Final Individual Project

Lynx Dataset Analysis

Jaideep Kulkarni

Student ID: 110295747

Introduction

"Lynx" dataset is Annual Lynx Trappings from year 1821 to 1934. These were recorded in Canada. These readings were taken from Brockwell & Davis (1991). This dataset is taken into consideration to explain illustrative methods involving time series analysis. Many of the methods used are associated with auto-regressive and periodic models. The dataset was constructed with the help of Time Series and Forecasting Methods by Brockwell & Davis (1991), Second edition. Springer. Series G (page 557).

"Lynx" is a time series dataset which we use in R. It starts from year 1821 to year 1934 with 114 readings.

Analysis

We first take the lynx data in R using "data()" function.

If we investigate dataset by running the code "lynx". We can see the information of the dataset as well as the values in it.

If we look at the lynx dataset by plotting, we can see that the graph of the dataset is periodic, and seasonality is not consistent. As we take the logarithm of lynx dataset, we get the graph which is stationary. "log" is used in the dataset to stationarize it. As we look in the plot, we can see the difference.

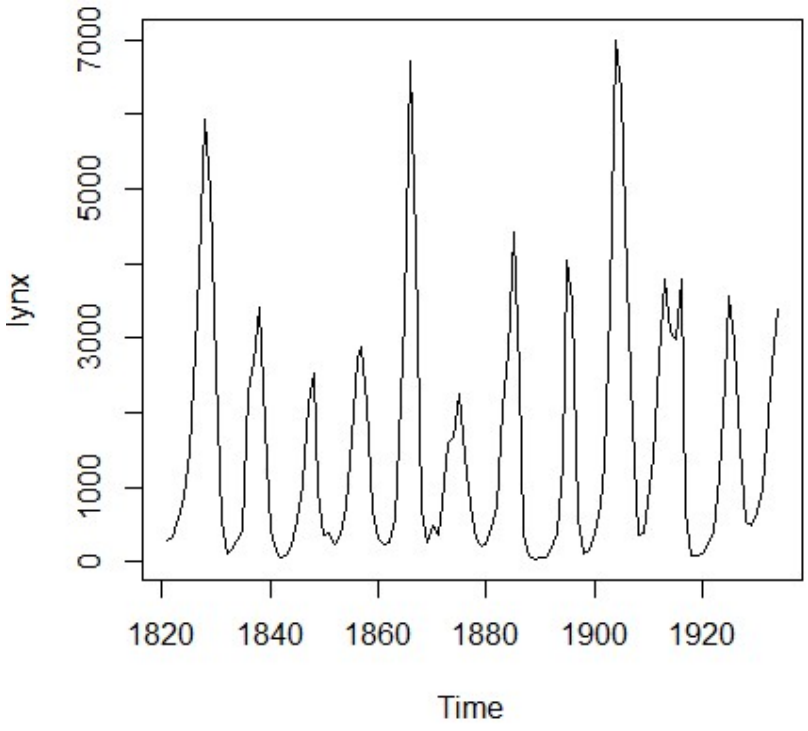


Fig. 1 Lynx dataset plot

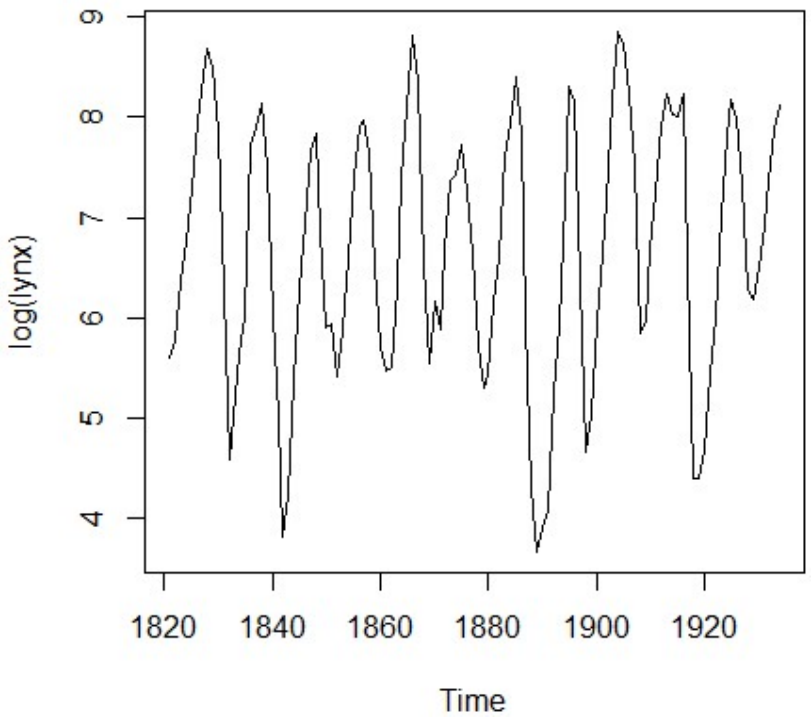


Fig 2. Log(lynx) dataset plot

Figure 1 represents the plotting of Lynx dataset whereas figure 2 represents the logarithm of lynx dataset.

Note: “log()” is a function used in R to find the logarithmic of any values. It can be used as log(x, base = 10) where “x” represents the numeric or complex vector and “base” represents the numeric or complex. With the help of “base”, we calculate the logarithm of “x”.

Now as we need to remove the seasonality of the dataset, we use the function “diff()”.

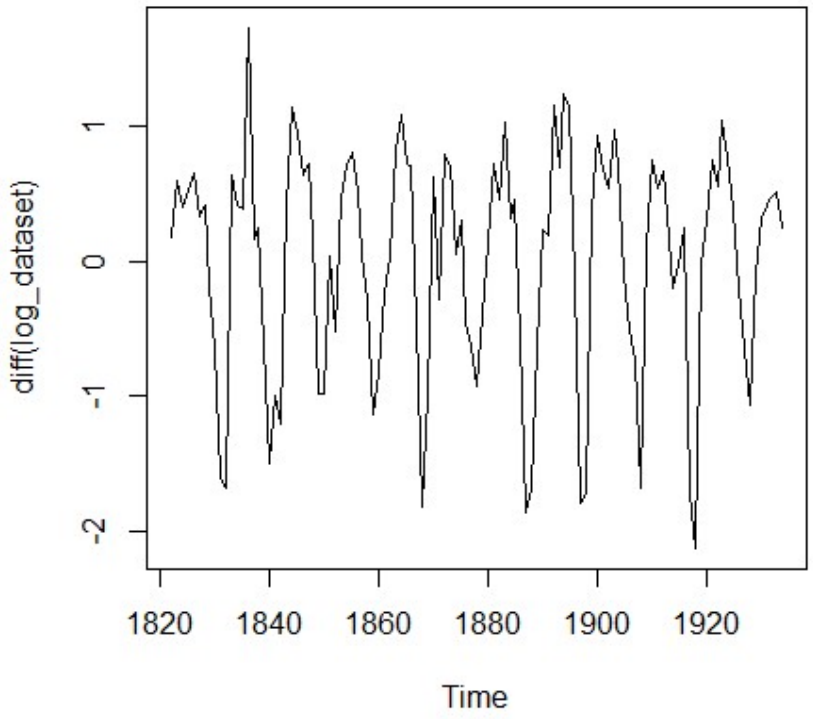


Fig 3. Plot after applying diff to the logarithm of the lynx data

Now we apply Acf to logarithm of the dataset. We get the graph such as,

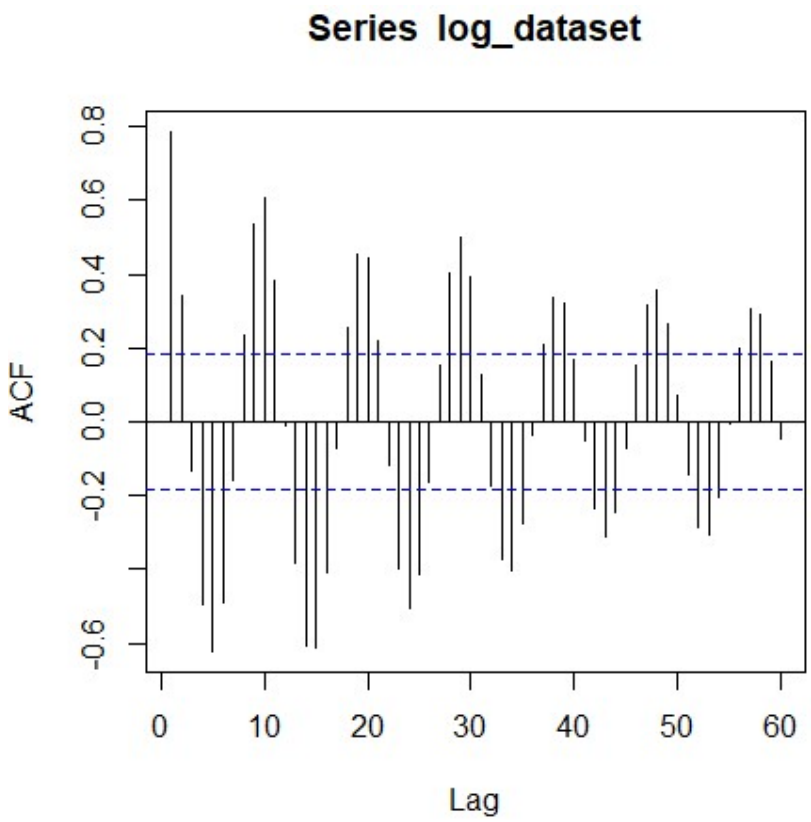


Fig 4. Acf of logarithm of lynx dataset

Acf function is used to find the correlations in the dataset. Autocorrelations or lagged correlations are used to find if the dataset is in relation with its past or an independent dataset. In the time series, the dataset is always in correlation with its past. As the time passes by, the values of the variable change over time. “Lag” in the x-axis is used in the function to display the plot clearly.

We can even find the Pacf of the lynx dataset. Pacf is used to compute a partial autocorrelation of the dataset. It measures linear correlation of the series and lagged version of itself.

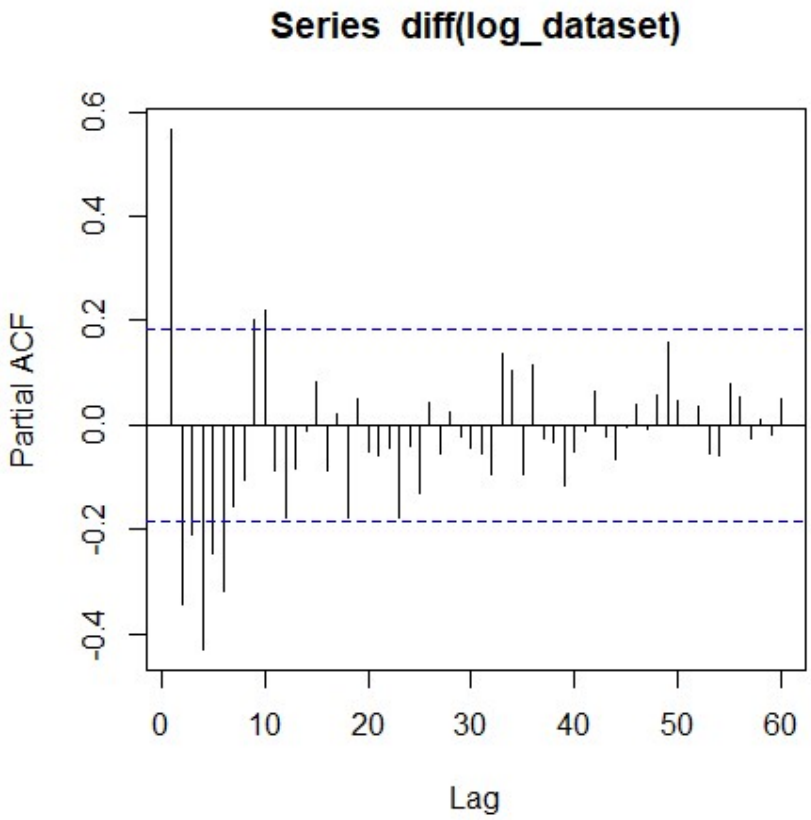


Fig 5. Pacf of diff of logarithm of lynx dataset

As Pacf plots represents the Partial correlation of the lynx dataset. The observations from the graph gives the results such as period is 10. The cycle stops after 10 and begins again from the beginning. Hence, S=10.

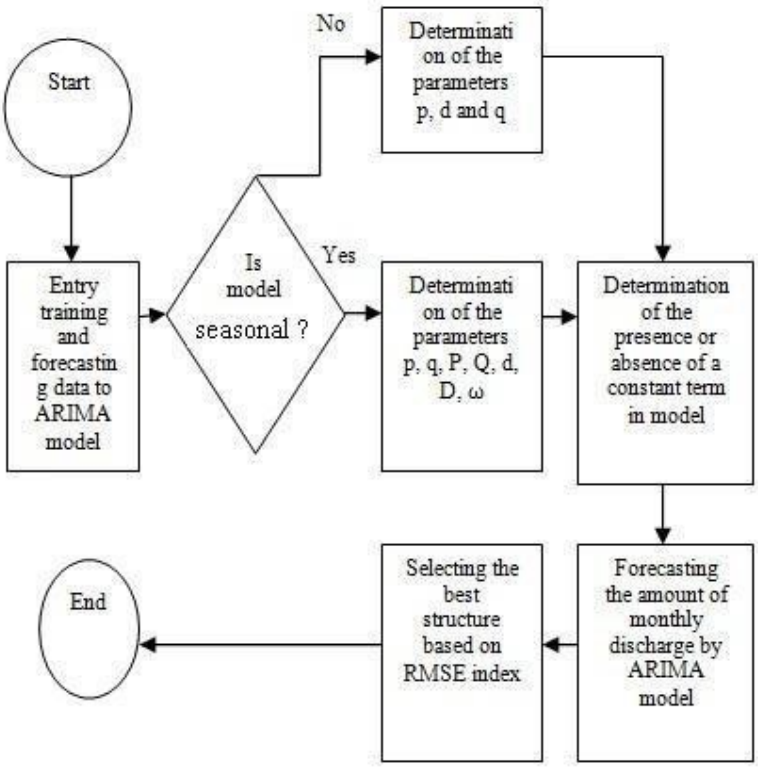


Fig 6. Workflow of ARIMA model

ARIMA is explained as Autoregressive Integrated Moving Average. This is used to better understand the dataset by using statistical analysis on the dataset. This uses one variable and compares it with regression analysis on other variables.

ARIMA model is used for seasonal and non-seasonal dataset. For seasonal dataset, we get the parameters p,q,d,P,Q,D,w. Here,

P – represents the number of lag observations in model

Q – order of moving average

D – Number of times observations are differenced

w – seasonal period

We calculate ACF and PACF for the different ARIMA models produced. When producing these models, we acquire different solutions. These solutions when compared with the AIC value we get; we can select the best model for the dataset.

The best model selected is ARIMA(2,0,4)(0,1,1)10. (After trail and error method by changing the values).

The coefficients which we get from the best ARIMA model is

ar1	ar2	ma1	ma2	ma3	ma4	ma1
1.3747	-0.7935	-0.1942	0.0539	0.4395	0.1081	-0.6619
0.1382	0.1091	0.1572	0.1335	0.1268	0.1203	0.1010

Now we predict the next 2 seasonal cycles i.e. 20 years from year 1934.

pred() function is used to predict the values based on linear object model. Prediction is done on the values from the dataset which are not present but can be visualised with the help of old entries. pred() is used from year 1934 to predict next 20 seasonal cycles till 1954. We can even plot the next 2 seasonal cycles with original series and determine the prediction plot.

```
Time Series:
Start = 1935
End = 1954
Frequency = 1
 [1] 3667.5212 2932.1352 1035.4448  252.7202  237.5621  352.7275  564.1793  937.0795 1828.3966
[10] 2749.6017 3439.9646 2995.5460 1074.7280  259.3341  238.9843  348.4178  552.1040  918.5351
[19] 1809.6423 2754.2421
```

These are the final predicted values we get by using predict function.

The equation we form from the ARIMA model is

$(1-1.3747B+0.7935B^2)\ (1-B)\ (1-B^{10})\ \log x_t = (1-0.6619B)\ (1-0.1942B-0.0539B^2-0.4395B^3-0.1081B^4)\ w_t$

We arrive this equation with the help of simple seasonal and non-seasonal difference

$$(1-\phi_1 B)\ (1-\Phi_1 B^4)\ (1-B)\ (1-B^4)y_t = (1+\theta_1 B)\ (1+\Theta_1 B^4)e_t.$$

(Non-seasonal AR(1)) (Seasonal AR(1)) (Non-seasonal difference) (Seasonal difference) (Non-seasonal MA(1)) (Seasonal MA(1))

Plotting both the original dataset with the predicted values, the graph we get is,

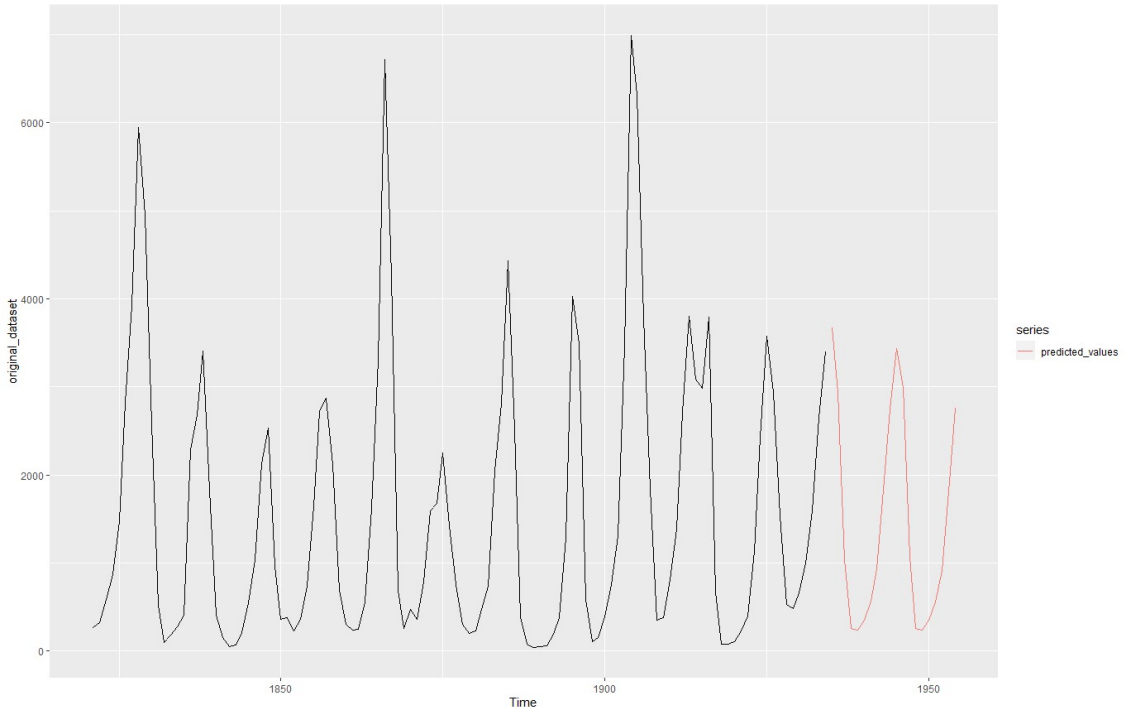


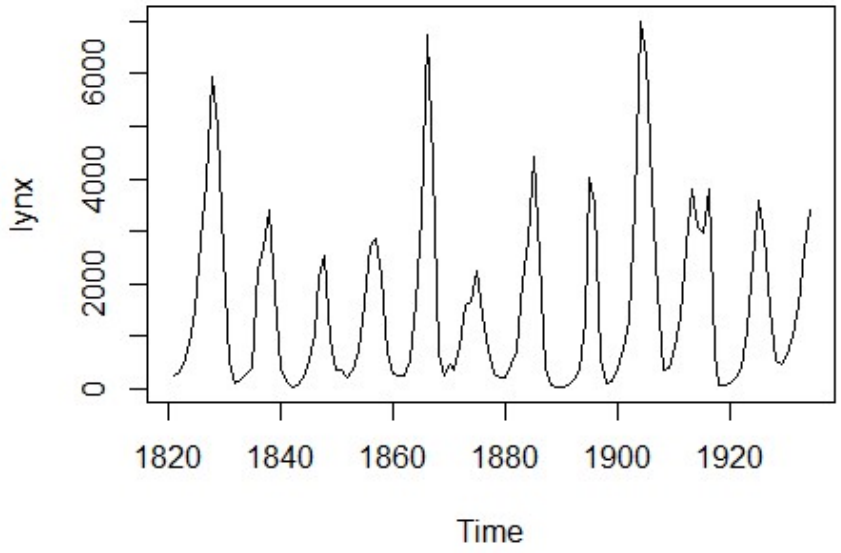
Fig 7. Original and Predicted Values Graph

Conclusion

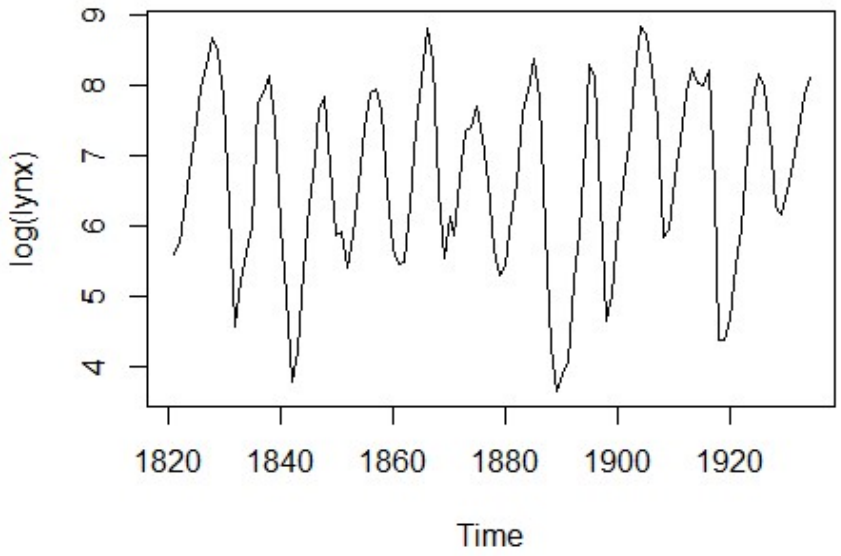
We can conclude from the following figures and the results that the predicted values and joining them with the original values using predicted values and the model we developed through ARIMA model of order (2,0,4) and seasonal (0,1,1), the time series data “Lynx” is periodic which can be stationarized and seasonal differencing can be removed. Trail and error method is best way to derive the model which is suitable for the time series dataset.

Appendix

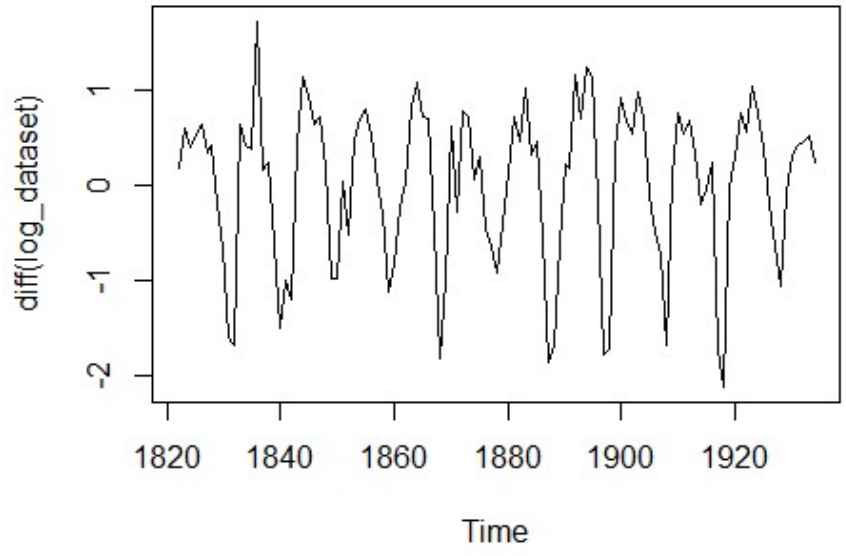
```
library(forecast)

library(ggplot2)

data("lynx")
original_dataset<- lynx
plot(lynx)
```



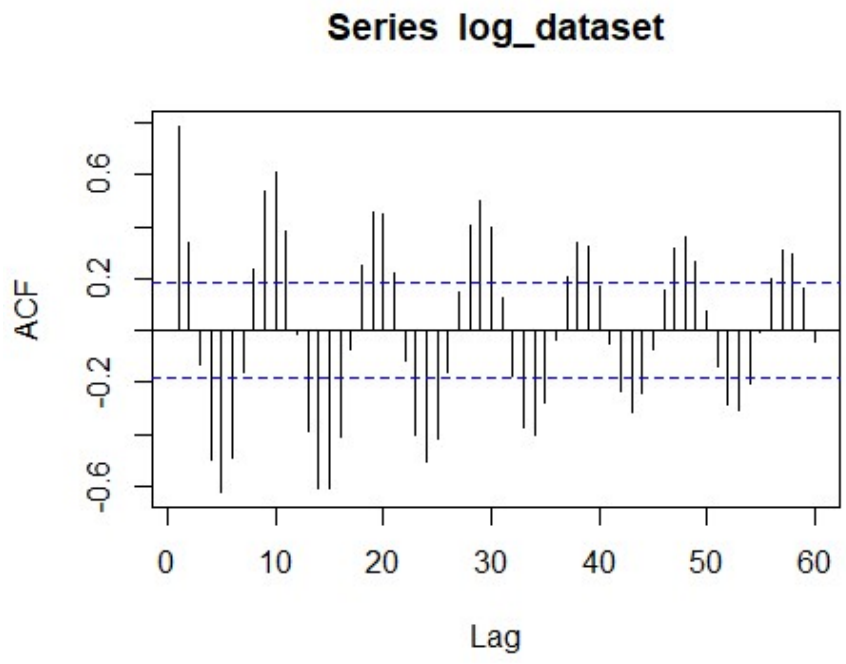

```
log_dataset<- log(lynx)
plot(log(lynx))
```



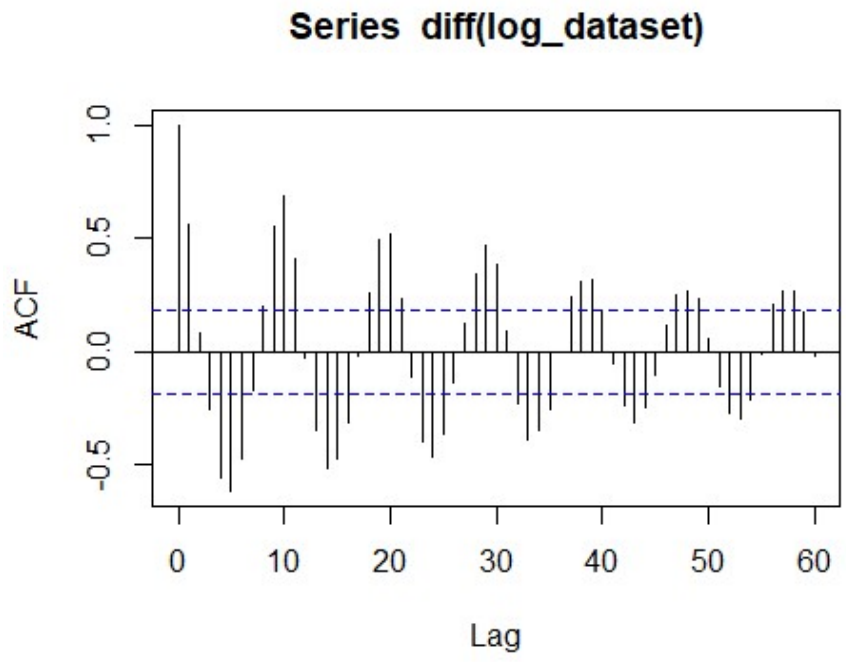

```
plot(diff(log_dataset))
```



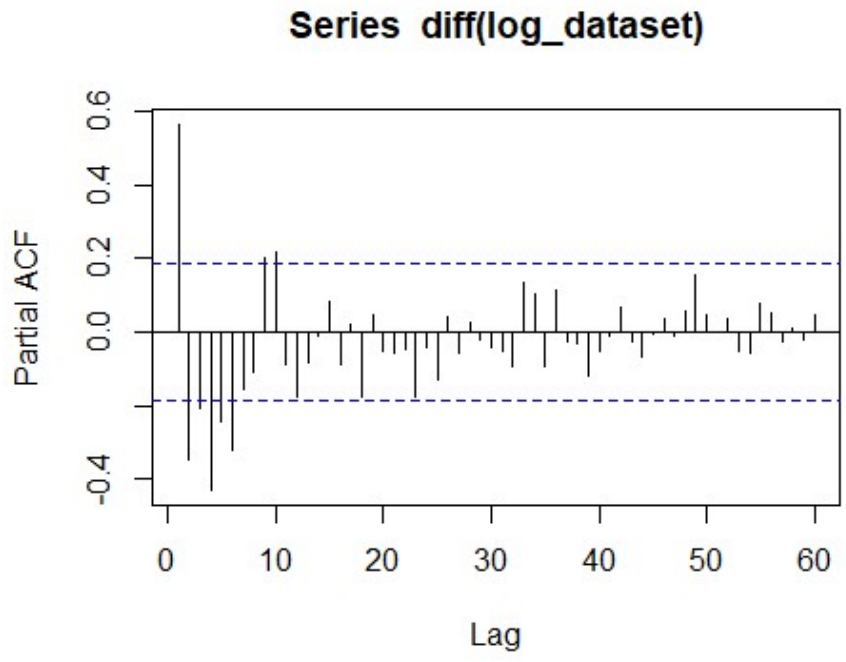

```
Acf(log_dataset, lag.max = 60)
```



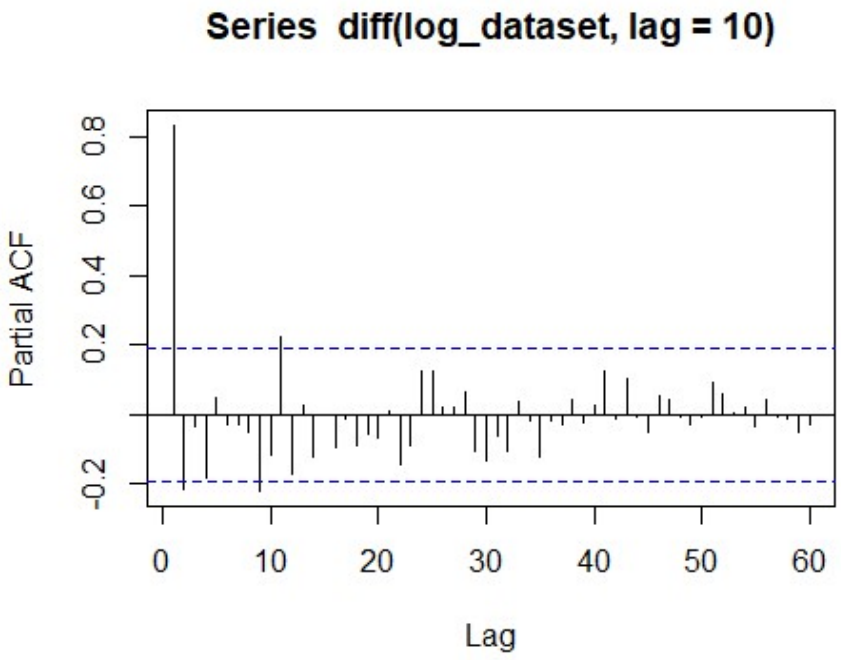

```
acf(diff(log_dataset),lag.max=60)
```



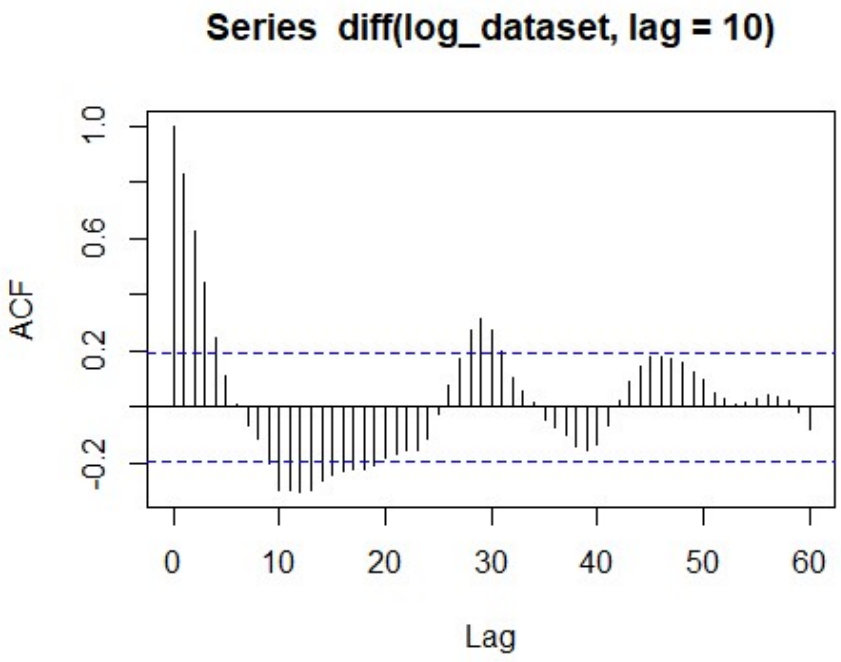

```
pacf(diff(log_dataset),lag.max=60)
```




```
pacf(diff(log_dataset,lag=10),lag.max=60)
```




```
acf(diff(log_dataset,lag=10),lag.max=60)
```

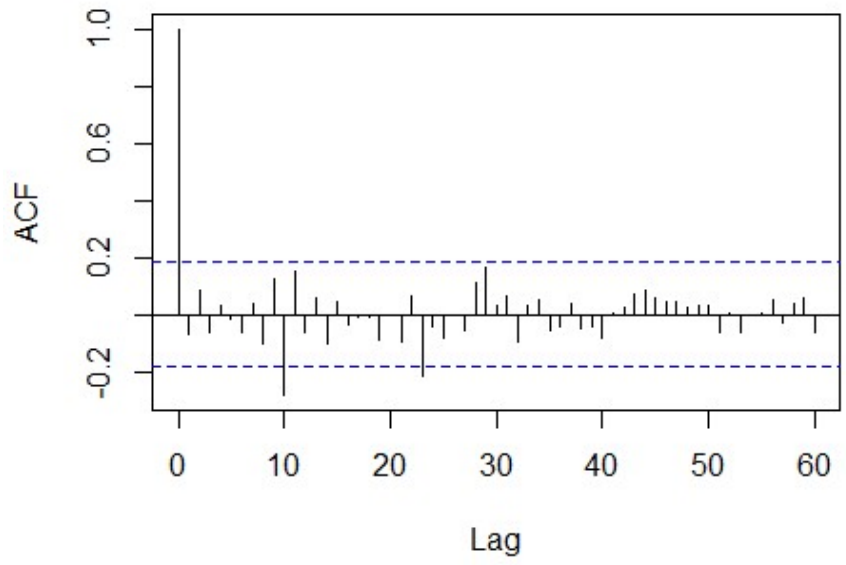



```
arima(log_dataset,order=c(9,0,0),seasonal=list(order=c(0,1,0),period=10))

##
## Call:
## arima(x = log_dataset, order = c(9, 0, 0), seasonal = list(order = c(0,
1, 0),
```

```
##     period = 10))
##
## Coefficients:
##          ar1      ar2     ar3      ar4     ar5     ar6      ar7     ar8
##       1.0432  -0.3045  0.2204  -0.2714  0.0050  0.0786  -0.0774  0.2471
## s.e.  0.0960   0.1412  0.1430   0.1430  0.1464  0.1469   0.1463  0.1426
##           ar9
##       -0.2895
## s.e.   0.0984
##
## sigma^2 estimated as 0.2924:  log likelihood = -84.83,  aic = 189.66
```

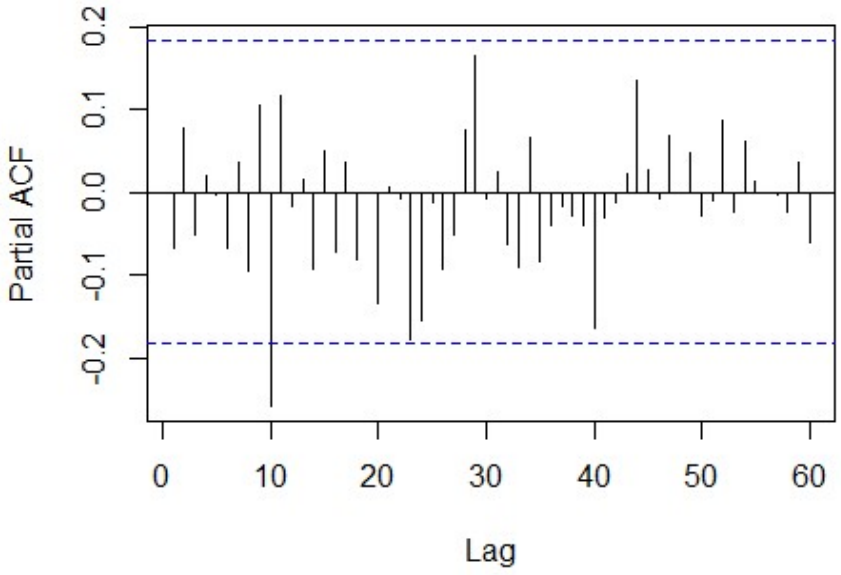
```
acf(arima(log_dataset,order=c(9,0,0),seasonal=list(order=c(0,1,0),period=10))$resid,lag.max=60)
```

lataset, order = c(9, 0, 0), seasonal = list(order = c(0, 1



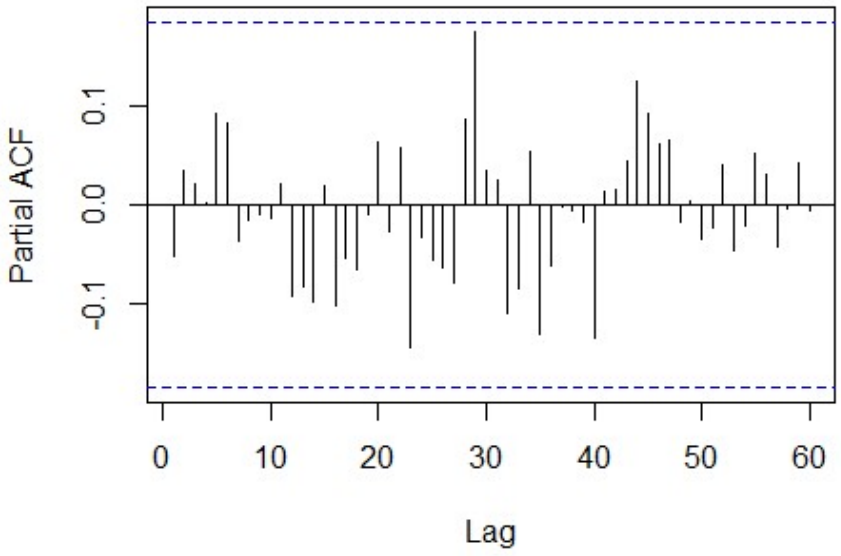

```
pacf(arima(log_dataset,order=c(9,0,0),seasonal=list(order=c(0,1,0),period=10))$resid,lag.max=60)
```

lataset, order = c(9, 0, 0), seasonal = list(order = c(0, 1



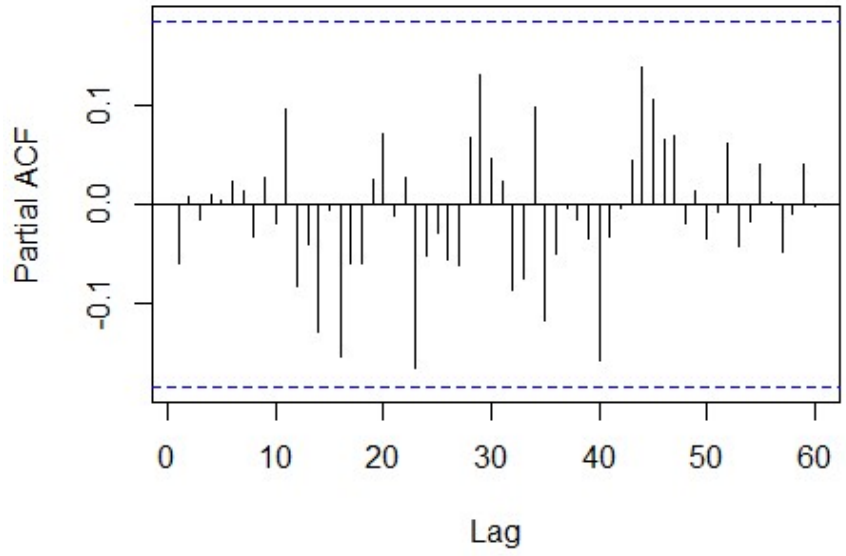

```
pacf(arima(log_dataset,order=c(3,0,5),seasonal=list(order=c(0,1,1),period=
10))$resid,lag.max=60)
```

lataset, order = c(3, 0, 5), seasonal = list(order = c(0, 1



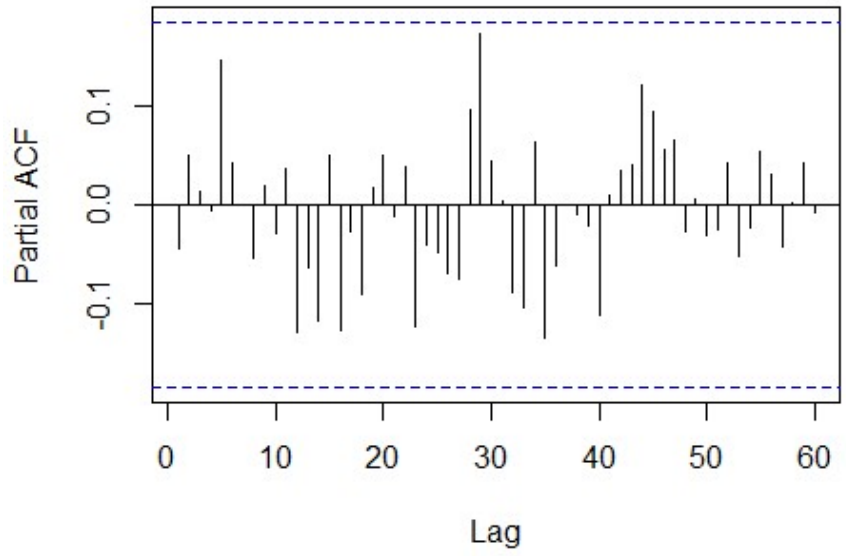

```
pacf(arima(log_dataset,order=c(2,0,5),seasonal=list(order=c(0,1,1),period=
10))$resid,lag.max=60)
```

lataset, order = c(2, 0, 5), seasonal = list(order = c(0, 1



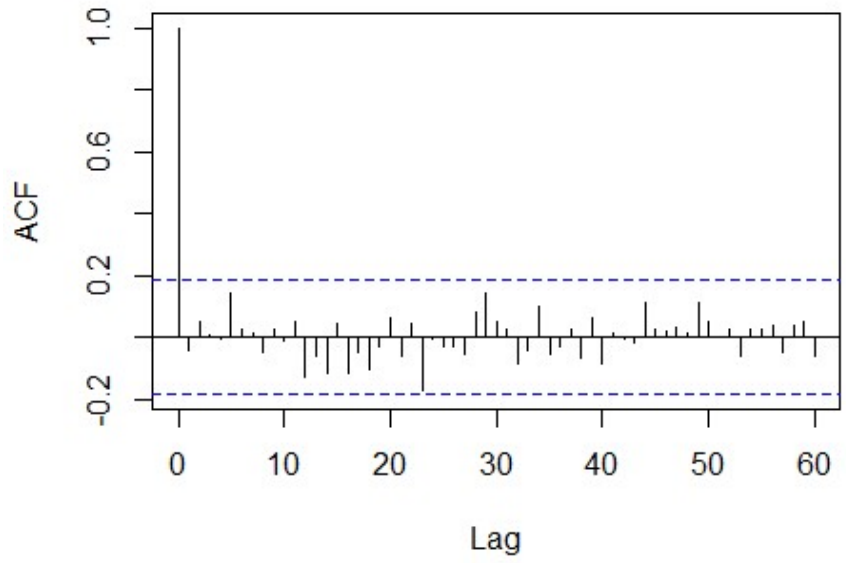

```
pacf(arima(log_dataset,order=c(2,0,4),seasonal=list(order=c(0,1,1),period=
10))$resid,lag.max=60)
```

lataset, order = c(2, 0, 4), seasonal = list(order = c(0, 1

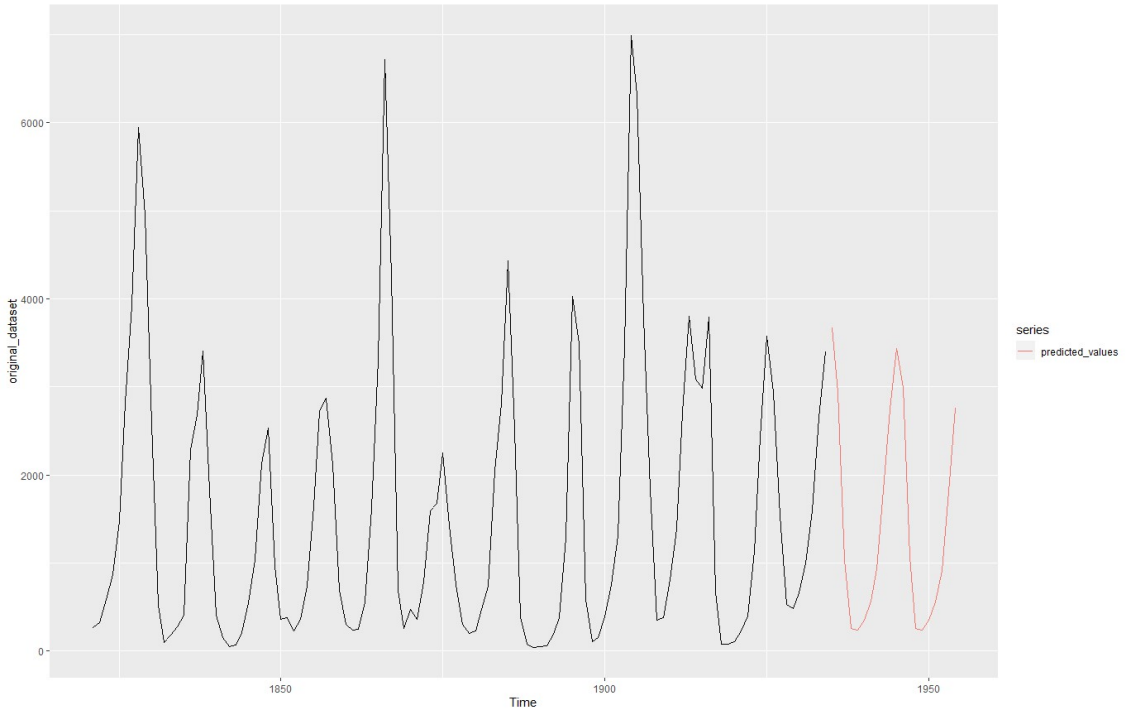



```
acf(arima(log_dataset,order=c(2,0,4),seasonal=list(order=c(0,1,1),period=1
0))$resid,lag.max=60)
```

lataset, order = c(2, 0, 4), seasonal = list(order = c(0, 1




```
arima(log_dataset,order=c(2,0,4),seasonal=list(order=c(0,1,1),period=10))

##
## Call:
## arima(x = log_dataset, order = c(2, 0, 4), seasonal = list(order = c(0,
1, 1),
##     period = 10))
##
## Coefficients:
##          ar1      ar2      ma1     ma2     ma3     ma4     sma1
##       1.3747  -0.7935  -0.1942  0.0539  0.4395  0.1081  -0.6619
## s.e.  0.1382   0.1091   0.1572  0.1335  0.1268  0.1203   0.1010
##
## sigma^2 estimated as 0.2297:  log likelihood = -74.56,  aic = 165.12

predicted_values <- exp(predict(arima(log_dataset,order=c(2,0,4),seasonal=
list(order=c(0,1,1),period=10)),n.ahead=20)$pred)
predicted_values

## Time Series:
## Start = 1935
## End = 1954
## Frequency = 1
##  [1] 3667.5212 2932.1352 1035.4448  252.7202  237.5621  352.7275  564.1
793
##  [8]  937.0795 1828.3966 2749.6017 3439.9646 2995.5460 1074.7280  259.3
341
## [15]  238.9843  348.4178  552.1040  918.5351 1809.6423 2754.2421

autoplot(original_dataset) + autolayer(predicted_values)
```



References

"How to Predict New Data Values with R" by Andrie de Vries, Joris Meys

Brockwell, P. J. and Davis, R. A. (1996). Introduction to Time Series and Forecasting. Springer, New York. Sections 3.3 and 8.3.

Introduction to Time Series in R by Alli Cramer