Exercise 9

Philipp Drebes

21.04.2023

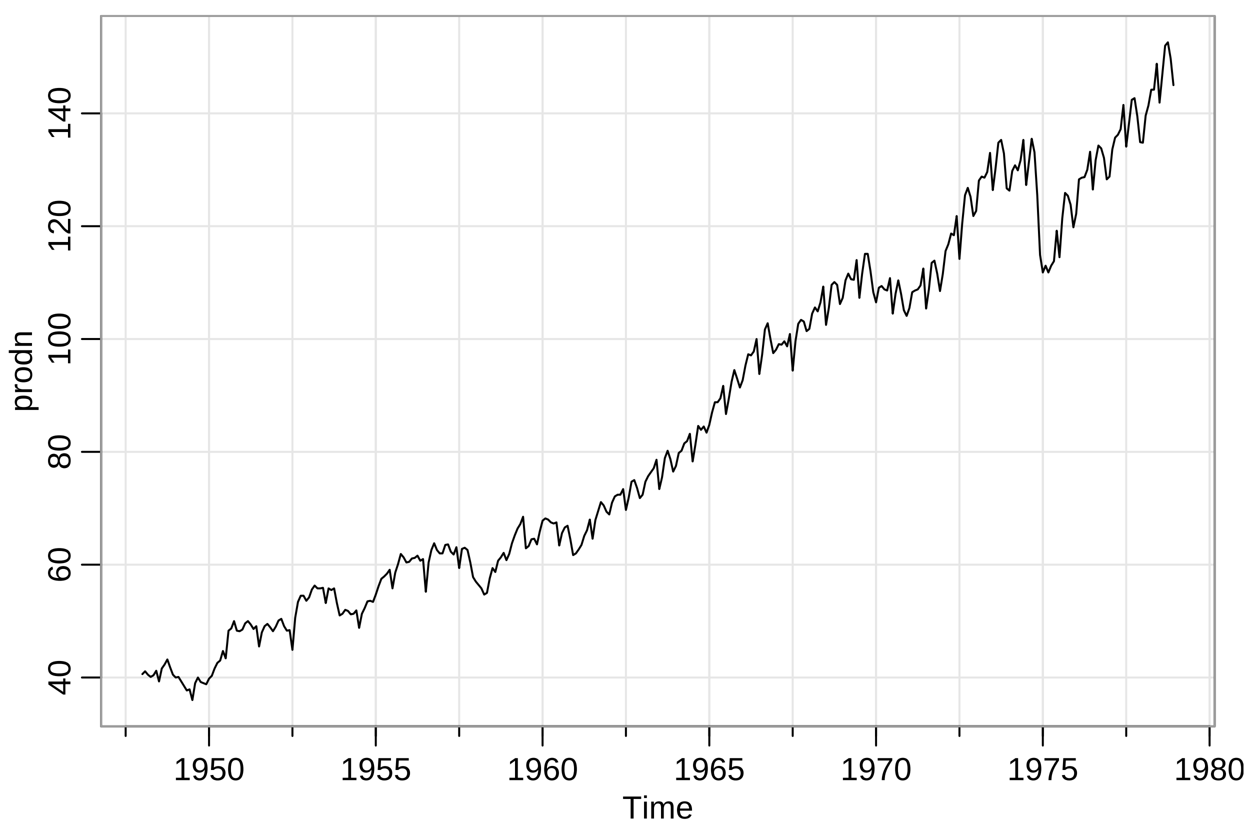
Exercise 9.1

In this exercise, we look at the time series `prodn`, which is available in the package `astsa`. It contains monthly data about the Federal Reserve Board Production Index from 1948-1978, in total the time series contains data for n = 372 months.

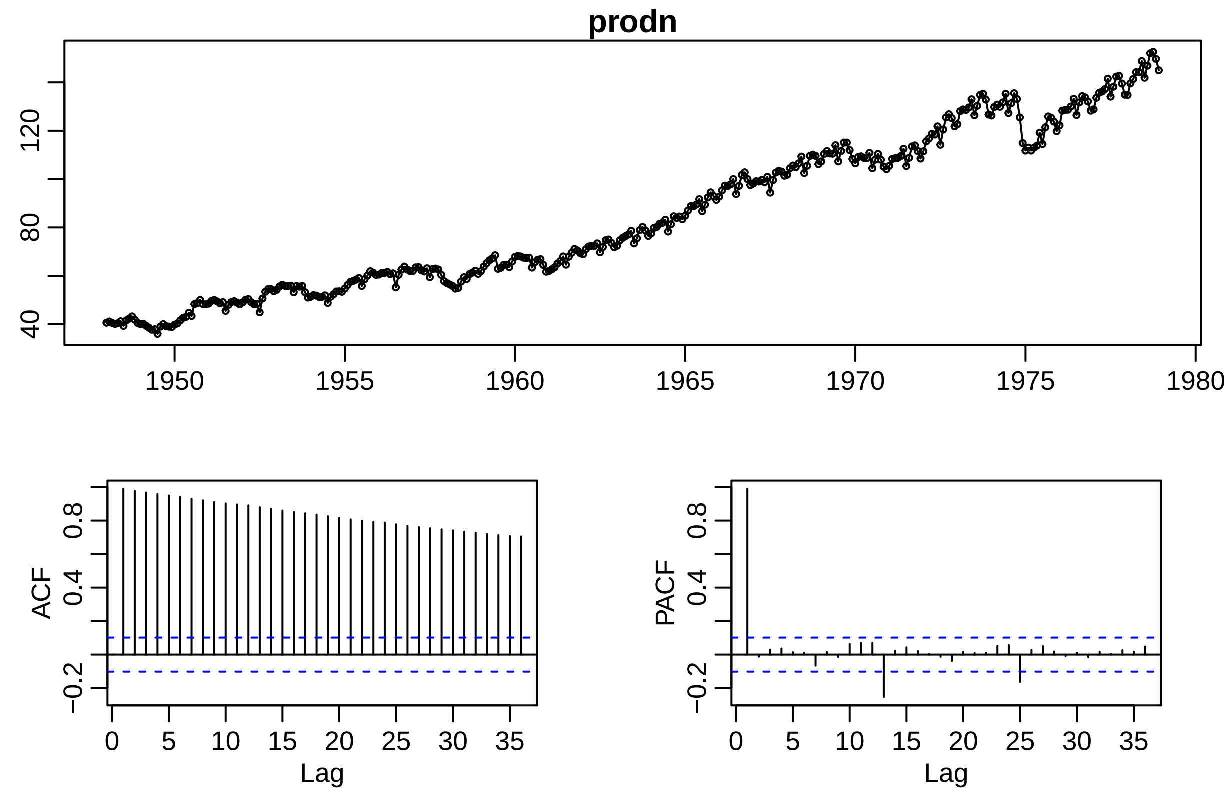
a) Plot the time series. What kind of non-stationarity is evident?

```
library(astsa, quietly = T)
library(forecast, quietly = T)

tsplot(prodn)
```

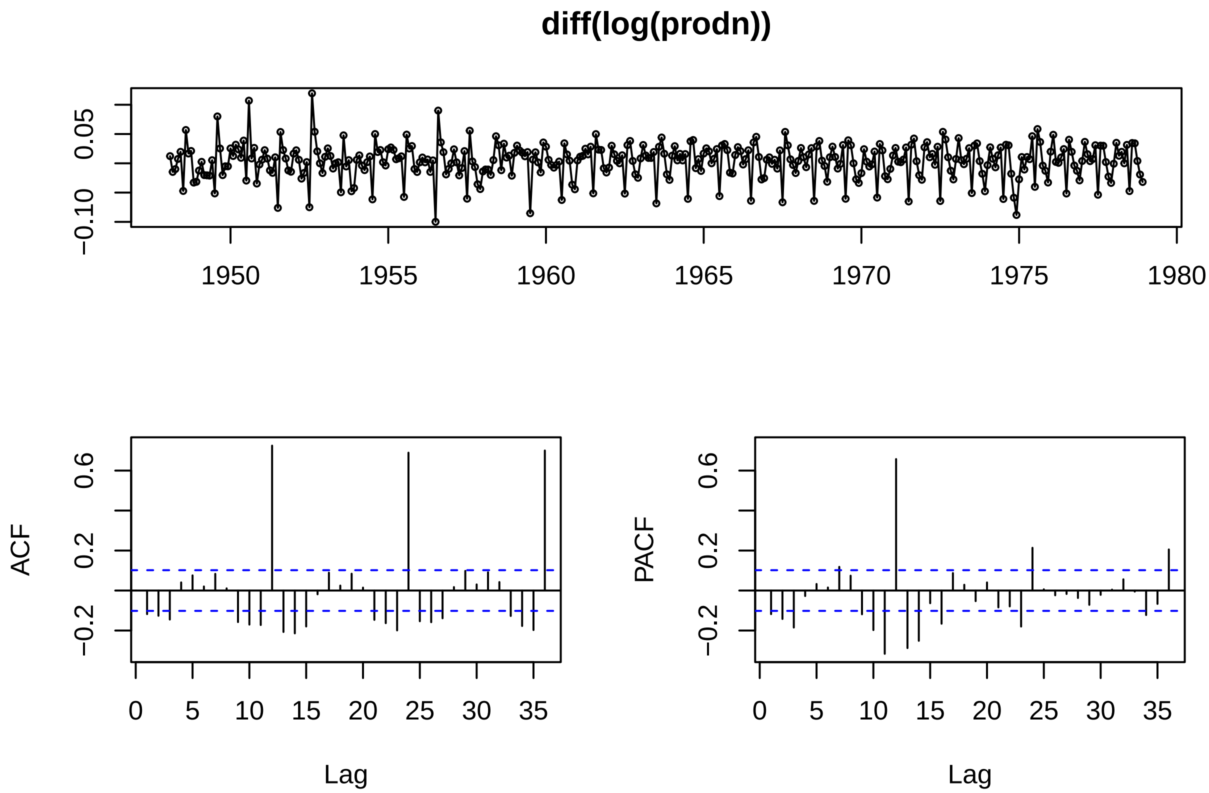



```
tsdisplay(prodn)
```

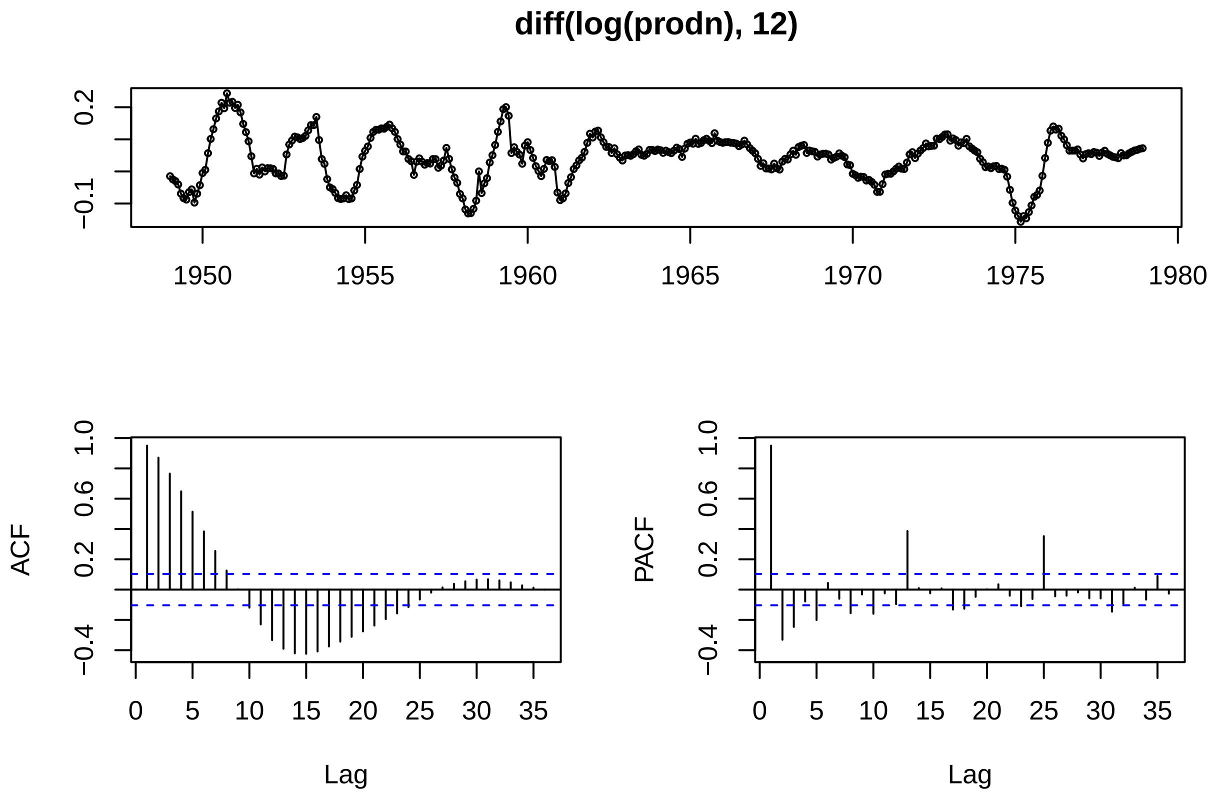


The ACF shows a slow decay, which implies that there is a trend. If there was also seasonality, the ACF would show an oscillatory behavior, which it does not.

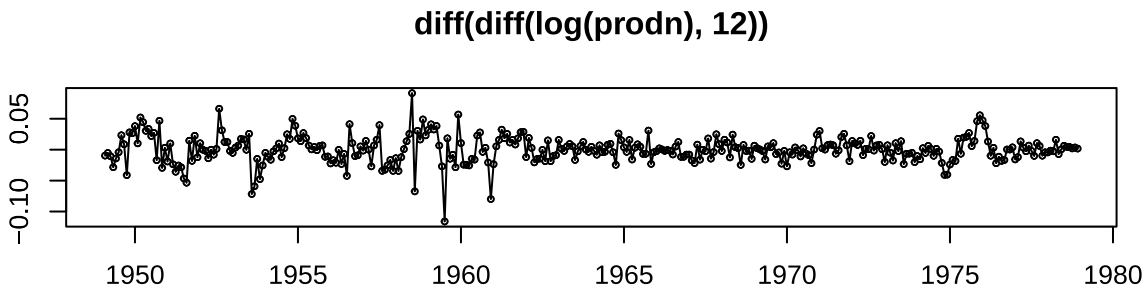
```
tsdisplay(diff(log(prodn)))
```

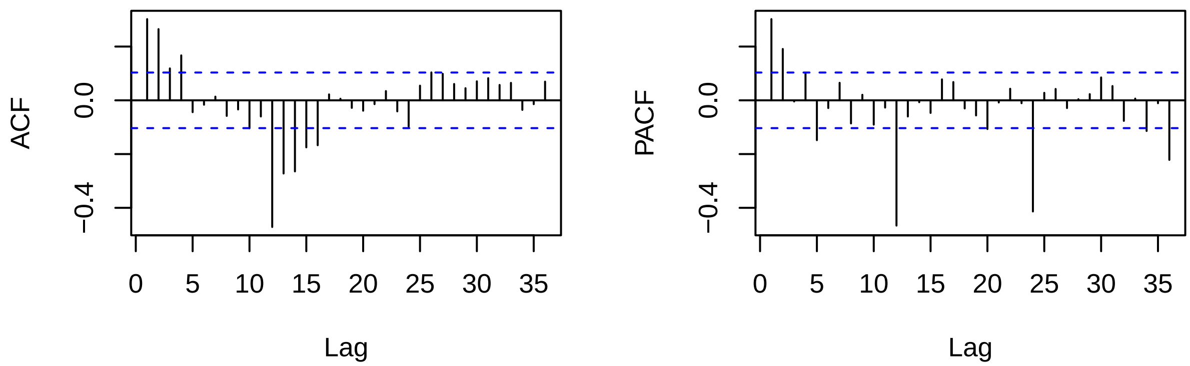



```r
tsdisplay(diff(log(prodn), 12)) # remove seasonality
```

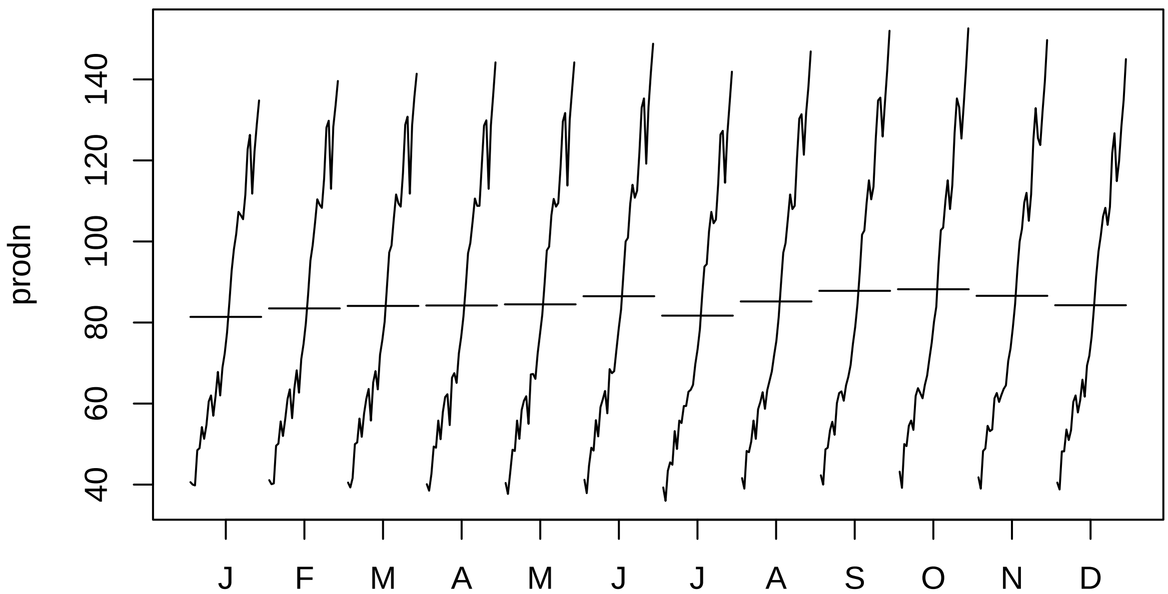



```r
tsdisplay(diff(diff(log(prodn), 12))) # remove trend
```

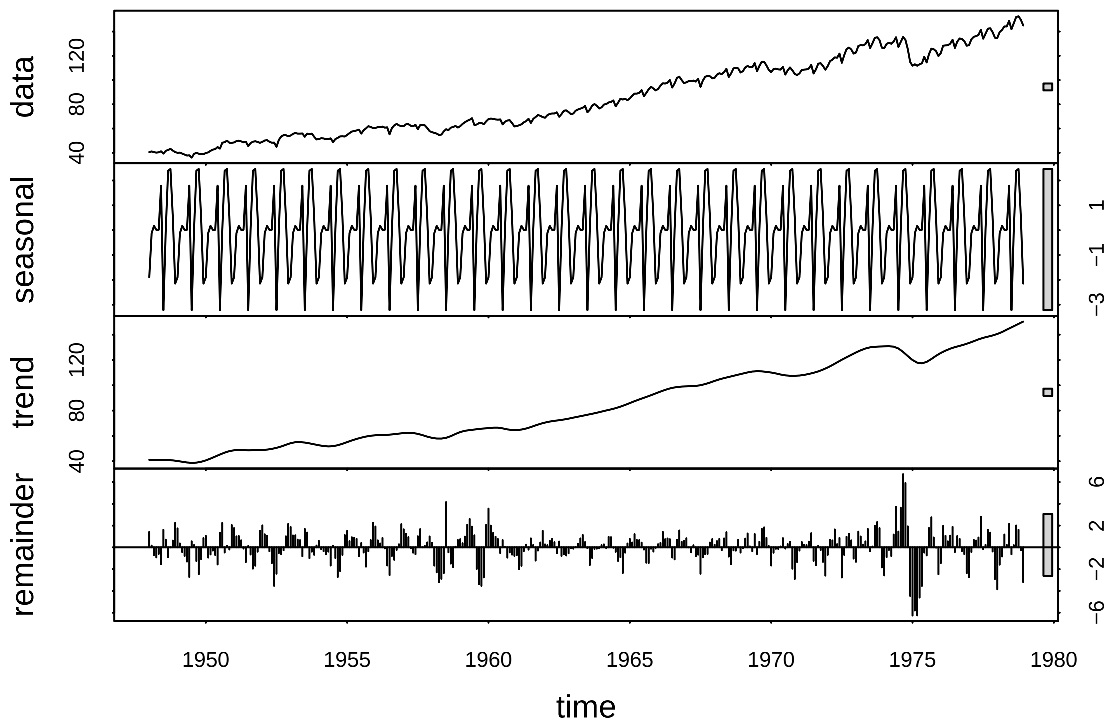





```
monthplot(prodn)
```




```r
plot(stl(prodn, s.window = 'periodic'))
```



b) How can the time series be made stationary?

By applying a SARIMA$(1,1,0)(1,1,0)^{12}$ model.

c) Based on your considerations in b), what kind of model would you fit to the original time series `prodn`? Try different fits and choose your favorite.

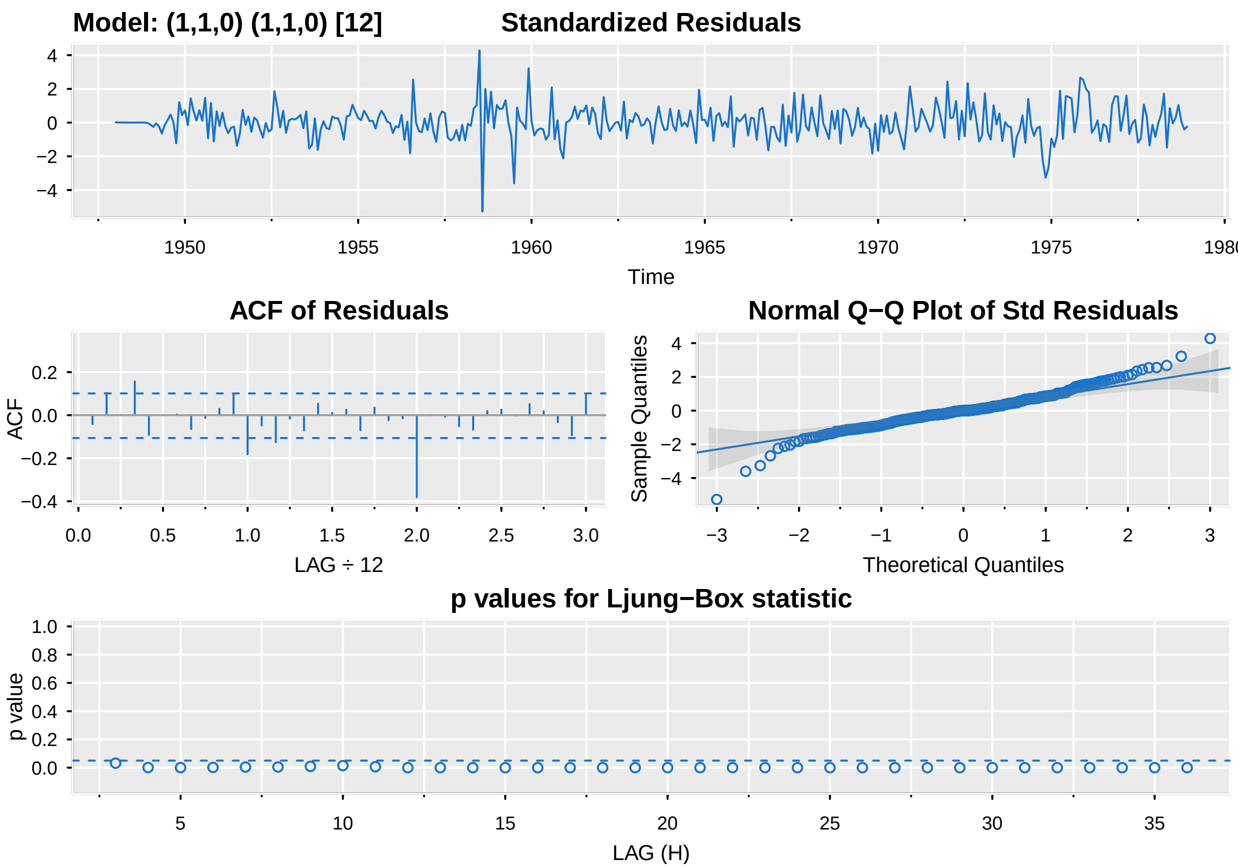
```
fit.110 <- arima(prodn, order = c(1, 1, 0), seasonal = c(1, 1, 0))
fit.110
```

```
##
## Call:
## arima(x = prodn, order = c(1, 1, 0), seasonal = c(1, 1, 0))
##
## Coefficients:
##          ar1     sar1
##       0.3530  -0.4147
## s.e.  0.0493   0.0474
##
## sigma^2 estimated as 1.813:  log likelihood = -617.36,  aic = 1240.72
```

```
sarima(prodn, 1, 1, 0, 1, 1, 0, 12, gg = TRUE, col = 4)
```

```
## initial  value 0.474717
## iter   2 value 0.309458
## iter   3 value 0.309455
## iter   4 value 0.309454
## iter   4 value 0.309454
## iter   4 value 0.309454
```

```
## final  value 0.309454
## converged
## initial  value 0.300777
## iter   2 value 0.300729
## iter   3 value 0.300727
## iter   3 value 0.300727
## iter   3 value 0.300727
## final  value 0.300727
## converged
```



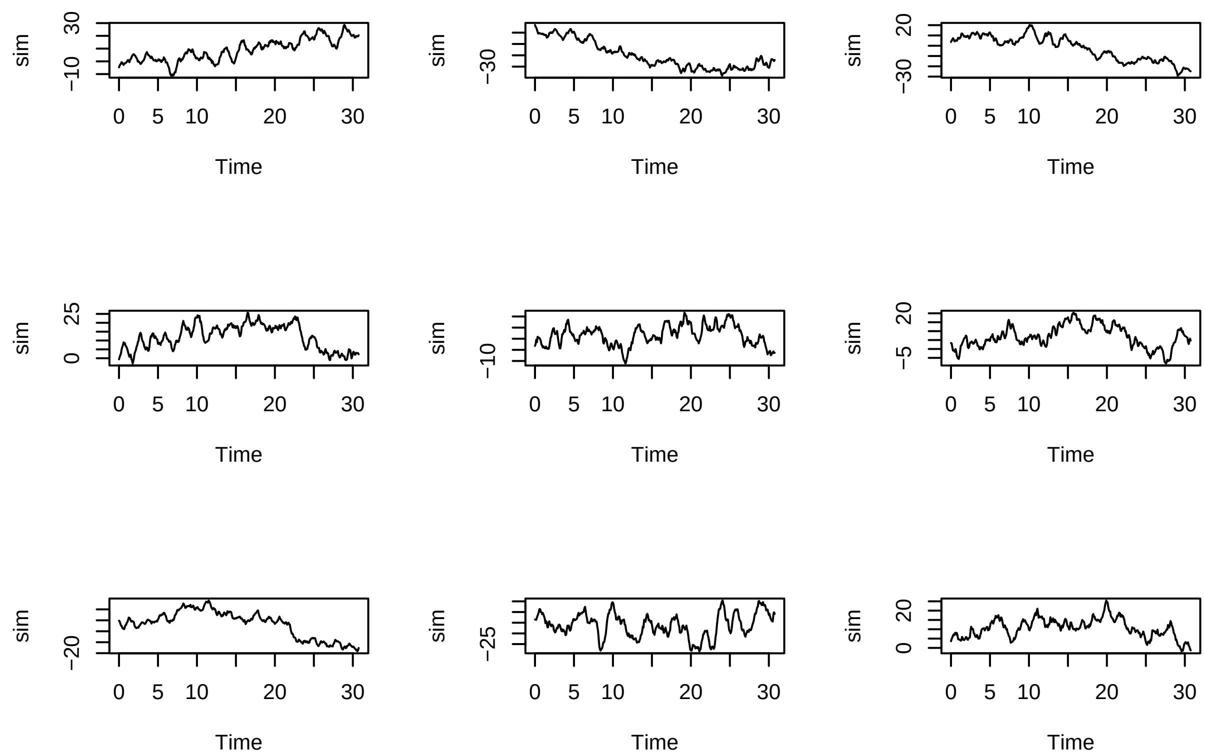

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control = list(trace =
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1     sar1
##       0.3530  -0.4147
## s.e.  0.0493   0.0474
##
## sigma^2 estimated as 1.813:  log likelihood = -617.36,  aic = 1240.72
##
## $degrees_of_freedom
## [1] 357
```

```
##
## $ttable
##      Estimate     SE t.value p.value
## ar1    0.3530 0.0493  7.1651       0
## sar1  -0.4147 0.0474 -8.7553       0
##
## $AIC
## [1] 3.456045
##
## $AICc
## [1] 3.456139
##
## $BIC
## [1] 3.488496
```

```
set.seed(3)
par(mfrow = c(3, 3))
summary(fit.110)
```

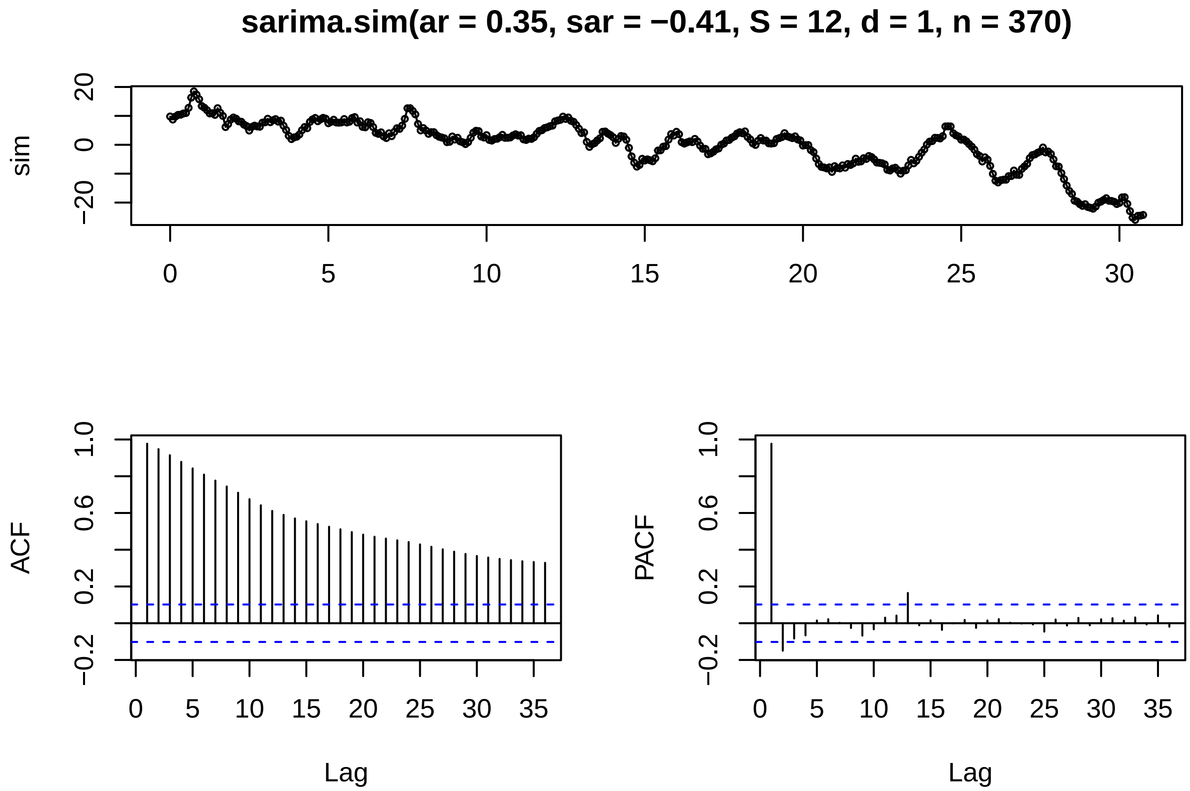
```
##
## Call:
## arima(x = prodn, order = c(1, 1, 0), seasonal = c(1, 1, 0))
##
## Coefficients:
##          ar1     sar1
##       0.3530  -0.4147
## s.e.  0.0493   0.0474
##
## sigma^2 estimated as 1.813:  log likelihood = -617.36,  aic = 1240.72
##
## Training set error measures:
##                      ME     RMSE       MAE        MPE     MAPE      MASE
## Training set 0.02312018 1.322627 0.9462347 0.03040184 1.206129 0.4810264
##                     ACF1
## Training set -0.04127911
```

```
replicate(9,
      plot(sarima.sim(ar = 0.35, sar = -0.41, S = 12, d = 1, n = 370) , ylab = 'sim'))
```




```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
##
## [[5]]
## NULL
##
## [[6]]
## NULL
##
## [[7]]
## NULL
##
## [[8]]
## NULL
##
## [[9]]
## NULL
```

```r
tsdisplay(sarima.sim(ar = 0.35, sar = -0.41, S = 12, d = 1, n = 370) , ylab = 'sim')
```



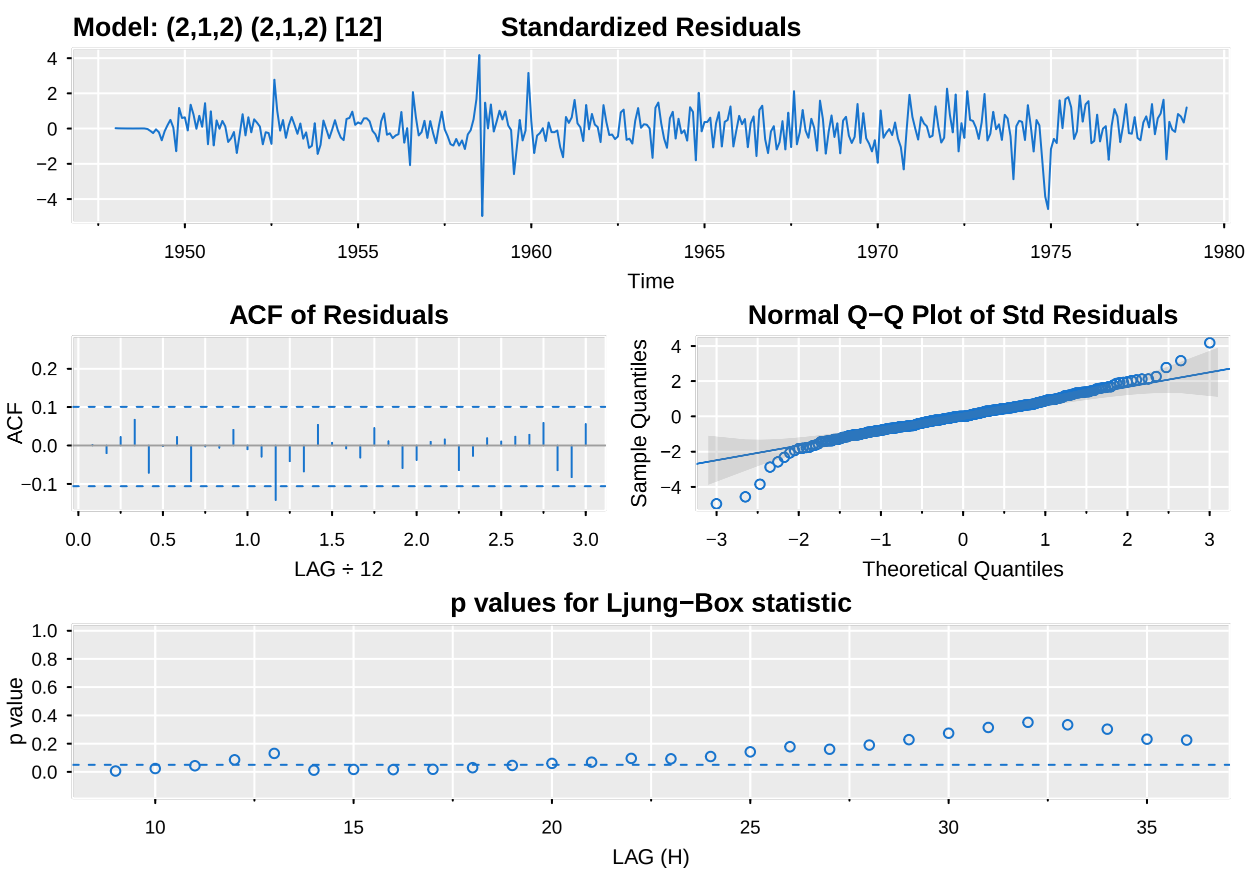
In most of the simulations, the ACF is clearly different to the original.

```r
fit.212 <- sarima(prodn, 2, 1, 2, 2, 1, 2, 12, gg = TRUE, col = 4)
```

```
## initial  value 0.480262
## iter   2 value 0.320770
## iter   3 value 0.201735
## iter   4 value 0.184543
## iter   5 value 0.174271
## iter   6 value 0.173883
## iter   7 value 0.172984
## iter   8 value 0.172916
## iter   9 value 0.172828
## iter  10 value 0.172722
## iter  11 value 0.172533
## iter  12 value 0.172392
## iter  13 value 0.172303
## iter  14 value 0.172242
## iter  15 value 0.172138
## iter  16 value 0.171959
## iter  17 value 0.171758
## iter  18 value 0.171597
## iter  19 value 0.171557
## iter  20 value 0.171491
## iter  21 value 0.171293
```

```
## iter  22 value 0.171073
## iter  23 value 0.170838
## iter  24 value 0.170713
## iter  25 value 0.170647
## iter  26 value 0.170613
## iter  27 value 0.170547
## iter  28 value 0.170364
## iter  29 value 0.170295
## iter  30 value 0.170278
## iter  31 value 0.170273
## iter  32 value 0.170272
## iter  33 value 0.170271
## iter  34 value 0.170270
## iter  35 value 0.170269
## iter  36 value 0.170269
## iter  37 value 0.170269
## iter  38 value 0.170269
## iter  39 value 0.170269
## iter  40 value 0.170269
## iter  41 value 0.170269
## iter  42 value 0.170269
## iter  43 value 0.170269
## iter  44 value 0.170269
## iter  45 value 0.170269
## iter  46 value 0.170269
## iter  46 value 0.170269
## iter  46 value 0.170269
## final  value 0.170269
## converged
## initial  value 0.163790
## iter   2 value 0.163559
## iter   3 value 0.163436
## iter   4 value 0.163329
## iter   5 value 0.163260
## iter   6 value 0.163041
## iter   7 value 0.162797
## iter   8 value 0.162053
## iter   9 value 0.160725
## iter  10 value 0.159828
## iter  11 value 0.159063
## iter  12 value 0.157449
## iter  13 value 0.156532
## iter  14 value 0.156065
## iter  15 value 0.155528
## iter  16 value 0.155173
## iter  17 value 0.153590
## iter  18 value 0.153470
## iter  19 value 0.153460
## iter  20 value 0.153434
## iter  21 value 0.153426
## iter  22 value 0.153422
## iter  23 value 0.153418
## iter  24 value 0.153412
## iter  25 value 0.153364
```

```
## iter  26 value 0.153353
## iter  27 value 0.153348
## iter  28 value 0.153347
## iter  29 value 0.153340
## iter  30 value 0.153335
## iter  31 value 0.153333
## iter  32 value 0.153333
## iter  33 value 0.153333
## iter  33 value 0.153333
## iter  33 value 0.153333
## final  value 0.153333
## converged
```

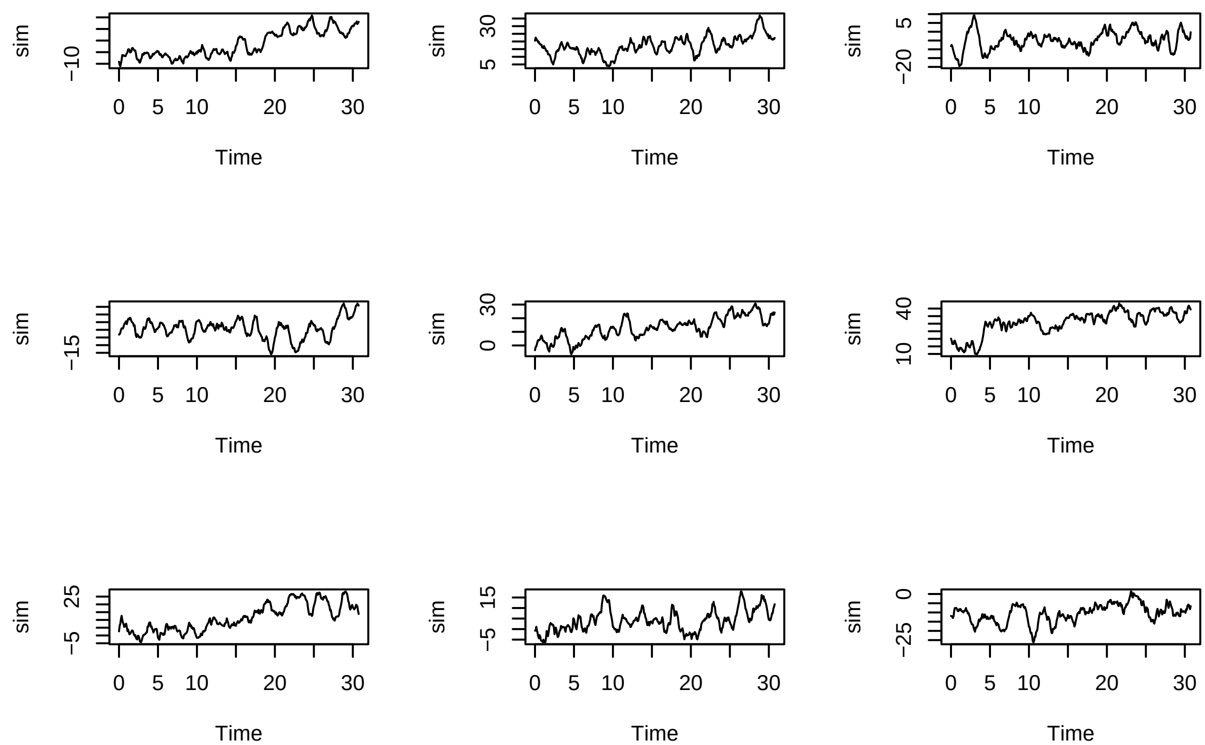



```
set.seed(2)
par(mfrow = c(3, 3))
fit.212
```

```
## $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 = list(trace =
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1     ar2     ma1      ma2     sar1     sar2     sma1    sma2
##      -0.2786  0.3396  0.5851  -0.0522  0.3822  -0.2970  -1.1301  0.4891
```

```
## s.e.   0.2202  0.1644  0.2331   0.1752  0.1604   0.0748   0.1631  0.1146
##
## sigma^2 estimated as 1.317:  log likelihood = -564.45,  aic = 1146.89
##
## $degrees_of_freedom
## [1] 351
##
## $ttable
##      Estimate     SE t.value p.value
## ar1   -0.2786 0.2202 -1.2649  0.2067
## ar2    0.3396 0.1644  2.0659  0.0396
## ma1    0.5851 0.2331  2.5103  0.0125
## ma2   -0.0522 0.1752 -0.2978  0.7660
## sar1   0.3822 0.1604  2.3823  0.0177
## sar2  -0.2970 0.0748 -3.9698  0.0001
## sma1  -1.1301 0.1631 -6.9299  0.0000
## sma2   0.4891 0.1146  4.2676  0.0000
##
## $AIC
## [1] 3.194682
##
## $AICc
## [1] 3.195828
##
## $BIC
## [1] 3.292036
```

```
replicate(9,
      plot(sarima.sim(ar = c(-0.3, 0.3),
                      ma = c(0.6, 0),
                      sar = c(0.4, -0.3),
                      sma = c(-1.1, 0.5),
                      S = 12, d = 1, n = 370) , ylab = 'sim'))
```




```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
##
## [[5]]
## NULL
##
## [[6]]
## NULL
##
## [[7]]
## NULL
##
## [[8]]
## NULL
##
## [[9]]
## NULL
```

```r
tsdisplay(sarima.sim(ar = c(-0.3, 0.3),
                     ma = c(0.6, 0),
                     sar = c(0.4, -0.3),
                     sma = c(-1.1, 0.5),
                     S = 12, d = 1, n = 370) , ylab = 'sim')
```