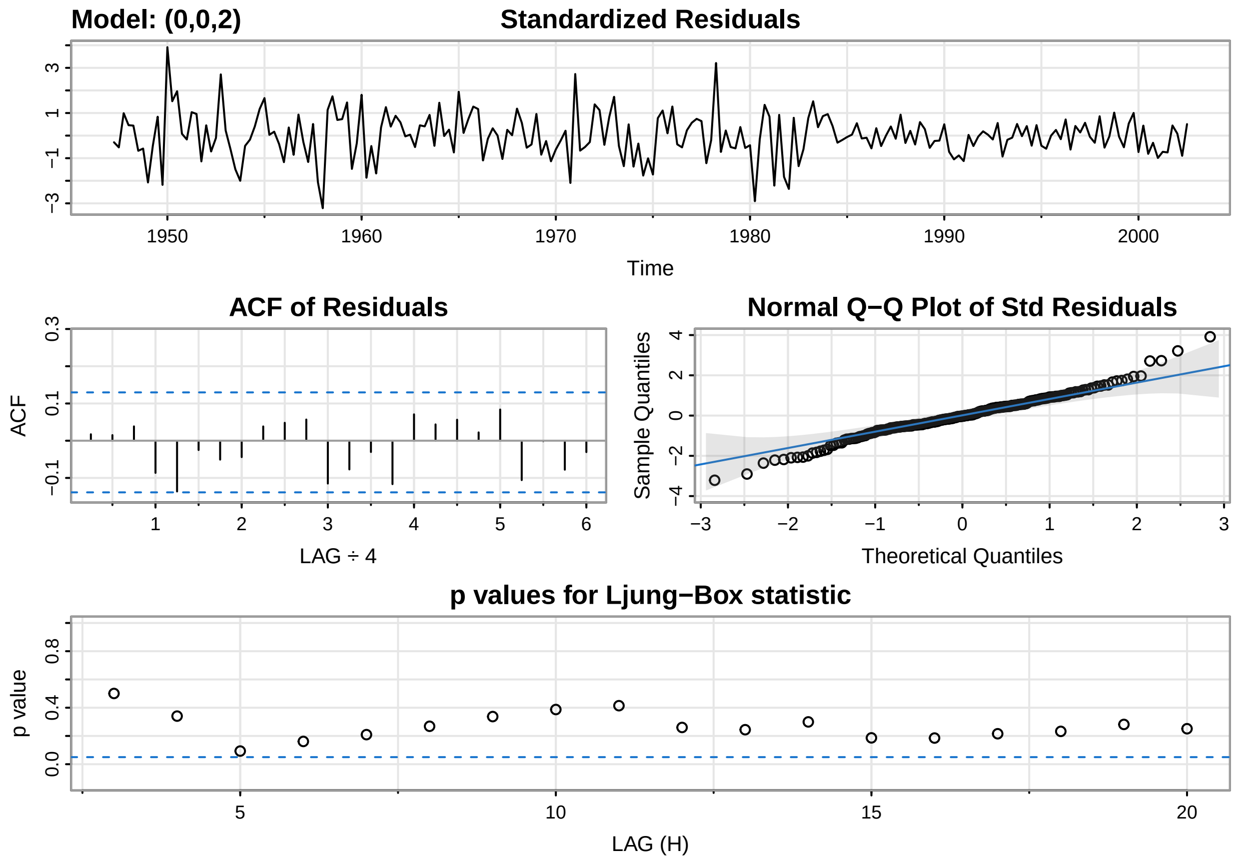
```
## [1] 220
##
## $ttable
##        Estimate     SE t.value p.value
## ar1      0.3467 0.0627  5.5255       0
## xmean    0.0083 0.0010  8.5398       0
##
## $AIC
## [1] -6.44694
##
## $AICc
## [1] -6.446693
##
## $BIC
## [1] -6.400958
```

$$x_t = .008_{(.001)}(1 - .347) + .347_{(.063)} x_{t-1} + \hat{w}_t \tag{6}$$

with $\hat{\sigma}_w = 0.0095$; note that the constant is .008(1-.347)=.005

```
sarima(gnpgr, 0,0,2) #MA(2)
```

```
## initial  value -4.591629
## iter   2 value -4.661095
## iter   3 value -4.662220
## iter   4 value -4.662243
## iter   5 value -4.662243
## iter   6 value -4.662243
## iter   6 value -4.662243
## iter   6 value -4.662243
## final  value -4.662243
## converged
## initial  value -4.662022
## iter   2 value -4.662023
## iter   2 value -4.662023
## iter   2 value -4.662023
## final  value -4.662023
## converged
```




```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##          ma1     ma2   xmean
##       0.3028  0.2035  0.0083
## s.e.  0.0654  0.0644  0.0010
##
## sigma^2 estimated as 8.919e-05:  log likelihood = 719.96,  aic = -1431.93
##
## $degrees_of_freedom
## [1] 219
##
## $ttable
##       Estimate     SE t.value p.value
## ma1     0.3028 0.0654  4.6272  0.0000
## ma2     0.2035 0.0644  3.1594  0.0018
## xmean   0.0083 0.0010  8.7178  0.0000
```

```
##
## $AIC
## [1] -6.450133
##
## $AICc
## [1] -6.449637
##
## $BIC
## [1] -6.388823
```

$$x_t = 0.008_{(0.001)} + .303_{(.065)} \hat{w}_{t-1} + .204_{(.064)} \hat{w}_{t-2} + \hat{w}_t \tag{7}$$

with $\hat{\sigma}_w = .0094$

Diagnostics for ARIMA(0,0,2):

- Inspection of the time plot of the standardized residuals shows no obvious patterns. Notice that there are outliers, however, with a few values exceeding 3 standard deviations in magnitude.
- The ACF of the standardized residuals shows no apparent departure from the model assumptions, and the Q-statistic is never significant at the lags shown.
- The normal Q-Q plot of the residuals shows departure from normality at the tails due to the outliers that occurred primarily in the 1950s and the early 1980s.
- The model appears to fit well except for the fact that a distribution with heavier tails than the normal distribution should be employed

```
ARMAtoMA(ar=.35, ma=0, 10) # prints psi-weights
```

```
##  [1] 3.500000e-01 1.225000e-01 4.287500e-02 1.500625e-02 5.252187e-03
##  [6] 1.838266e-03 6.433930e-04 2.251875e-04 7.881564e-05 2.758547e-05
```

Notice that, in this example, not including a constant leads to the wrong conclusions about the nature of the U.S. economy. Not including a constant assume the average quarterly growth rate is zero, whereas the U.S. GNP average quarterly growth rate is about 1% (as you can see from the U.S. GNP quarterly growth rate plot).

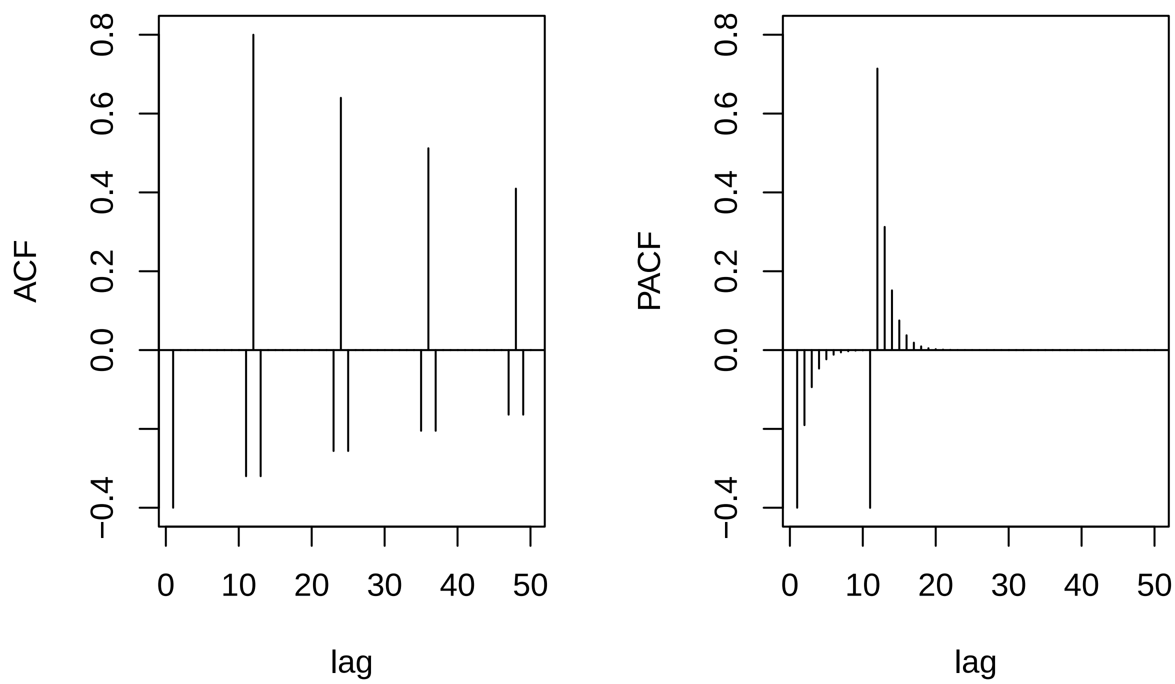
```
# MA(2) w/o constant, you should not use this
# sarima(gnpgr, 0, 0, 2, no.constant = TRUE)
```

7 Section 3.9

7.1 Example 3.44

$ARMA(0,1)x(1,0)_{12}$

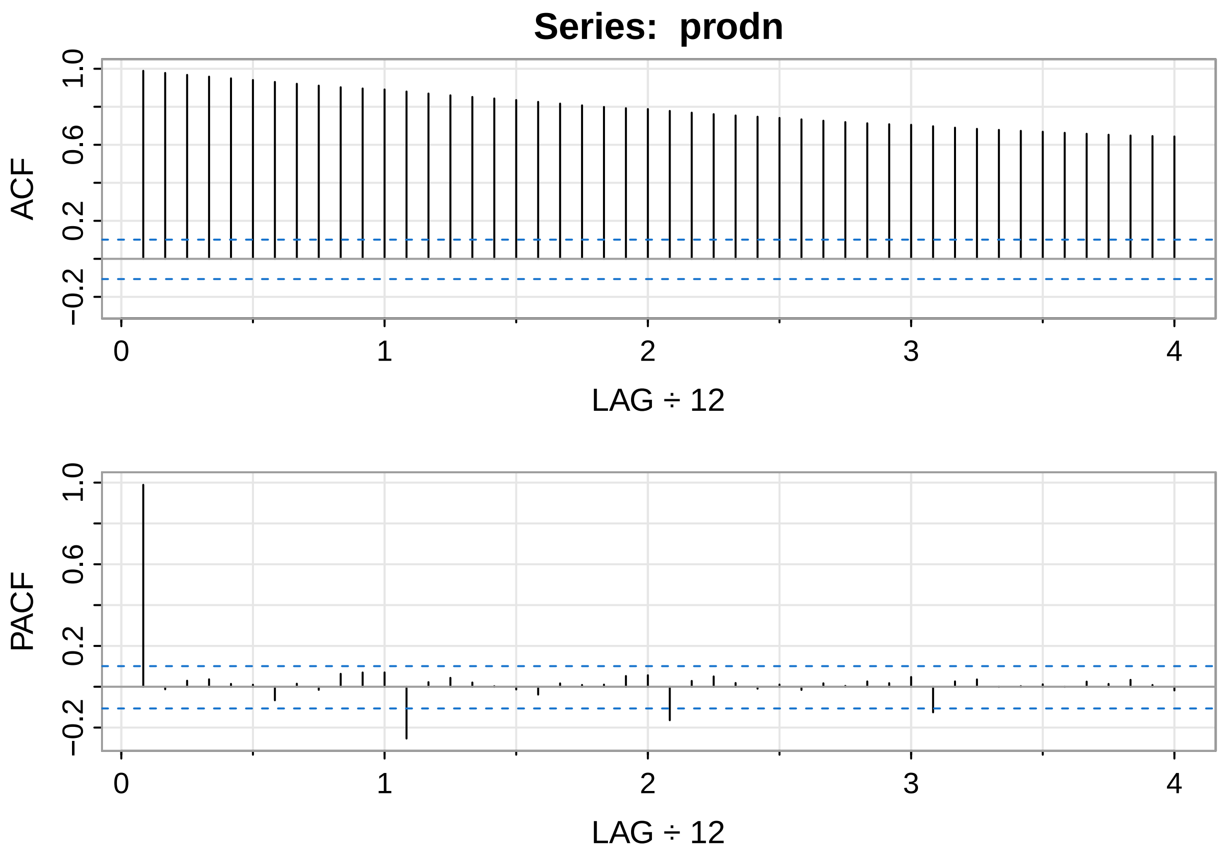
```
phi = c(rep(0,11),.8)
ACF = ARMAacf(ar=phi, ma=-.5, 50)[-1] # [-1] removes 0 lag
PACF = ARMAacf(ar=phi, ma=-.5, 50, pacf=TRUE)
par(mfrow=c(1,2))
plot(ACF, type="h", xlab="lag", ylim=c(-.4,.8)); abline(h=0)
plot(PACF, type="h", xlab="lag", ylim=c(-.4,.8)); abline(h=0)
```



7.2 Example 3.46

The Federal Reserve Board Production Index

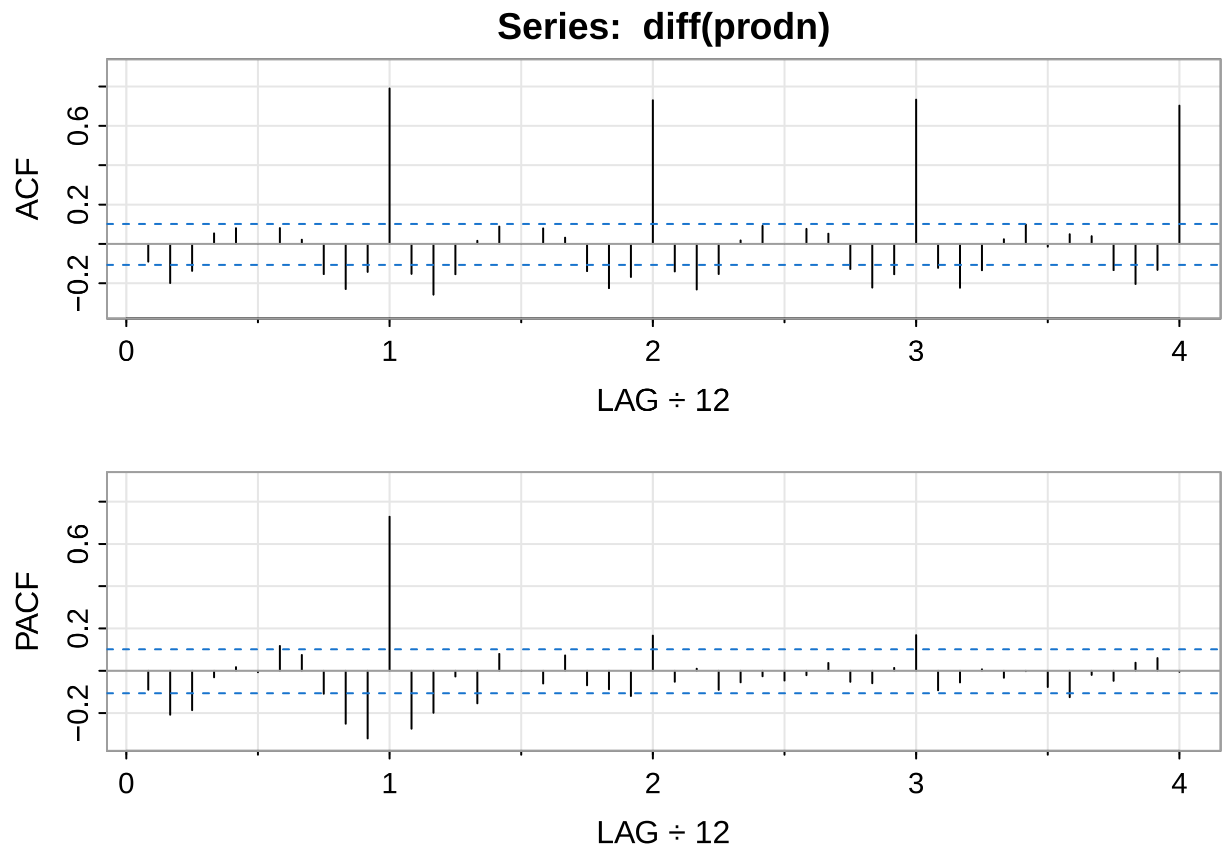
```
# ACF and PACF of the production series
acf2(prodn, 48)
```




```
##       [,1]  [,2] [,3] [,4] [,5] [,6]  [,7] [,8]  [,9] [,10] [,11] [,12] [,13]
## ACF  0.99  0.98 0.97 0.96 0.95 0.94  0.93 0.92  0.91  0.90  0.90  0.89  0.88
## PACF 0.99 -0.01 0.03 0.04 0.01 0.01 -0.07 0.01 -0.02  0.06  0.07  0.07 -0.25
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF   0.87  0.86  0.85  0.84  0.84  0.83  0.82  0.81  0.80  0.79  0.79  0.78
## PACF  0.02  0.04  0.02  0.00 -0.01 -0.04  0.02  0.01  0.01  0.05  0.06 -0.16
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
## ACF   0.77  0.76  0.75  0.75  0.74  0.73  0.73  0.72  0.71  0.71  0.71  0.70
## PACF  0.03  0.05  0.02 -0.01  0.01 -0.02  0.02  0.00  0.03  0.02  0.05 -0.13
##      [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF   0.69  0.68  0.68  0.67  0.67  0.66  0.66  0.65  0.65  0.65  0.64
## PACF  0.03  0.04  0.00  0.00  0.01  0.00  0.03  0.01  0.03  0.01 -0.02
```

The slow decay in the ACF, and the fact that the PACF at the first lag is nearly 1, all indicate nonstationary behavior.

```r
# ACF and PACF of differenced production
acf2(diff(prodn), 48)
```

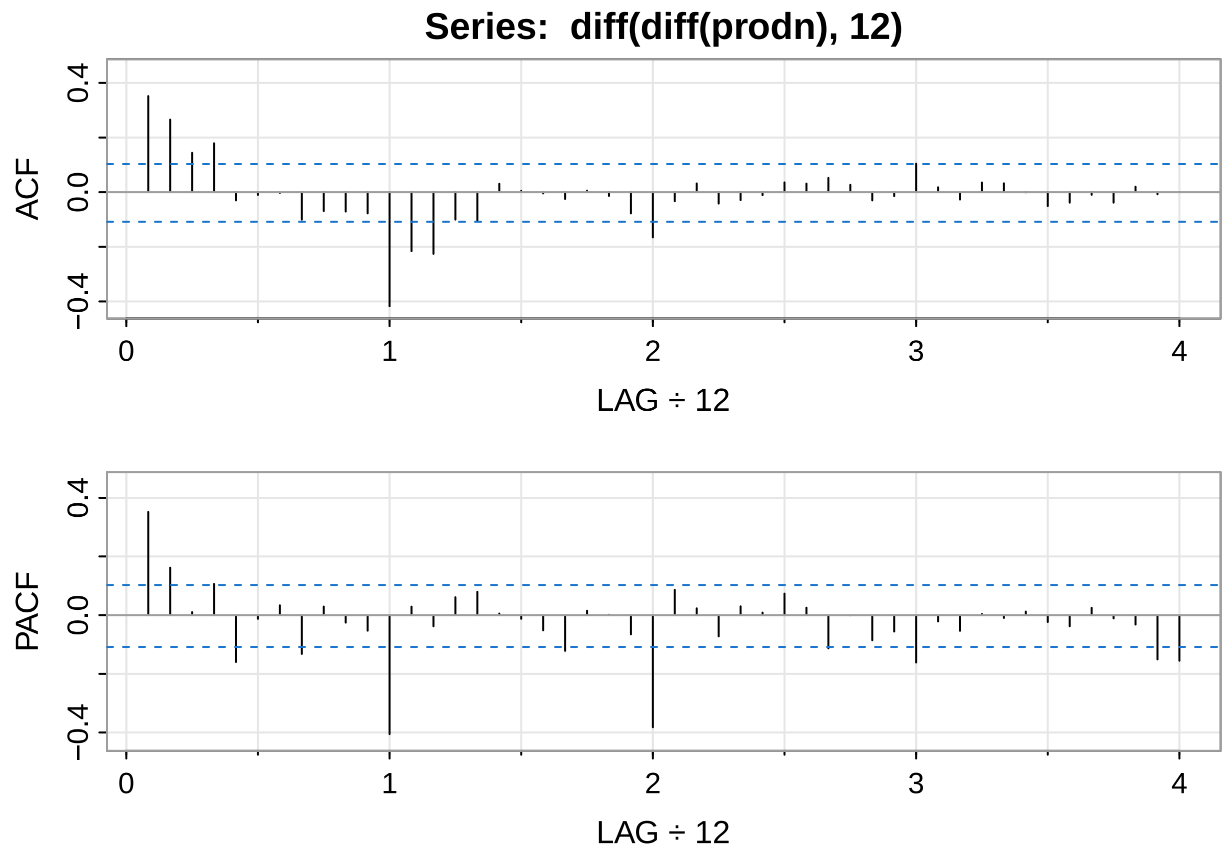



```
##       [,1]  [,2]  [,3]  [,4] [,5]  [,6] [,7] [,8]  [,9] [,10] [,11] [,12] [,13]
## ACF  -0.09 -0.20 -0.14  0.05 0.08  0.00 0.08 0.02 -0.15 -0.23 -0.14  0.79 -0.15
## PACF -0.09 -0.21 -0.19 -0.03 0.02 -0.01 0.12 0.07 -0.11 -0.25 -0.32  0.73 -0.27
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF  -0.26 -0.15  0.02  0.09     0  0.08  0.03 -0.14 -0.23 -0.17  0.73 -0.14
## PACF -0.20 -0.03 -0.15  0.08     0 -0.06  0.07 -0.07 -0.09 -0.12  0.17 -0.05
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
## ACF  -0.23 -0.15  0.02  0.09  0.00  0.08  0.05 -0.13 -0.22 -0.15  0.73 -0.12
## PACF  0.01 -0.09 -0.05 -0.03 -0.05 -0.02  0.04 -0.05 -0.06  0.01  0.17 -0.09
##      [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF  -0.22 -0.13  0.02   0.1 -0.01  0.05  0.04 -0.13 -0.20 -0.13   0.7
## PACF -0.06  0.01 -0.03   0.0 -0.08 -0.12 -0.02 -0.05  0.04  0.06   0.0
```

$(1 - B)x_t$

Noting the peaks at seasonal lags, h = 1s, 2s, 3s, 4s where s = 12 (i.e., h = 12, 24, 36, 48) with relatively slow decay suggests a seasonal difference

```
acf2(diff(diff(prodn), 12), 48)
```




```
##       [,1] [,2] [,3] [,4]  [,5]  [,6] [,7]  [,8]  [,9] [,10] [,11] [,12] [,13]
## ACF  0.35 0.27 0.14 0.18 -0.03 -0.01 0.00 -0.10 -0.07 -0.07 -0.08 -0.42 -0.22
## PACF 0.35 0.16 0.01 0.11 -0.16 -0.01 0.03 -0.13  0.03 -0.03 -0.05 -0.41  0.03
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF  -0.23 -0.10 -0.11  0.03  0.00  0.00 -0.03  0.01 -0.01 -0.08 -0.17 -0.03
## PACF -0.04  0.06  0.08  0.01 -0.01 -0.05 -0.12  0.02  0.00 -0.07 -0.38  0.09
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
## ACF   0.03 -0.04 -0.03 -0.01  0.04  0.03  0.05  0.03 -0.03 -0.01  0.10  0.02
## PACF  0.02 -0.07  0.03  0.01  0.07  0.03 -0.11  0.00 -0.09 -0.06 -0.16 -0.02
##      [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF  -0.03  0.04  0.03  0.00 -0.05 -0.04 -0.01 -0.04  0.02 -0.01  0.00
## PACF -0.05  0.00 -0.01  0.01 -0.02 -0.04  0.03 -0.01 -0.03 -0.15 -0.16
```

$(1 - B^{12})(1 - B)x_t$

First, concentrating on the seasonal (s = 12) lags, the characteristics of the ACF and PACF of this series tend to show a strong peak at h = 1s in the autocorrelation function, with smaller peaks appearing at h = 2s,3s, combined with peaks at h = 1s,2s,3s,4s in the partial autocorrelation function. It appears that either

- (i) the ACF is cutting off after lag 1s and the PACF is tailing off in the seasonal lags,
- (ii) the ACF is cutting off after lag 3s and the PACF is tailing off in the seasonal lags, or
- (iii) the ACF and PACF are both tailing off in the seasonal lags.

This suggests either

- (i) an SMA of order Q = 1
- (ii) an SMA of order Q = 3
- (iii) an SARMA of orders P = 2 (because of the two spikes in the PACF) and Q = 1

Next, inspecting the ACF and the PACF at the within season lags, h = 1,…,11, it appears that either

- (a) both the ACF and PACF are tailing off, or
- (b) that the PACF cuts off at lag 2.

This result indicates that we should either consider fitting a model

- (a) with both p > 0 and q > 0 for the nonseasonal components, say p = 1,q = 1, or
- (b) p = 2, q = 0.

It turns out that there is little difference in the results for case (a) and (b), but that (b) is slightly better, so we will concentrate on case (b).

Fitting the three models suggested by these observations we obtain:

- (i) $ARIMA(2,1,0) \times (0,1,1)_{12}$:

  AIC= 1.372, AICc= 1.378, BIC= .404
- (ii) $ARIMA(2,1,0) \times (0,1,3)_{12}$:

  AIC= 1.299, AICc= 1.305, BIC= .351
- (iii) $ARIMA(2,1,0) \times (2,1,1)_{12}$:

  AIC= 1.326, AICc= 1.332, BIC= .379

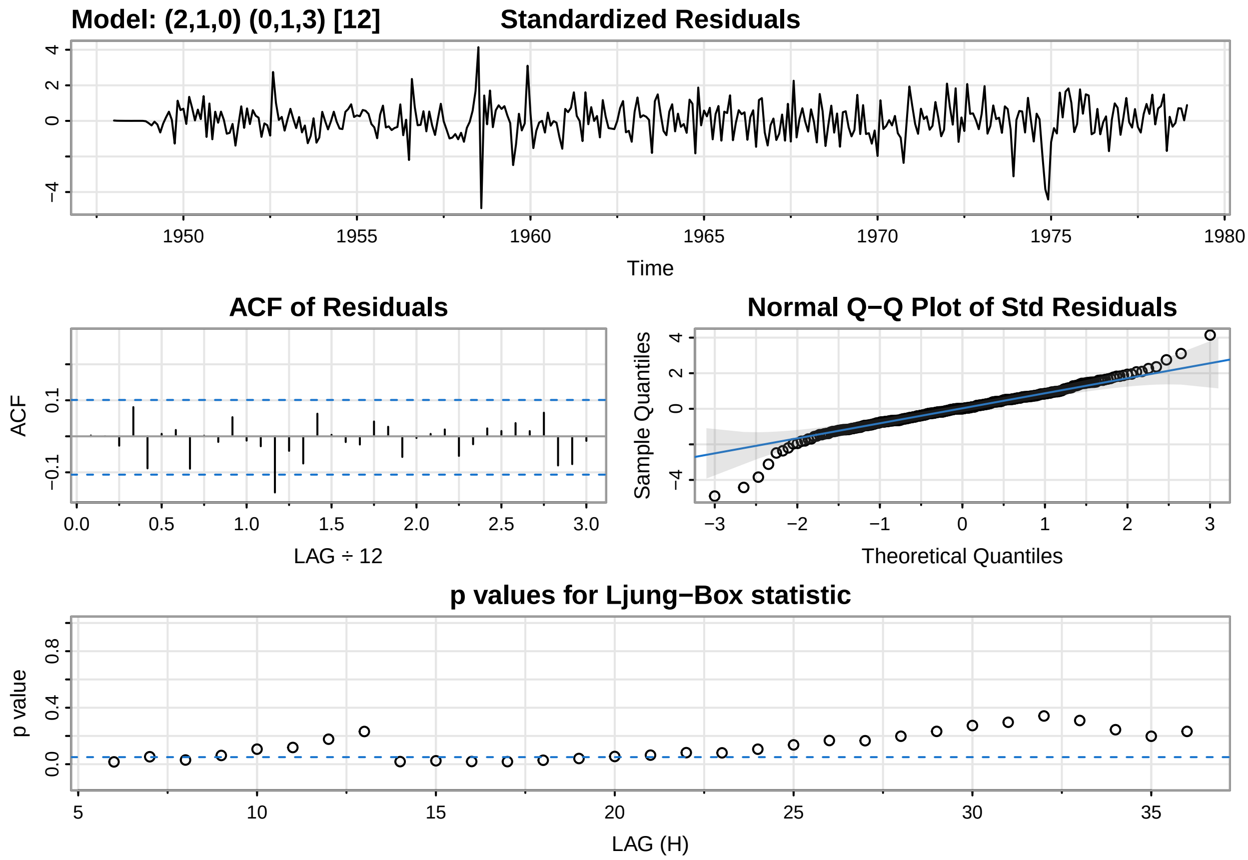
The $ARIMA(2,1,0) \times (0,1,3)_{12}$ is the preferred model, and the fitted model in this case is

$$(1 - .30_{(.05)}B - .11_{(.05)}B^2)\nabla_{12}\nabla\hat{x}_t = (1 - .74_{(.05)}B^{12} - .14_{(.06)}B^{24} + .28_{(.05)}B^{36})\hat{w}_t \tag{8}$$

with $\hat{\sigma}_w^2 = 1.312$

```
sarima(prodn, 2, 1, 0, 0, 1, 3, 12) # fit model (ii)
```

```
## initial  value 0.464774
## iter   2 value 0.217306
## iter   3 value 0.192321
## iter   4 value 0.162972
## iter   5 value 0.152650
## iter   6 value 0.149833
## iter   7 value 0.149626
## iter   8 value 0.149315
## iter   9 value 0.149303
## iter  10 value 0.149298
## iter  11 value 0.149298
## iter  11 value 0.149298
## iter  11 value 0.149298
## final  value 0.149298
## converged
## initial  value 0.152048
## iter   2 value 0.152048
## iter   3 value 0.152045
## iter   4 value 0.152043
## iter   5 value 0.152043
## iter   6 value 0.152043
## iter   6 value 0.152043
## iter   6 value 0.152043
## final  value 0.152043
## converged
```



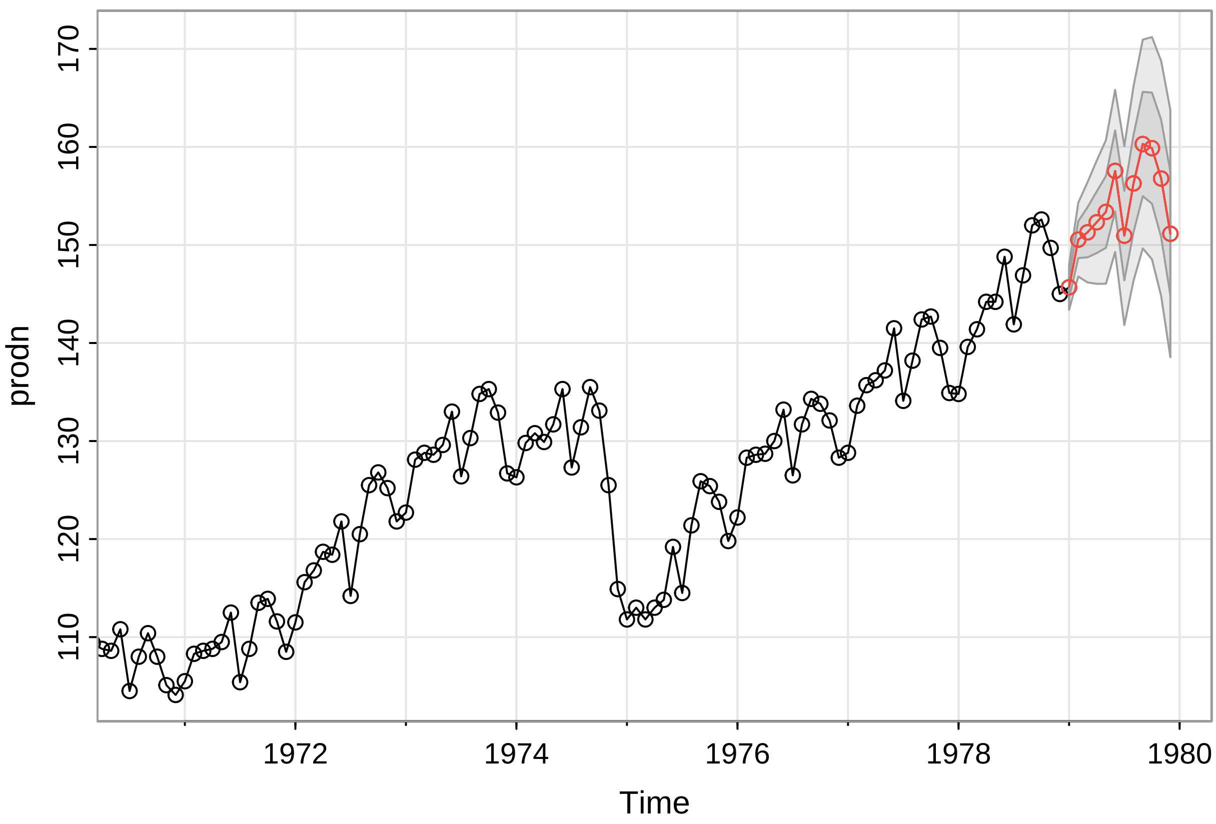

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control =
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1     ar2     sma1     sma2    sma3
##       0.3038  0.1077  -0.7393  -0.1445  0.2815
## s.e.  0.0526  0.0538   0.0539   0.0653  0.0526
##
## sigma^2 estimated as 1.312:  log likelihood = -563.98,  aic = 1139.97
##
## $degrees_of_freedom
## [1] 354
##
## $ttable
##      Estimate     SE  t.value p.value
## ar1    0.3038 0.0526   5.7708  0.0000
## ar2    0.1077 0.0538   2.0030  0.0459
## sma1  -0.7393 0.0539 -13.7175  0.0000
```

```
## sma2  -0.1445 0.0653  -2.2119  0.0276
## sma3   0.2815 0.0526   5.3475  0.0000
##
## $AIC
## [1] 3.17539
##
## $AICc
## [1] 3.175863
##
## $BIC
## [1] 3.240292
```

The diagnostics for the fit are displayed. We note the few outliers in the series as exhibited in the plot of the standardized residuals and their normal Q-Q plot, and a small amount of autocorrelation that still remains (although not at the seasonal lags) but otherwise, the model fits well.

Finally, forecasts based on the fitted model for the next 12 months.

```
sarima.for(prodn, 12, 2, 1, 0, 0, 1, 3, 12) # forecast
```




```
## $pred
##            Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 1979 145.6808 150.5469 151.2870 152.3131 153.3714 157.5532 150.9487 156.2781
```

```
##           Sep      Oct      Nov      Dec
## 1979 160.2958 159.8744 156.7729 151.1447
##
## $se
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 1979 1.145488 1.882207 2.551461 3.139668 3.663008 4.133720 4.562358 4.957034
##           Sep      Oct      Nov      Dec
## 1979 5.323957 5.667865 5.992404 6.300412
```