Introduction

I have got the dataset about ATM withdrawals at UK banks. My individual datasets are N67 and N107. The tasks of the coursework are to analyse the data, develop several models, evaluate and select the best model and give reasonable forecast data. So, this report is comprised of four parts: data analysis, model development, model evaluation and forecast and conclusion.

1.Data analysis

The data sets I need to analyse are N67 and N107. First, I need to extract the two data sets separately and store them in the csv file. In the R software, I imported the required data and used the function to detect the dataset. There are 30 null values in the two data. In order to make the data continuous, I use the average of the data to replace the null value. Next, I import the data of N67 and N107 in time series and draw their images as it shown in figure 1.1 and 1.3. Through the image, I can see that the data of N67 does not show a strong trend over time. At the same time, there was a peak in the data in mid-1997. By plotting the histogram of N67 (Figure 1.2), the data of N67 has a fat tail in its distribution. For the data N107, N107 showed two peaks in the second half of 1996 and the end of 1997, respectively. And these two peaks seem to have a periodic pattern. By plotting the histogram of N107(Figure 1.4), the distribution of N107 data is closer to the normal distribution than N67, and it also has the case of fat tail.

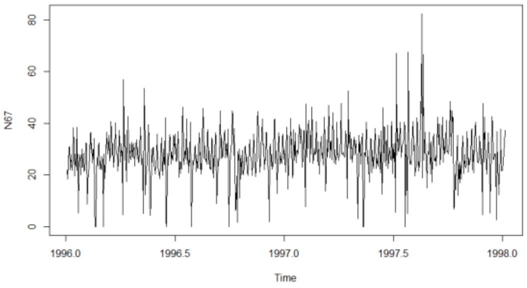


Figure 1.1

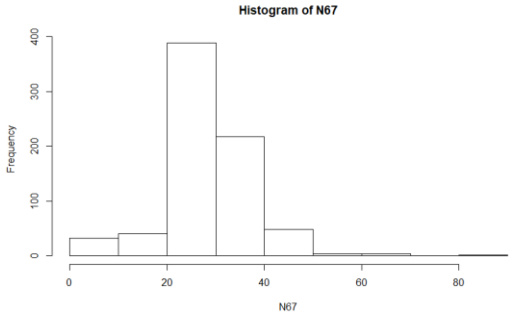


Figure 1.2

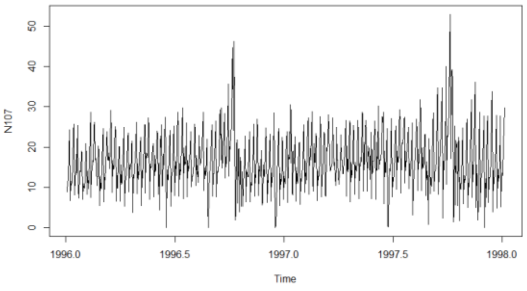


Figure 1.3

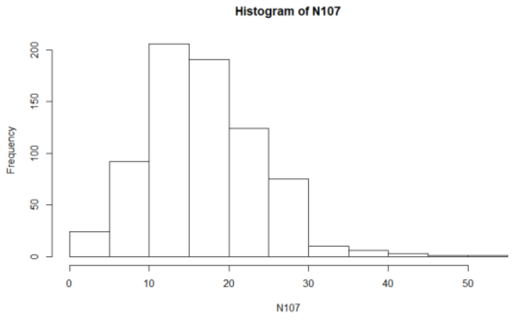


Figure 1.4

To further explore the level, trend, seasonality and regular component of the data, I need to decompose the data. There are two models that are useful for describing the relationship between these components: the multiplicative model and the additive model. According to the analysis results in Figure 1.5 and 1.6 below, N67 showed a slight upward trend in both of the two models. And with regard to seasonality, the N67 also showed a certain seasonality. According to the decomposition result of N107 (Figure 1.7 and 1.8), the trend of N107 also showed a slight rise with time. At the same time, the N107 showed a strong seasonality. In terms of the irregular component, all the decomposition results of the N107 look closer to white noise. The N67's irregular component contains more outliers.

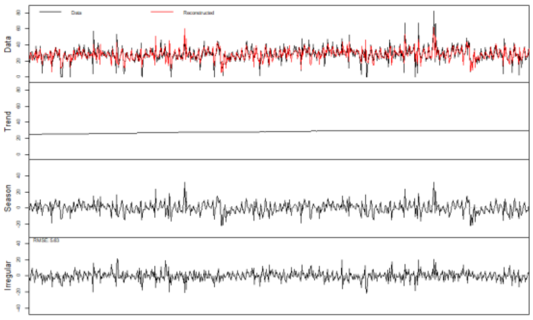


Figure 1.5

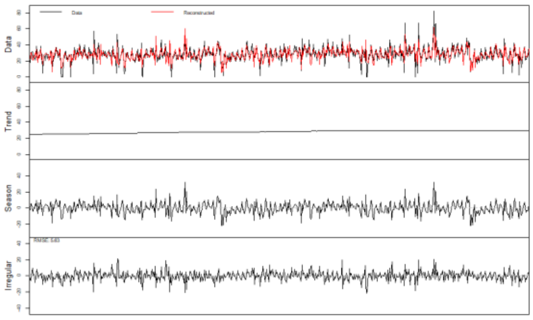


Figure 1.6

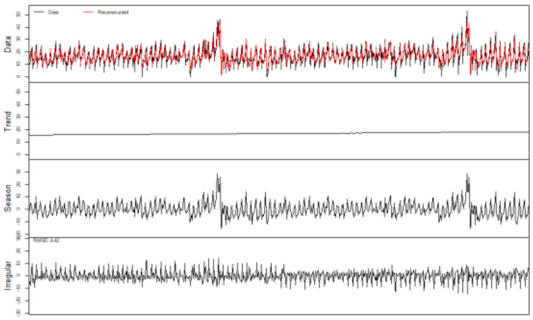


Figure 1.7

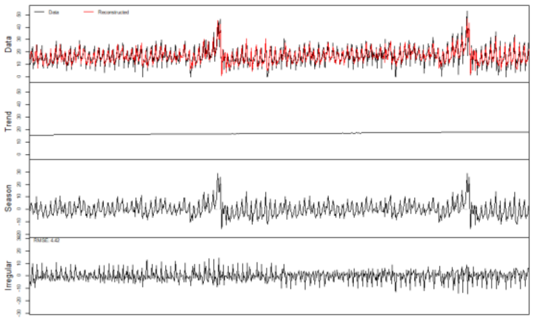


Figure 1.8

Next, I perform ACF and PACF analysis on N67 and N107, and the results are shown in the figure below. According to the figure 1.9, I can see that the ACF of N67 drops slowly and has obvious seasonality in the ACF graph, which indicates that N67 is not stationary and needs differential processing. Similarly, in the processing of N107 data (Figure 1.10), I can still see the same situation, which means that both data need to be differentially processed.

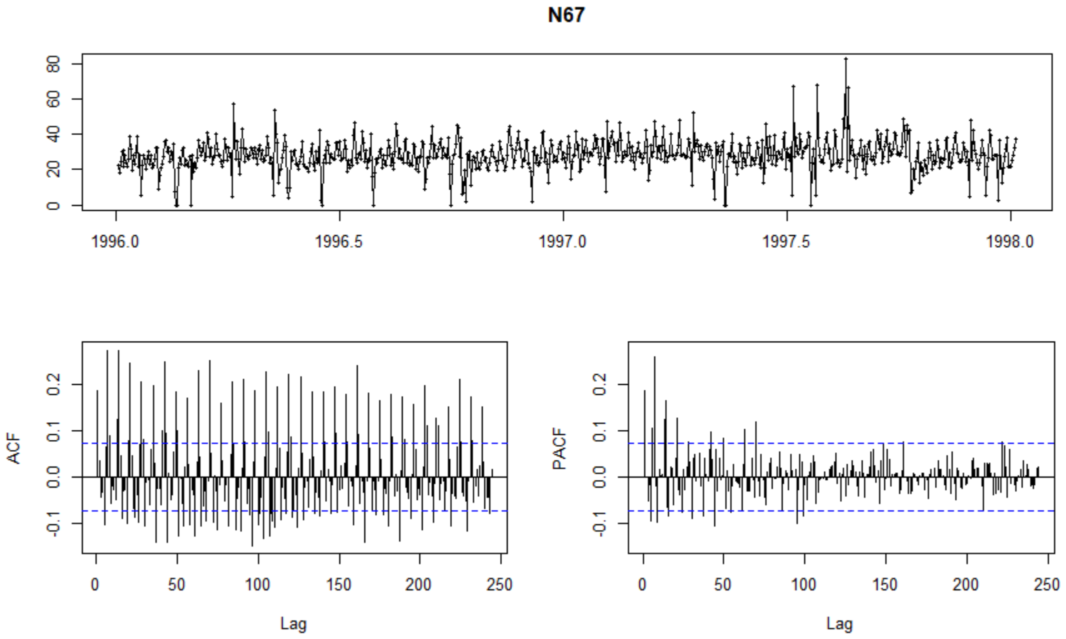


Figure 1.9

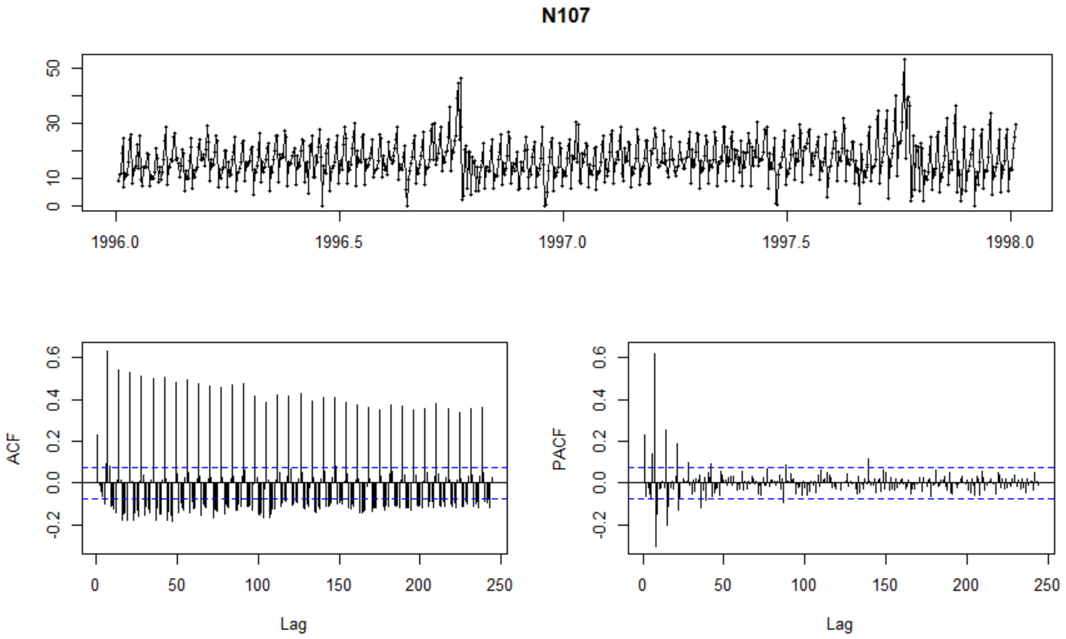


Figure 1.10

In order to verify the above statement, I can use the KPSS test to check whether the data is stationary. The KPSS test results for the two data sets are given in the appendix 1 and 2. I can see that the two data are not stationary and require differential processing. First, I need to perform the first-order differential processing on the two data, and then observe that the two data seem to follow the seasonality of the period of 7, so seasonal differential processing (lag=7). After that, I can see that the ACF diagrams of N67 and N107 are as follows (figure 1.11 and 1.12). The differentially processed data showed a better result. This shows that both data have a seasonality of lag=7. This result can be used as an important reference in the establishment of the ARIMA model.

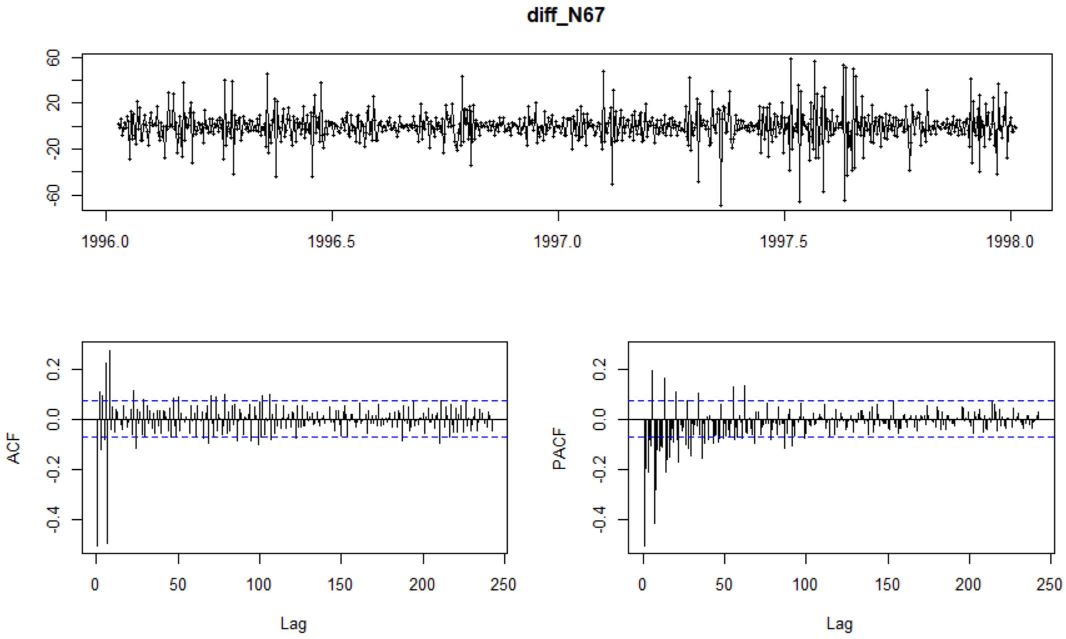


Figure 1.11

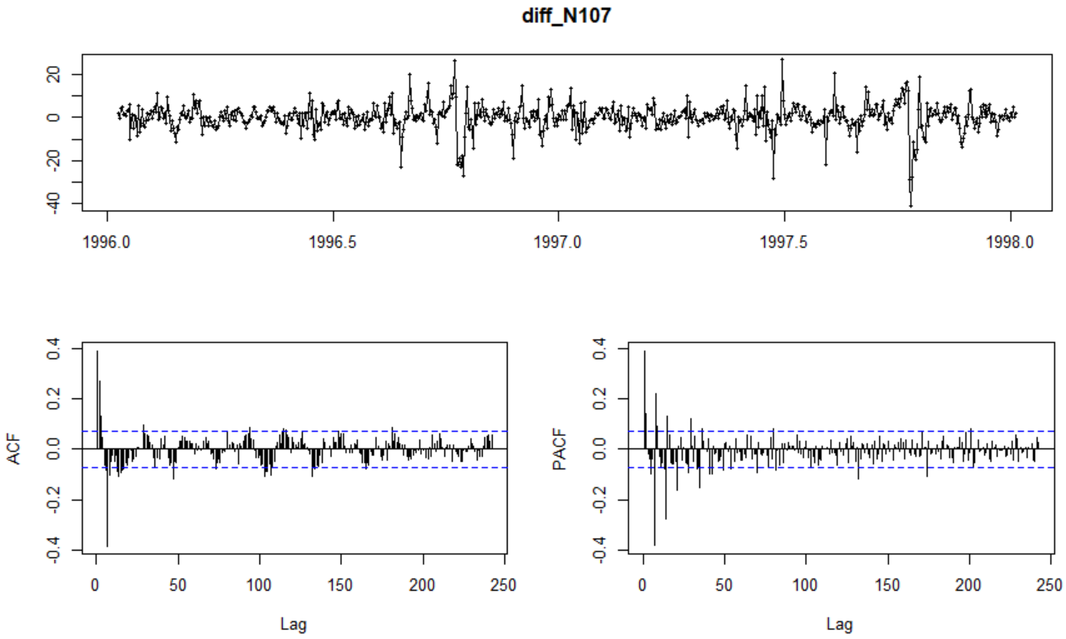


Figure 1.12

Next, in order to compare the performance of the two disassembled data models, I obtained the irregular component by splitting the two data with the two models. Then the ACF analysis of the four groups of irregular components. The results are shown in the figure below. And I can see that the analysis results are basically the same. This shows that for the two sets of data, the two disassembly methods have little difference, and they can be used as a reference for model establishment.

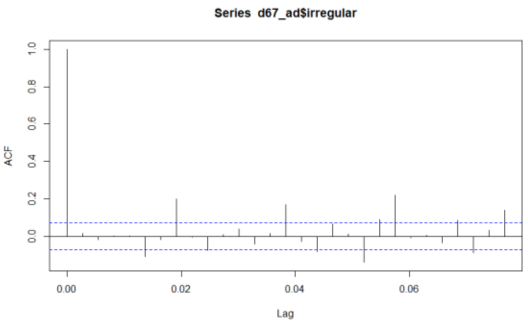


Figure 1.13

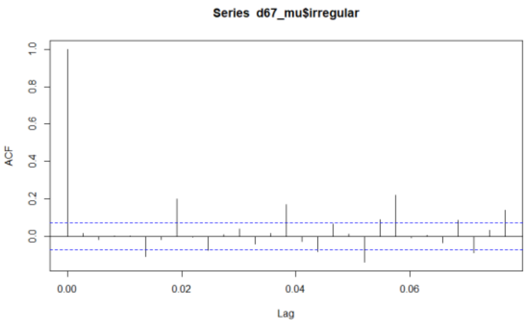


Figure 1.14

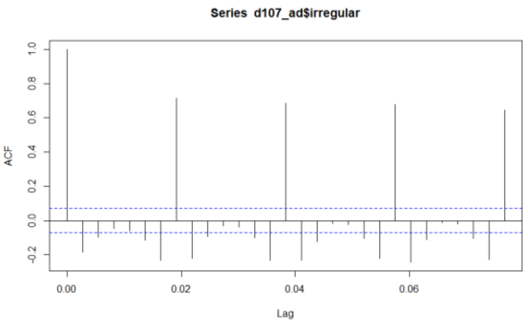


Figure 1.15

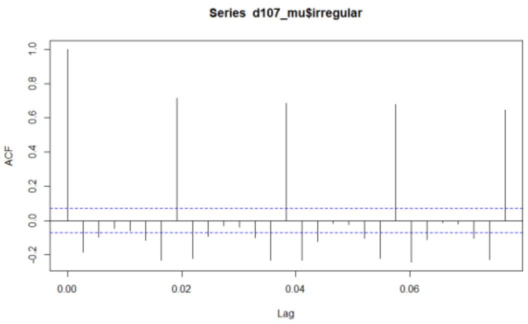


Figure 1.16

2. Model development

Before the model is built, I need to divide the data set into a training set and a test set with a ratio of 7:3. The training set is used for data modelling. The test set is used to test the accuracy of the predicted results produced by the established model. At the same time, in order to compare the relative advantages of the established model with the naïve prediction, I also need to make naïve predictions for the two training sets. Naïve predicts the results as follows (table 1 and figure 2.1 and 2.2). Through the image, it can be found that the naïve prediction only selects the last digit of the training set as the predicted value for prediction, and the prediction result is not ideal.

	N67	N107
Naïve mean	26.604	17.645

Table 2.1 Naïve forecast

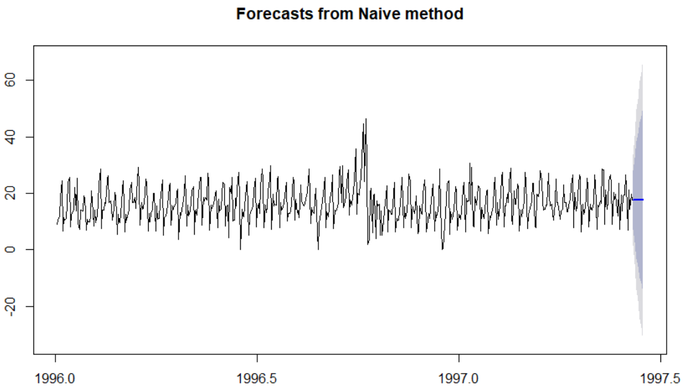


Figure 2.1 N67 naïve forecast

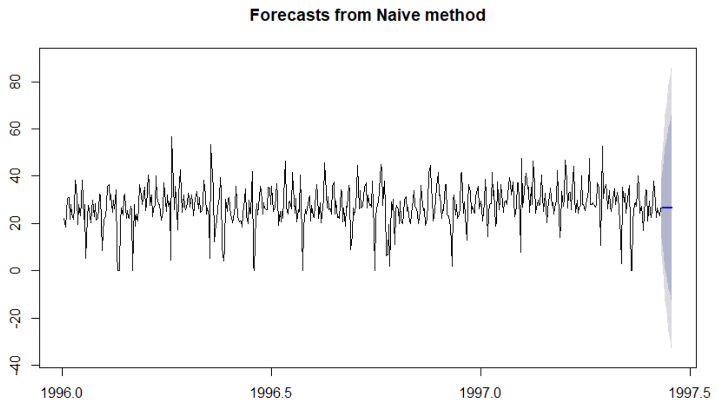


Figure 2.2 N107 naïve forecast

2.1 Exponential smooth

In order to get better prediction results, I try to use the exponential smoothing model for prediction. Since the components required for data fitting can be divided into error, trend, and season, I can adjust the combination of these components and the way of combining them to get different exponential smooth models. First, I use ets function. By adjusting the parameters in the function to "ANN" and "MNN", I could get the SES model with additive error and multiplicative error. Next, I can further add the trend factor by adjusting the parameters. When adding the trend parameter, I can set whether the type of the trend is damped. At the same time, the way to add the trend could also be either addictive or multiplicative. Finally, the season factor can also be added to the model by adjusting the parameters. Since the ets function helps us get the best alpha value, I don't need to do a lot of testing to get the corresponding value. All I need to do is to adjust the parameters provided when the model is built, and add error, trend, and season to the model for testing. After obtaining the required model, the best model can be found by testing the error rate with test set.

For the N67 dataset, since the dataset has data with a value of 0, the multiplicative model does not apply to this dataset. So, I will only discuss additive-related models here. In our manual modelling process, I add error, damped trend, non-damped trend and season to the model one by one, I can get 6 models. The AIC values of these six models are as follows. From the AIC value I can see that the ANN model performs best. That is to say, for the ES model, when the model only considers the additive error, the model is most suitable for the data set.

	ANN	AAdN	AAN	ANA	AAA	AAdA
AIC	5471.532	5477.691	5476.912	5490.297	5496.955	5498.08

Table 2.2

Similarly, I can automatically get the corresponding best model by adjusting the model parameter to ZZZ. And the best model I get from the function is the same model as the best model I got manually. This model takes additive error as the only variable, and the alpha value of this model is 0.0265.

Next, I analyse the residual of the model. Ideally, the model's residual should be white noise, and all valid information should be included in the model. But in general, there is still valid information left in the residuals. For the ES model of the N67, I firstly draw the residual histogram of the model. I can see that in the histogram, the residual of the model is not a standard normal distribution. In the histogram, the fat tail appears at the right end of the residual distribution. To further validate our idea, I use QQ plot to plot the distribution of the residuals against the standard normal distribution. I can see that the residual data in the model is not completely white noise in the ideal state. This shows that the model does not contain all the information I expect. Next, I draw the ACF plot and the PACF plot of the residual. Through the chart, I can clearly see that there is still some seasonal information in the residual. This shows that the exponential smoothing model does not explain the seasonality very well.

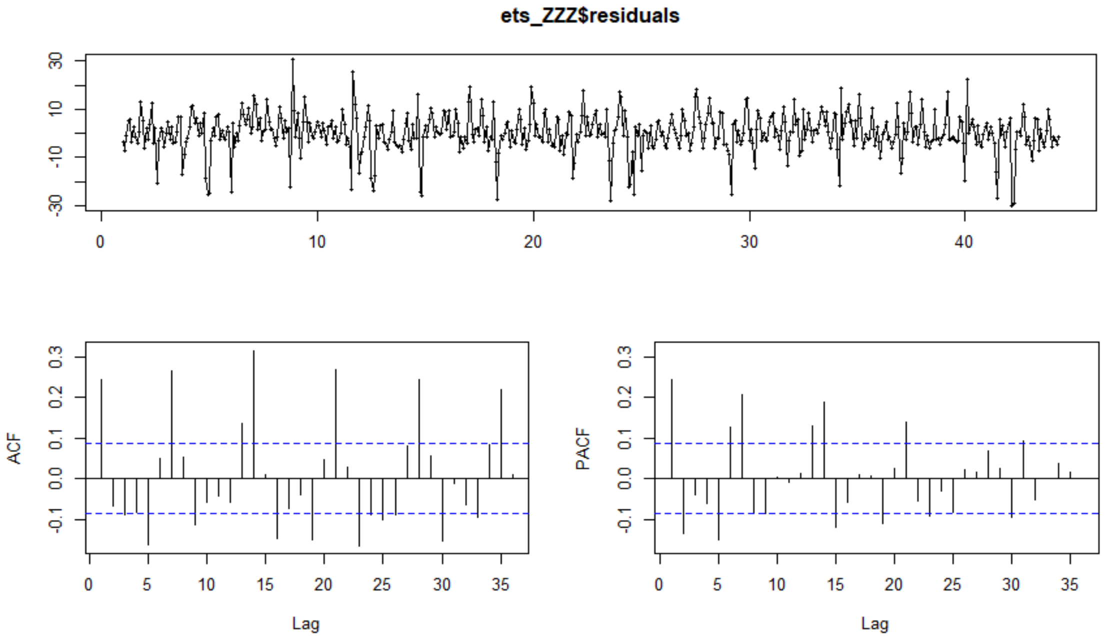
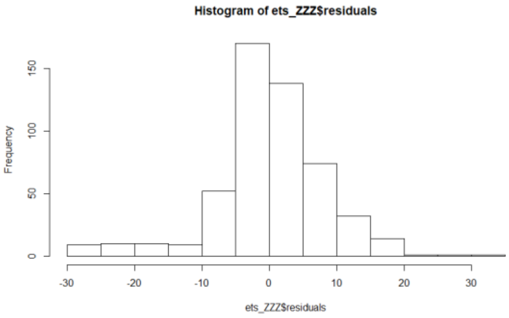


Figure 2.3 N67 residual ACF&PACF



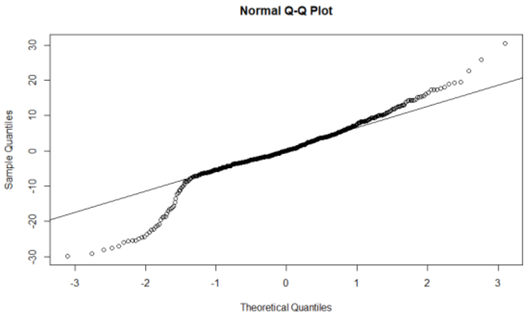


Figure 2.4 N67 residual

Figure 2.5 N67 residual

For the N107 data set, the steps of manually deriving the model are essentially the same as those for N67. Since there is still data equal to 0 in the N107 data set. So, for the n107 dataset, I still can't use the multiplicative model. In the process of manual modelling, I can get six corresponding models by adding errors, trend, season and other components to the model. Based on the AIC values obtained for each model, I can see that, like the previous model, the ANN performed better.

	ANN	AAdN	AAN	ANA	AAA	AAdA
AIC	5217.172	5220.473	5220.473	5239.153	5244.288	5242.773

Table 2.3

By setting the model parameter to ZZZ, the R software will automatically filter out the best model for us. The best model that is automatically filtered is consistent with the best model results I manually screened. The alpha value corresponding to the model is 0.0196.

Next, I will analyse the residuals of this model. First, I will draw the histogram of the residuals. From the histogram I see that for the N107 dataset, the residual of this model is closer to the normal distribution. However, there is still a fat tail. From the QQ plot of the residual and the standard normal distribution, this data basically conforms to the normal distribution. But this does not mean that the residual does not contain any useful information. After I draw the residual ACF and PACF diagrams, I can see that there is still a strong seasonality in the residuals through the images in the figure. This shows that the seasonality of the data is not well explained by the model.

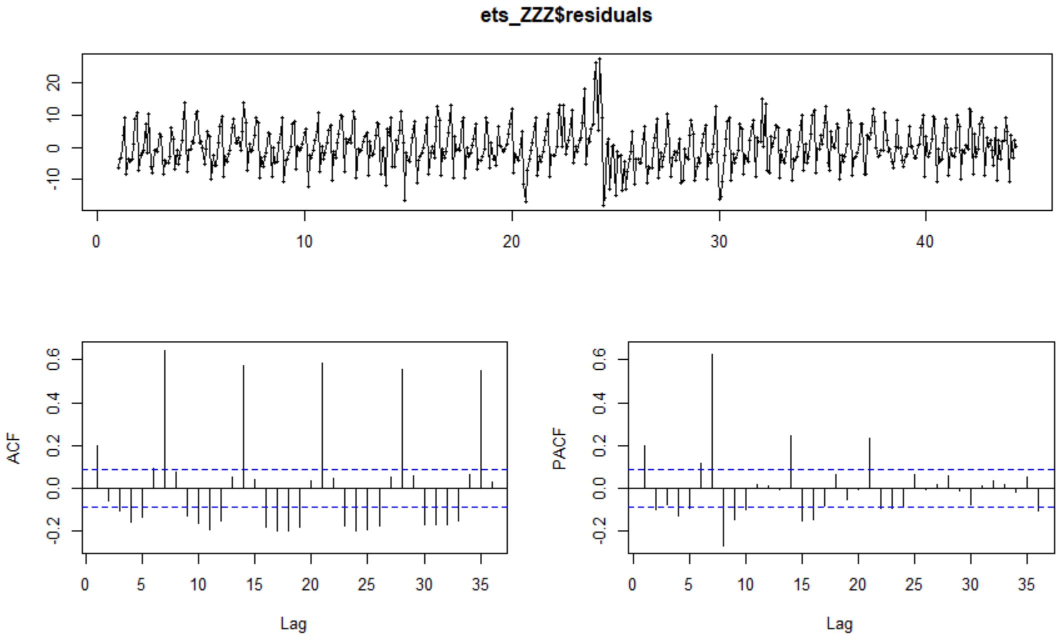
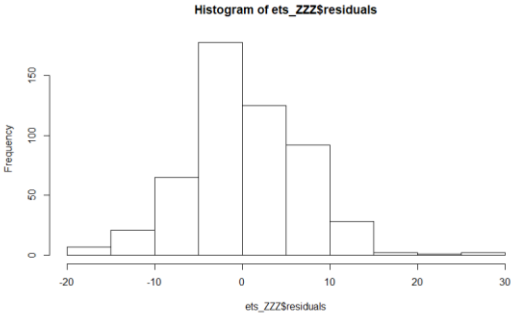


Figure 2.6 N107 residual



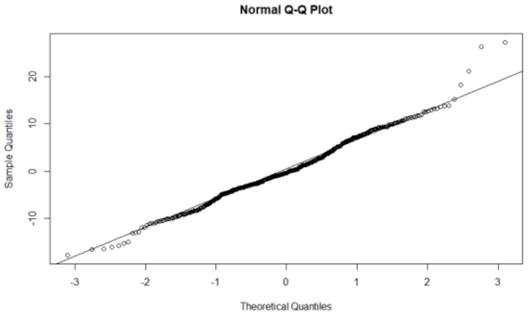


Figure 2.7 N107 residual

Figure 2.8 N107 residual

2.2 ARIMA

First, because the ARIMA model requires data stationary, I need to test the data for stationarity before I model the data. In this report, the stationarity test tool I used is the KPSS test. For the N67 dataset, I first perform a KPSS test on the training set of the N67 data. According to the test results (appendix 3), I know that this data is not stationary. In order to make the data smooth, first I make a first-order difference. Then I performed KPSS test on the first-order difference data and found that the data reached a smooth state after the first-order difference (appendix 4). I draw the ACF map and the PACF graph of the differential data. Through the figure 2.9 I can see that the data after the differencing still has strong seasonality and can identify the order of the AR is 6. Next, since our data is calculated in days, I try to lag the data as 7 Differential, then the data after the difference is analysed by ACF and PACF. After the data passes the difference of 7 lag, the seasonality of the data disappears in the ACF graph. This shows that I have found the right seasonality of the train data.

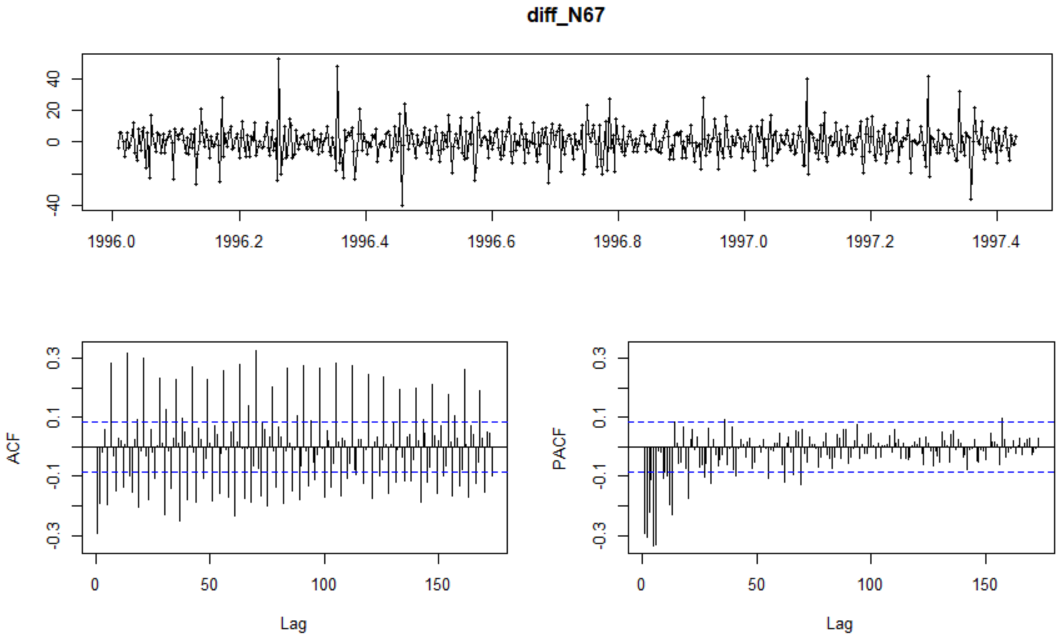


Figure 2.9

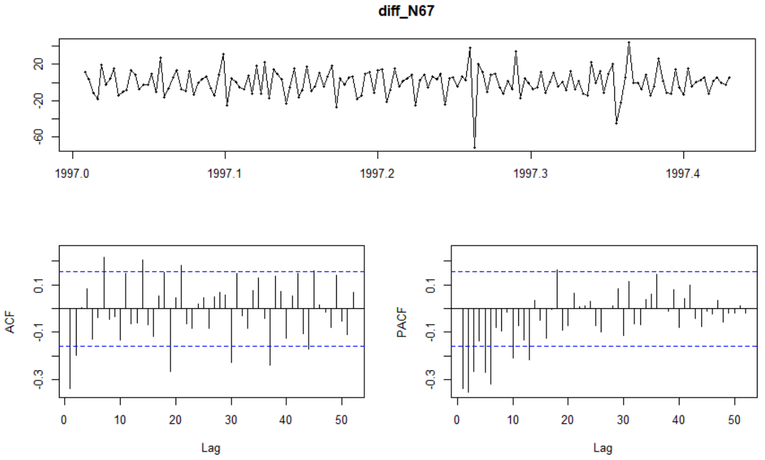


Figure 2.10

The manual modelling method I use is Box-Jenkins Methodology. Based on the previous analysis, I can determine that the model to be tested is ARI(6,1)(0,1,0)[7]. After testing the model, the residuals obtained by our model are as figure 2.11. According to the ACF graph of the residual, I can see that the order of the MA of the residual should be 1. So I need to continue adding parameters in the next step, the model should be SARIMA(6,1,1)(0,1,0) [7]. The figure 2.12 below shows the residual obtained after the test. From the figure 2.12, I can observe that the ACF reversal starts from 2 in the ACF diagram, and the peak appears at lag=7. So the model I next try is SARIMA(6,1,1)(0,1,2)[7]. According to the ACF plot of the residual (figure 2.13), I can see that the residual is close to white noise. Through the histogram of the model residual and the QQ plot, it can be found that the residual of the model has a fat tail, which indicates that there are some data components in the residual that cannot be explained by the model. The AIC value of this model is 3480.395. The coefficient of the model is showed in the table 2.4.

Ar1	Ar2	Ar3	Ar4	Ar5
0.25157438	0.05193240	-0.09131516	-0.01467513	-0.01118426
Ar6	Ma1	Sma1	Sma2	
0.01165665	-0.96469584	-1.01478022	0.01480029	

Table 2.4

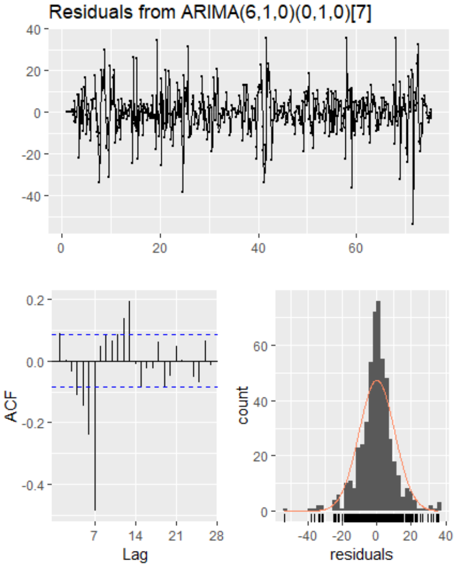


Figure 2.11

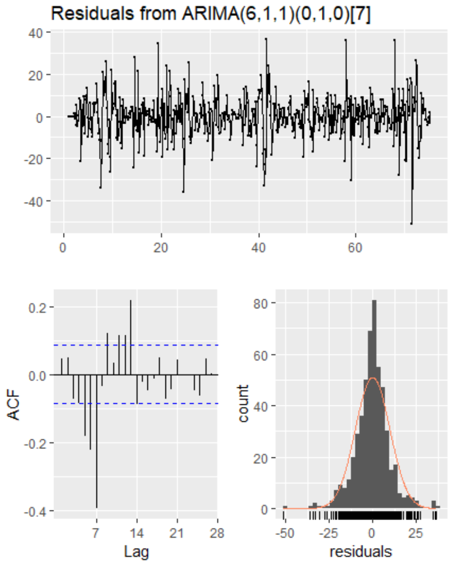


Figure 2.12

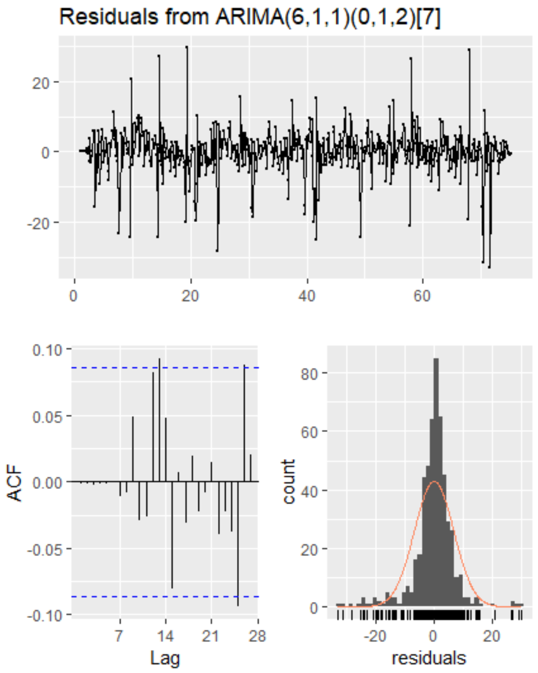


Figure 2.13

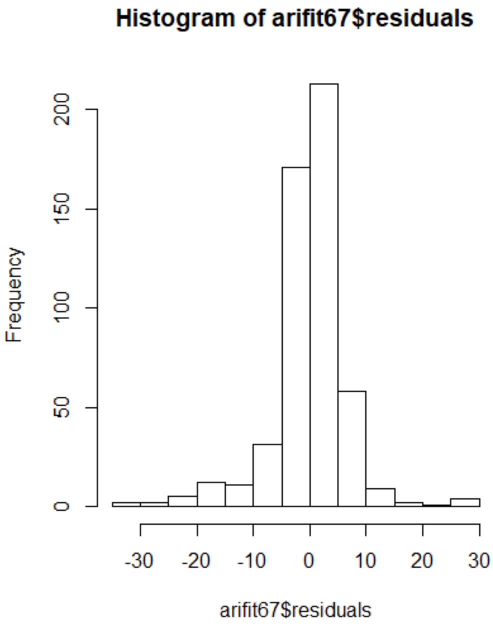


Figure 2.14

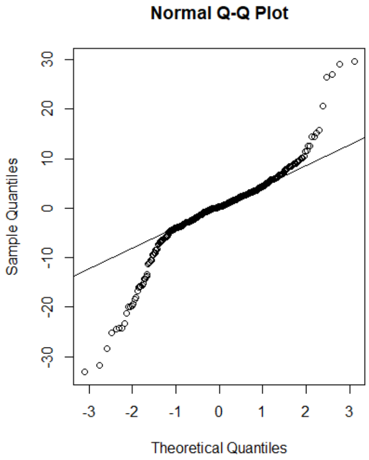
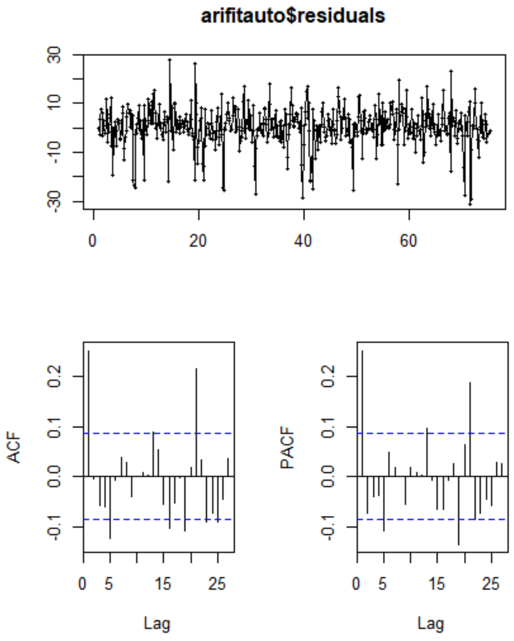


Figure 2.15

Next, the best model I get using the automated modelling tool is ARIMA(0,1,1)(0,0,2)[7]. However, the corresponding ACF and PACF graphs show that there is still valid information in the residual. The ACF and PACF of the residual of this model exceed the confidence interval when lag=1. The AIC value of this model is 3634.084. The QQ plot of the residual is shown below, and it is obvious that the interpretation effect is worse than the manually established model. The coefficient of the model is showed in the table 2.5.

Ma1	Sma1	Sma2
-0.9889	0.1633	0.2206

Table 2.5



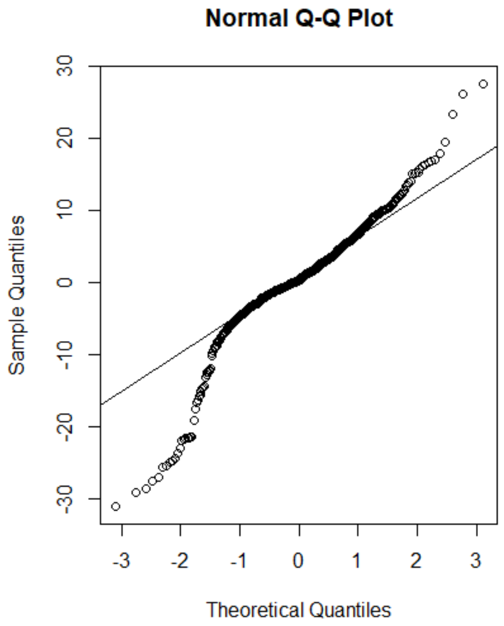
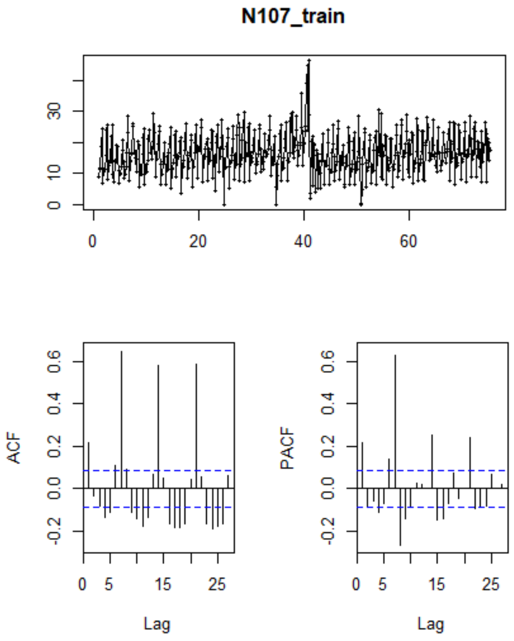


Figure 2.16

Figure 2.17

For the N107 dataset, I use a similar process for analysing. First, I performed a stationarity test on the N107 training set. The test results (appendix 5) show that the training set of N107 is stationary. Next, I draw the ACF and PACF diagrams of the N107 training set. The N107 training set shows a certain seasonality. Next, I make a difference of lag equal to 7 for the N107 training set. After the scoring, I observed the seasonal disappearance by observing the ACF and PACF maps of the N107 training set, indicating that I found the correct lag order.



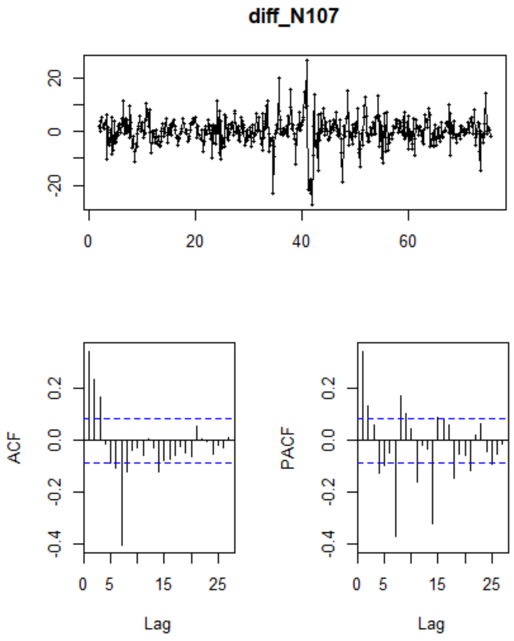


Figure 2.18

Figure 2.19

According to Box-Jenkins Methodology, our first model tested is ARIMA (2,0,0) (0,1,0), and the residuals were obtained as follows. From the figure 2.20, I can know that for the MA model, there is no corresponding information in the residual. So, I can keep the order of the MA to zero. At the same time, I also found a spike in lag=7. Next, I test ARIMA (2,0,0) (7,1,0). The new model residuals are as figure 2.21. I can see that the new model ACF and PACF graph performance are close to white noise. By plotting the QQ plot and the histogram of the model residuals, I can see that compared to the standard normal distribution. The distribution of model residuals has a fat tail. This shows that there is still valid information in the residual and the model estimate is too high. The AIC of this model is 2908.664. The coefficient of the model is showed in the table 2.6

Ar1	Ar2	Sar1	Sar2	Sar3
0.3138754	0.1741133	-0.7185188	-0.608838	-0.4043277
Sar4	Sar5	Sar6	Sar7	
-0.3441025	-0.2346628	-0.1893583	-0.1235131	

Table 2.6

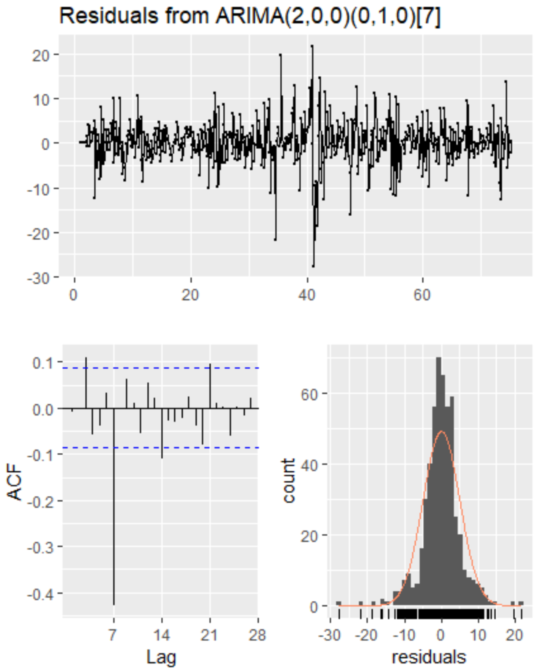


Figure 2.20

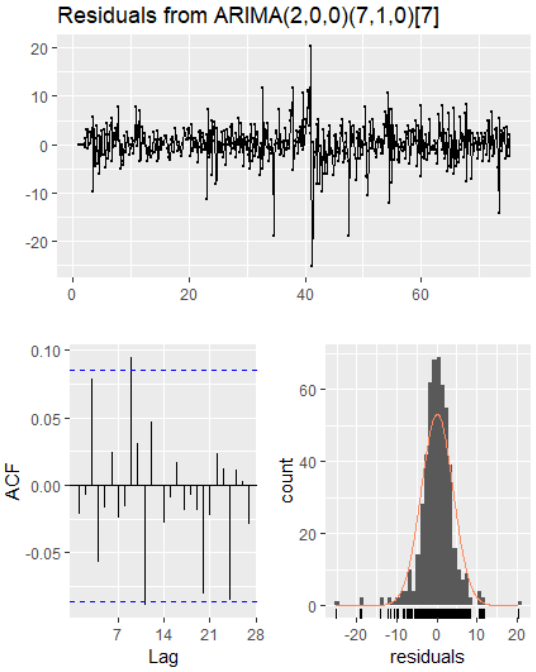
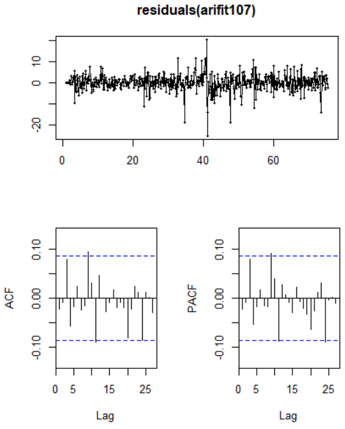
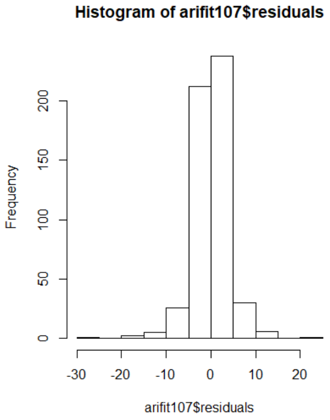


Figure 2.21





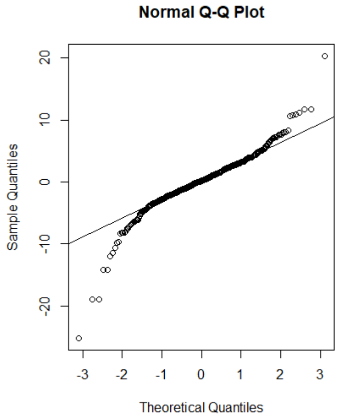


Figure 2.22

Figure 2.23

Figure 2.24

Through automatic modelling, the resulting model is ARIMA (1,0,1) (1,1,0). The analysis of the residuals is as follows. I can see that this model has some spikes in the ACF graph that exceed the confidence interval. This shows that there is some information that the model cannot explain. The distribution of residuals also has a fat tail phenomenon. At the same time, the AIC value of this model is 3015.35. The overall performance is not as good as the results of manual model. The coefficient of the model is showed in the table 2.7.

Ar1	Ma1	Sar1
0.6736	-0.3494	-0.4376

Table 2.7

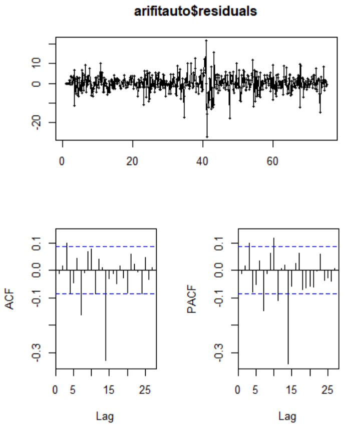


Figure 2.25

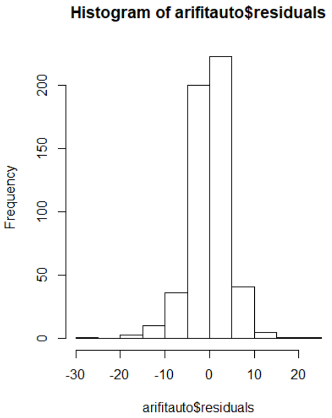


Figure 2.26

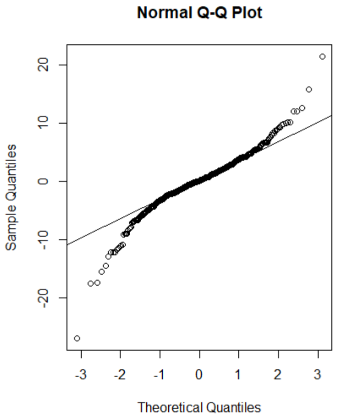


Figure 2.27

2.3 Regression

When performing a regression, I use the seasonal dummies, autoregressive Lags, and time trend as independent variables for fitting. In the previous analysis, both the N67 and N107 data sets showed a certain degree of seasonality. This seasonal length is at 7. Therefore, a total of 7 seasonal dummy is prepared. For autoregressive lags, I have prepared a total of 7 orders of lag data.

Firstly, in order to accommodate the length of the seasonal dummy. I adjust the length of the training set to 518 and the length of the test set to 217. Before the fitting, I draw the ACF and PACF maps of the new training set, and I can find that the data has a seasonality with a length of 7. This confirms that I have prepared enough dummy data. Then I combine the N67 training set with the 7 seasonal dummies to form a new data set and transform the data set into a time series. Then, I use the lm () function to fit the data and see the relevant summary (appendix 6). By summarizing I can see that in these data, D2, D4, D5 and D6 passed the significance test. Then I will keep these four variables for the next step of fitting. Next, I add 7 Autoregressive Lags to the fitted data frame, and fit the four variables obtained from the previous fitting together with lags. The results are shown in appendix 7. In this fitting, only D2, D4 , D6, L1_N67, and L2_N67 passed the test, so these variables are retained. Next, I add a sequence from 1 to 518 as an indicator of time trend to the fitted data, and fit the five variables that Ire last fitted through the test with the time trend variable. The result of the fitting is as shown in appendix 8. As you can see from the chart, only D2, D4, D6, L1_N67 and l2

passed the test. So, I will fit again through all the variables tested, and I can get the final fitting function, the result is in appendix9.
The model we get is N67= 14.50409-2.84239*D2+ 8.84076*D4 -7.90517*D6+ 0.35917 *L1_N67+ 0.12012*L2_N67.

Next, I perform a residual analysis on the manually obtained fitting results. It can be seen from the ACF and PACF plots of the model residual that the residual of the fitted model is very close to white noise. Through the histogram of the residual graph and the QQ plot, I can see that the distribution of the residuals is very similar to the normal distribution, but there is a case where the distribution is left-biased. But overall the residual distribution approximates a normal distribution. At the same time, the AIC value of this model is 3467.906.

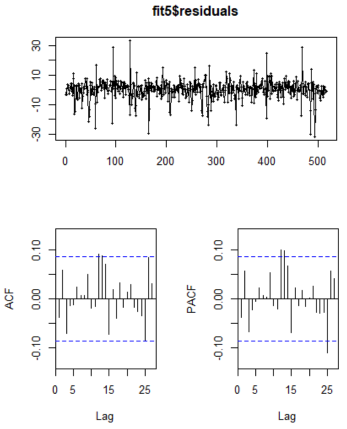


Figure 2.28

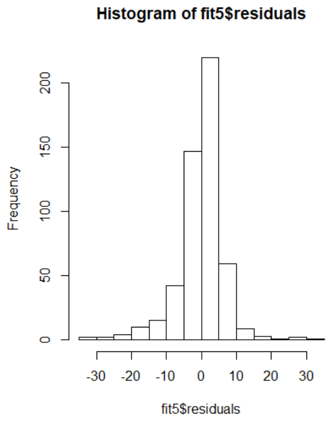


Figure 2.29

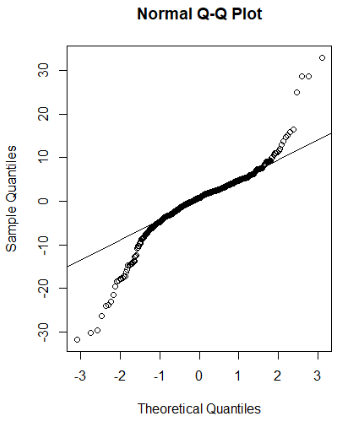


Figure 2.30

Next, I will use the step function to get the result of the automatic fitting. First, I get a model with an independent variable as a fixed intercept. Then all the independent variables are fitted to get a model. Then use the step function to automatically select the variable to get the model that is automatically fitted. A summary of this model is shown in the appendix 10. I can see that two of the variables in the model did not pass the significance test. The model's AIC value is 3547.988. In contrast, the residual distribution of the autofit model exhibits a right-biased shape. It can be seen from the QQ plot that the residuals of the auto-fit model and the normal distribution are not as similar as the manual model. At the same time, there are more spikes in the ACF graph and PACF that exceed the confidence interval. This means that there is some information in the residual that the model cannot explain. The model we get is shown in the table 2.8.

intercept	D4	L1_N67	D6
-3.353720147	31.229265284	0.264998993	15.647561200
D2	L2_N67	D3	D5
19.223897299	0.066229086	23.321029338	24.169245038
D7	D1	N67_trend	L3_N67
23.440882985	21.853125909	0.005438929	-0.080071584

Table 2.8

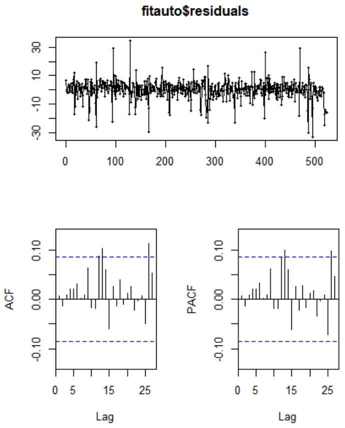


Figure 2.31

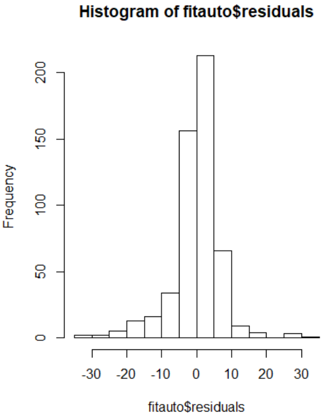


Figure 2.32

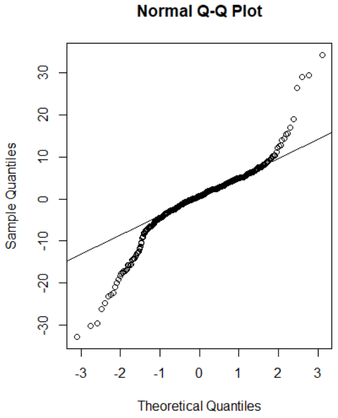


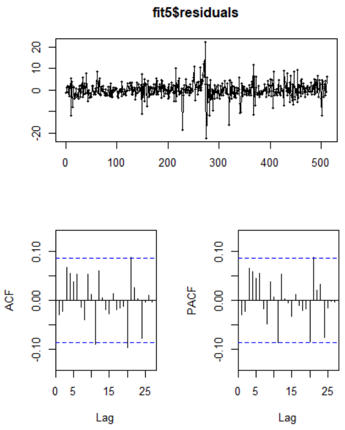
Figure 2.33

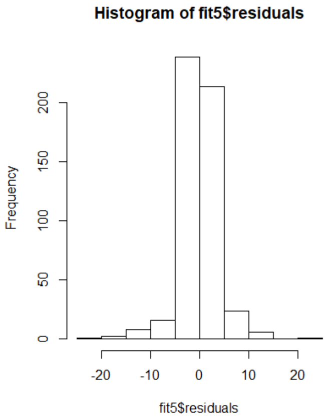
Similarly, for the N107 data set I used a similar approach to fit. According to the previous analysis, the fitting data required for N107 is the same as that of N67. First, the N107 was fitted using seasonal dummies and the results were fitted as a graph. In the first fit, D3, D4, D5, and D6 passed the significance test. And the summary is in appendix 11. I add new lags data to the fitted data and fit them again along with the data passed through the test in the first fit. The result of the second fit is as shown in the appendix 12. In the second fit, D4, D5, D6, L1_N107, L2_N107 L3_N107, L4_N107 and L7_N107 passed the significance test. Keep these variables and add the time trend to the fitted data for a third fit. The results of the third fit are shown in the appendix 13. I can see that the time trend did not pass the significance test. So, I will remove this argument. The model obtained by fitting the significant independent variables again is the model obtained by manual fitting. The final result is shown in the appendix 14. By testing the correlation between variables, it is found that there is no obvious correlation between each independent variable. The AIC value of the manual fit model is 2828.026. the model we get is shown in the table 2.9

intercept	D4	D5	D6
6.9190830	5.3864434	2.6906550	-12.0530307
L1_N107	L2_N107	L4_N107	L7_N107
0.3890872	0.2269110	-0.1275792	0.1198021

Table 2.9

Next, I analyse the residuals of the model. From the ACF diagram and the PACF diagram I can see that the residual ACF is basically consistent with white noise, but there are still some points beyond the confidence interval. In the histogram and QQ plot diagram, the model residual has a fat tail phenomenon. Explain that the model does not explain all the information perfectly.





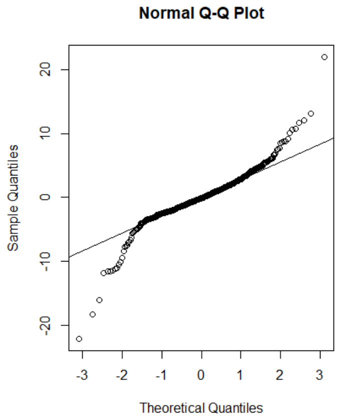
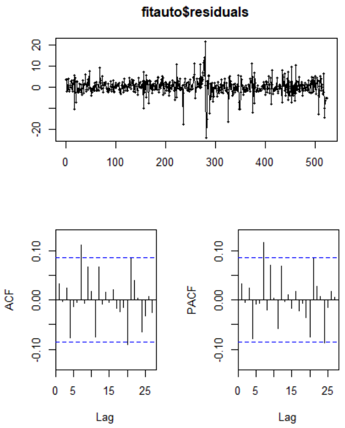


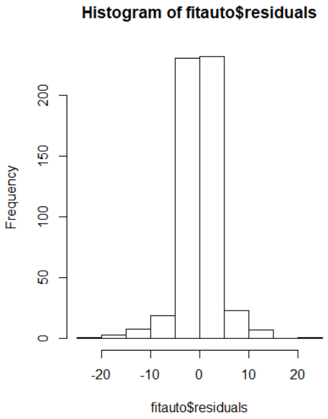
Figure 2.34

Figure 2.35

Figure 2.36

The method of automatic fitting of N107 data is exactly the same as that of N67. The resulting model results are shown below. The model has an AIC value of 2926.703. By observing the ACF and PACF plots, the residuals of the autofit model showed more spikes beyond the confidence interval compared to the results of the manual fit. At the same time, by observing the histogram of the residual and the QQ plot, the residual distribution of the automatic fitting model is also more different from the normal distribution. This shows that the manual fit model performs better in interpreting the data information.





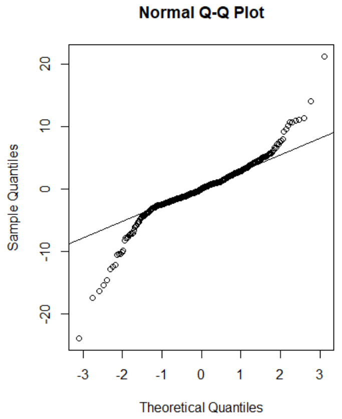


Figure 2.37

Figure 2.38

Figure 2.39

3.Model evaluation

I now build six models through various modelling methods. Next, I will use the rolling origin method to calculate the sMAPE of each model to compare the differences between the predictions and test sets generated by each model. The sMAPE indicator has the advantage of scale independent compared to other indicators. Moreover, this indicator is less biased than MAPE and is less possible to have zero denominator. But the downside of this indicator is that it is not intuitive enough. And compare these six models with the naïve prediction model using the GMRAE indicator. Finally, the best prediction model is obtained for prediction.

For the N67 dataset, the model derived manually from the exponential smoothing model is identical to the model automatically modelled. So, for this data set, I have built a total of five models. I use these five models to obtain the same prediction data length as the test set, and calculate the corresponding sMAPE and GMRAE. The results are in table 3.1.

N67	ES	ARIMA	ARIMA auto	regression	Regression auto
SMAPE	0.2562842	0.285787	0.2565249	0.303472	0.30468
GMRAE	0.9292107	0.776605	0.9507673	0.88126	0.84001

Table 3.1

According to the table 3.1, I can know that for ARIMA, although the results of automatic modelling have less error. But its predictions are too similar to naïve predictions, which makes the results of automated modelling meaningless. Moreover, the error rate of the manual model is not much different from the error of the automatic model. For regression, the results of manual modelling work better. Next, I will use these three models for prediction.

The prediction data and images of the exponential smoothing model are in table 3.2 and figure 3.1.

1	2	3	4	5	6	7
27.58096	27.58096	27.58096	27.58096	27.58096	27.58096	27.58096
8	9	10	11	12	13	14
27.58096	27.58096	27.58096	27.58096	27.58096	27.58096	27.58096

Table 3.2

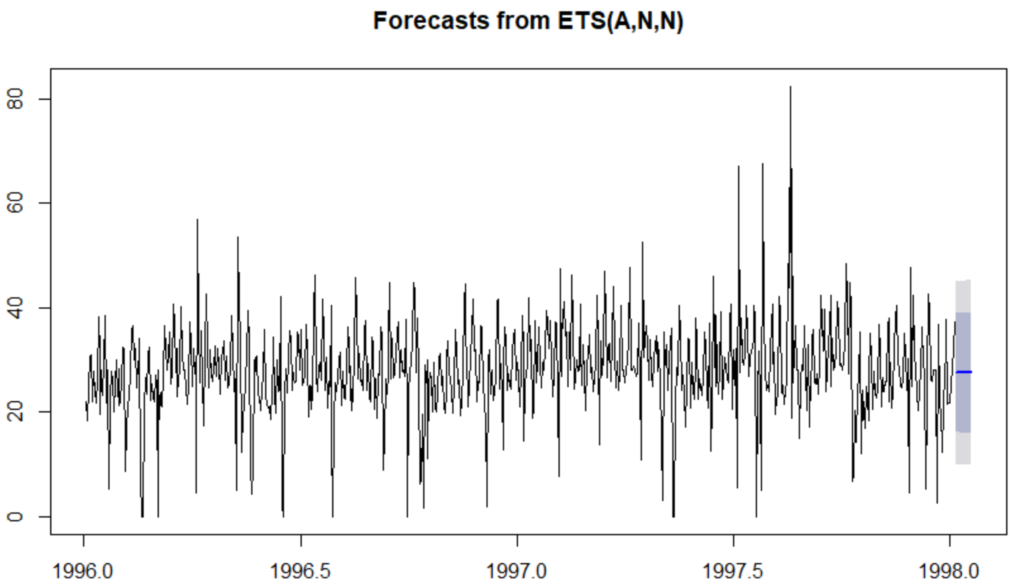


Figure 3.1

The predicted data and graphics of the ARIMA model are as follows.

1	2	3	4	5	6	7
36.04350	31.81332	22.55860	26.83313	26.01593	24.2625	27.53499
8	9	10	11	12	13	14
36.24374	31.85509	22.49140	26.8119	26.01774	24.29325	27.56565

Table 3.3

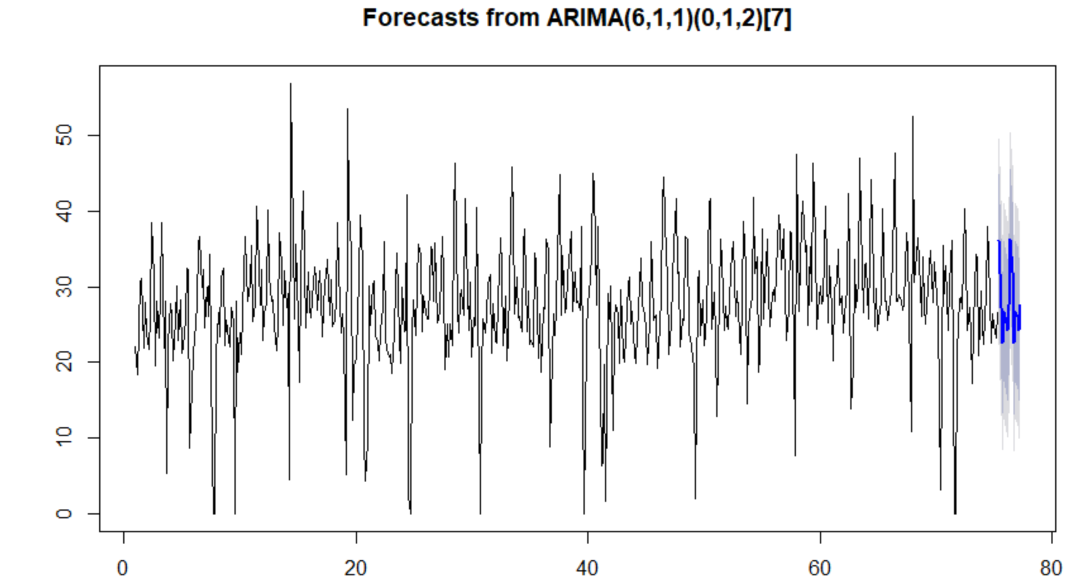


Figure 3.2

The prediction data and figures of the regression model are as follows

1	2	3	4	5	6	7
18.34511	21.69916	23.91428	33.94733	27.61970	20.85018	25.02138
8	9	10	11	12	13	14
26.83836	22.29169	24.95284	34.43661	30.02026	20.94884	24.33972

Table 3.4

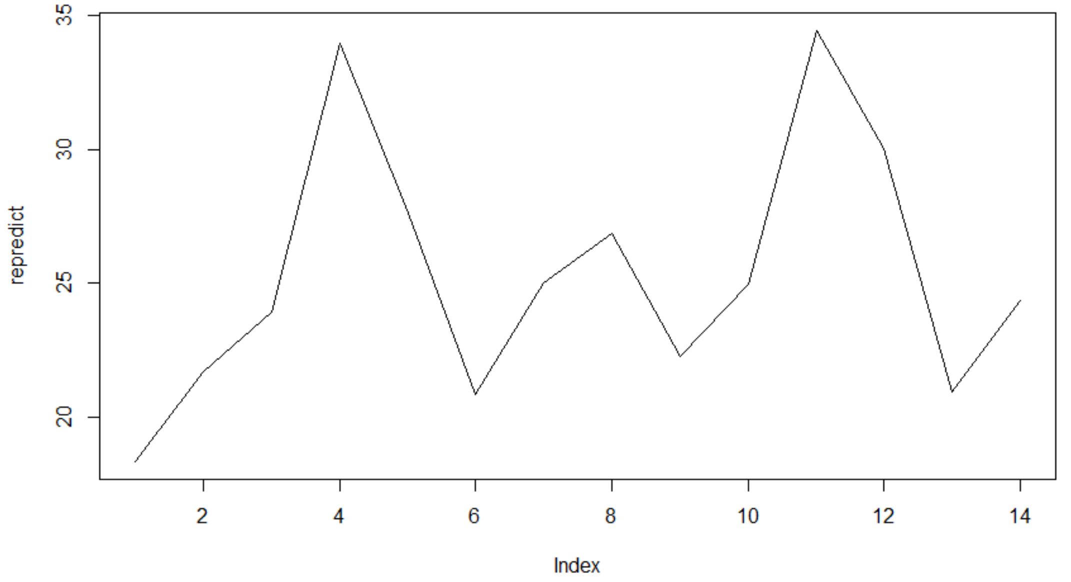


Figure 3.3

It can be seen from the above results that the results of the exponential smoothing model and the naïve prediction are not much different and cannot achieve the purpose of prediction. The regression model has a higher sMAPE value, but it can give relatively reasonable prediction results. However, the error rate of regression is larger than that of the ARIMA model. So for the N67 model, the best model should be ARIMA(6,1,1)(0,1,2).

For the N107 data, the results obtained manually and automatically due to the exponential smoothing model are the same. Therefore, I have established a total of five models as an alternative to the best model. By comparing with the naïve prediction and by the error analysis of the rolling origin, I can get the sMAPE and GMRAE of the five models as follows.

107	ES	ARIMA	ARIMA auto	regression	Regression auto
SMAPE	0.4277951	0.49214	0.4987292	0.48775	0.48807
GMRAE	0.621542	0.53215	0.6197502	0.52734	0.5386

Table 3.5

From the above results, it can be known that for the ARIMA model, the results obtained manually are better. For regression, the difference between the two models is not large, but the model error obtained manually is smaller and performs better than the naïve model. Next, I will use these three models for prediction for further analysis.

The predicted data and images of the exponential smoothing model are as follows

1	2	3	4	5	6	7
17.23525	17.23525	17.23525	17.23525	17.23525	17.23525	17.23525
8	9	10	11	12	13	14
17.23525	17.23525	17.23525	17.23525	17.23525	17.23525	17.23525

Table 3.6

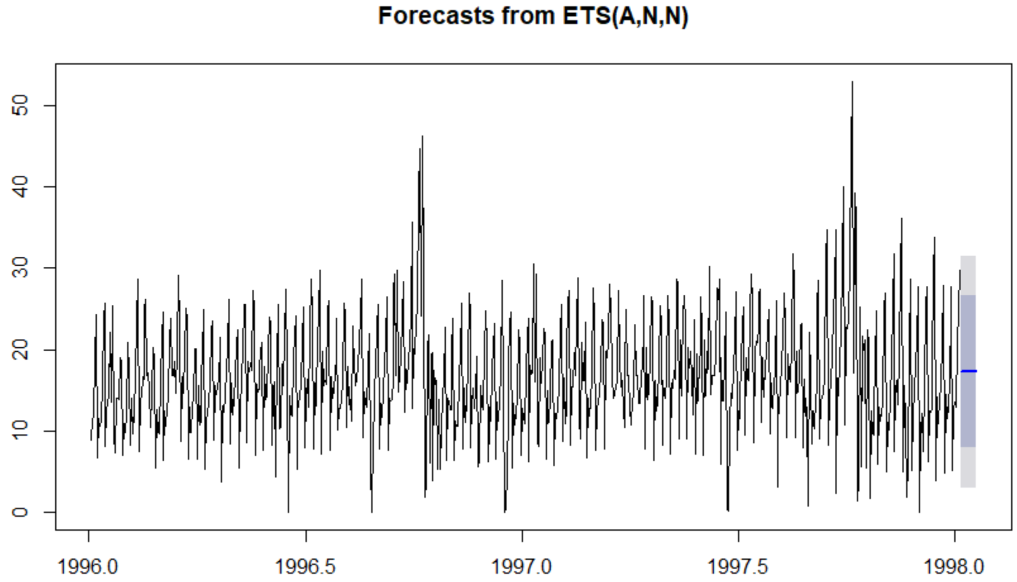


Figure 3.4

The predicted data and images of the ARIMA model are as follows.

1	2	3	4	5	6	7
21.44738	5.77785	18.41018	16.99461	17.15581	30.59186	16.00342
8	9	10	11	12	13	14
29.27868	8.75897	8.07814	18.94797	15.858988	18.693005	22.548

Table 3.7

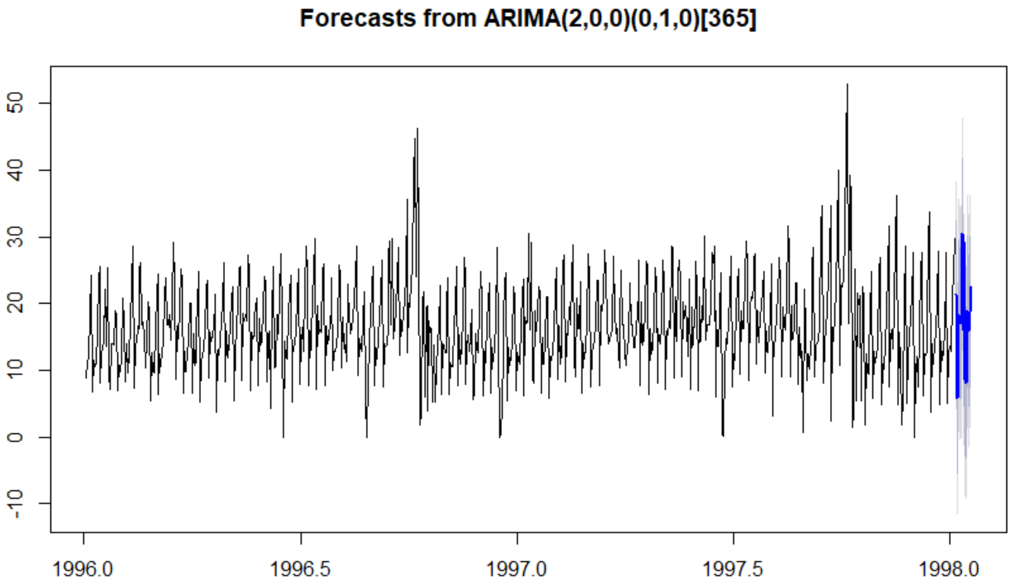

Forecasts from ARIMA(2,0,0)(0,1,0)[365]

Figure 3.5

The prediction data and images of the regression model are as follows.

1	2	3	4	5	6	7
7.891037	10.88256	13.39484	18.9641	17.82692	6.91766	13.13452
8	9	10	11	12	13	14
11.945625	12.193311	14.678824	21.837343	24.048961	9.444926	15.174289

Table 3.8

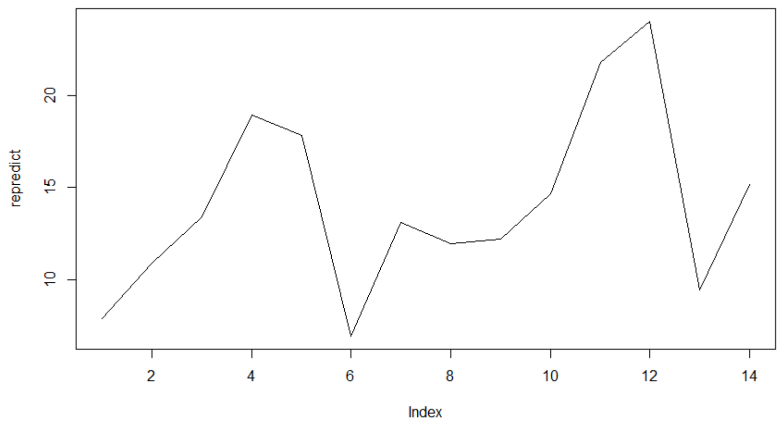


Figure 3.6

Based on the above results, it is clear that the manually established regression is the best model among these models. The prediction results of the ES model are not as significant as the regression results predicted by naïve. According to our analysis above, the forecast changes given by ES are too flat. This makes the ES model not a good predictor in terms of prediction. For this data set, the ARIMA model performed Ill. However, the ARIMA model has a higher error rate than regression. And in these models, manual regression has the best performance compared to naïve prediction. In summary, for the N107 dataset, the best model I get is the manual regression model.

4.Conclusion

In summary, for a given two data sets, the models that best fit the respective data set are different. For the N67, the manual ARIMA synthesis seems to perform best. For the N107 data, the regression model has obvious advantages in all aspects. In terms of model selection, the deviation caused by the fixed prediction starting point can be well overcome by the method of rolling origin. By this method we can calculate the deviation of the model's estimator from the test set. This bias estimate is an less-biased estimate due to the nature of sMAPE. These two tools can more accurately evaluate the performance of the model through reasonable cooperation.

Through this assignment, I gained a new perspective on data analysis. I learned to decompose the data one by one before predicting the data, which provides useful ideas for analysing complex data in my future career. At the same time, I should learn to flexibly adjust each variable to get a better predictive model when I analyse data next time.

In addition, I also found that depending on the division of the training set and the test set, for some time series with weak stability, the original time series may be unstable and the training set may be stable. For example, in the N107 data, we found that the data was not stable when analysing the overall data. But when we divided the training set and the test set by a ratio of 7:3, the data of the training set was found to be stable data after KPSS test. This will have an impact on the quality of the established model. Therefore, the division ratio of the training set and the test set is a factor that needs to be carefully considered. In addition, the size of the sample will also have an impact on this phenomenon. If the sample size is large enough, the training set has a greater chance of containing the information needed for modelling.

Also, I found that in all the resulting models, the results of automatic software fitting were generally weaker than those of manual modelling. This shows that depending on the algorithm of the software, it is possible that the results obtained are not satisfactory.

Appendix

```
1. KPSS Test for Level Stationarity

data:  N67
KPSS Level = 0.91273, Truncation lag parameter =
6, p-value = 0.01


2. KPSS Test for Level Stationarity

data:  N107
KPSS Level = 0.52066, Truncation lag parameter =
6, p-value = 0.03701


3. KPSS Test for Level Stationarity

data:  N67_train
KPSS Level = 0.78424, Truncation lag parameter =
6, p-value = 0.01


4. KPSS Test for Level Stationarity

data:  diff_N67
KPSS Level = 0.010811, Truncation lag parameter =
6, p-value = 0.1


5. KPSS Test for Level Stationarity

data:  N107_train
KPSS Level = 0.31901, Truncation lag parameter =
6, p-value = 0.1


6. Call:
lm(formula = N67 ~ ., data = redata)

Residuals:
    Min      1Q  Median      3Q     Max 
-32.936  -2.327   0.896   3.751  26.735 

Coefficients: (1 not defined because of singularities)
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  26.6360     0.8363  31.849  < 2e-16 ***
D1           -0.8198     1.1827  -0.693 0.488558    
D2           -2.5454     1.1827  -2.152 0.031854 *  
D3            0.7268     1.1827   0.615 0.539134    
D4            9.4304     1.1827   7.973 1.02e-14 ***
D5            5.0371     1.1827   4.259 2.45e-05 ***
D6           -4.3289     1.1827  -3.660 0.000278 ***
D7                NA         NA      NA       NA    
---
Signif. codes:  
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.194 on 511 degrees of freedom
Multiple R-squared:  0.2707,  Adjusted R-squared:  0.2622 
F-statistic: 31.62 on 6 and 511 DF,  p-value: < 2.2e-16
```

```
7. Call:
lm(formula = N67 ~ D2 + D4 + D5 + D6 + L1_N67 + L2_N67 + L3_N67 +
    L4_N67 + L5_N67 + L6_N67 + L7_N67, data = redata)

Residuals:
    Min      1Q  Median      3Q     Max
-32.793  -2.438   0.802   3.466  33.413

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 13.960941   1.978565   7.056 5.58e-12 ***
D2          -2.764784   1.146727  -2.411  0.01626 *
D4           9.237477   1.106626   8.347 6.52e-16 ***
D5           1.870931   1.216407   1.538  0.12465
D6          -7.561923   1.133647  -6.670 6.62e-11 ***
L1_N67       0.331280   0.043149   7.678 8.26e-14 ***
L2_N67       0.132577   0.044040   3.010  0.00274 **
L3_N67      -0.024027   0.045206  -0.532  0.59530
L4_N67      -0.006403   0.041022  -0.156  0.87602
L5_N67       0.014158   0.045617   0.310  0.75641
L6_N67       0.058561   0.043409   1.349  0.17792
L7_N67      -0.020853   0.043534  -0.479  0.63214
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.172 on 513 degrees of freedom
Multiple R-squared:  0.3653,  Adjusted R-squared:  0.3517
F-statistic: 26.84 on 11 and 513 DF,  p-value: < 2.2e-16


8. Call:
lm(formula = N67 ~ D2 + D4 + D6 + L1_N67 + L2_N67 + N67_trend,
    data = redata)

Residuals:
    Min      1Q  Median      3Q     Max
-31.689  -2.664   0.805   3.595  34.431

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 14.3321733  1.3537050  10.587  < 2e-16 ***
D2          -2.8360637  0.9306678  -3.047  0.00243 **
D4           8.8434620  0.9353111   9.455  < 2e-16 ***
D6          -7.8848952  1.0073026  -7.828 2.82e-14 ***
L1_N67       0.3584820  0.0377513   9.496  < 2e-16 ***
L2_N67       0.1187557  0.0405501   2.929  0.00355 **
N67_trend    0.0008509  0.0020783   0.409  0.68241
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.172 on 518 degrees of freedom
Multiple R-squared:  0.359,   Adjusted R-squared:  0.3516
F-statistic: 48.35 on 6 and 518 DF,  p-value: < 2.2e-16

9. Call:
lm(formula = N67 ~ D2 + D4 + D6 + L1_N67 + L2_N67, data = redata)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-31.511  -2.725   0.834   3.647  34.334 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 14.50409    1.28590  11.279  < 2e-16 ***
D2          -2.84239    0.92979  -3.057  0.00235 ** 
D4           8.84076    0.93454   9.460  < 2e-16 ***
D6          -7.90517    1.00528  -7.864 2.18e-14 ***
L1_N67       0.35917    0.03768   9.531  < 2e-16 ***
L2_N67       0.12012    0.04038   2.975  0.00307 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.166 on 519 degrees of freedom
Multiple R-squared:  0.3588,	Adjusted R-squared:  0.3526 
F-statistic: 58.08 on 5 and 519 DF,  p-value: < 2.2e-16

10. Call:
lm(formula = N67 ~ D4 + L1_N67 + D6 + D2 + L2_N67 + D3 + D5 + 
    D7 + D1 + N67_trend + L3_N67, data = redata)

Residuals:
    Min      1Q  Median      3Q     Max 
-32.887  -2.112   0.639   3.018  32.002 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -3.353720   2.783011  -1.205  0.22873    
D4          31.229265   2.993850  10.431  < 2e-16 ***
L1_N67       0.264999   0.043408   6.105 2.03e-09 ***
D6          15.647561   3.155514   4.959 9.66e-07 ***
D2          19.223897   2.977547   6.456 2.49e-10 ***
L2_N67       0.066229   0.044722   1.481  0.13925    
D3          23.321029   2.971780   7.847 2.49e-14 ***
D5          24.169245   3.129844   7.722 6.04e-14 ***
D7          23.440883   3.075693   7.621 1.22e-13 ***
D1          21.853126   3.010359   7.259 1.45e-12 ***
N67_trend    0.005439   0.002052   2.650  0.00829 ** 
L3_N67      -0.080072   0.042195  -1.898  0.05830 .  
---
Signif. codes:  
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.785 on 513 degrees of freedom
Multiple R-squared:  0.4319,	Adjusted R-squared:  0.4197 
F-statistic: 35.45 on 11 and 513 DF,  p-value: < 2.2e-16


11. Call:
lm(formula = N107 ~ ., data = redata)

Residuals:
     Min       1Q   Median       3Q      Max 
-17.9245  -2.0261  -0.1925   1.8624  31.2333 

Coefficients: (1 not defined because of singularities)
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  15.0817     0.4923  30.638  < 2e-16 ***
D1           -1.3636     0.6962  -1.959  0.05068 .  
D2           -1.1829     0.6962  -1.699  0.08990 .  
D3            2.2802     0.6962   3.275  0.00113 ** 
```

```
D4             7.9633     0.6962  11.439  < 2e-16 ***
D5             8.4688     0.6962  12.165  < 2e-16 ***
D6            -6.5906     0.6962  -9.467  < 2e-16 ***
D7                 NA         NA      NA       NA    
---
Signif. codes:  
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.235 on 511 degrees of freedom
Multiple R-squared:  0.5841,	Adjusted R-squared:  0.5793 
F-statistic: 119.6 on 6 and 511 DF,  p-value: < 2.2e-16
```

12. Call:

```
lm(formula = N107 ~ D3 + D4 + D5 + D6 + L1_N107 + L2_N107 + L3_N107 
+ 
    L4_N107 + L5_N107 + L6_N107 + L7_N107, data = redata)

Residuals:
     Min       1Q   Median       3Q      Max 
-24.1306  -1.6782   0.0416   1.8790  20.4346 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)   5.100727   0.960819   5.309 1.65e-07 ***
D3            2.456291   0.914680   2.685  0.00748 ** 
D4            7.259564   0.936377   7.753 4.87e-14 ***
D5            4.968382   0.965613   5.145 3.81e-07 ***
D6          -10.624628   0.870667 -12.203  < 2e-16 ***
L1_N107       0.329772   0.039536   8.341 6.84e-16 ***
L2_N107       0.211415   0.029597   7.143 3.14e-12 ***
L3_N107       0.074770   0.028558   2.618  0.00910 ** 
L4_N107      -0.057996   0.043248  -1.341  0.18051    
L5_N107       0.017283   0.041912   0.412  0.68024    
L6_N107       0.002475   0.044266   0.056  0.95543    
L7_N107       0.072752   0.041376   1.758  0.07929 .  
---
Signif. codes:  
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.919 on 513 degrees of freedom
Multiple R-squared:  0.6705,	Adjusted R-squared:  0.6634 
F-statistic: 94.88 on 11 and 513 DF,  p-value: < 2.2e-16
```

13. Call:

```
lm(formula = N107 ~ D4 + D5 + D6 + L1_N107 + L2_N107 + L3_N107 + 
    L4_N107 + L7_N107 + N107_trend, data = redata)

Residuals:
     Min       1Q   Median       3Q      Max 
-21.3965  -1.7931  -0.0875   1.7885  21.6851 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)   7.547246   0.879913   8.577  < 2e-16 ***
D4            5.280852   0.606589   8.706  < 2e-16 ***
D5            3.074095   0.708927   4.336 1.75e-05 ***
D6          -11.107810   0.764995 -14.520  < 2e-16 ***
L1_N107       0.329679   0.037109   8.884  < 2e-16 ***
L2_N107       0.169951   0.029417   5.777 1.32e-08 ***
```

```
L3_N107       0.051046   0.027567   1.852   0.0646 .  
L4_N107      -0.147152   0.027375  -5.375 1.17e-07 ***
L7_N107       0.145559   0.036155   4.026 6.54e-05 ***
N107_trend    0.001177   0.001143   1.030   0.3035    
---
Signif. codes:  
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.789 on 508 degrees of freedom
  (7 observations deleted due to missingness)
Multiple R-squared:  0.669,	Adjusted R-squared:  0.6631 
F-statistic: 114.1 on 9 and 508 DF,  p-value: < 2.2e-16


14. Call:
lm(formula = N107 ~ D4 + D5 + D6 + L1_N107 + L2_N107 + L4_N107 + 
    L7_N107, data = redata)

Residuals:
    Min      1Q  Median      3Q     Max 
-23.821  -1.793   0.006   1.955  20.685 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)   6.91908    0.85300   8.111 3.66e-15 ***
D4            5.38644    0.62249   8.653  < 2e-16 ***
D5            2.69065    0.72615   3.705 0.000234 ***
D6          -12.05303    0.78128 -15.427  < 2e-16 ***
L1_N107       0.38909    0.03720  10.459  < 2e-16 ***
L2_N107       0.22691    0.02893   7.843 2.54e-14 ***
L4_N107      -0.12758    0.02766  -4.613 5.02e-06 ***
L7_N107       0.11980    0.03660   3.273 0.001135 ** 
---
Signif. codes:  
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.973 on 517 degrees of freedom
Multiple R-squared:  0.6586,	Adjusted R-squared:  0.6539 
F-statistic: 142.5 on 7 and 517 DF,  p-value: < 2.2e-16
```