Time series forecasting with AR, MA, AR(I)MA models

Usue Mori

30 de octubre de 2023

In this tutorial we will learnt to perform forecasts on time series using ARIMA models.

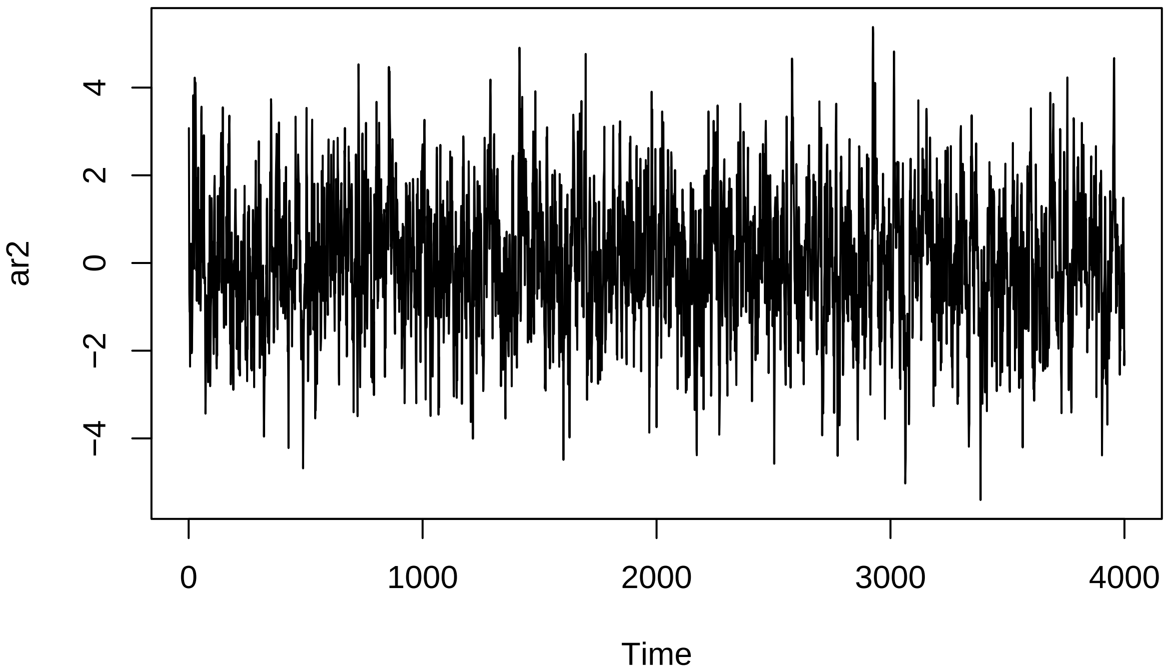
1. Simulation of ARIMA processes

In R, we can simulate or generate time series that follow different ARIMA processes using the function `arima.sim` from package `forecast`. In the following lines, you can find some examples of usage of this function:

- To simulate and plot an AR(2) process of parameters 0.9 and -0.2:

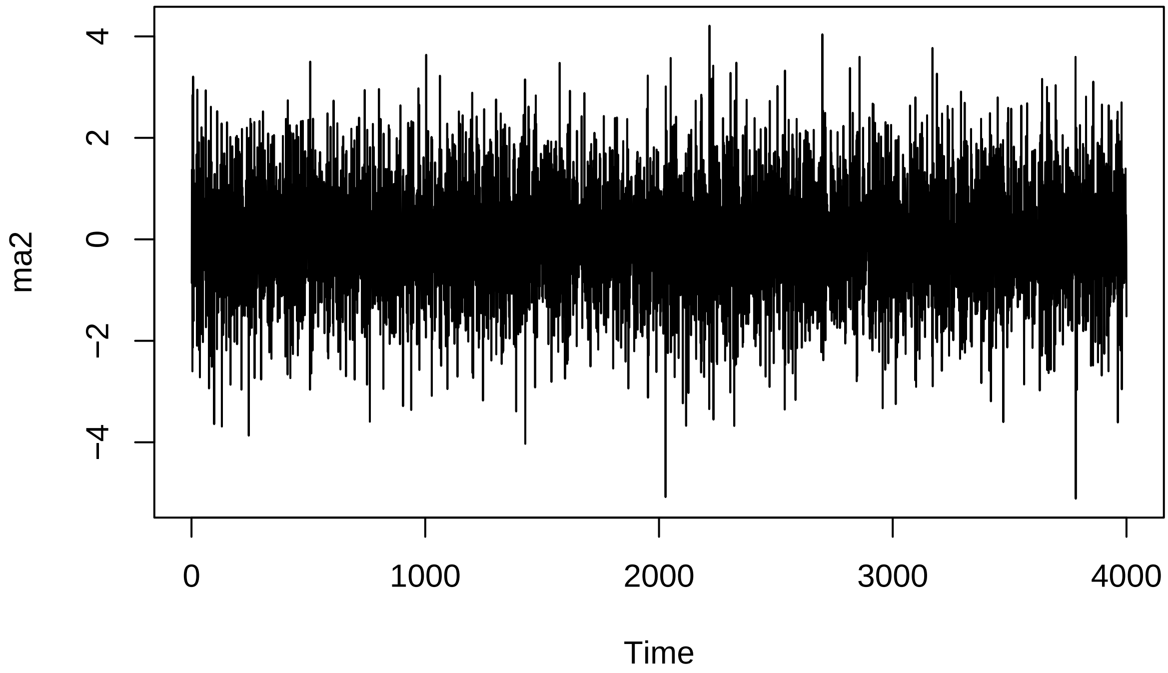
```
library(forecast)
set.seed(102)
```

```
ar2 <- arima.sim(model=list(ar=c(.9,-.2)), n=4000)
ts.plot(ar2)
```



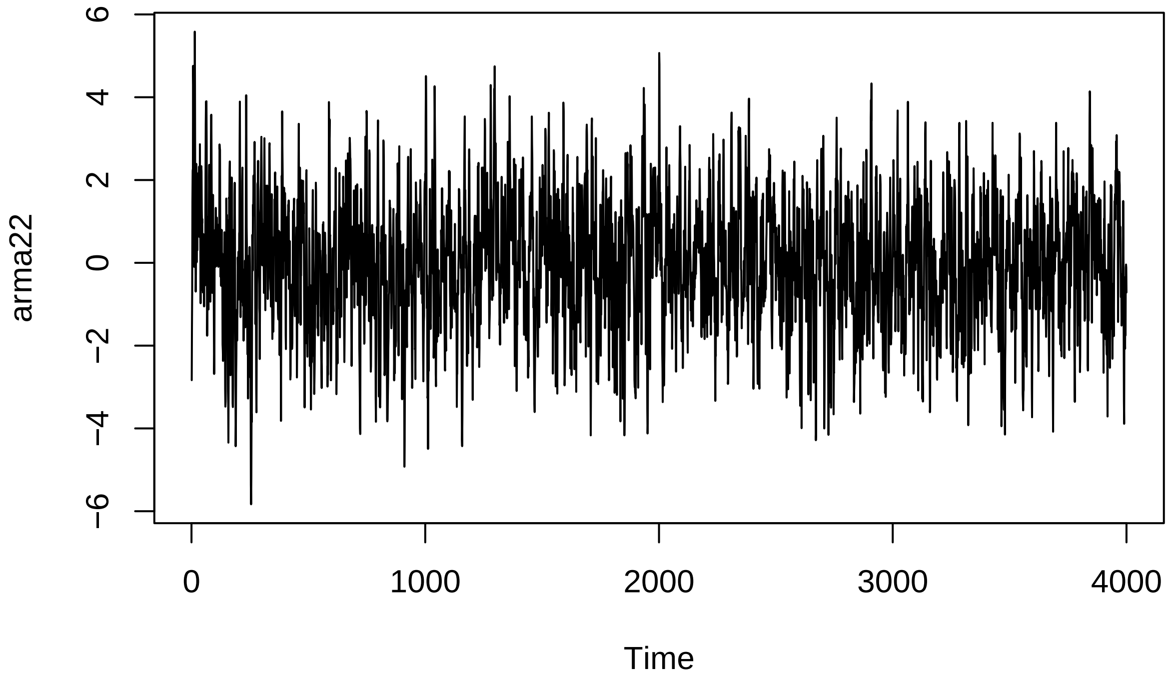
- To simulate and plot an MA(2) process of parameters -0.7 and 0.1:

```r
ma2 <- arima.sim(model=list(ma=c(-.7,.1)), n=4000)
ts.plot(ma2)
```



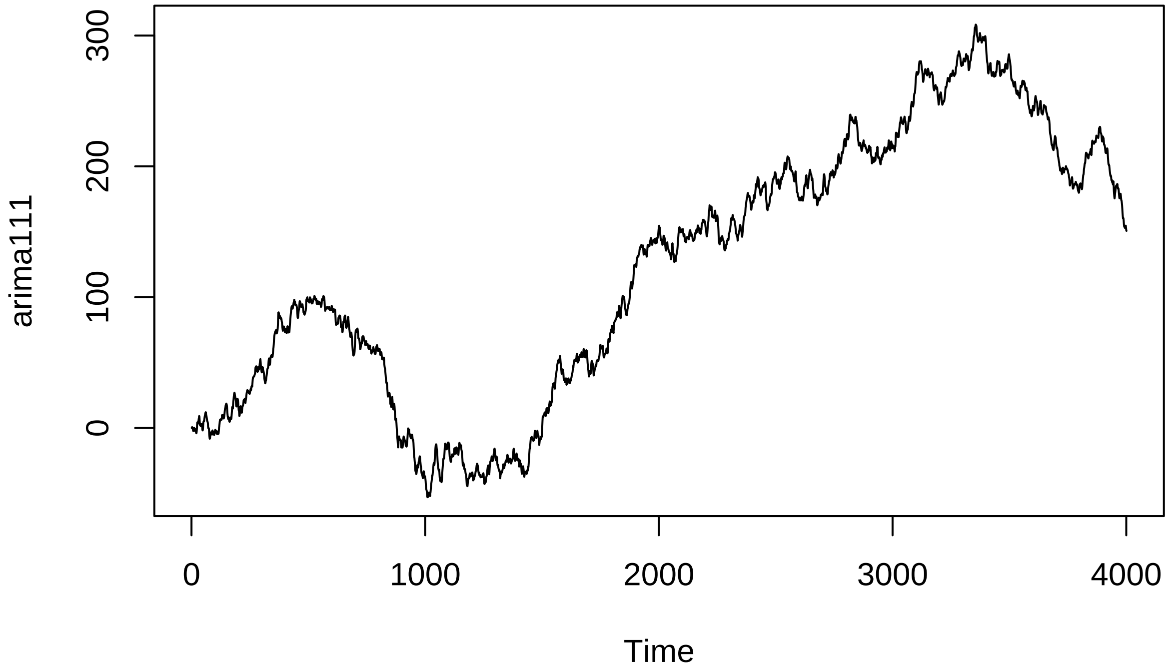
- To simulate an ARMA(2,2) process with AR parameters (0.4, 0.1) and MA parameters (0.5, 0.2):

```r
arma22 <- arima.sim(list(order=c(2, 0, 2), ar=c(0.4, 0.1), ma=c(0.5, 0.2)), n=4000)
ts.plot(arma22)
```



- Finally, let's simulate and ARIMA(1,1,1) with AR paramater 0.4 and MA parameter 0.5:

```r
arima111 <- arima.sim(list(order=c(1, 1, 1), ar=c(0.4), ma=c(0.5)), n=4000)
ts.plot(arima111)
```



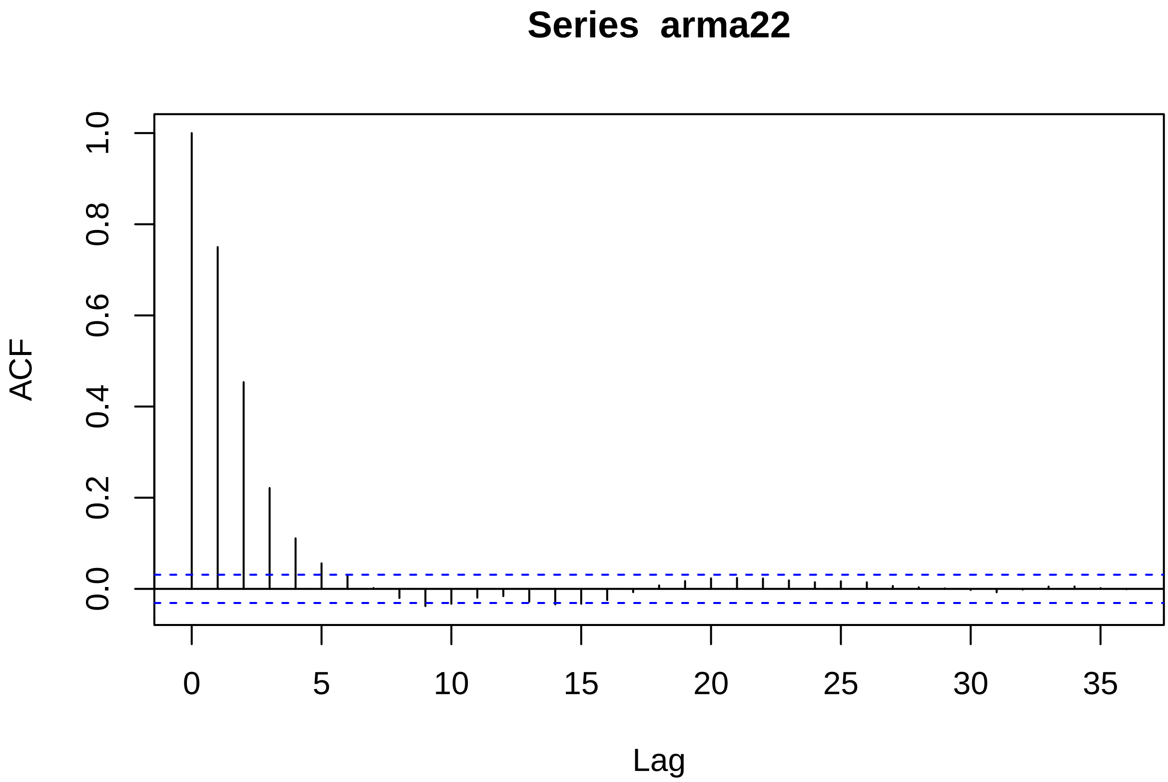
NOTE: If you write `arima.sim` in the R console, you can see the code of the `arima.sim` function and understand how these simulations are carried out by using the definitions of the different types of processes.

2. Fitting an AR(I)MA model to our series

In this section we will learn to fit ARMA models to time series. As simple examples, we will use the series we have simulated in the previous section.

Recall that in order to adjust a model of type AR, MA or ARMA, if necessary, we must first decompose the series in order to obtain a stationary time series. Additionally, once we obtain a stationary time series, if there is no temporal correlation between the observations (i.e. they are independent), it does not make sense to adjust a model of this type. We can analyze the temporal correlation between the observation by using the autocorrelation function, which analyzes the correlation between series X_t and its lagged versionsX_{t-i}. This is implemented in the `acf` function in R:

```r
acf(arma22)
```

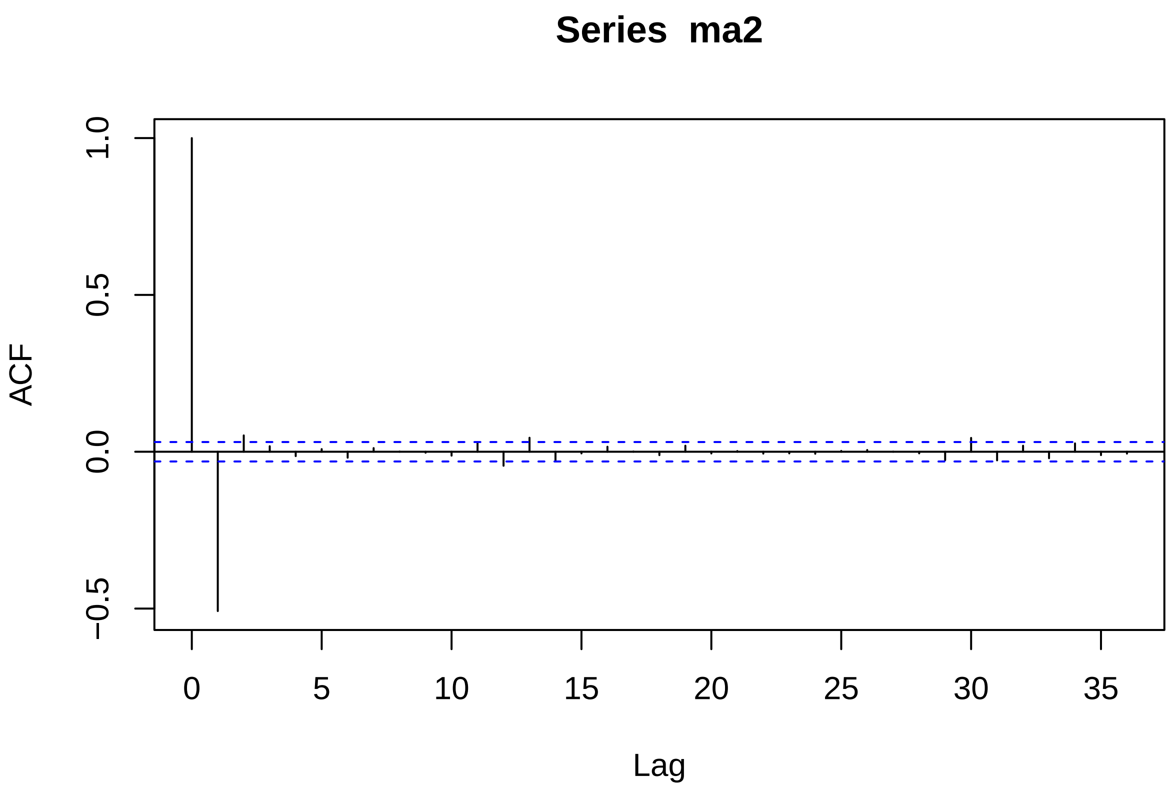


In this plot, each vertical lines represents the autocorrelation at a certain lag k. We can see that lags 1 trhough 5 have a correlation that exceeds the confidence intervals, so we can conclude that there is some significant correlation between the observations of the series. In this context, it makes sense to adjust a temporal model to the data.

2.1. Fitting a MA model to our series

We can fit a MA model to our data easily by using the `forecast` package. Additionally, in order to decide which p parameter can be suitable, we can analyze the correlogram of the autocorrelation (ACF) function. This function, as stated previously, measures the correlation of a time series X_t with its lagged versions X_{t-i}. It can be proved that the ACF of a MA(q) process is 0 for all lags which are larger that q. So, if a MA(q) is suitable to represent our series, the sample ACF values should be inside the confidence interval for all lags $k > q$, and it makes sense to choose q as the smallest lag that fulfills this condition:

```
acf(ma2)
```



In this case, the ACF correlogram would suggest a model of type MA(2) because after lag 2, all the autocorrelation values are inside the confidence interval. Note that this is not an exact method and will not always provide clear candidates for q, especially for real data. Once we have decided the order of the model, we can fit it to our data as follows:

```r
arima(ma2, order=c(0,0,2))
```

```
Call:
arima(x = ma2, order = c(0, 0, 2))

Coefficients:
          ma1     ma2  intercept
      -0.6996  0.0877    -0.0036
s.e.   0.0157  0.0161     0.0062

sigma^2 estimated as 1.018:  log likelihood = -5712.08,  aic = 11432.16
```

A more quantitative method to select the best parameters for our model is to use criteria based on goodness of fit, such as the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC). We can choose the parameter and fit the model automatically by using the `auto.arima` function:

```r
model.ma <- auto.arima(ma2, max.q=5, max.p=0, d=0, seasonal=FALSE)
model.ma
```

```
Series: ma2
ARIMA(0,0,2) with zero mean
```

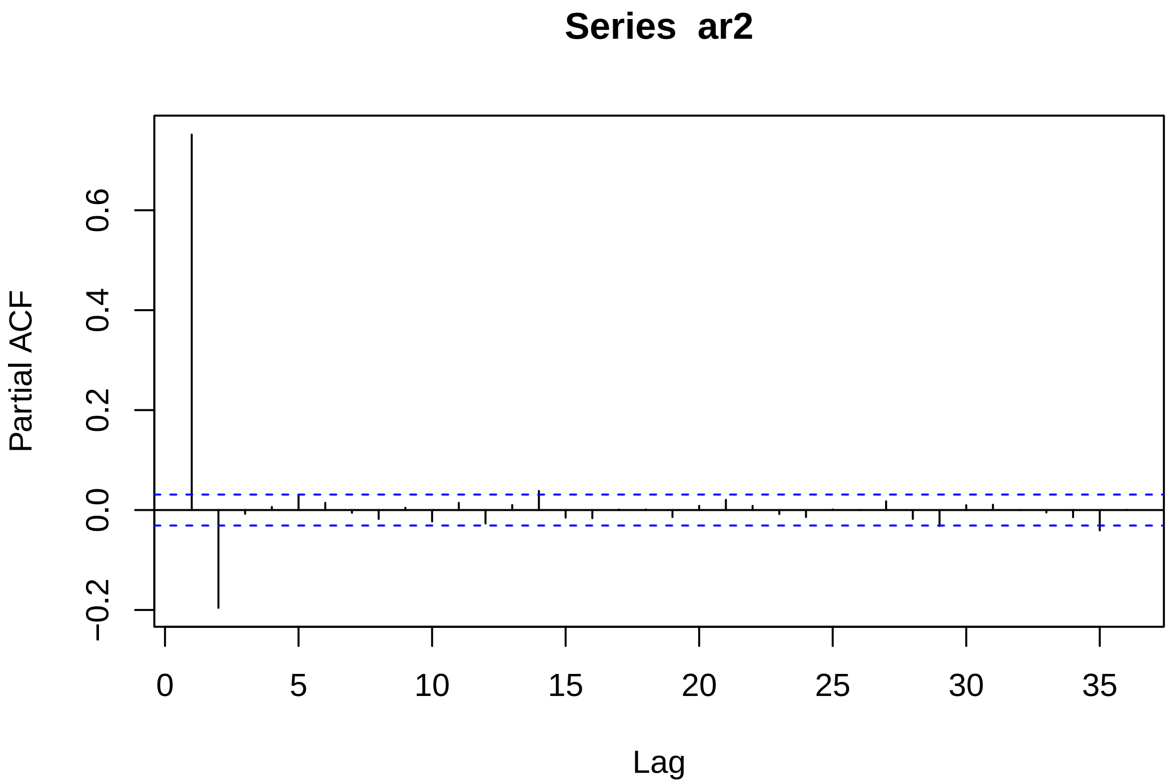
```
Coefficients:
          ma1     ma2
      -0.6996  0.0878
s.e.   0.0157  0.0161

sigma^2 = 1.019:  log likelihood = -5712.25
AIC=11430.5   AICc=11430.5   BIC=11449.38
```

We confirm that the MA(2) model is the model that best fits the data.

2.2. Fitting an AR model to our series

Similar to what we have done for MA models, we can also fit AR models to our time series by using the `forecast` package. In order to choose a suitable p parameter for the model, we can use the partial autocorrelation function (PACF). This function measures the correlation of a time series X_t with its lagged versions X_{t-i} after removing the effect of more recent lags $j < i$. The partial autocorrelation of an AR(p) process is zero at lags greater than p. As such, if an AR(p) process is suitable to model my series, the sample partial autocorrelation should not be significant for lags $k > p$. and it makes sense to choose p as the smallest lag that fulfills this condition:

```
pacf(ar2)
```



As expected, the correlogram suggests a model of type AR(2), which can be fitted as follows:

```
arima(ar2, order=c(2,0,0))
```

```
Call:
```

```
arima(x = ar2, order = c(2, 0, 0))

Coefficients:
         ar1      ar2  intercept
      0.9002  -0.1966     0.0241
s.e.  0.0155   0.0155     0.0526

sigma^2 estimated as 0.9745:  log likelihood = -5624.65,  aic = 11257.3
```

Now, we repeat the process but using goodness of fit:

```
model.ar <- auto.arima(ar2, max.q=0, max.p=5, d=0, seasonal=FALSE)
model.ar
```

```
Series: ar2
ARIMA(2,0,0) with zero mean

Coefficients:
         ar1      ar2
      0.9002  -0.1966
s.e.  0.0155   0.0155

sigma^2 = 0.9751:  log likelihood = -5624.76
AIC=11255.52   AICc=11255.53   BIC=11274.4
```

As can be seen, the same model has been chosen.

2.3. Fitting an AR(I)MA model to our series

Finally, we can also fit models of type ARMA or ARIMA (for non stationary time series). For models with both $p, q > 0$ the ACF and PACF plots are not usually helpful. As such we will search among some candidate models using goodness of fit. For example, for the `arma11` model:

```
model.arima <-auto.arima(arima111, max.q=5, max.p=5, seasonal=FALSE)
model.arima
```

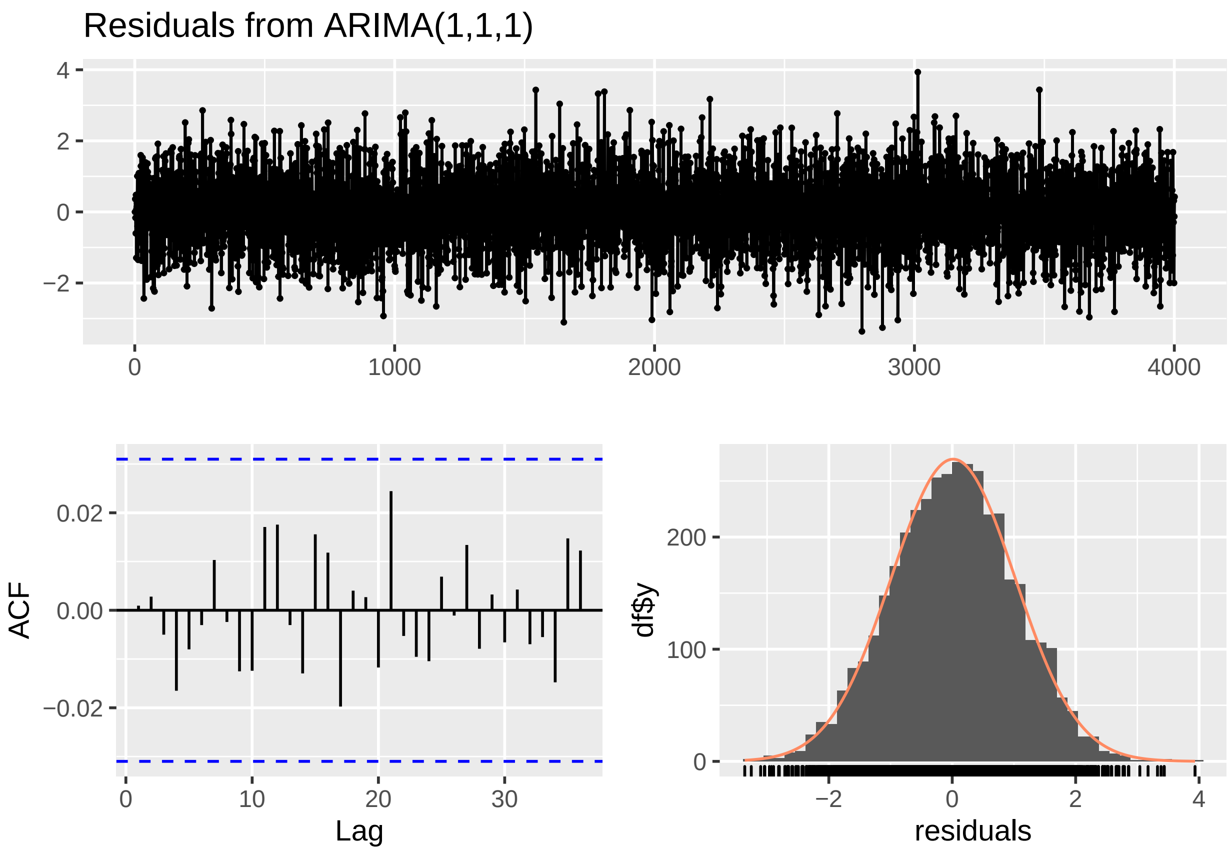
```
Series: arima111
ARIMA(1,1,1)

Coefficients:
         ar1     ma1
      0.4140  0.4884
s.e.  0.0192  0.0185

sigma^2 = 1.011:  log likelihood = -5697.46
AIC=11400.92   AICc=11400.92   BIC=11419.8
```

The method recommends an ARIMA(1,1,1) model.

3. Analyzing the residuals of the model

Before making any predictions with the fitted model, we must analyze the residual and see if they seem to follow a white noise process. The function `checkresiduals` analyzes the probability distribution and the ACF of the residuals and also performs the Ljung-Box test to check whether the residuals still contain any autocorrelation (H0: the data are independent). Additionally, we can also perform a Shapiro-Wilks test to check for normality.

```
checkresiduals(model.arima)
```




```
	Ljung-Box test

data:  Residuals from ARIMA(1,1,1)
Q* = 3.2159, df = 8, p-value = 0.9201

Model df: 2.   Total lags used: 10
```

```
shapiro.test(model.arima$residuals)
```

```
	Shapiro-Wilk normality test

data:  model.arima$residuals
W = 0.99958, p-value = 0.5667
```

As can be seen, all the autocorrelation values are inside the confidence intervals, which indicates that there is no evident autocorrelation. Additionally, the Ljung-Box test issues a high p-value which tells us that we can not discard independence. Finally, we can see in the histogram that the residuals seem to follow a normal distribution, and we reassure this with the high p-value of the Shapiro-Wilks test.

4. Making forecasts with a model of type AR(I)MA

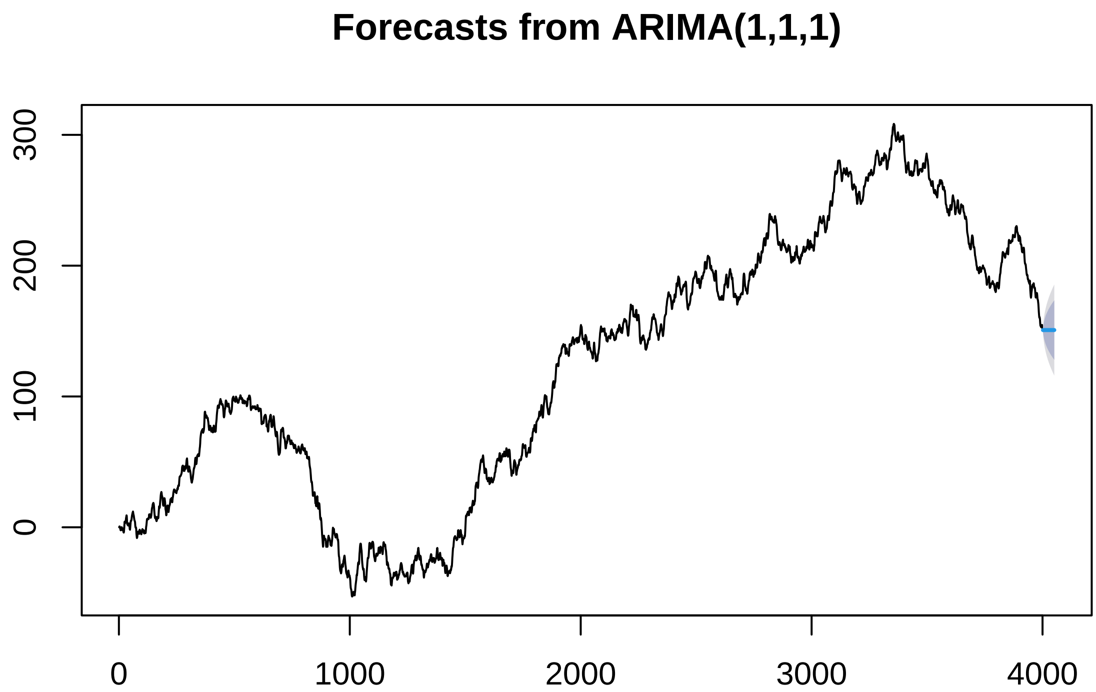
To perform predictions using the fitted model, we can use the `forecast` function of the package with the same name. We must indicate the number of data points in the future for which we want to make forecasts (`h` or forecasting horizon). For example, to predict 50 future data points for the `arima111` series using the fitted model we can proceed as follows:

```r
predictions <- forecast(model.arima, h = 50)
predictions
```

```
     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
4002       150.7368 149.4481 152.0255 148.7659 152.7077
4003       150.7544 147.9848 153.5241 146.5186 154.9902
4004       150.7617 146.7277 154.7957 144.5922 156.9312
4005       150.7647 145.6574 155.8720 142.9538 158.5757
4006       150.7660 144.7311 156.8008 141.5365 159.9955
4007       150.7665 143.9127 157.6203 140.2845 161.2485
4008       150.7667 143.1758 158.3577 139.1573 162.3761
4009       150.7668 142.5019 159.0317 138.1267 163.4069
4010       150.7668 141.8781 159.6556 137.1726 164.3610
4011       150.7668 141.2949 160.2388 136.2807 165.2530
4012       150.7669 140.7454 160.7883 135.4404 166.0933
4013       150.7669 140.2245 161.3092 134.6437 166.8900
4014       150.7669 139.7281 161.8056 133.8845 167.6492
4015       150.7669 139.2531 162.2806 133.1580 168.3757
4016       150.7669 138.7969 162.7368 132.4604 169.0733
4017       150.7669 138.3575 163.1762 131.7883 169.7454
4018       150.7669 137.9331 163.6006 131.1393 170.3944
4019       150.7669 137.5223 164.0114 130.5110 171.0227
4020       150.7669 137.1238 164.4099 129.9017 171.6320
4021       150.7669 136.7367 164.7970 129.3096 172.2241
4022       150.7669 136.3600 165.1737 128.7335 172.8002
4023       150.7669 135.9929 165.5408 128.1720 173.3617
4024       150.7669 135.6347 165.8991 127.6241 173.9096
4025       150.7669 135.2847 166.2490 127.0890 174.4447
4026       150.7669 134.9425 166.5912 126.5656 174.9681
4027       150.7669 134.6076 166.9261 126.0534 175.4803
4028       150.7669 134.2794 167.2543 125.5515 175.9822
4029       150.7669 133.9577 167.5760 125.0594 176.4743
4030       150.7669 133.6420 167.8917 124.5766 176.9571
4031       150.7669 133.3320 168.2017 124.1025 177.4312
4032       150.7669 133.0274 168.5063 123.6367 177.8970
4033       150.7669 132.7280 168.8057 123.1788 178.3549
4034       150.7669 132.4335 169.1002 122.7284 178.8054
4035       150.7669 132.1436 169.3901 122.2850 179.2487
4036       150.7669 131.8582 169.6756 121.8485 179.6852
4037       150.7669 131.5770 169.9567 121.4184 180.1153
```

```
4038       150.7669 131.2998 170.2339 120.9946 180.5391
4039       150.7669 131.0266 170.5071 120.5767 180.9570
4040       150.7669 130.7571 170.7766 120.1646 181.3692
4041       150.7669 130.4912 171.0425 119.7579 181.7759
4042       150.7669 130.2287 171.3050 119.3564 182.1773
4043       150.7669 129.9695 171.5642 118.9601 182.5737
4044       150.7669 129.7135 171.8202 118.5686 182.9651
4045       150.7669 129.4606 172.0731 118.1818 183.3519
4046       150.7669 129.2107 172.3230 117.7995 183.7342
4047       150.7669 128.9636 172.5701 117.4217 184.1120
4048       150.7669 128.7193 172.8144 117.0480 184.4857
4049       150.7669 128.4777 173.0560 116.6785 184.8552
4050       150.7669 128.2386 173.2951 116.3129 185.2208
4051       150.7669 128.0021 173.5316 115.9512 185.5825
```

```r
plot(predictions)
```



As can be seen, in addition to the point estimations we can also obtain the confidence intervals of the predictions.

5. Exercises

In the file named Data/ParoCCAA.Rdata we have the unenployment rates in the period 1977-2009 in the different autonomous communities. Take the column which corresponds to Catalonia and answer the following questions:

- Check if the series is stationary and if it is not, difference it until you obtain a stationary time series (done in previous lab).
- Draw the PACF correlogram to select a candidate order of an AR model for the stationary time series obtained in the previous section. Then, use the `auto.arima` function to select p automatically and fit a suitable AR model. What can you conclude?

- Draw the ACF correlogram to select a candidate order of an MA model for the stationary time series obtained in the previous section. Then, use the `auto.arima` function to select q automatically and fit a suitable MA model. What can you conclude?
- Use `auto.arima` to select the most suitable ARMA model. Compare the AIC_c values of the fitted AR, MA and ARMA models and select the most suitable.
- Use `auto.arima` to fit an ARIMA model to the initial time series (without differencing).
- Analyze the residuals of this last chosen model. What can you conclude?
- Calculate the predictions of the series for the following 5 trimesters and represent them graphically.

6. Extra work

- Use the first 70 % of the unenployment time series of Catalonia to fit an ARIMA model and the leftover 30 % to evaluate the 1-step predictions. Use the RMSE to evaluate the predictions. See `https://otexts.com/fpp2/forecasting-on-training-and-test-sets.html` on information about how to do this in different ways. How is this procedure different to what we have done previously with the `forecast` function?

7. References

- P. J. Brockwell and R. A. Davis. Introduction to Time Series and Forecasting. Springer Verlag, 1996.
- P. S.P. Cowpertwait and A. V. Metcalfe. Introductory Time Series with R (Use R). Springer, 2009.
- R. H. Shumway and D. S. Stoffer. Time Series Analysis and Its Applications With R Examples. Springer Verlag, 2006.