BAN430 Forecasting
Project 2

Bergen
April 8, 2019

Forecasting Project 2

Group: 9

Candidates: 1003 1087

NORWEGIAN SCHOOL OF ECONOMICS

Table of Contents

Part 1. Australian Private Final Consumption

Data Summary

The data for part 1 is the quarterly Australian private final consumption expenditure. The data set contains a total of 127 observations, from 1959 quarter 4 to 1991 quarter 2. On quick checking, we observe no missing value or meaningless data. Then we divide the data into training and test set. The training set includes 115 observation from 1959Q4 to 1988Q2. The test set is the rest 12 observations, stats from 1988 Q3, ends at 1991 Q2.

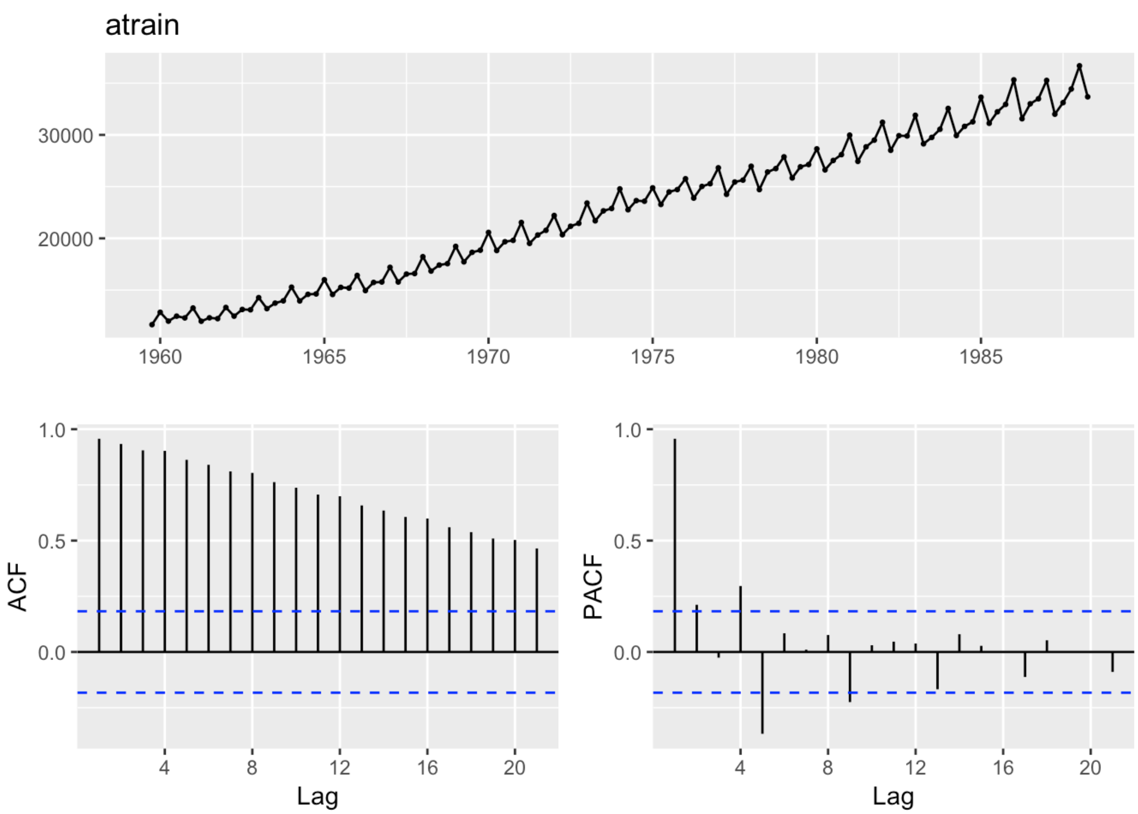


Figure 1: Plot of training series and its correlograms

From the plot, we can see that the train data has a linear upward trend with a strong seasonality from 1959 to 1991. Though having seasonal variations, the final consumption expenditure has been growing over time in general. However, these 30 years have seen a small change of the seasonality. The consumption expenditure in quarter four is surpassing quarter three instead of being equal or similar to it as in the 60s and 70s. Moreover, we can observe that the seasonal variation is enlarging over time. It can cause the problem of heteroscedasticity while fitting models.

The correlograms of the training data seen a clear decay of ACF. It dies out with along the increasing of lags, and it is out of 95% significance area up to the 21st lag. It shows that the series is strongly correlated to its lags. While in the PACF correlogram, it spikes at lag 1 and cut-off afterwards, though lag 2, 4, 5, is slightly out of the significant area. These lags still correlate with the series after its previous lags have been controlled. Furthermore, some decay but still quite obvious spikes of PACF at lag 5, 9, 13 and even 17 indicate that the seasonality of the series. Overall, the correlograms of quarterly Australian private final consumption expenditure data have typical features of a growing, non-stationary series.

Identify and Estimate the ARIMA model

From the analysis of ACF and PACF previous, we learned that ACF of the series decay along the time and its PACF cuts off after lag1, which suggests this series might follow an AR process. Since a strong seasonality can be observed, a seasonal AR model may better fit this data.

Stationarity

However, our data is not stationary series that is needed to fit AR models. We need to differentiate the series to make it stationary. So, it raises a question, how many times does it need to be differentiated to make the series stationary. We used two function *ndiffs()* and *nsdiffs()* to identify the number of regular differences required and seasonal difference required respectively. Both functions give one as answer. However, when we apply *nsdiffs()* to regular differenced data, it returns 1. While if we apply the *ndiffs()* function to the seasonally differenced data, the function returns 0. This means that we should better take one seasonal difference. If we take the first difference, we still need a seasonal difference afterwards.

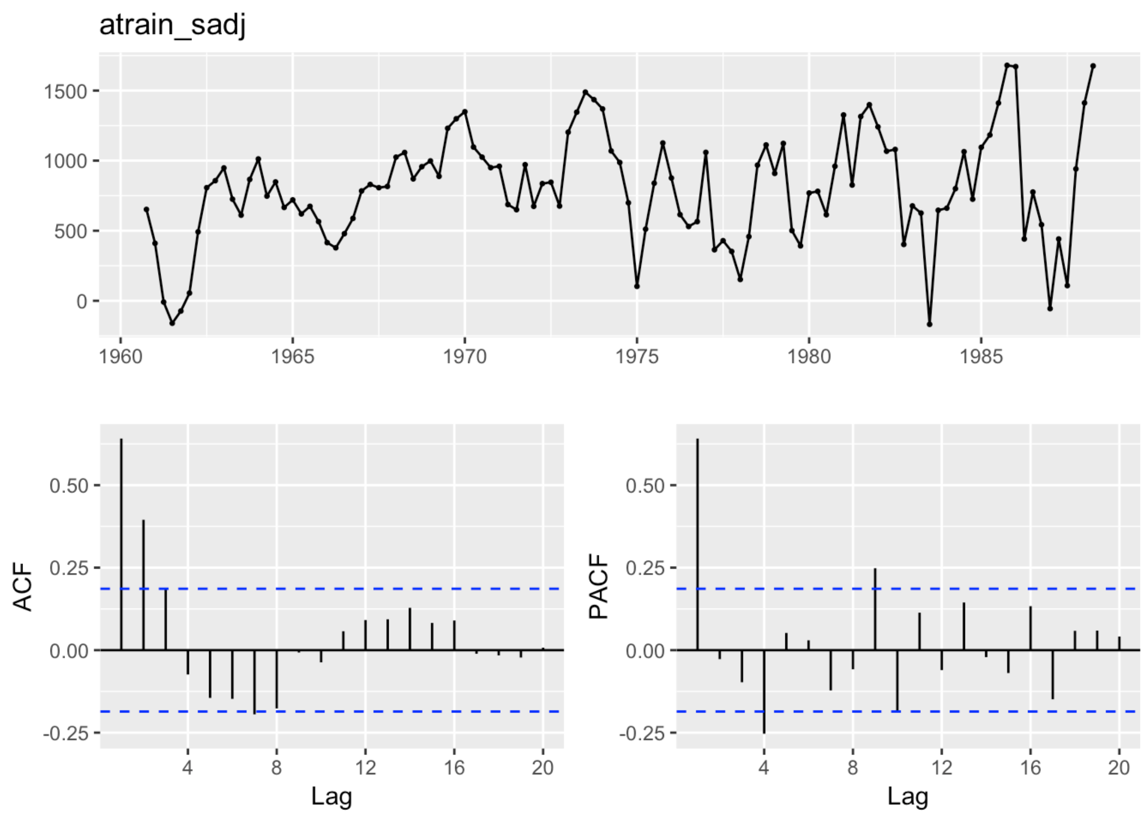


Figure 2: Plot of training series with one seasonal differentiation

From the plot, we can see that the train data differentiated with lag equal to 4(one seasonal differentiation) is not trendy any more. Except for the first few lags, the ACF and PACF of most lags are inside 95% significant area. Since it is subjective to judge stationarity of series based on plots, we also conduct the KPSS test and the ADF test to check its stationarity.

The t-statistic for the KPSS test is 0.2258, smaller than the critical value at a 10% significance level. Therefore, we cannot reject the null hypothesis of the KPSS test that the series is stationary. As for the ADF test, t-statistic of the seasonal differenced series with a constant is -4.09, which is higher than its critical value at 1%. It means we can reject the null hypothesis that the series have unit roots and say of the first order seasonal differenced series is stationary with a 99% confidence level.

Box-cox transformation

As we previously mentioned, the enlarging seasonal variation of the series may cause a problem of heteroscedasticity. In order to avoid that, we use **Box-cox transformation** to transform data. We used a function called *BoxCox.lambda()* to find the best lambda λ to make the series' variation at the same level.

The λ turns out to be 0.096, which is very close to 0 and go into similar effect as a logarithm transformation. We can see from the plot that after being transformed, instead of increasing along time, the size of the seasonal variation is at the same level across the whole series. It means λ=0.096 is suitable λ for our data set, and this transformation would make it easier to build the model.

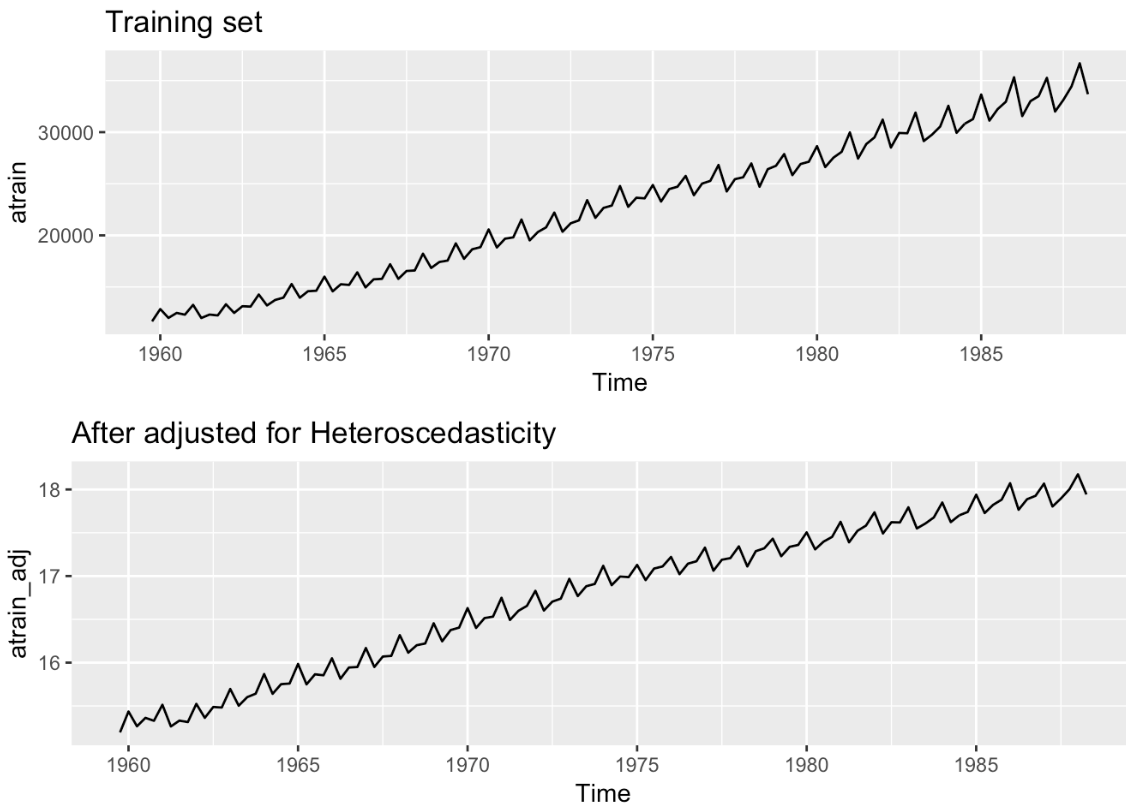


Figure 3: Comparison of the training series and its Box-cox transformed form

Now since we understand that the model should be a seasonal AR model, and we know the differentiation to make the series stationary, and we also identify the lambda to let the series get rid of heteroscedasticity, we can start to choose the right model.

First we tried a model ARIMA(1,0,0)(0,1,0)[4], with λ=0.096. The following are the correlograms of this model.

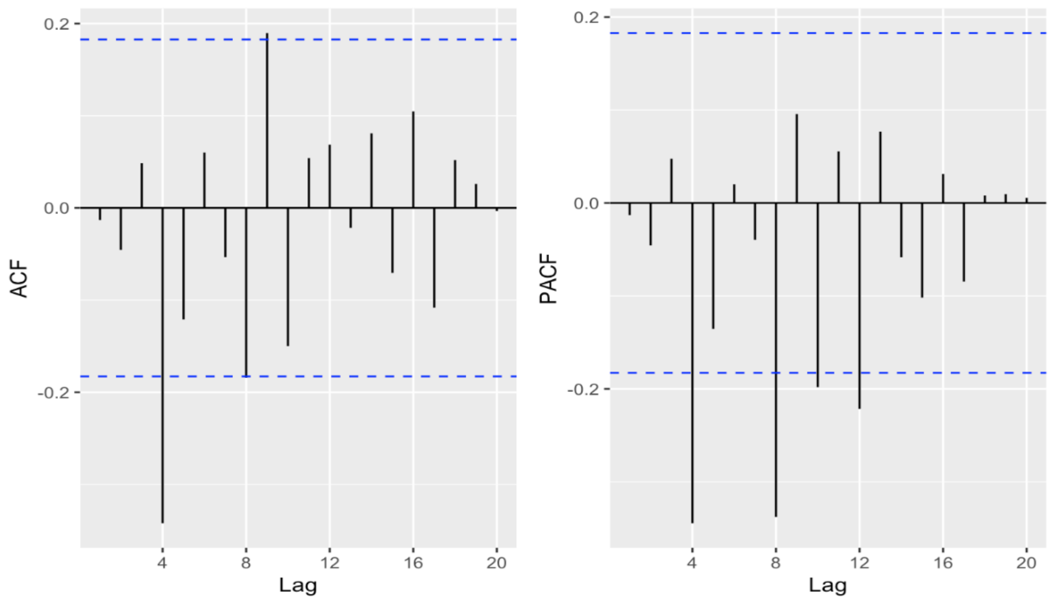


Figure 4 : ACF and PACF of residual of model ARIMA(1,0,0)(0,1,0)[4]

We can see that lag 4 and lag 8 spike in both ACF and PACF. Lag 12 spike in PACF as well. This means there still are seasonal information contains in the residual. Therefore, we tried some other models with larger seasonal AR term and non-seasonal terms. Note that a changing of lambda will change the scale of information criteria, i.e. AIC, AICc and BIC. So, we assigned λ=0.096 when fitting all the models.

Table 1: Information criteria for each model

Model	AIC	AICc	BIC
ARIMA(1,0,0)(0,1,0)[4]	-433.84	-433.72	-428.42
ARIMA(1,0,0)(1,1,0)[4]	-446.44	-446.22	-438.31
ARIMA(1,0,0)(2,1,0)[4]	**-461.54**	**-461.16**	**-450.69**
ARIMA(2,0,0)(2,1,0)[4]	-1025.6	-1025.03	-1012.05

The AIC, AICc and BIC of the model ARIMA(2,0,0)(2,1,0)[4] are abnormally small. We consider it may be a result of overfitting. In order to avoid the risk of having overfitting models, we do not consider ARIMA(2,0,0)(2,1,0)[4] though it has the lowest AIC, AICc and BIC. Excluding ARIMA(2,0,0)(2,1,0)[4], the model chosen by information criteria is ARIMA(1,0,0)(2,1,0)[4] with a lambda equals to 0.096.

One thing worth mention is that choosing lags or models is more like art rather than science. Here we used information criteria like AIC and BIC to choose the model, but the result is not absolute. There may exist some other model that score even worse in these criteria but behave better in forecasting.

Analyse the estimated model

The following is the equation of model ARIMA(1,0,0)(2,1,0)[4]:

$$(\phi B)(1-\Phi_1 B^4)(1-\Phi_2 B^4)(1-B^4)y_t = \varepsilon_t$$

Table 2 : Coefficients for model ARIMA(1,0,0)(2,1,0)[4]

	AR1 ϕ	**SAR1 Φ_1**	**SAR2 Φ_2**
Coefficients	0.989	-0.508	-0.419

AR1 term has the most significant coefficient. The model is heavily influenced by its last lag. The two seasonal AR term is also quite significant and has a big impact on y on a seasonal basis.

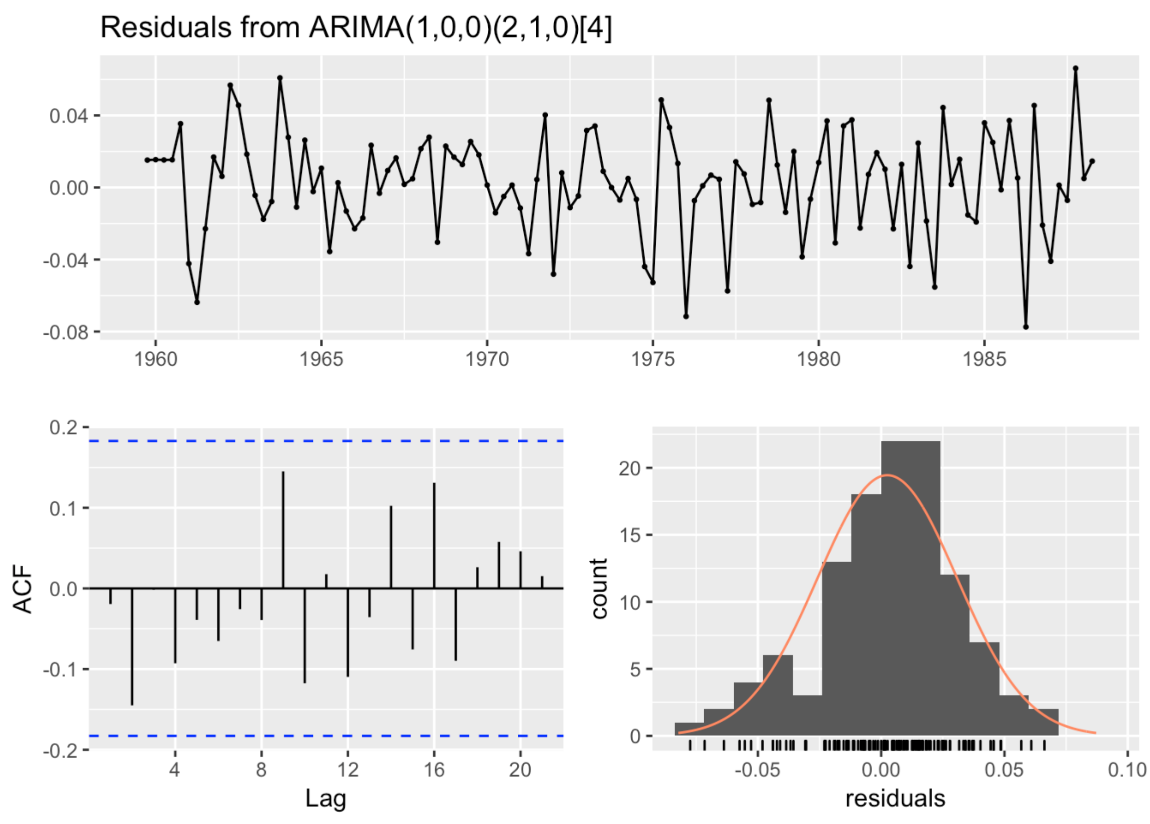


Figure 5 : ACF and PACF of residual of model ARIMA(1,0,0)(0,1,0)[4]

The residual of model ARIMA(1,0,0)(2,1,0)[4] looks relatively stationary and normally distributed. There is no visible pattern can be observed. The ACF correlogram shows that we have control of all lags. All of them are inside the 95% significant area.

We conducted the Ljung-Box test on the residual. A test p-value equals to 0.4695 indicates that we cannot reject the null hypothesis that the residuals are independently distributed. Which means there no autocorrelation exists in the residuals. The model ARIMA(1,0,0)(2,1,0)[4] with a lambda equals to 0.096 carries most of the information that the consumption series contains.

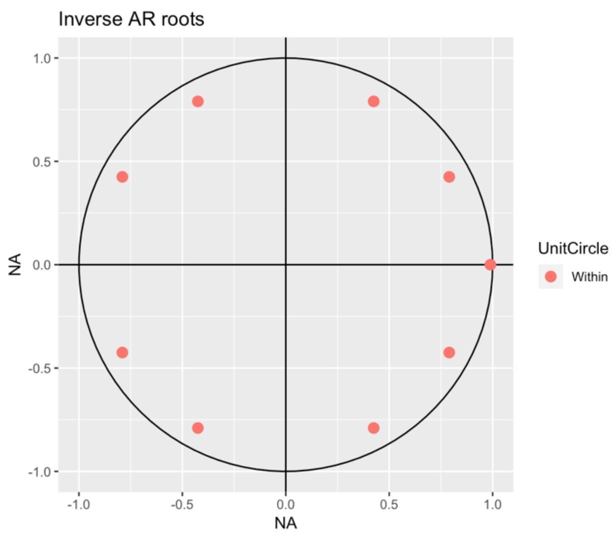
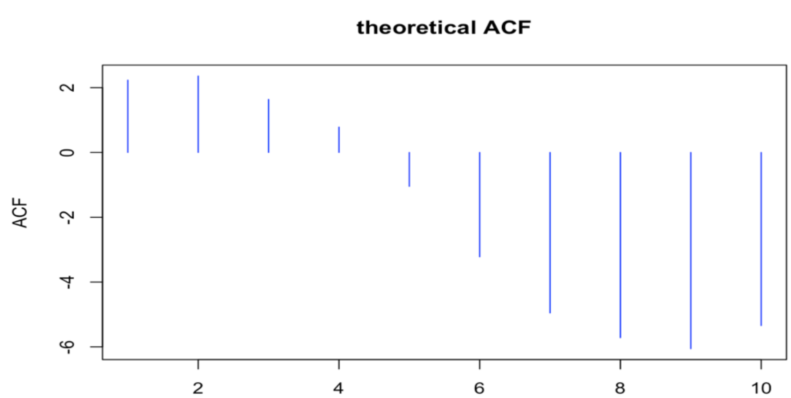


Figure 6: Plot of Unit Circle

We also check the stability of ARIMA(1,0,0)(2,1,0)[4] model. Although very close the unit circle, all the roots of this AR model are distributed within the unit circle. The model does not have unit roots, hence, is stable.

Comparison of theoretical and sample ACF and PACF



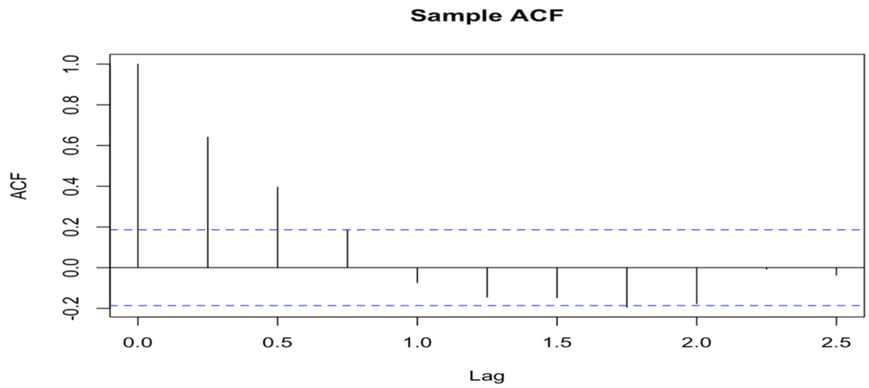


Figure 7: Comparison of theoretical and sample ACF

Here we plot the theoretical autocorrelation function for ARIMA(1,0,0)(2,1,0)[4] model comparing the ACF of seasonal differenced training data. We can see that though having different scale, both ACF of theoretical and sample follows a decaying trend. Both dies down to zero and turn to be negative at lag 4, 5.

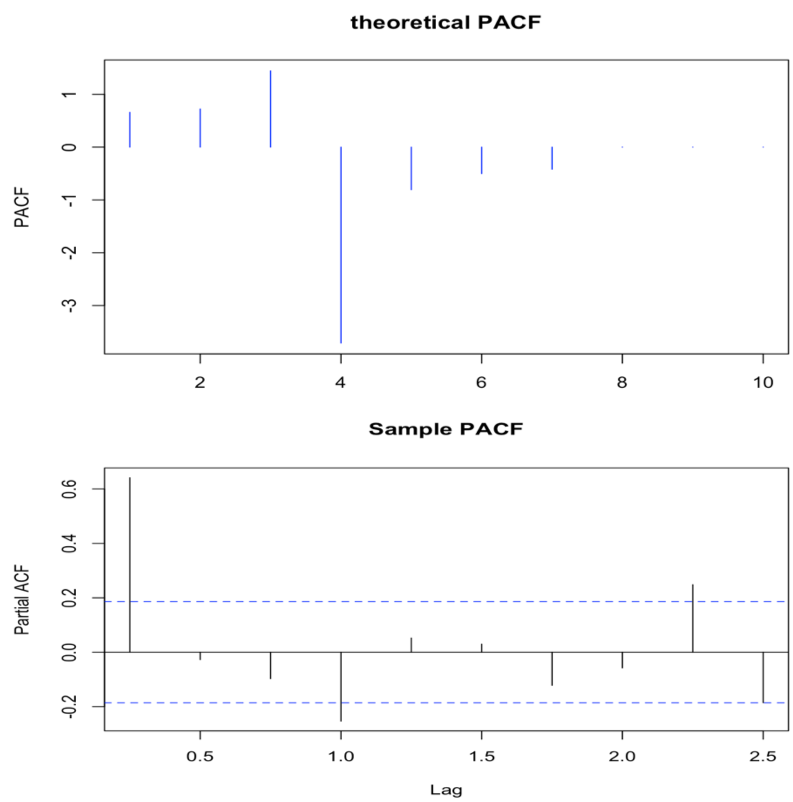


Figure 8: Comparison of theoretical and sample PACF

Both theoretical and sample partial autocorrelation function dies down to zero quickly and fluctuate around zero — both spike at lag 4.

Overall, though having certain differences, the ACF and PACF of the ARMA process and the seasonal differenced series have many similarities. We can conclude that the ARIMA(1,0,0)(2,1,0)[4] model describe the fitted series quite well.

Forecasting

Using ARIMA(1,0,0)(2,1,0)[4] estimated by training data, we conduct the forecasting of Australian private final consumption expenditure from 1988 Q3 to 1991 Q2.

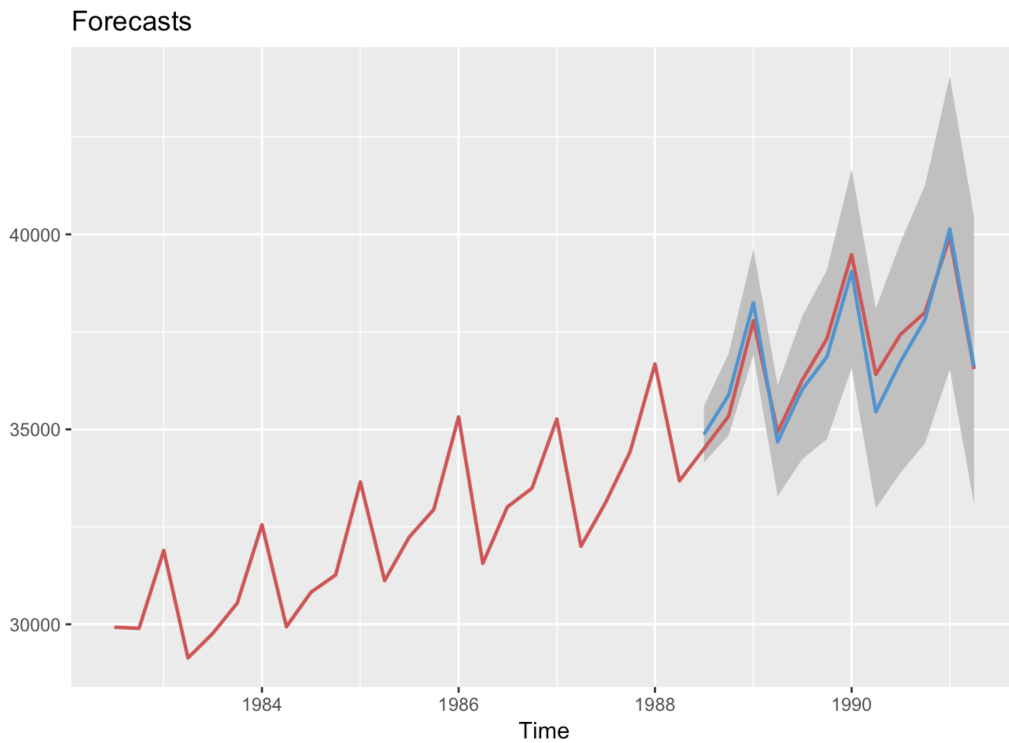


Figure 9: Forecasting the last three years' expenditure

The red line is the actual consumption expenditure, the blue line is the point forecasted and the grey area stands for a 95% confidence interval. In general, the forecasting captures the trend and the seasonality of the test set and performs pretty well. The predictive interval enlarges along time, which makes sense since we adopt the dynamic forecasting approach here. For example, using the number, we predicted to forecast the number in the next period, without feeding in real observations in the test set. In other words, we accumulate the uncertainty while forecasting the period long from now. As for the point forecasting, we can see that the mean of the forecasting has a similar value of the test set from 1988 Q3 to 1990Q2 while in 1990 Q3, the forecasted number is about 3% lower than the real value. Although having a minor deviation at quarter three, the forecasted number at 1990 Q4, 1991 Q1 and Q2 are almost the same as the actual consumption expenditure.

The well-performed forecasting not only shows that we probably choose the right model but also because this time-series is suited for using an ARMA to describe.

Table 3 : Train and test error for model ARIMA(1,0,0)(2,1,0)[4]

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	17.39	263.21	193.34	0.09	0.84	0.24	-0.12	NA
Test set	133.11	469.78	404.34	0.35	1.10	0.50	0.66	0.23

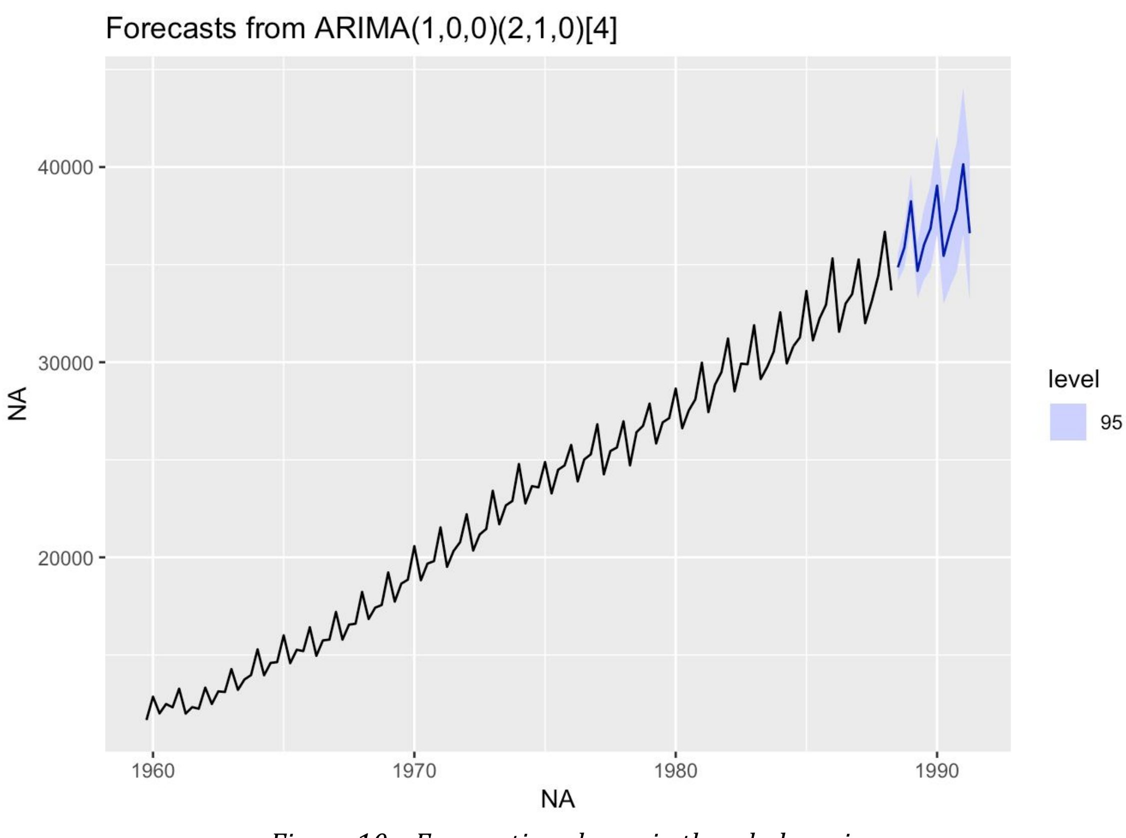


Figure 10 : Forecasting shown in the whole series

Part 2. Annual lynx trapping in Canada

Data Summary

The lynx series shown in *Figure 12* contains the number of lynx trapped per year in the MacKenzie River district of Northern Canada. The dataset has 114 observations from 1821 to 1934. First, we get a quick glimpse of the data to check if there are any missing values or meaningless values (for example, negative values). We find no missing values or negative values. Second, we look at the summary of this dataset as shown in *Table 4*. This dataset has a significant variation between the big numbers and small numbers. The minimum value is 39 while the maximum is 6991. The median value is 771 and the Mean value is 1538. We can see from the histogram (*Figure 11*) that this variable has a skewed distribution. In most years, under 1000 lynxes were trapped. At last, we plot the whole series as shown in *Figure 12*, in which we observe longer-than-seasonal periodicity. There are spikes in every 10 or so years.

Then, we divide the data into a training set and a test set where the test set is the last 12 observations.

Table 4: Summary of the lynx series

```
Annual.number.of.lynx.trapped..MacKenzie.River..1821.1934
Min.   :  39.0
1st Qu.: 348.2
Median : 771.0
Mean   :1538.0
3rd Qu.:2566.8
Max.   :6991.0
```

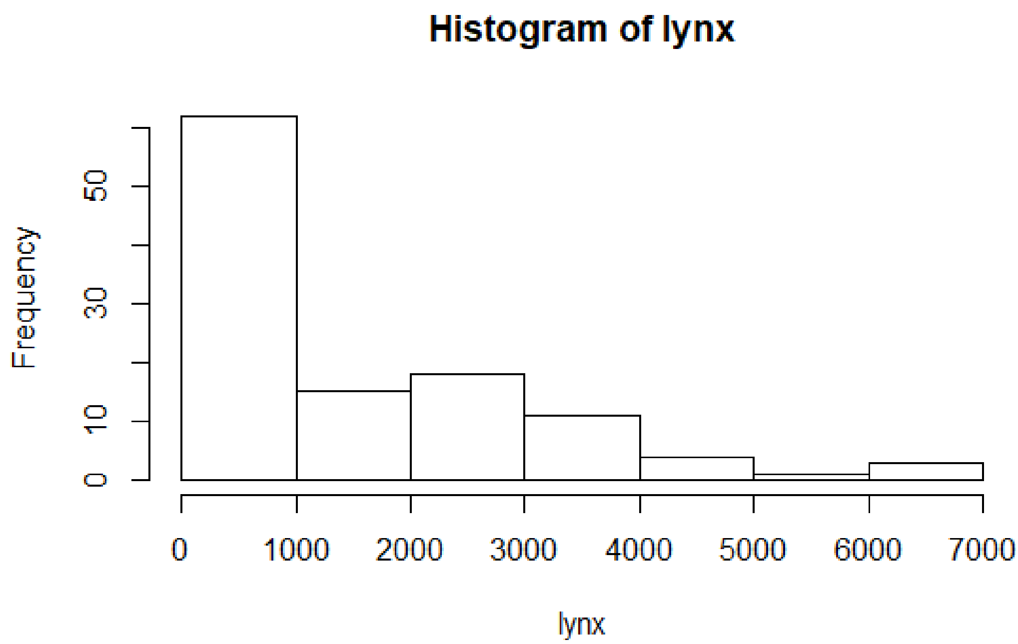


Figure 11: Histogram of the lynx series

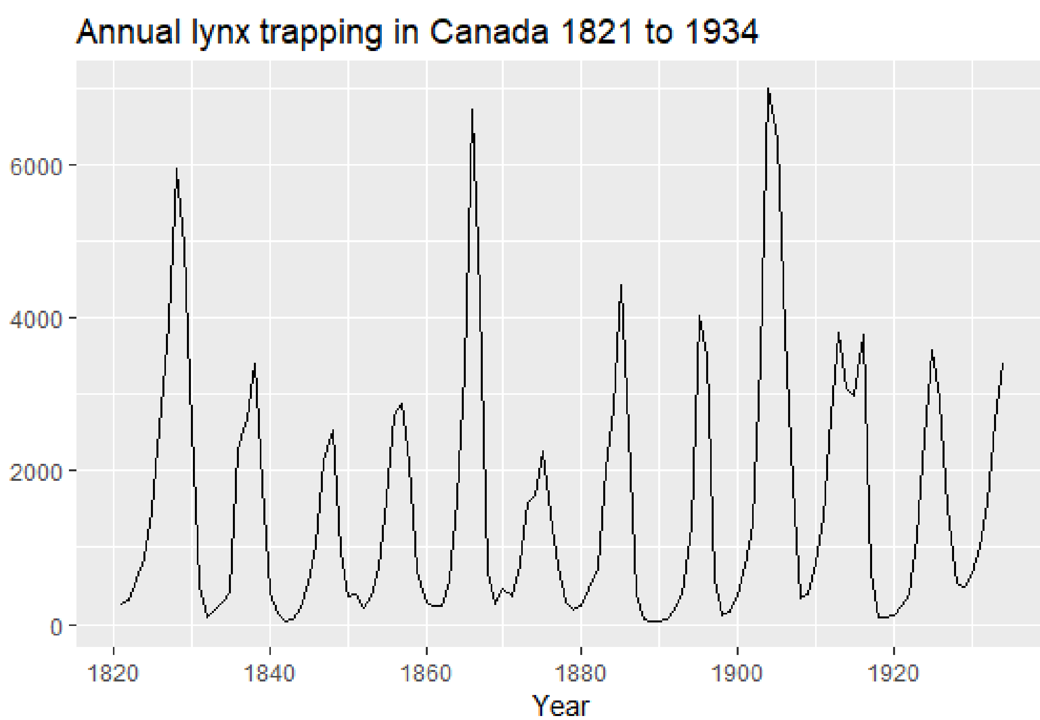


Figure 12: Time plot of the lynx series

Identify and Estimate the ARIMA model

Unit root test

We tested the stationarity of the lynx series with three different methods and all the results (*Table 5*) show that the series is stationary. In the ADF test, the t-statistic is smaller than the 1% critical value, rejecting the null hypothesis that the series is nonstationary. Therefore, the series is stationary. In the KPSS test, the test statistic is smaller than the 10% critical value, failing to reject the null hypothesis that the series is stationary. Thus, the series is stationary, which is the same as in the ADF test. In the Elliot, Rothenberg and Stock test, the test statistic is smaller than the 1% critical value, rejecting the null hypothesis that the series is nonstationary. Therefore, the series is stationary, which is the same as in the ADF test and KPSS test.

Table 5: Unit root test of the lynx series

Unit Root Test Method	test-statistic	1%	5%	10%	Null hypothesis	Conclusion
ADF	-4.53	-2.58	-1.95	-1.62	Unit root	Stationary
KPSS	0.05	0.739	0.463	0.347	Stationary	Stationary
ERS	-3.76	-2.59	-1.94	-1.62	Unit root	Stationary

ACF and PACF plots

The ACF and PACF plots for the lynx data series are shown in *Figure 13*. The ACF is sinusoidal and becomes insignificant after around 80 lags. While the PACF has significant spikes at lag 1, lag 2, lag 4 and lag 8. Therefore, the lynx series may follow an ARIMA(8,0,0) model.

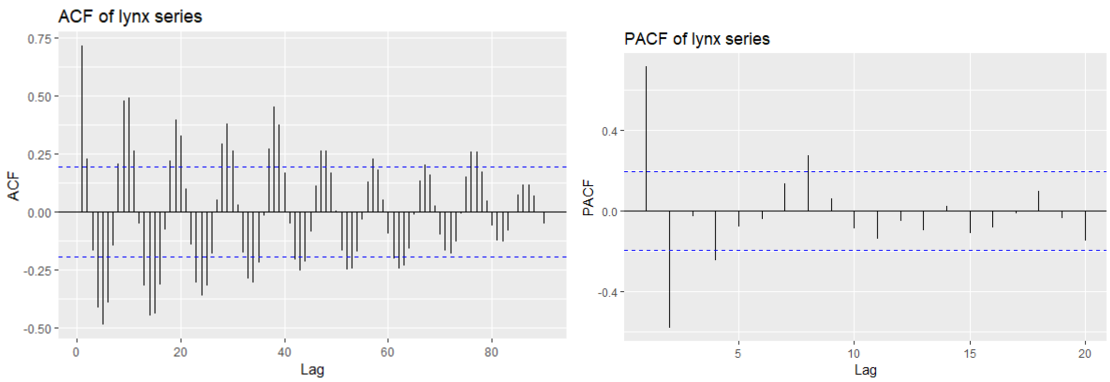


Figure 13: ACF and PACF of the lynx series

Choosing the right model

In order to make sure the forecast values and the prediction intervals stay positive, we make a Box-Cox transformation. Using the function BoxCox.lambda, we get the lambda=0.135 which minimizes the coefficient of variation for subseries of lynx data. Then, we fit an ARIMA(8,0,0) model and the variations including ARIMA(7,0,0), ARIMA(9,0,0), ARIMA(8,0,1) using the training set. Of the four models, the ARIMA(8,0,0) has the smallest AICc value, as is shown in *Table 6*.

Table 6: AICc of the models

```
         model     AICc
1 ARIMA(8,0,0) 350.3758
2 ARIMA(7,0,0) 351.4071
3 ARIMA(9,0,0) 351.7151
4 ARIMA(8,0,1) 352.5764
```

Check the residuals

The residuals should satisfy these requirements to ensure the model yields good forecasts:

(1) Uncorrelated. From the ACF of residuals, we find each spike is within the required limits, indicating no significant correlation in the residuals. In addition to looking at the ACF plot, we also conduct the Ljung-Box test with 20 degrees of freedom, which is calculated by the maximum lag subtracts the number of coefficients in the model. The p-value is 0.09 and is above 0.05, thus, we can conclude the residuals seem like a white noise series.

(2) Zero mean and constant variance. The mean of the residuals is -0.016, which is close to 0. From the time plot of the residuals, the in-sample forecast errors seem to have roughly the same variance over time, although the size of the fluctuations in the subseries 1840-1860 may be slightly less than that at time 1900-1920. Therefore, the variance of the residuals can be treated as constant.

(3) Normal distribution. The histogram of the residuals suggests that the residuals may follow a normal distribution, although slightly skewed compared to the overlaid normal curve.

In summary, the ARIMA(8,0,0) model is an adequate estimation of the lynx data, because the forecast errors are normally distributed with mean zero and constant variance, furthermore, the residuals are uncorrelated.

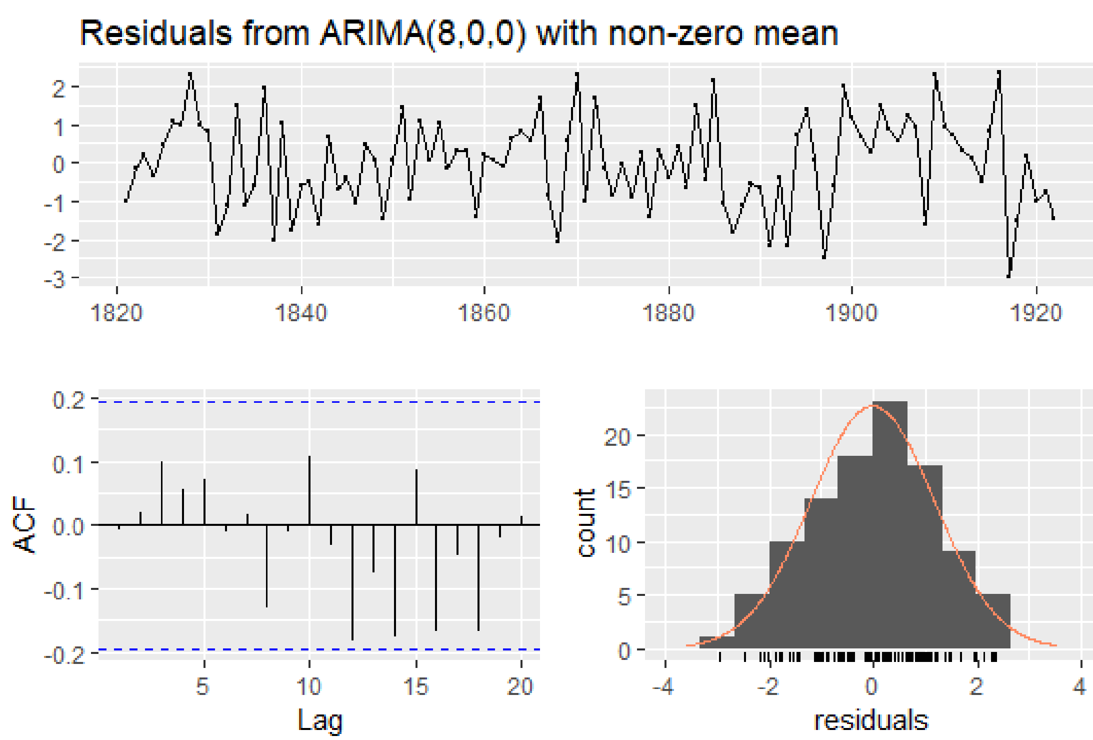


Figure 14: Residuals from ARIMA(8,0,0)

Table 7: Ljung-Box test of the residuals from ARIMA(8,0,0)

```
Ljung-Box test
data:  Residuals from ARIMA(8,0,0) with non-zero mean
Q* = 28.906, df = 20, p-value = 0.08962
Model df: 9.   Total lags used: 29
```

The points representing characteristic roots are within the unit circle (*Figure 15*), indicating that the model ARIMA(8,0,0) is stationary.

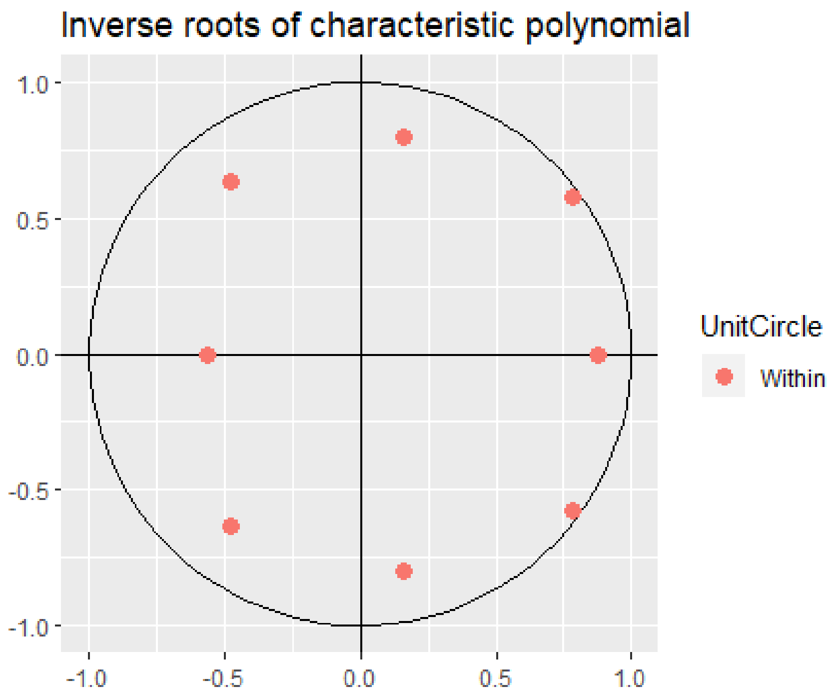


Figure 15: Inverse roots of characteristic polynomial of ARIMA(8,0,0) model

Estimate the model

The ARIMA(8,0,0) model can be written as below, where ε_t is white noise:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \phi_3 y_{t-3} + \phi_4 y_{t-4} + \phi_5 y_{t-5} + \phi_6 y_{t-6} + \phi_7 y_{t-7} + \phi_8 y_{t-8} + \varepsilon_t$$

After getting the estimated coefficients, the model can be written as below, in which we can tell that the value of lag 1 has the most significant positive effect, the value of lag 2 has the most important negative effect and the influence of lag 7 is almost zero.

$$y_t = 11 + 1.2y_{t-1} - 0.8y_{t-2} + 0.4y_{t-3} - 0.4y_{t-4} + 0.2y_{t-5} - 0.1y_{t-6} + 0y_{t-7} + 0.2y_{t-8} + \varepsilon_t$$

Table 8: The coefficients of the ARIMA(8,0,0) model

Use the model to forecast the last 12 observations

Because we use Box-Cox transformation, the forecast values and the prediction intervals are all positive. We can tell from *Figure 16* that the test set falls into the 80% confidence level, however, the point forecasts are quite far away from the true values though roughly capture the cyclic pattern. The forecast errors are evaluated as the mean error, root mean squared error, mean absolute error and so on. For the test set, the ME, RMSE and MAE are all quite large, indicating the ARIMA model can be improved. Therefore, we will fit an NNAR model and make combinations of the ARIMA model and the NNAR model in later sections.

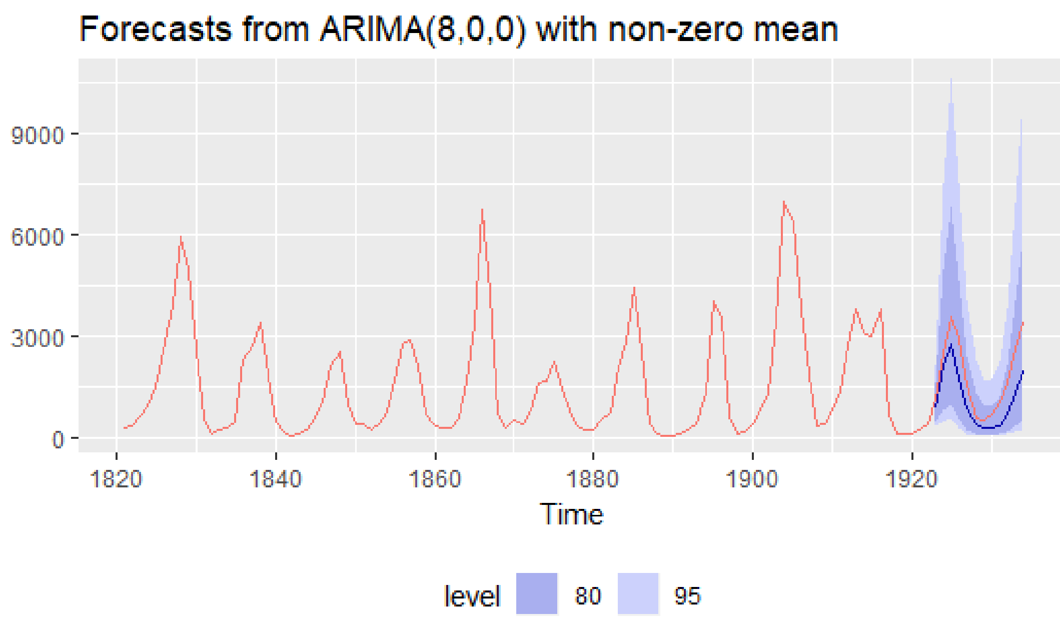


Figure 16: Forecasts plot from ARIMA(8,0,0) model

Table 9: Forecasts accuracy of ARIMA(8,0,0) model

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	128.9238	802.5360	496.9920	-15.64023	44.85589	0.5925511
Test set	718.7567	838.9263	718.7567	42.88163	42.88163	0.8569556

Use a neural network autoregressive model

We fit a neural network autoregressive model with one hidden layer. There are p lagged inputs ($y_{t-1}, y_{t-1}, \dots y_{t-p}$) in the input layer and k nodes in the hidden layer, and the output is y_t. Because there is no seasonality in the lynx series, we don't need to specify seasonal lags. The optimal number of p is the number of lags in the AR(p) model, which is 8 in our case. The default k is set to $p/2$ (rounded to the nearest integer), which is 4 in our case. Therefore, we get the NNAR(8,4) model, indicating 8 inputs in the input layer and 4 nodes in the hidden layer. In order to make the results reproducible, we set a random seed, which is 1 in our case, to make sure the neural network autoregressive model yields same results every time.

Then we check the residuals of the NNAR model and conclude the model has captured the information in the lynx series. First, we find each spike is within the required limits from the ACF of residuals, indicating no significant correlation in the residuals. Second, the mean of residuals is -0.2, which is close to zero. At last, the histogram of the residuals suggests that the residuals may be a normal distribution, although slightly left-skewed compared to the overlaid normal curve.

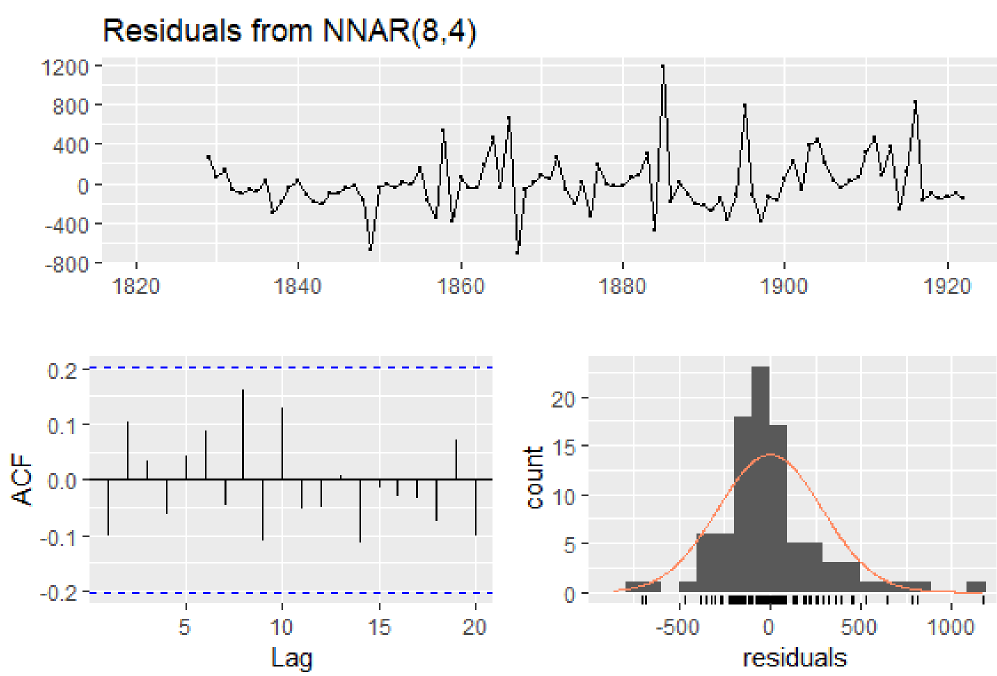


Figure 17: Residuals from NNAR(8,4)

Finally, we forecast the last 12 observations. Compared to the forecasts from the ARIMA model, the point forecasts from the NNAR model is more accurate and capture the cycles well. Most of the test set fall into the 95% confidence interval even though the prediction intervals are narrow.

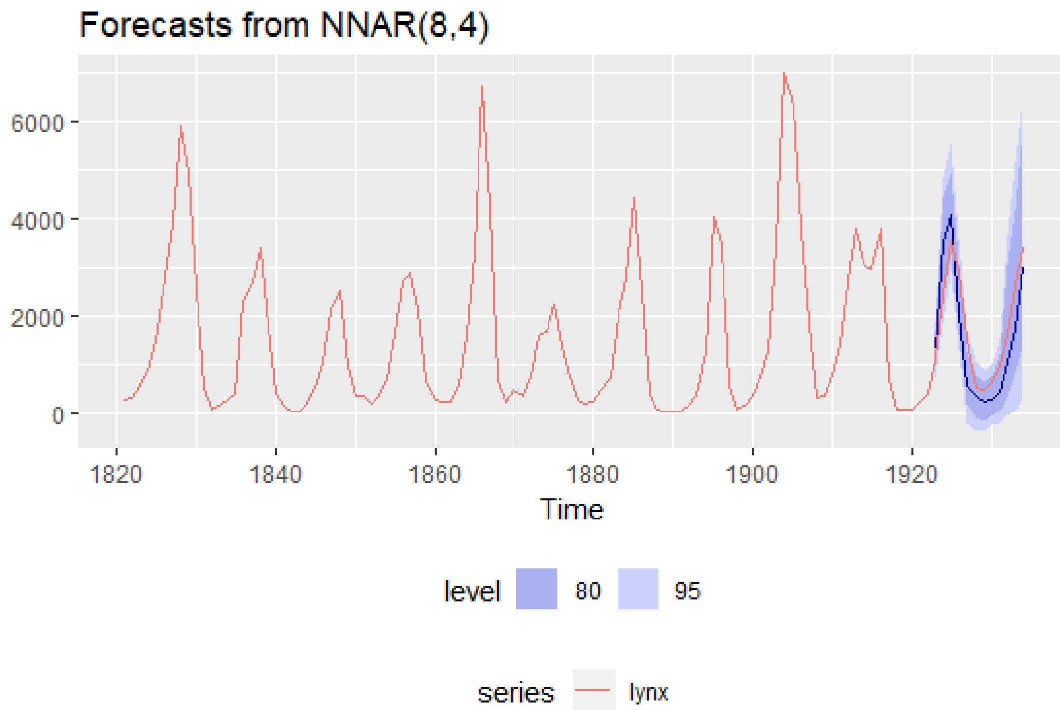


Figure 18: Forecasts plot from NNAR(8,4)

Table 10: Forecast accuracy of NNAR(8,4)

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	-0.2024641	284.5571	195.9303	-37.63312	49.41176	0.2336029
Test set	259.4860878	639.7627	562.0290	23.43350	36.35266	0.6700932

Combine the forecasts in two ways

So far, we have estimated the ARIMA model and the NNAR model, in which the NNAR model has more accurate forecasts. We try to make the combinations of these two forecasts using the simple average and the regression to see if we can improve forecast accuracy.

In order to get the combination by average, we simply take the mean of the two forecasts. The combination by average is useful when we don't know the accuracy of the individual forecasts, or all the models have similar forecasts accuracy.

To get the combination by regression, we need to specify the weights of the two forecasts. At first, we fit a linear regression without the intercept term, in which the fitted values of the two models are independent variables, the actual value of the training set is the dependent variable. Secondly, remember to take the subset of the variables because there are missing fitted values of the first eight observations from the NNAR(8,4) model. At last, scale the coefficients of the regression so that the sum of the coefficients is 1. Therefore, the weight of forecasts from the ARIMA model is -0.23, while the weight of forecasts from the NNAR model is 1.23. Negative weight doesn't mean the ARIMA model is of no value in the combination, it can be used to offset the forecast errors of the NNAR model.

Evaluate the four forecasts and plot the forecasts

According to the evaluation of the four forecasts, the ARIMA yields the worst estimation with all the accuracy measures. Different accuracy measures get different conclusion about which model is best, for example, the combination by average performs best in mean absolute error (MAE) and mean absolute percentage error (MAPE), while the combination by regression is best in mean error (ME), mean percentage error (MPE) and mean absolute scaled error (MASE). Since root mean squared error (RMSE) is most widely used, we conclude that the NNAR model has the most accurate forecasts.

Table 11: Evaluation of the four forecasts

	ME	RMSE	MAE	MPE	MAPE	MASE
ARIMA	718.7567	838.9263	718.7567	42.88163	42.88163	0.8569556
NNAR	259.4861	639.7627	562.0290	23.43350	36.35266	0.6700932
Average	489.1214	660.1121	554.9715	33.15757	35.86522	0.4841695
Regression	154.0467	691.4367	580.6034	18.96860	37.19326	0.4736730

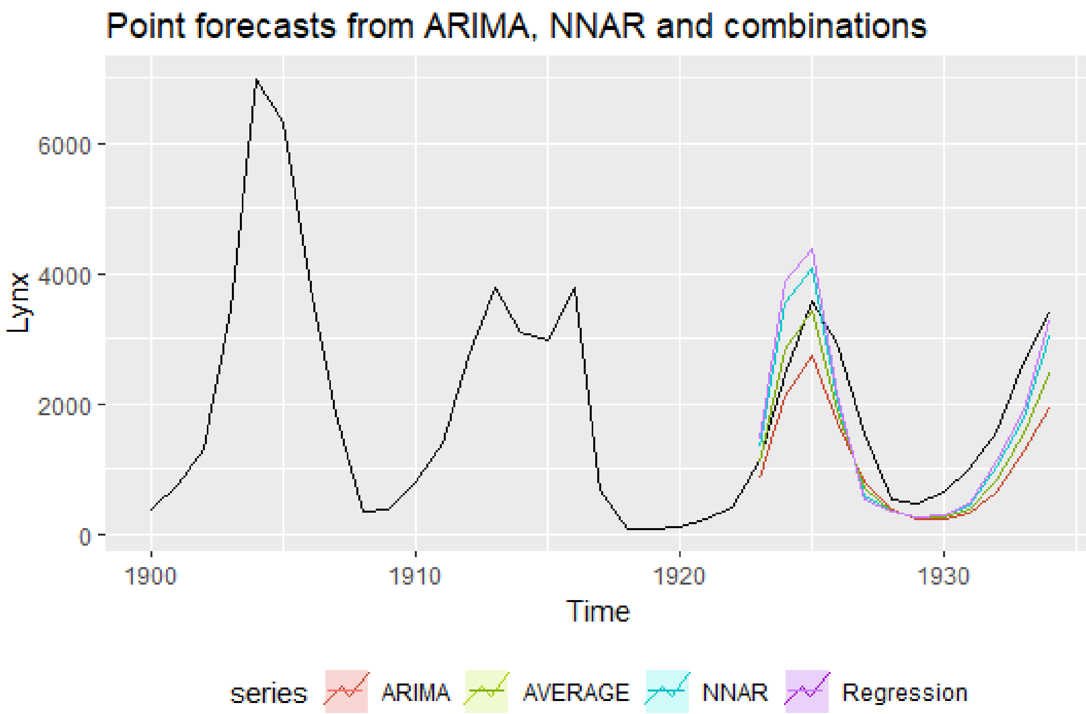


Figure 19: Plot from the four forecasts

Reference

Adrian Trapletti and Kurt Hornik (2018). tseries: Time Series Analysis and Computational Finance. R package version 0.10-46.

Hilde C. Bjørnland, Leif Anders Thorsrud (2015) Applied time series for macroeconomics, 2nd edition.

Hyndman, R.J., & Athanasopoulos, G. (2018) Forecasting: principles and practice, 2nd edition,OTexts: Melbourne, Australia. OTexts.com/fpp2. Accessed on 5. April.2019

Hyndman RJ, Khandakar Y (2008). "Automatic time series forecasting: the forecast package for R." _Journal of Statistical Software_, *26*(3), 1-22.
<URL:http://www.jstatsoft.org/article/view/v027i03>.

Niels Holtrop. (2014). Finding effective weights to combine forecasts, A search for effective weights to combine volatility forecasts, ERASMUS SCHOOL OF ECONOMICS

Petropoulos F, Razbash S, Wang E, Yasmeen F (2019). _forecast: Forecasting functions for time series and linearmodels_. R package version 8.5,
<URL:http://pkg.robjhyndman.com/forecast>.