SARIMA Models

Lesson 3.3

Introduction

So far, we have restricted our attention to non-seasonal data and non-seasonal ARIMA models. However, ARIMA models are also capable of modelling a wide range of seasonal data.

A seasonal ARIMA model is formed by including additional seasonal terms in the ARIMA models we have seen so far. It is written as follows:

$$ARIMA \quad \underbrace{(p, d, q)}_{\text{Non-seasonal part}} \quad \underbrace{(P, D, Q)_m}_{\text{Seasonal part}}$$

where $m =$ number of observations per year. We use uppercase notation for the seasonal parts of the model, and lowercase notation for the non-seasonal parts of the model.

The seasonal part of the model consists of terms that are similar to the non-seasonal components of the model, but involve backshifts of the seasonal period. For example, an $ARIMA(1,1,1)(1,1,1)_4$ model (without a constant) is for quarterly data ($m = 4$), and can be written as

$$(1 - \phi_1 B)(1 - \phi_1 B^4)(1 - B)(1 - B^4)y_t = (1 + \theta_1 B)(1 + \theta_1 B^4)\epsilon_t$$

ACF/PACF

The seasonal part of an AR or MA model will be seen in the seasonal lags of the PACF and ACF. For example, an $ARIMA(0,0,0)(0,0,1)_{12}$ model will show:

- a spike at lag 12 in the ACF but no other significant spikes;
- exponential decay in the seasonal lags of the PACF (i.e., at lags 12, 24, 36, ...).

Similarly, an $ARIMA(0,0,0)(1,0,0)_{12}$ model will show:

- exponential decay in the seasonal lags of the ACF;
- a single significant spike at lag 12 in the PACF.

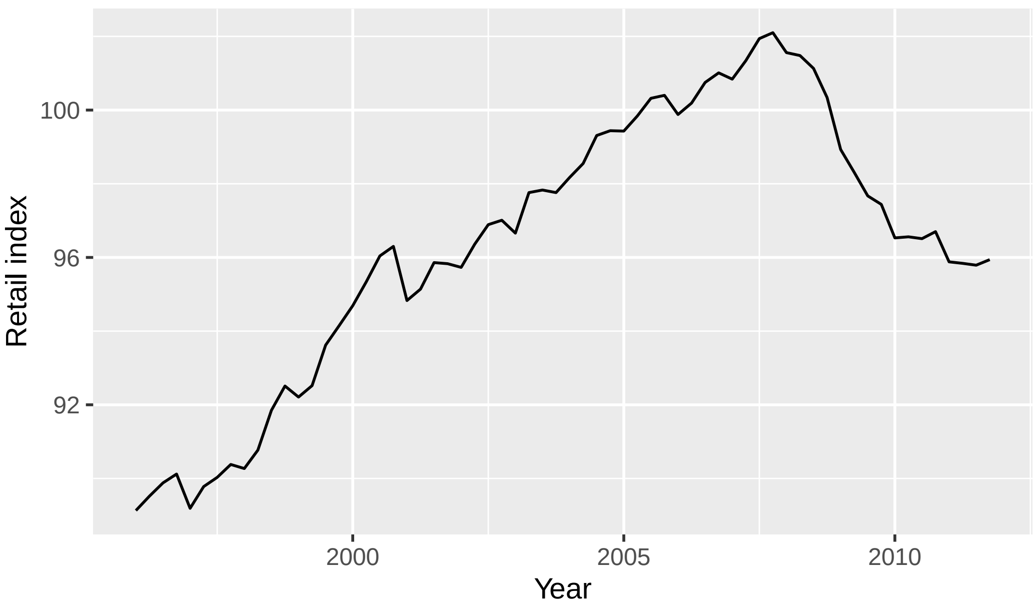
In considering the appropriate seasonal orders for a seasonal ARIMA model, restrict attention to the seasonal lags.

The modelling procedure is almost the same as for non-seasonal data, except that we need to select seasonal AR and MA terms as well as the non-seasonal components of the model. The process is best illustrated via examples.

Example: European quarterly retail trade

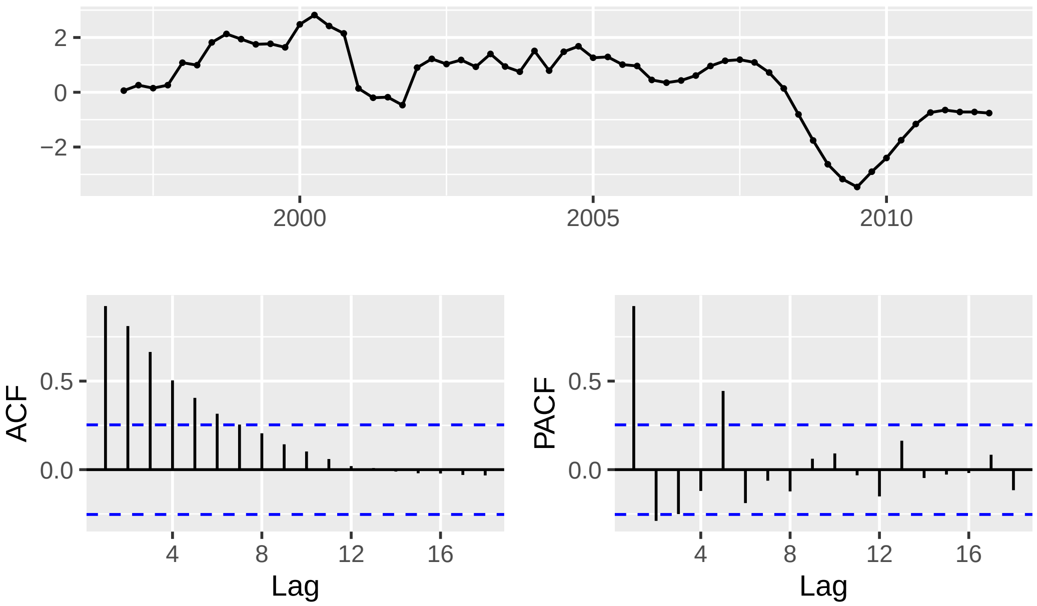
We will describe the seasonal ARIMA modelling procedure using quarterly European retail trade data from 1996 to 2011.

```r
library(tidyverse)
library(fpp2)
autoplot(euretail) + ylab("Retail index") + xlab("Year")
```



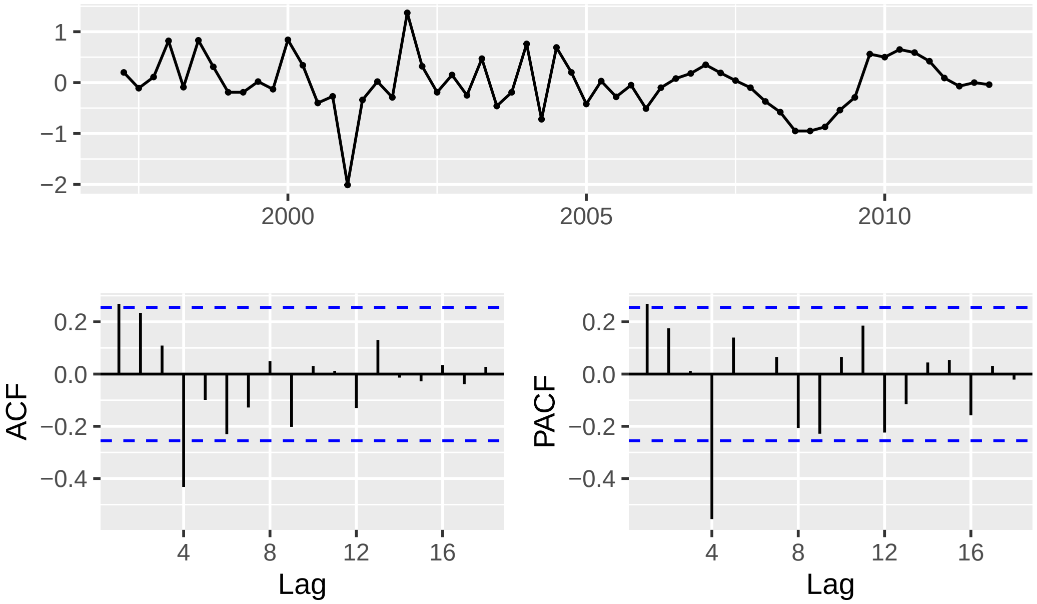
The data are clearly non-stationary, with some seasonality, so we will first take a seasonal difference. The seasonally differenced data are shown in the figure below.

```
euretail %>% diff(lag=4) %>% ggtsdisplay()
```



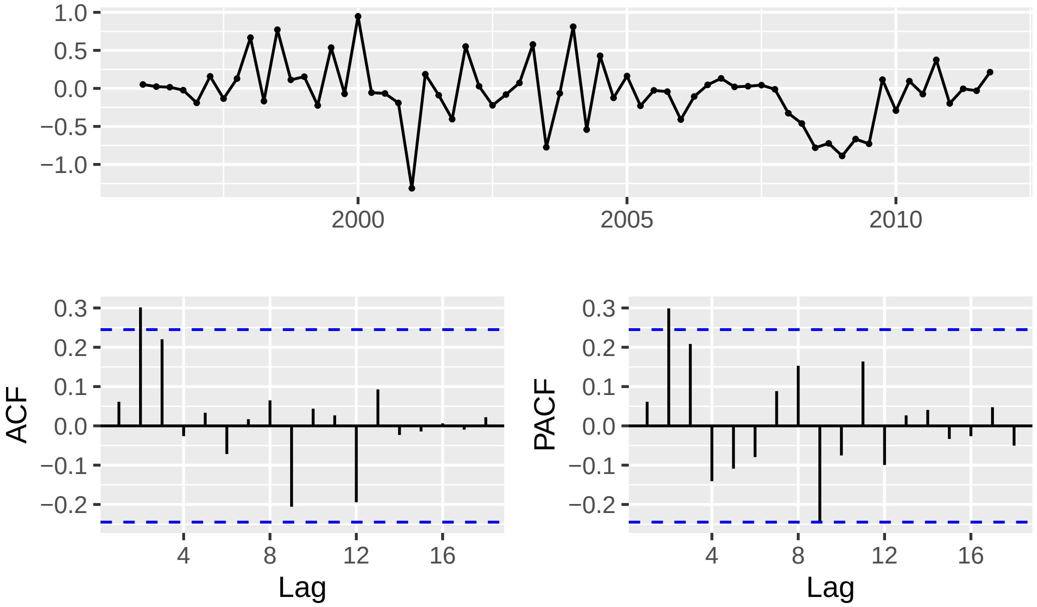
The plot above appear to be non-stationary, so we take an additional first difference, the plot is shown below..

```
euretail %>% diff(lag=4) %>% diff() %>% ggtsdisplay()
```



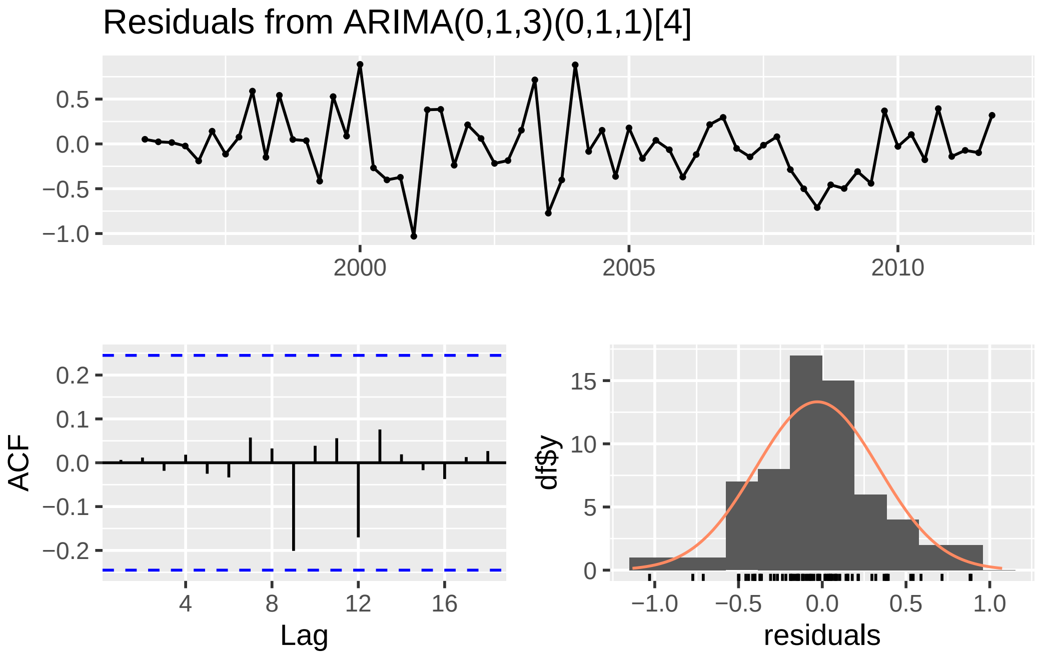
Looking at the ACF and PACF plots above, the significant spike at lag 1 in the ACF suggests a non-seasonal MA(1) component, and the significant spike at lag 4 in the ACF suggests a seasonal MA(1) component. Consequently, we begin with an $ARIMA(0,1,1)(0,1,1)_4$ model, indicating a first and seasonal difference, and non-seasonal and seasonal MA(1) components. The residuals for the fitted model are shown in Figure 8.20. (By analogous logic applied to the PACF, we could also have started with an $ARIMA(1,1,0)(1,1,0)_4$ model.)

```r
euretail %>%
  Arima(order=c(0,1,1), seasonal=c(0,1,1)) %>%
  residuals() %>% ggtsdisplay()
```



Both the ACF and PACF show significant spikes at lag 2, and almost significant spikes at lag 3, indicating that some additional non-seasonal terms need to be included in the model. The AIC_c of the $ARIMA(0,1,2)(0,1,1)_4$ model is 74.36, while that for the $ARIMA(0,1,3)(0,1,1)_4$ model is 68.53. We tried other models with AR terms as well, but none that gave a smaller AIC_c value. Consequently, we choose the $ARIMA(0,1,3)(0,1,1)_4$ model. Its residuals are below. All the spikes are now within the significance limits, so the residuals appear to be white noise. The Ljung-Box test also shows that the residuals have no remaining autocorrelations.

```r
fit3 <- Arima(euretail, order=c(0,1,3), seasonal=c(0,1,1))
checkresiduals(fit3)
```



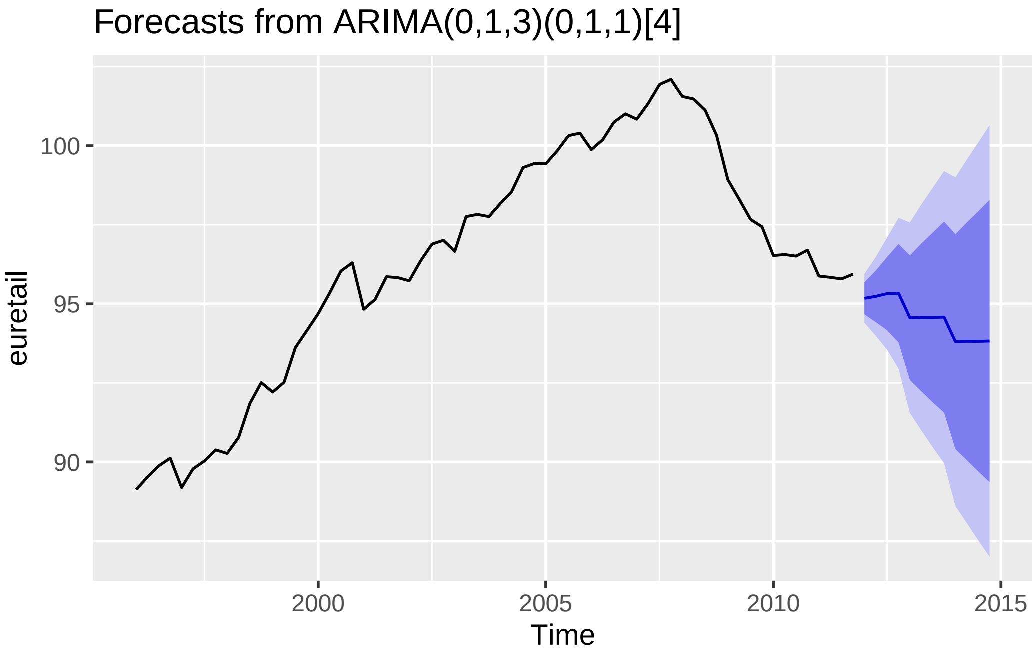
```
    Ljung-Box test

data:  Residuals from ARIMA(0,1,3)(0,1,1)[4]
Q* = 0.51128, df = 4, p-value = 0.9724

Model df: 4.   Total lags used: 8
```

Thus, we now have a seasonal ARIMA model that passes the required checks and is ready for forecasting. Forecasts from the model for the next three years are shown in Figure 8.22. The forecasts follow the recent trend in the data, because of the double differencing. The large and rapidly increasing prediction intervals show that the retail trade index could start increasing or decreasing at any time — while the point forecasts trend downwards, the prediction intervals allow for the data to trend upwards during the forecast period.

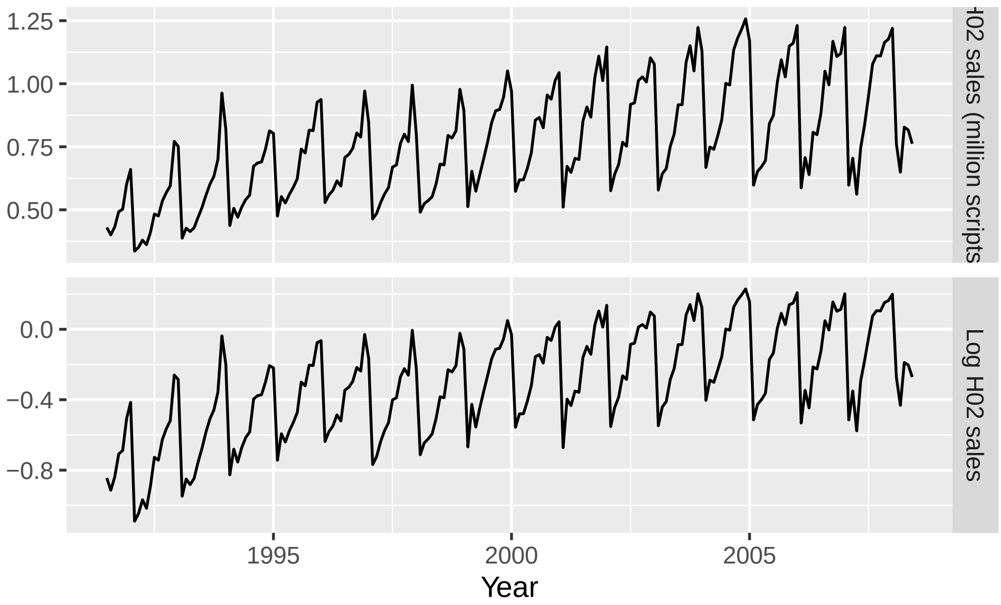
```
fit3 %>% forecast(h=12) %>% autoplot()
```



Second example

Our second example is more difficult. We will try to forecast monthly corticosteroid drug sales in Australia. These are known as H02 drugs under the Anatomical Therapeutic Chemical classification scheme.

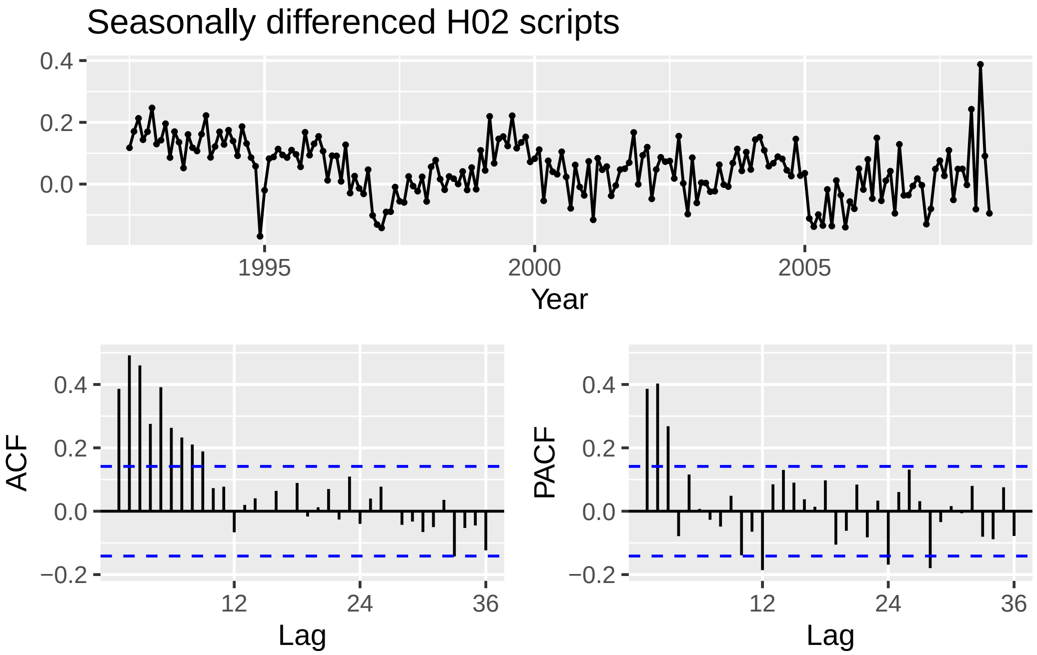
```
lh02 <- log(h02)
cbind("H02 sales (million scripts)" = h02,
      "Log H02 sales"=lh02) %>%
  autoplot(facets=TRUE) + xlab("Year") + ylab("")
```



As shown above, the plot of the original data from July 1991 to June 2008 shows a small increase in the variance with the level, so we take logarithms to stabilise the variance. The plot of the log transformed data is also shown in the same plot.

The data are strongly seasonal and obviously non-stationary, so seasonal differencing will be used. The seasonally differenced data are shown in below. It is not clear at this point whether we should do another difference or not. We decide not to, but the choice is not obvious.

```
lh02 %>% diff(lag=12) %>%
  ggtsdisplay(xlab="Year",
    main="Seasonally differenced H02 scripts")
```



In the plots of the seasonally differenced data, there are spikes in the PACF at lags 12 and 24, but nothing at seasonal lags in the ACF. This may be suggestive of a seasonal AR(2) term. In the non-seasonal lags, there are three significant spikes in the PACF, suggesting a possible AR(3) term. The pattern in the ACF is not indicative of any simple model.

Consequently, this initial analysis suggests that a possible model for these data is an $ARIMA(3,0,0)(2,1,0)_{12}$. We fit this model, along with some variations on it, and compute the $AICc$ values shown in the following table.

Model	**AICc**
$ARIMA(3,0,1)(0,1,2)_{12}$	-485.5
$ARIMA(3,0,1)(1,1,1)_{12}$	-484.2
$ARIMA(3,0,1)(0,1,1)_{12}$	-483.7
$ARIMA(3,0,1)(2,1,0)_{12}$	-476.3
$ARIMA(3,0,0)(2,1,0)_{12}$	-475.1
$ARIMA(3,0,2)(2,1,0)_{12}$	-474.9
$ARIMA(3,0,1)(1,1,0)_{12}$	-463.4

Of these models, the best is the $ARIMA(3,0,1)(0,1,2)_{12}$ model since it has the smallest $AICc$ value.

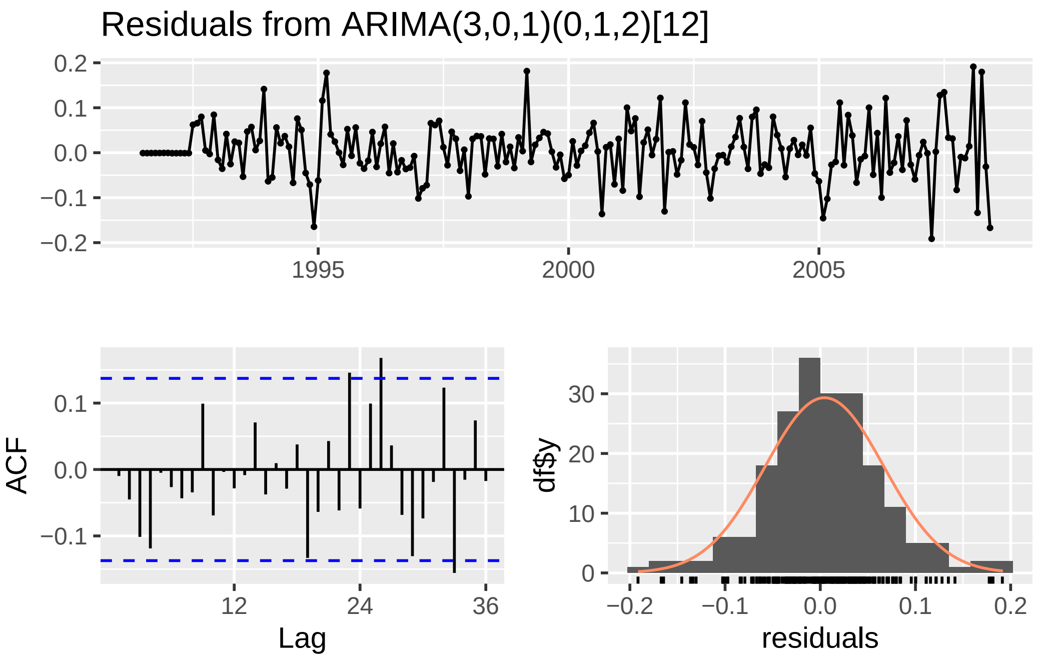
```
(fit <- Arima(h02, order=c(3,0,1), seasonal=c(0,1,2),
  lambda=0))
```

```
Series: h02
ARIMA(3,0,1)(0,1,2)[12]
Box Cox transformation: lambda= 0

Coefficients:
          ar1     ar2     ar3     ma1     sma1     sma2
      -0.1603  0.5481  0.5678  0.3827  -0.5222  -0.1768
s.e.   0.1636  0.0878  0.0942  0.1895   0.0861   0.0872

sigma^2 = 0.004278:  log likelihood = 250.04
AIC=-486.08   AICc=-485.48   BIC=-463.28
```

```
checkresiduals(fit, lag=36)
```



```
    Ljung-Box test

data:  Residuals from ARIMA(3,0,1)(0,1,2)[12]
```

```
Q* = 50.712, df = 30, p-value = 0.01045

Model df: 6.   Total lags used: 36
```

We can also try the automatic ARIMA algorithm. Running auto.arima() with all arguments left at their default values led to an $ARIMA(4,1,1)(0,1,2)_{12}$ model. However, the model still fails the Ljung-Box test for 36 lags. Sometimes it is just not possible to find a model that passes all of the tests.

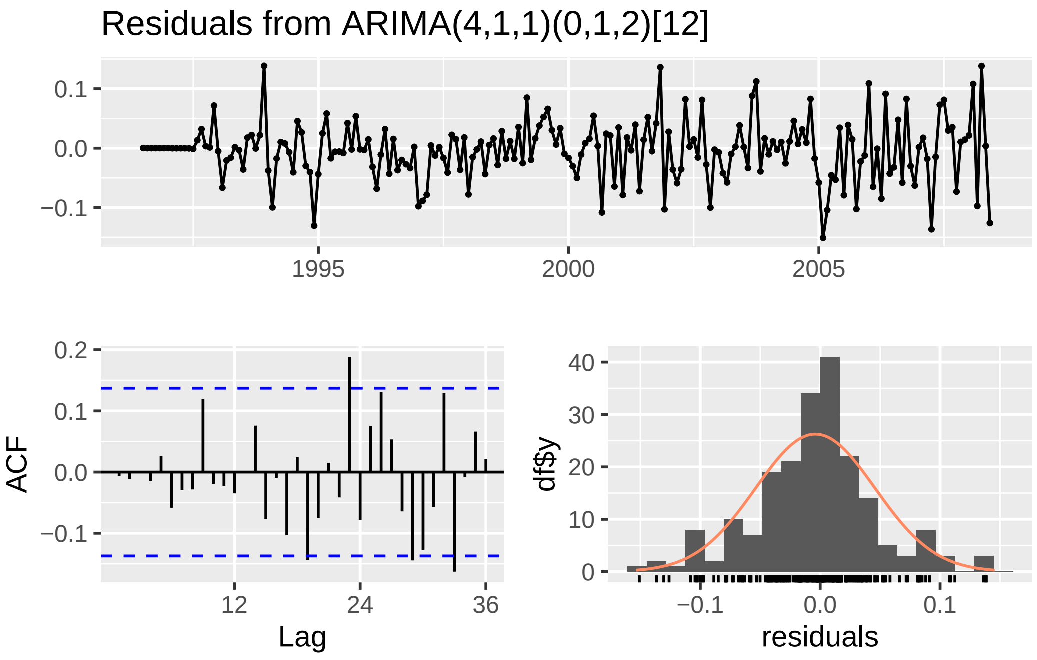
```
(fit <- auto.arima(h02))
```

```
Series: h02
ARIMA(4,1,1)(0,1,2)[12]

Coefficients:
         ar1     ar2     ar3      ar4      ma1     sma1     sma2
      0.0888  0.3386  0.2302  -0.2233  -0.9068  -0.4798  -0.1624
s.e.  0.1063  0.0976  0.0894   0.0850   0.0853   0.0913   0.0930

sigma^2 = 0.00276:  log likelihood = 291.7
AIC=-567.4   AICc=-566.6   BIC=-541.38
```

```
checkresiduals(fit, lag=36)
```



```
    Ljung-Box test

data:  Residuals from ARIMA(4,1,1)(0,1,2)[12]
Q* = 54.62, df = 29, p-value = 0.002743

Model df: 7.   Total lags used: 36
```

The best way forward is to try other models in the vicinity of the model generated by the *auto.arima*() function.