Project - Electric Power Consumption Analysis

1MS014 Analysis of Time Series

Aviral Jain
aviral.jain.8329@student.uu.se

Aditya Khadkikar
adkh8153@student.uu.se

July 1, 2025

1 Introduction

This project is focused on analyzing the electrical power consumption in Tetouan city, Morocco using time-series modeling techniques. The full dataset contains columns like date, time, temperature, wind speed, and the feature that is of focus for this project is the power consumption of 3 different zones. They have been aggregated for getting the total power consumption of city across all zones.

Each data-point corresponds to a recorded value, with an interval of 10 minutes. For this study we have chosen one month of Power Consumption data (corresponding to 4380 data points, i.e. 43,800 minutes), allowing us to capture both short-term fluctuations and if there is seasonality, or a pattern in cyclic behaviour. Modelling the series data can aid in providing more accurate and robust forecasting of electricity demand across the city, and help provide insight for which periods of the day, or week, are consumption-intensive, and e.g. conduct regulation measures in the power grids.

In this report, we tried a series of autoregressive integrated moving average (ARIMA) models to fit to the data, as well as tried seasonality-based modelling methods. As a last step, we used the better model out of the ones tried, to forecast the power consumption for the next 2 days, or 288 future timesteps.

The dataset chosen for this project can be found at [1].

2 Analysing the Data

First, we have checked the data weather it is stationary or not. Then, we took a differencing of 1 to make the data stationary. To confirm it we used the Dickey-Fuller test whose results are shown in Figure 3. Additionally, we logarithm-transformed the data, as the magnitude of the data was very high initially (in 10,000s).

Then we made ACF and PACF plots for the differenced data to find the most optimal AR/MA or both that would be best fit for the data. In this case AR(2) suits the best.

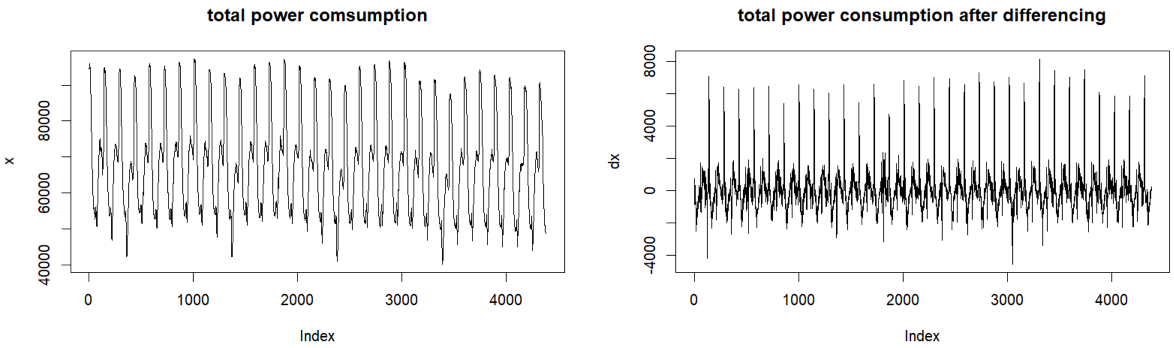


Figure 1: Actual data and Differenced Data for making data stationary.

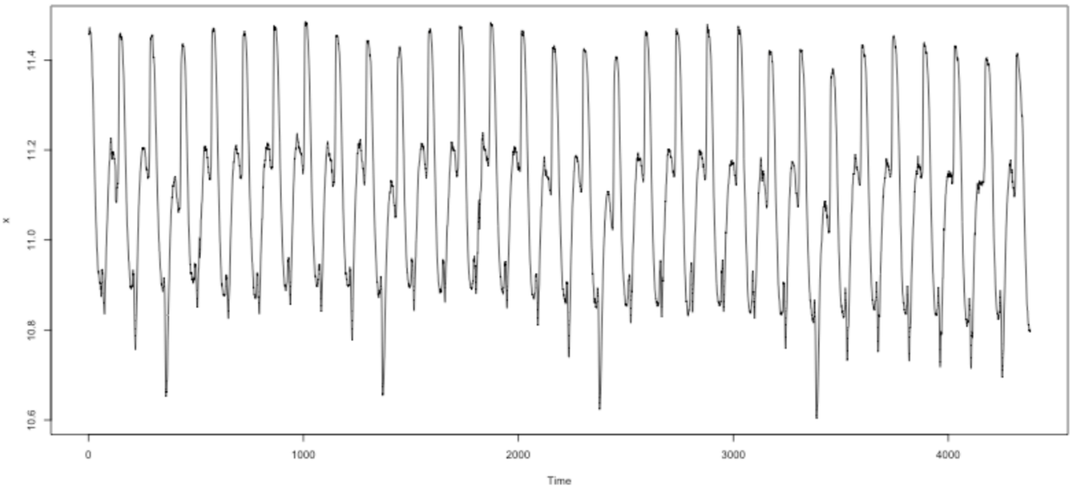
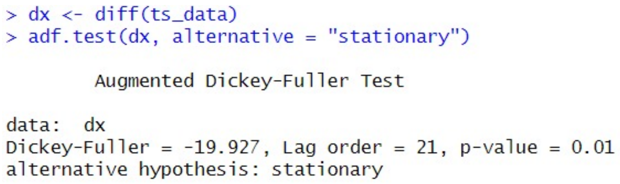


Figure 2: Logarithm-transformed time-series data of power consumption.



```
> dx <- diff(ts_data)
> adf.test(dx, alternative = "stationary")

        Augmented Dickey-Fuller Test

data:  dx
Dickey-Fuller = -19.927, Lag order = 21, p-value = 0.01
alternative hypothesis: stationary
```

Figure 3: dickey-fuller test results

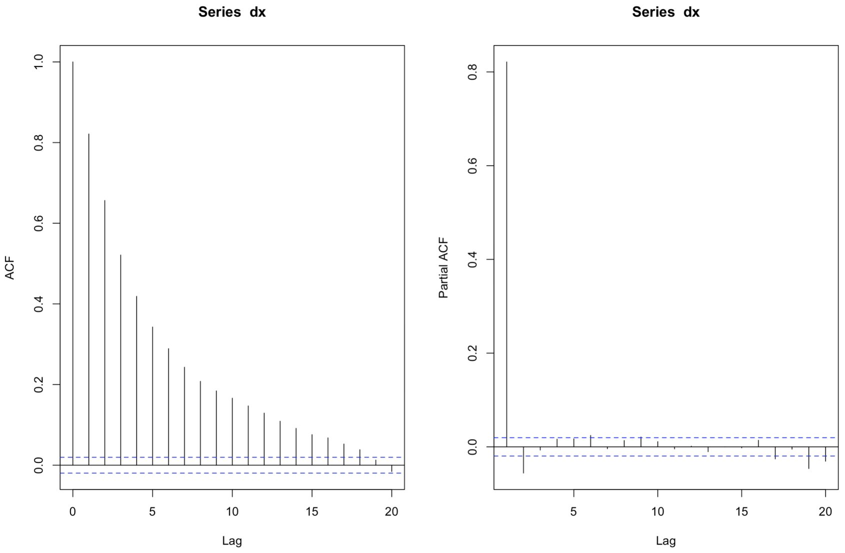


Figure 4: ACF/PACF of the Aggregated Power Consumption across all zones differenced by 1.

3 Model Fitting

From analyzing the ACF and PACF, the values decay slowly for the ACF graph, and cut off in the PACF graph at lag = 2, suggesting potentially an autoregressive (AR) 2.

We started by applying ARIMA(2,1,0) as we saw stationarity in data after 1 differencing, but after inspecting the data we have seen a strong seasonality after 144 points which depicts daily seasonality. Based on this, we enhanced the model by introducing seasonal components using a SARIMA structure with a seasonal period of 144.

3.1 AR(2) Model with diff=1

Firstly, we attempted to fit an AR(2) model with a difference of lag 1, or in other words, an ARIMA(2,1,0) model. The AIC was found to be -28315.21, and the log-likelihood was 14160.6. Despite these, it performed poorly in the p-values, when observed in the Ljung-Box plots, for very small lags (lag=5 onwards, the points were sometimes below, and/or too close to the cutoff).

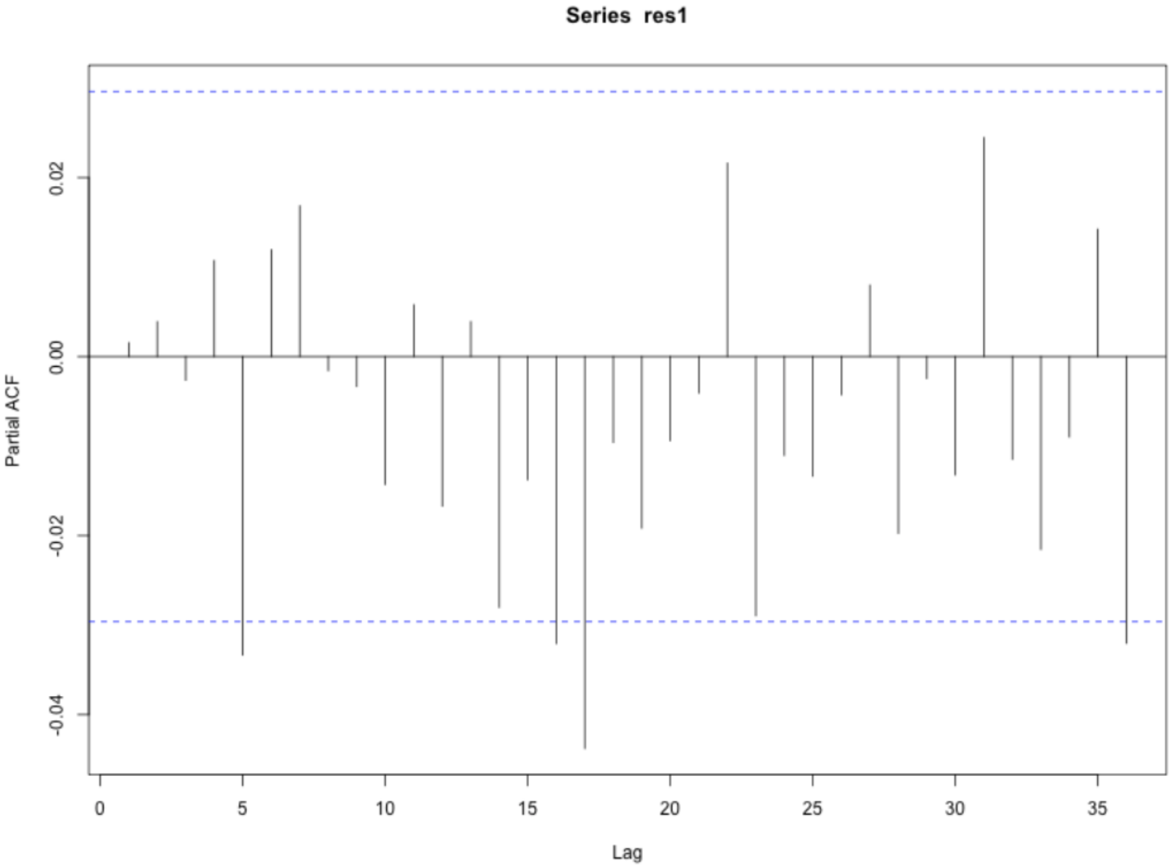


Figure 5: PACF of the ARIMA(2,1,0) model.

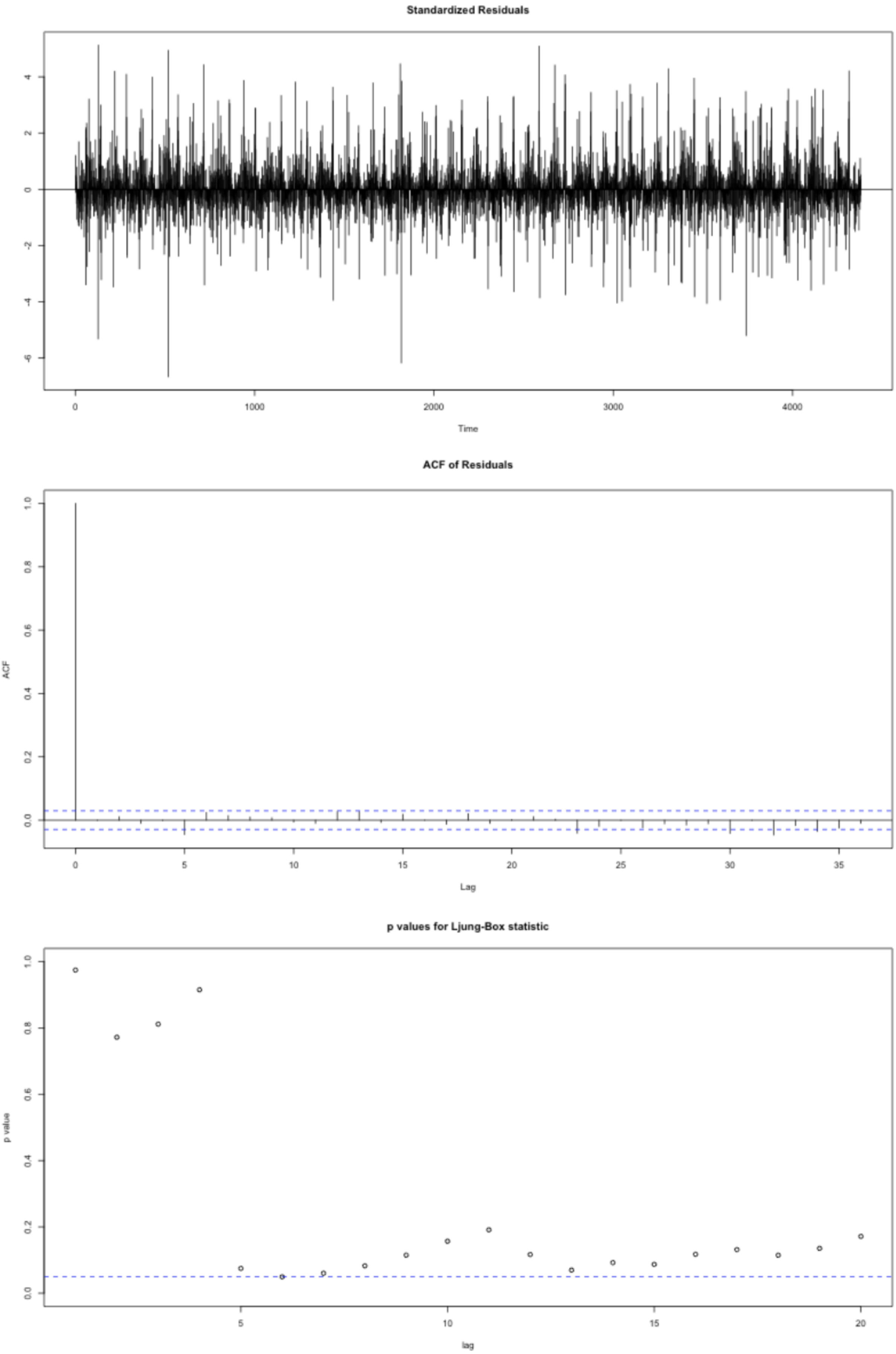


Figure 6: Residuals and the p-values for ARIMA(2,1,0) model.

3.2 Seasonality - Pure Seasonal Model

The model have all the p values to 0 as shown in Figure 7 which means that the model's residuals are not white noise which means the model has not captures the structure of data hence this model was not a good fit for the data at higher lags. Hence, more experimentation regarding purely seasonal model was not continued hereafter, and a hybrid seasonal ARIMA was investigated.

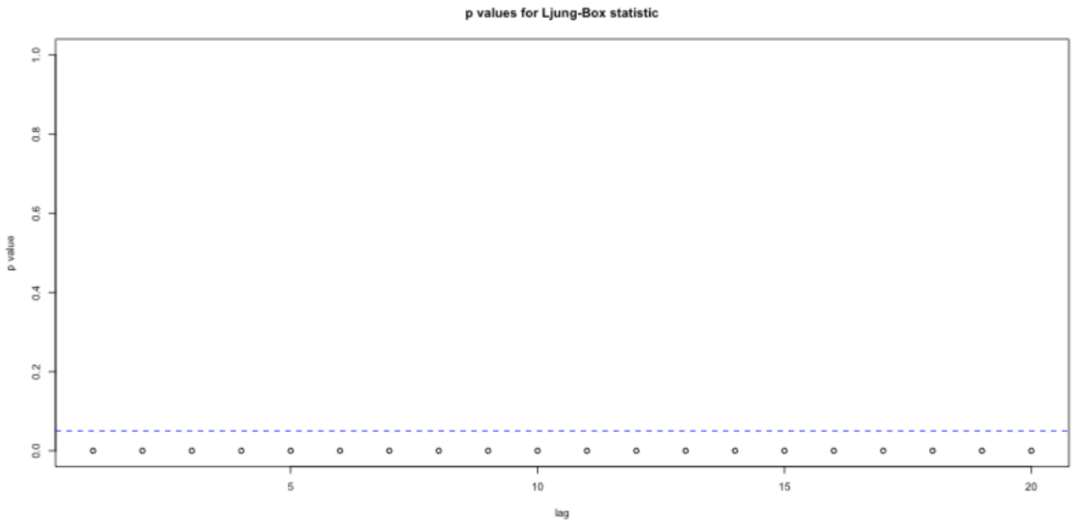
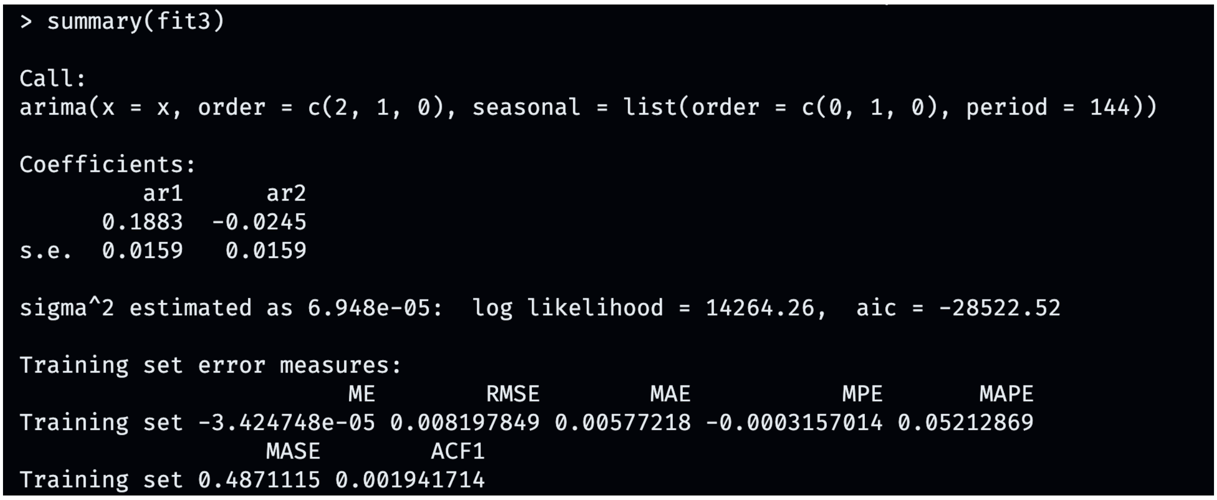


Figure 7: p values of ljung box statistics for purely seasonal SARIMA$((0, 0, 0), (2, 1, 0)_{144})$ model

3.3 Seasonality - SARIMA

After using different values for the seasonal- and non-seasonal parameters in AR/MA, and comparing the two models, the SARIMA model fits the data considerably well. There were 2 non-seasonal AR polynomials involved and 1 seasonal MA polynomial with differencing of 1, which produced a considerably more accurate forcasting and all the p values were above 0.05 which indicates that model captured the structure and patterns of the data.



```
> summary(fit3)

Call:
arima(x = x, order = c(2, 1, 0), seasonal = list(order = c(0, 1, 0), period = 144))

Coefficients:
         ar1      ar2
      0.1883  -0.0245
s.e.  0.0159   0.0159

sigma^2 estimated as 6.948e-05:  log likelihood = 14264.26,  aic = -28522.52

Training set error measures:
                        ME        RMSE        MAE           MPE       MAPE
Training set -3.424748e-05 0.008197849 0.00577218 -0.0003157014 0.05212869
                  MASE        ACF1
Training set 0.4871115 0.001941714
```

Figure 8: Model summary of SARIMA$((2, 1, 0), (0, 1, 0)_{144})$.

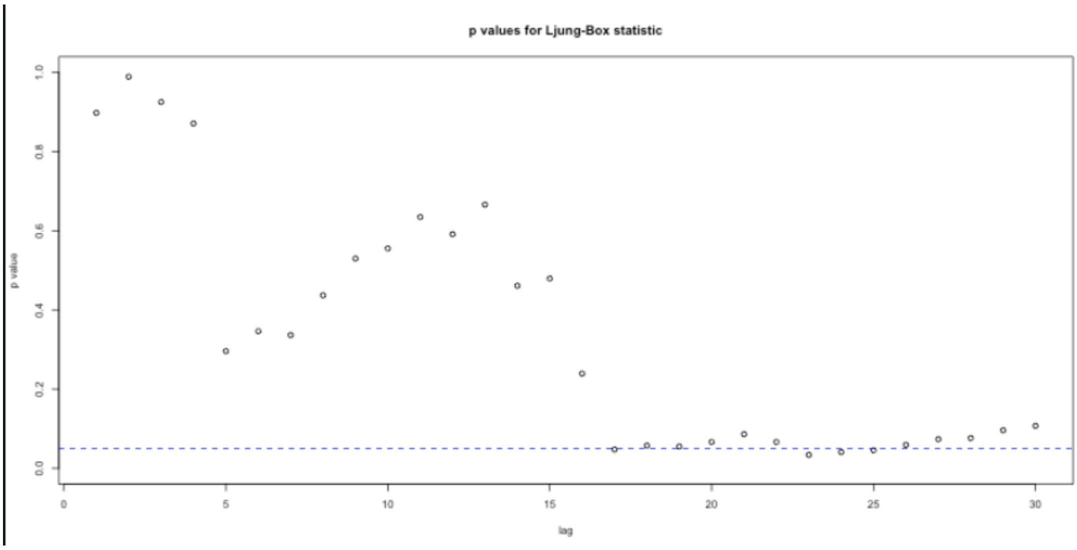
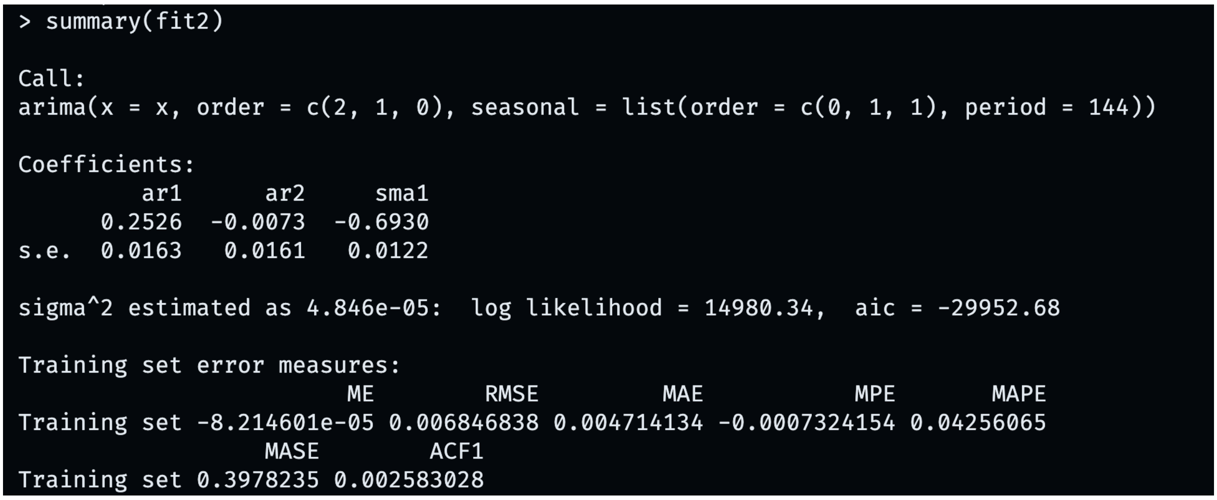


Figure 9: P-values of SARIMA$((2,1,0),(0,1,0)_{144})$.

As in Figure 9 and 12, upon inspecting the p-values from the Ljung-Box plots, the first model had some p-values going below the cutoff (at around lags 16-20), but SARIMA$((2,1,0),(0,1,1)_{144})$ had better p-values above the threshold for longer lags.



```
> summary(fit2)

Call:
arima(x = x, order = c(2, 1, 0), seasonal = list(order = c(0, 1, 1), period = 144))

Coefficients:
         ar1      ar2     sma1
      0.2526  -0.0073  -0.6930
s.e.  0.0163   0.0161   0.0122

sigma^2 estimated as 4.846e-05:  log likelihood = 14980.34,  aic = -29952.68

Training set error measures:
                        ME        RMSE         MAE           MPE       MAPE
Training set -8.214601e-05 0.006846838 0.004714134 -0.0007324154 0.04256065
                  MASE        ACF1
Training set 0.3978235 0.002583028
```

Figure 10: Model summary of SARIMA$((2,1,0),(0,1,1)_{144})$.

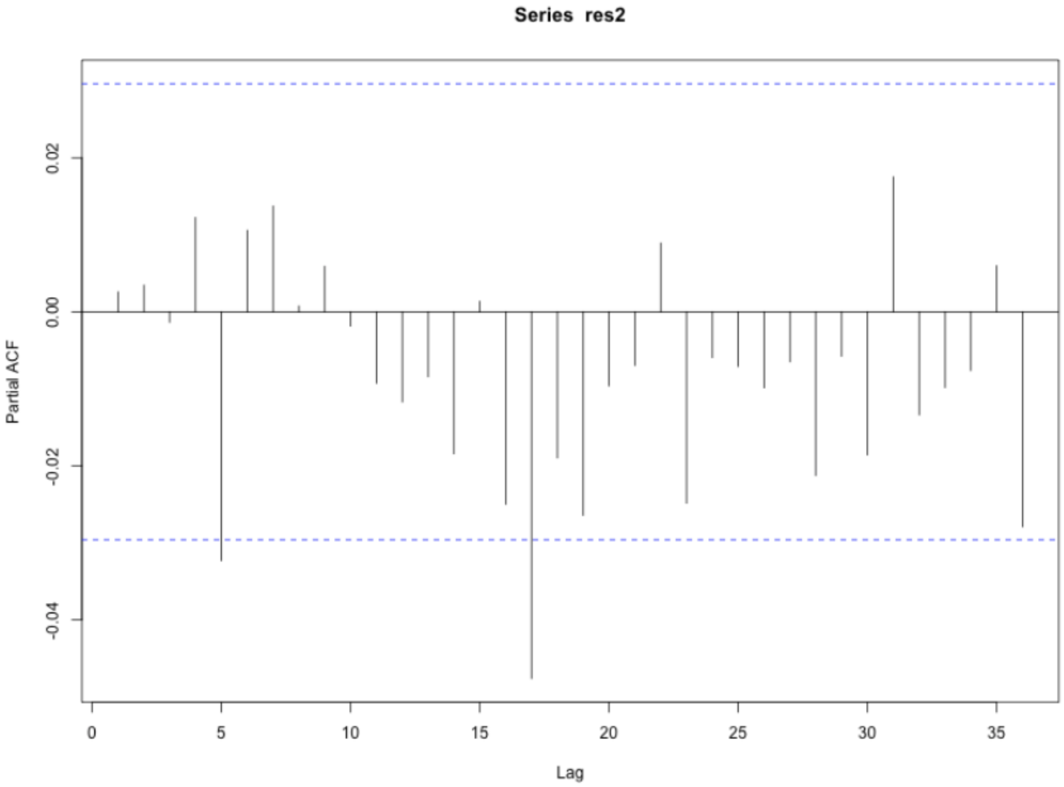


Figure 11: PACF for residuals of SARIMA$((2,1,0),(0,1,1)_{144})$.

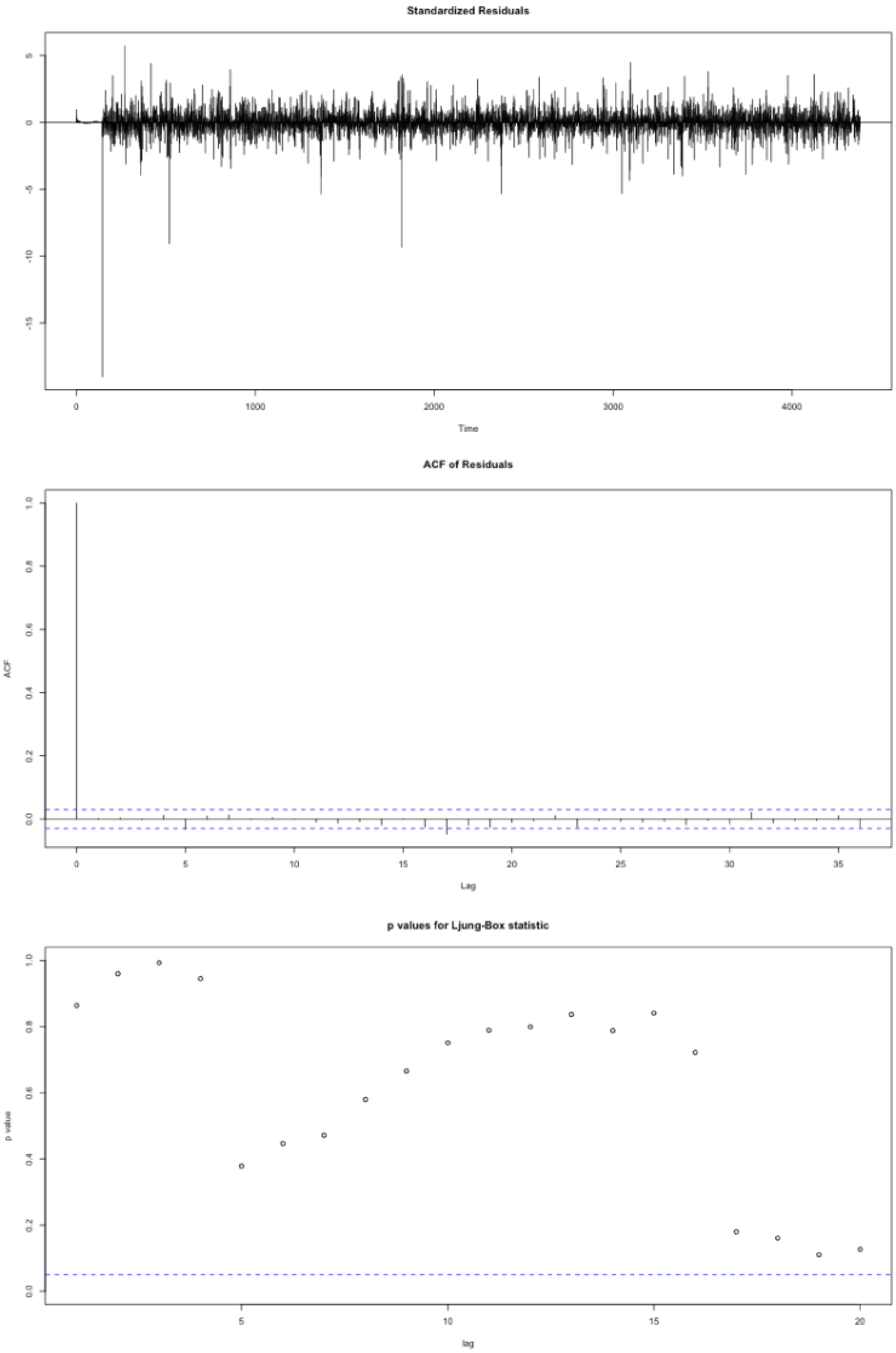


Figure 12: Plots showing standardized residuals, ACF of residuals and p-values of Ljung-Box Statistic for SARIMA$((2, 1, 0), (0, 1, 1)_{144})$.

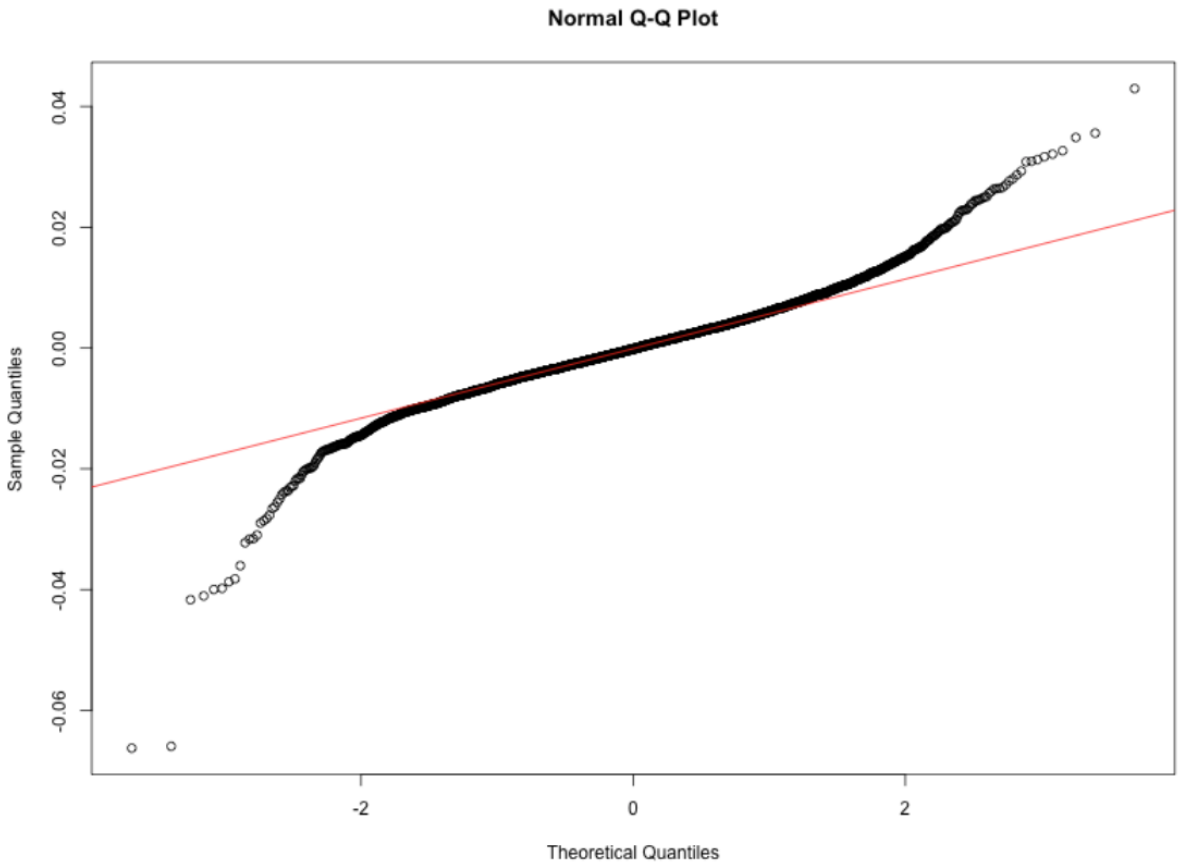


Figure 13: QQ plot for SARIMA$((2,1,0),(0,1,1)_{144})$

Increasing the AR degree from 2, and trying 3 and 4 did not yield better AICs values, hence it was decided to stay with the AR degree of 2.

4 Forecasting

The final model which we selected is SARIMA$((2,1,0),(0,1,1)_{144})$ which is a combination of a non-seasonal AR polynomial with differencing and a seasonal MA polynomial. the period is 144 which depicts 1440 minutes equals to 1 day. Figure 14 shows the actual data in black, while the forecasted data is represented by dotted blue lines.

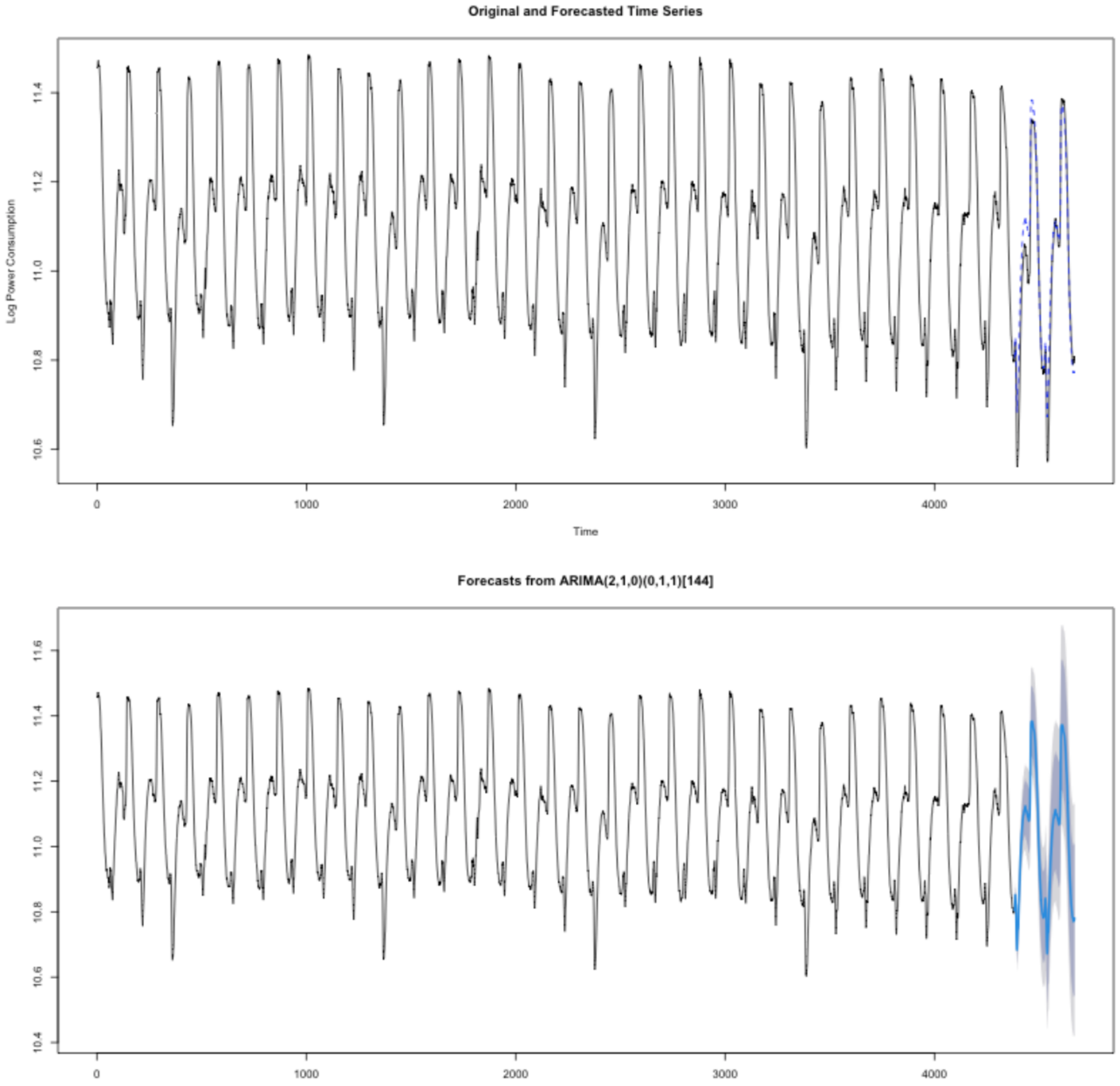


Figure 14: Forecasting with SARIMA(2,1,0)(0,1,1)[144]

5 Results and Conclusion

To conclude, a range of different autoregressive moving-average based models were fitted on a subset of a dataset containing total power consumption statistics across 3 zones in Tetouan, Morocco. The time-series data consisted of a total of 52,416 observations, each being in a 10-minute interval [1], and for this analysis project, we selected a 1-month window, from the 40,000th datapoint to 44,380th datapoint (4,380 timesteps of 10-seconds, being roughly equal to 1 month). Upon choosing a model adequately fitting to the data, from analyzing the AIC, Residual Plots, and the Ljung-Box p-values, the forecast was done for 2 days ahead (288 10-minute steps ahead).

From the ACF and PACF of the log-transformed data, a basic AR(2) with differencing of 1 was tried, followed by a purely seasonal model with the same parameters, but with a seasonal lag of 144. The reason of choosing 144 was that 1,440 minutes correspond to 1 day, which was observed as the seasonality quality from inspecting the data graph.

The basic AR model could not provide a reasonable forecast, and the seasonal patterns were not captured, and the p-values stayed above the 0.05 threshold only for small lag values. Upon trying the purely seasonal $(2,1,0)_{144}$ model, the p-values all went below the threshold, hence it was not a good fit for the dataset. Finally, two hybrid-based models were tried, being a SARIMA$((2,1,0),(0,1,0)_{144})$, and SARIMA$((2,1,0),(0,1,1)_{144})$. The AIC of the first model was -28522.52, and -29952.68 for the second model, which is better in the latter case. As in Figure 9 and 12, upon inspecting the p-values from the Ljung-Box plots, the first model had some p-values going below the cutoff, but SARIMA$((2,1,0),(0,1,1)_{144})$ had better p-values above the threshold for longer lags.

References

[1] fedesoriano. Electric power consumption. https://www.kaggle.com/datasets/fedesoriano/electric-power-consumption, 2022. Kaggle.

```
#########################
library(stats)
library(forecast)
library(tseries)
library(here)
#########################

data = read.csv(here(’powerconsumption.csv’), header=TRUE)
data$AggPowerConsumption = data$PowerConsumption_Zone1 + data$
   PowerConsumption_Zone2 + data$PowerConsumption_Zone3
x = ts(log(data$AggPowerConsumption[40000:44380]))

par(mfrow=c(3,1)) # Making the display of 2 plots
plot(x)

lag = 1
slag = 144 # seasonality of roughly 1440 minutes, roughly equal to a day
   .
dx = diff(x, lag=lag)

#########################

par(mfrow=c(2,2)) # Making the display of 2 plots
acf(x, lag.max = 30) # ACF
pacf(x, lag.max = 30) # PACF

par(mfrow=c(2,2)) # Making the display of 2 plots
acf(dx, lag.max = 30) # ACF
pacf(dx, lag.max = 30) # PACF

#########################
# Model Fitting
#########################

start.time <- Sys.time()
fit1 <- arima(x,order=c(0,0,0), seasonal=list(order=c(2,1,0),period=144)
   )
fit2 <- arima(x,order=c(2,1,0), seasonal=list(order=c(0,0,0),period=144)
   )
fit3 <- arima(x,order=c(2,1,0), seasonal=list(order=c(0,1,1),period=144)
   )
end.time <- Sys.time()

summary(fit1)
summary(fit2)
summary(fit3)
res1 <- residuals(fit1)
res2 <- residuals(fit2)
res3 <- residuals(fit3)

par(mfrow=c(2,2))
qqnorm(res1)
qqline(res1, col=’red’)
qqnorm(res2)
qqline(res2, col=’red’)
qqnorm(res3)
qqline(res3, col=’red’)
tsdiag(fit1, gof.lag = 30)
```

```r
pacf(res1)
tsdiag(fit2, gof.lag = 30)
pacf(res2)
tsdiag(fit3, gof.lag = 30)
pacf(res3)

time.taken <- round(end.time - start.time,2)
time.taken

#########################
# Forecasting
#########################

forecasted <- forecast(fit3, h=288)

par(mfrow=c(2,1))
x_test <- ts(log(data$AggPowerConsumption[40000:44668]))
plot(
    x_test,
    type="l", col="black", lty=1,
    xlim=c(0, length(x) + length(forecasted$mean)),
    ylab="Log Power Consumption",
    xlab="Time",
    main="Original and Forecasted Time Series"
)
lines(
    (length(x) + 1):(length(x) + length(forecasted$mean)),
    forecasted$mean,
    col="blue",
    lty=2
)
plot(forecasted)
```