STAT8008 Time Series Project

Paul Christopher, R00207143

12/17/2021

Introduction

The dataset household_power_consumption.txt contains 2,075,259 measurements gathered in a house located in Sceaux between December 2006 and November 2010 (47 months).[1] The measurements of the household are taken with a one-minute sampling rate over the almost 4 year period.

The analysis of this project will be focused on the monthly averaged global_active_power variable which measures the amount of electrical power consumed by the household.

Data cleaning and manipulation

The first task was to check the dataset for missing values. A summary of the missing values is given in the table and the associated graphs at Figure 1 and Figure 2.

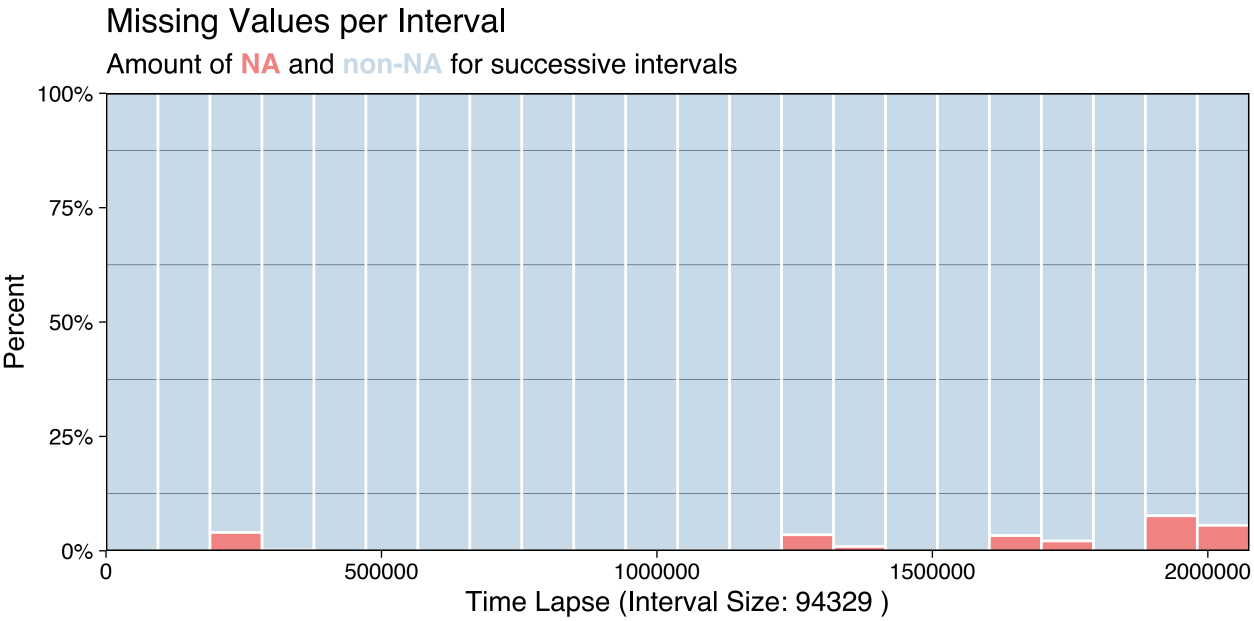


Figure 1: Distribution of missing values in the dataset.

Missing values of the global_active_power variable needed to be estimated by interpolation of previous and posterior obsrvations. The `imputeTS` package was used for the interpolation of missing values. It offers multiple state-of-the-art imputation algorithm implementations along with plotting functions for time series missing data statistics. See Moritz and Bartz-Beielstein (2017) for more information about this package.

[1] See the UCI Machine Learning Repository.

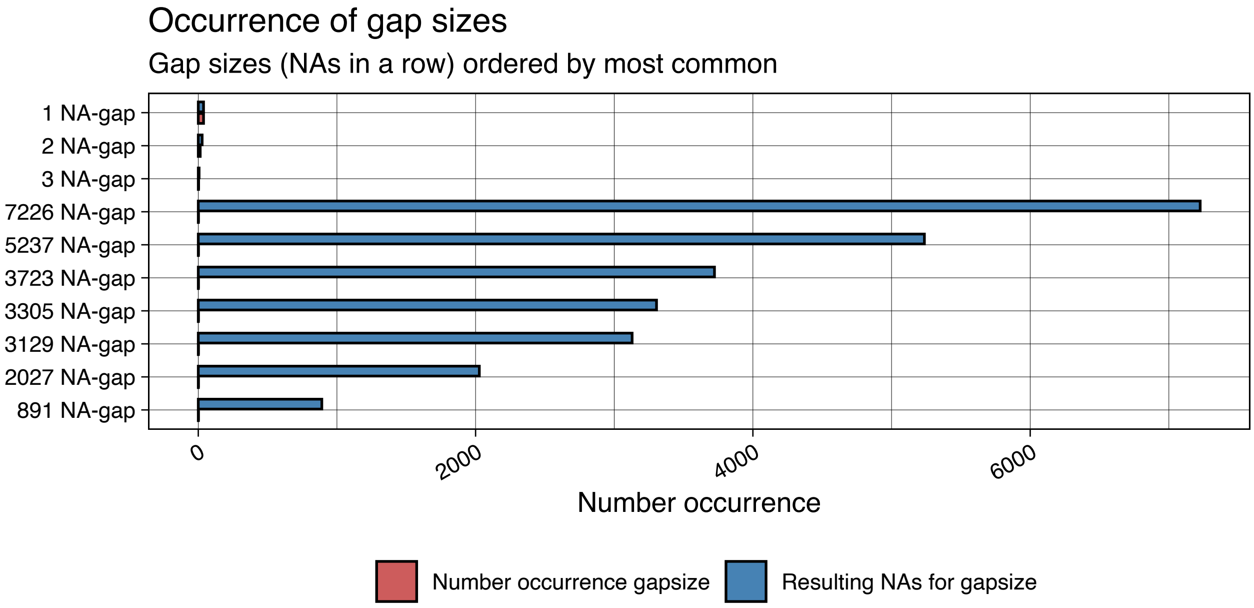


Figure 2: Distribution of missing values in dataset.

Data classed as arbitrary missing was that which included up to 83 consecutive missing values. However, as can be seen, most of the data was monotone missing, with the highest length of consecutively missing data being 7226 NA's in a row. This represented approximately 5 consecutive days worth of data. This represents a problem with the data and may skew the monthly averages to be calculated from the data. A possible solution may be to apply the Kalman imputaiton algorithm, which Wongoutong (2020) has shown to be the best algorithm in the `imputeTS` package for large monotone missing blocks.
However Moritz and Bartz-Beielstein (2017) warn that applying this imputation algorithm could take several days to complete on an ordinary computer. Therefore, the `na_ma()`[2] algorithm in `imputeTS` was used to impute the missing values, despite its less than optimal imputation algorithm.

Once missing data was replaced with imputed values as above, `lubridate` and `dplyr` were then used to create a new time series of the monthly power averages, as transformed from the minute-interval data, for analysis.

Preliminary analysis

A plot of the transformed time series is shown at Figure 3.

Summary statistics for the time series are as follows:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  276.5   969.6  1116.4  1100.6  1280.8  1901.6
```

A decomposition of the time series is shown at Figure 4.

It would appear that an additive model is appropriate given that the seasonal variance is relatively constant, albeit that in 2008 the seasonal trough was lower than usual.

There is strong seasonality, with peaks occurring in the winter. However, the overall trend is slightly downwards between 2007 and 2011. The greatest contribution to the overall series is the seasonal component, as can be seen from the range bars at the right of the decomposed series plots. A seasonplot is also shown at Figure 5.

[2] See Moritz and Bartz-Beielstein (2017).

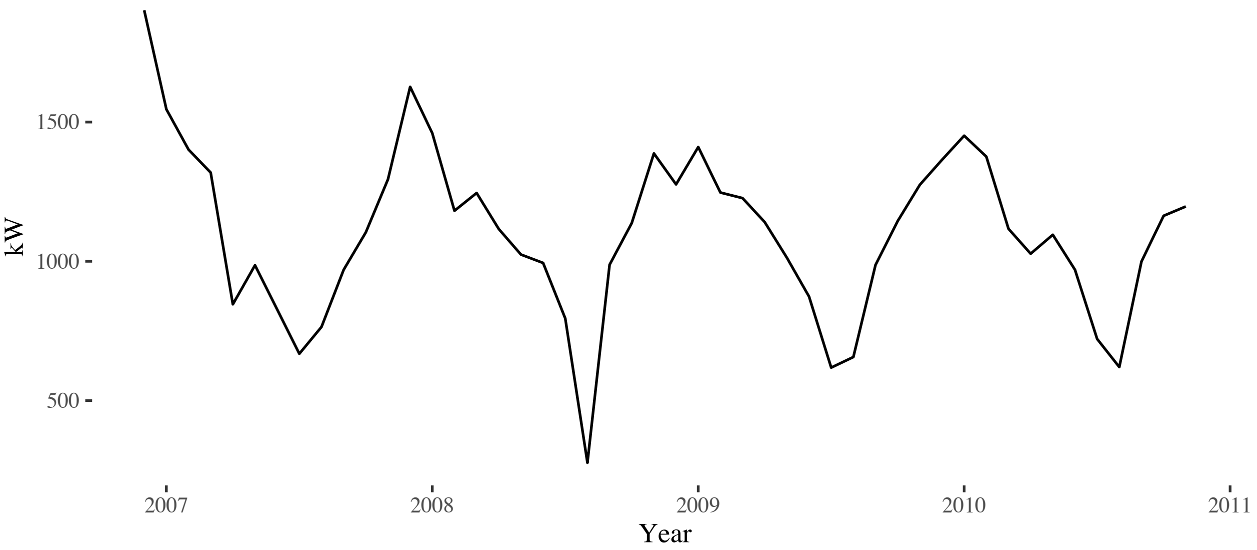


Figure 3: Plot of the time series.

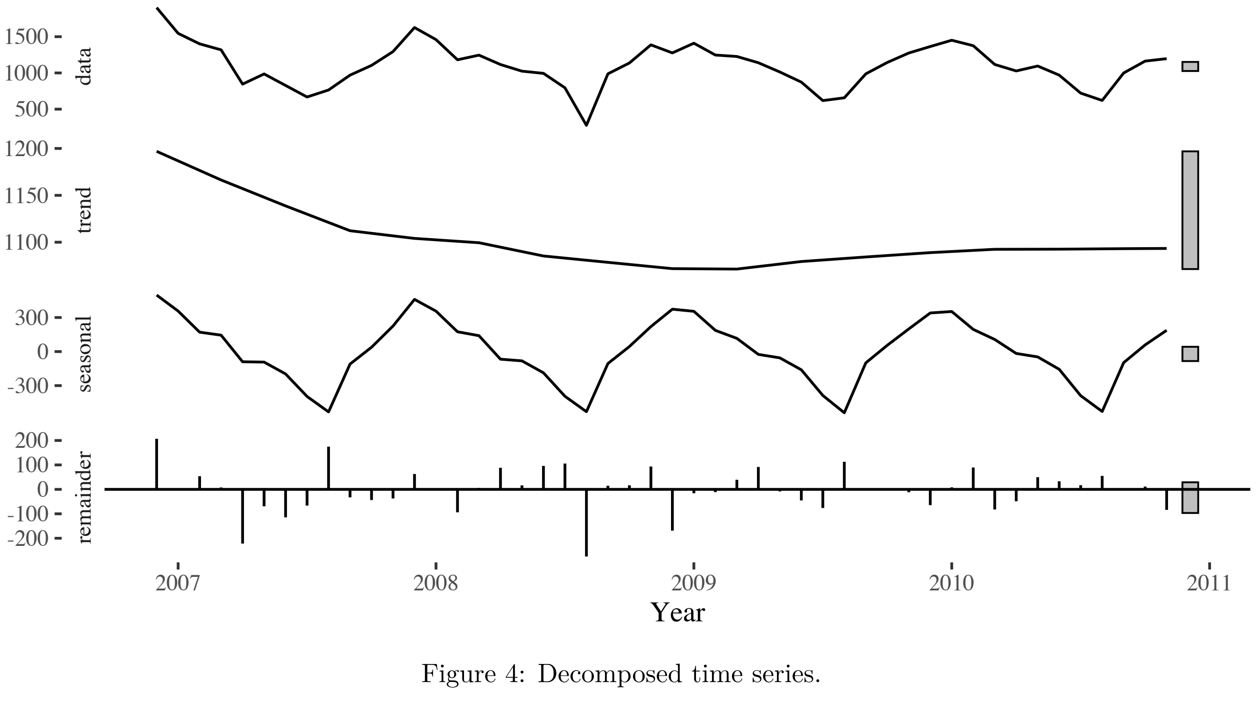


Figure 4: Decomposed time series.

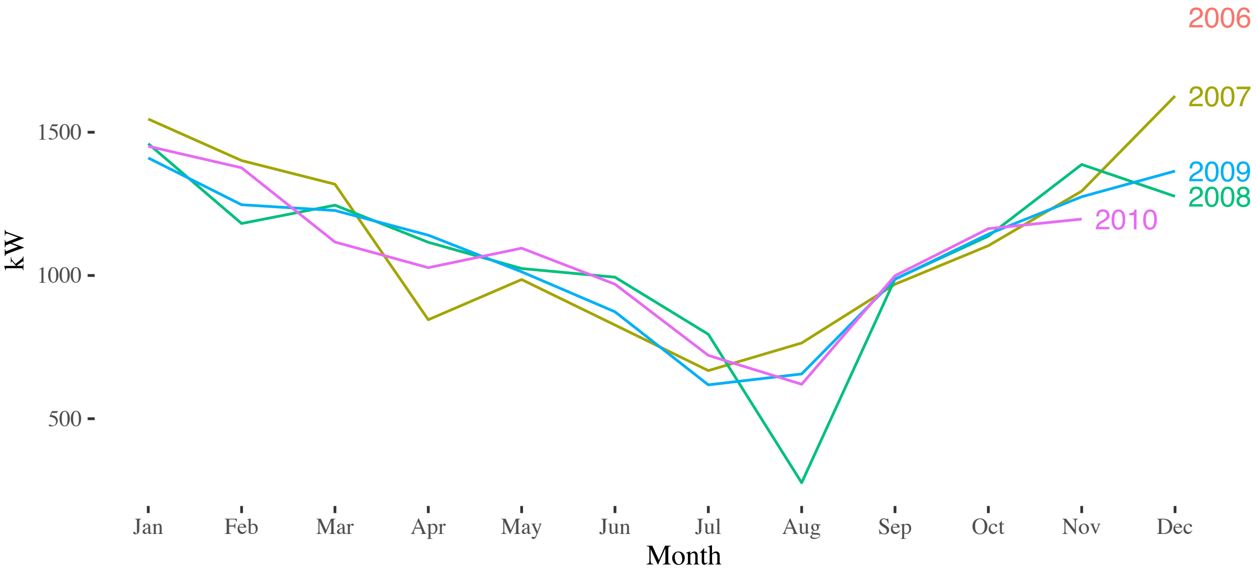


Figure 5: Seasonplot

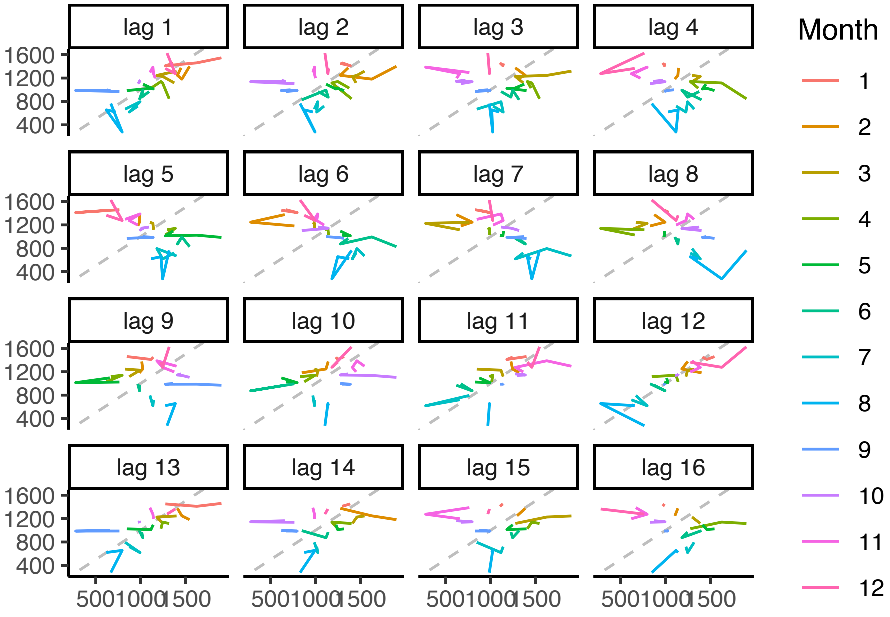


Figure 6: Lag plots of the time series

Another graphical tool to check for seasonality is the lag plot. A matrix of lag plots for different lags from 1 to 12 is shown at Figure 6. The relationship is strongly positive at lag 12, reflecting the strong seasonality in the data.

There a number of ways to address seasonality in time series data. For example, differencing can be used to eliminate seasonality effects. As the seasonality is annual, the periodicity of this seasonal component would be 12. A lag-12 seasonal difference operator can be defined as:

$$\nabla_{12} y_t = (1 - B^{12}) y_t = y_t - y_{t-12}$$

where y_t denotes the power consumption in month t, and y_{t-12} represents the power consumption 12 months earlier.

Time series modelling

Given the strong seasonal component in the data, the most appropriate classical method (*aka* exponential smoothing) to describe this data would be Holt-Winter's seasonal method.

The Holt-Winter's seasonal method comprises the forecast equation and three smoothing equations - one for the level ℓ_t, one for the trend b_t, and one for the seasonal component s_t, with corresponding smoothing parameters α, β^*andγ. The seasonality of the data is denoted by $m = 12$.

In the additive model, the seasonal variation is independent of the absolute level of the time series, but it takes approximately the same magnitude each year.

In the multiplicative model, the seasonal variation takes the same relative magnitude each year. This means that the seasonal variation equals a certain percentage of the level of the time series. The amplitude of the seasonal factor varies with the level of the time series.[3]

The additive method is preferred when the seasonal variations are roughly constant through the series. With this method, the seasonal component is expressed in absolute terms in the scale of the observed series, and in the level equation the series is seasonally adjusted by subtracting the seasonal component. Within each year, the seasonal component will add up to approximately zero.

The component form for the additive method is:

$$\begin{aligned}
\hat{y}_{t+h|t} &= \ell_t + hb_t + s_{t+h-m(k+1)} \\
\ell_t &= \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \\
s_t &= \gamma(y_t - \ell_{t-1} - b_{t-1} + (1-\gamma)s_{t-m})
\end{aligned}$$

The output for the additive model is shown below. The plot for the additive model is at Figure 7.

```
ETS(A,N,A)

Call:
 ets(y = ts)

  Smoothing parameters:
    alpha = 1e-04
    gamma = 1e-04

  Initial states:
    l = 1105.9703
    s = 231.8411 43.9348 -104.638 -523.7833 -392.8731 -200.4128
```

[3] Linde (2005).

```
          -9.5169 -8.9415 97.9742 142.984 347.6633 375.7682

  sigma:  135.9022

     AIC     AICc      BIC 
670.8111 685.8111 698.8791 

Training set error measures:
                   ME     RMSE      MAE      MPE     MAPE      MASE       ACF1
Training set -5.44064 114.3788 79.74902 -2.77284 9.012536 0.6976086 0.05396785
```

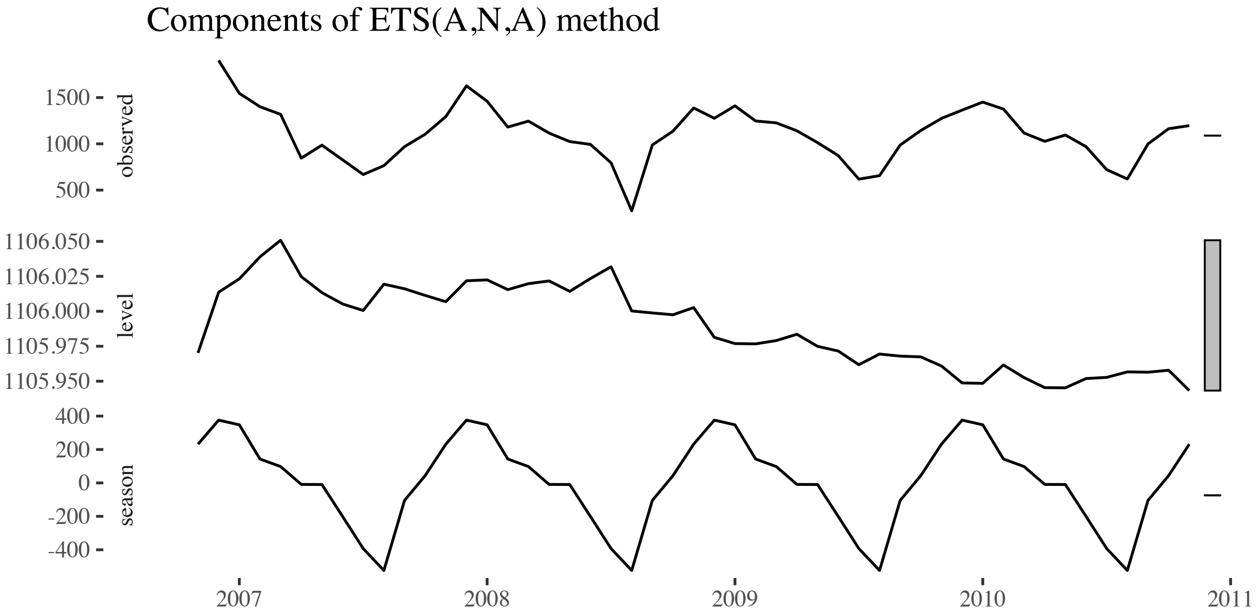


Figure 7: additive model

The `ets()` function uses the Akaike's Information Criterion corrected for small sample bias (AICc) for choosing the best model by default.The AICc is defined as follows:

$$\mathrm{AIC_c} = \mathrm{AIC} + \frac{2k(k+1)}{T-k-1},$$

where AIC is defined as:

$$\mathrm{AIC} = -2\log(L) + 2k,$$

where L is the likelihood of the model and k is the total number of parameters and initial states that have been estimated.[4]

The output for the multiplicative model is shown below:

```
ETS(M,N,M) 

Call:
 ets(y = ts, model = "MNM") 

  Smoothing parameters:
    alpha = 1e-04 
    gamma = 1e-04 
```

[4] Rob J. Hyndman and Athanasopoulos (2018).

```
  Initial states:
    l = 1107.2959
    s = 1.1633 1.0268 0.8836 0.5686 0.6322 0.8229
           0.9284 0.9423 1.1189 1.1742 1.3106 1.4283

  sigma:  0.1356

     AIC     AICc      BIC
677.2242 692.2242 705.2922

Training set error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE
Training set -6.739175 109.0767 71.98371 -2.962254 8.470601 0.6296812
                    ACF1
Training set -0.01936058
```

It is interesting that when you do not explicitly state the model, the function chooses the additive type of model automatically based on this criterion. As can be seen, the AICc for the default model is 685.81, whereas the AICc for the multiplicative model is 692.22.

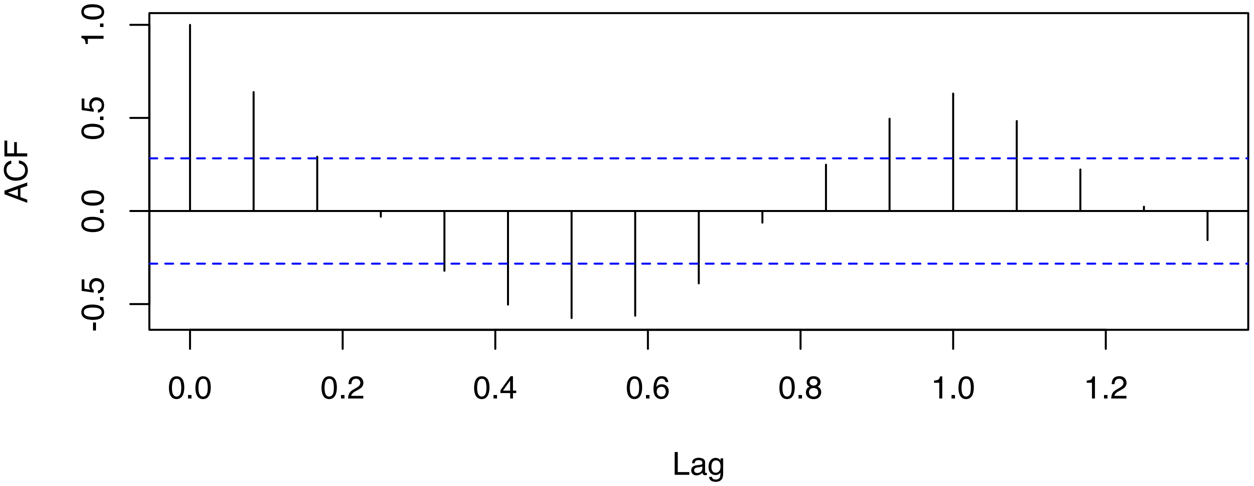
The fact that the `ets()` function automatically chose the additive model based on the AICc criterion is not surprising, given that the earlier visual inspection of the seasonal component of the decomposed series showed the seasonal component as having reasonably constant variance.

Is the time series stationary?

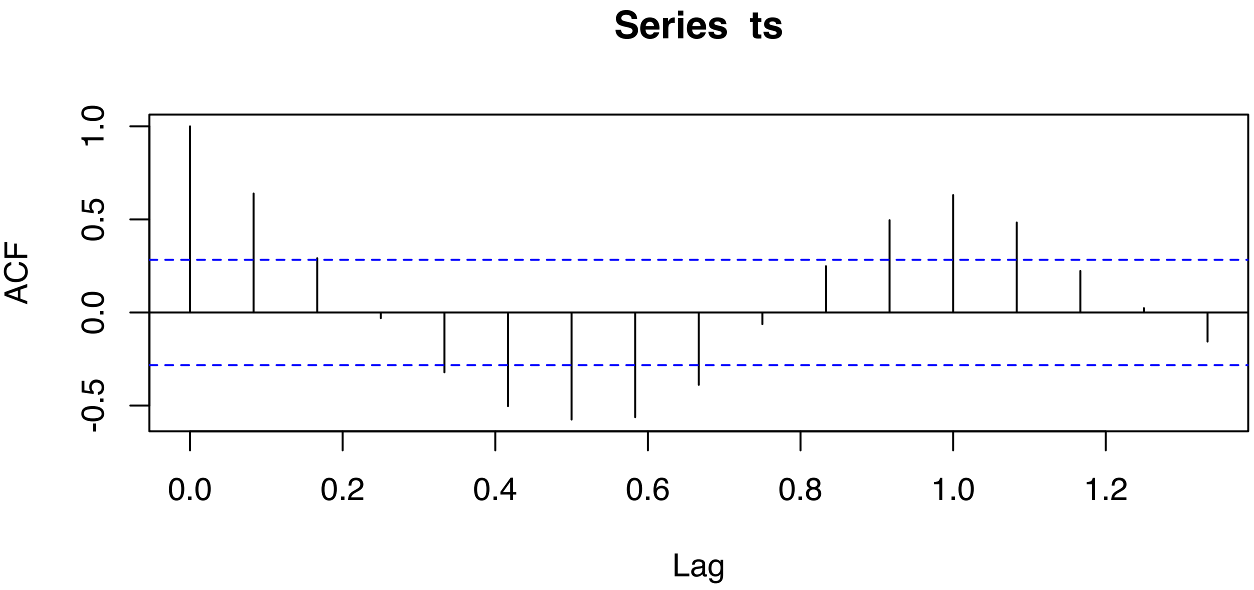
A time series is said to be *strictly stationary* if its properties are not affected by a change in the time origin. The stationarity assumption means that the probability distribution of y_t is the same for all time periods and can be written as $f(y)$. Weak stationarity is defined as follows: (1) the expected value of the time series does not depend on time and (2) the autocovariance function defined as $Cov(y_t, y_{t+k})$ for any lag `k` is only a function of `k` and not time.[5]

As any violation of stationarity creates estimation problems for ARIMA models, it is necessary, in the first instance, to check whether the time series is stationary and, in the second instance, if it is not stationary, to transform the original time series by, for example, differencing it.

One method of checking whether the series is stationary is to create a correlogram or auto-correlation function plot.



[5] Montgomery, Jennings, and Kulahci (2008).



The correlogram at **??** shows that auto-correlation functions drop off reasonably quickly but there is some cyclicality, especially around lag 12, which is unsurprising, given that the data is monthly and there is annual seasonality. The auto-correlation function plot is not strongly suggestive of a non-stationary time series, seasonality effects excluded.

However, the earlier decomposed plot of the time series seems to show a slight downward trend, particularly in the first half of the series, although the trend seems to stabilise after that. This is mildly indicative of non-stationarity.

The identification process can be assisted by another plot, the Partial Auto-Correlation Function plot.

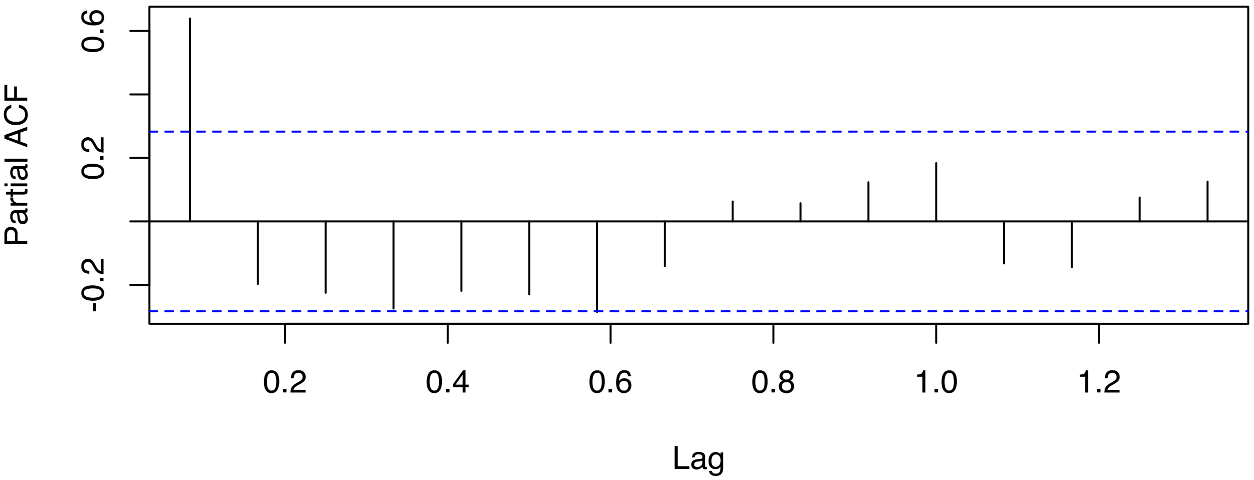


Figure 8: Partial Auto-Correlation Function plot

The output of the PACF plot suggests that an AR(1) process may be appropriate.

More formal statistical tests for assessing stationarity are the Augmented Dickey-Fuller, the Phillips-Perron and the KPSS unit root tests.

The Augmented Dickey-Fuller and Phillips-Perron tests use the following null and alternative hypotheses:

- H_0: The time series is non-stationary. In other words, it has some time-dependent structure and does not have constant variance over time.
- H_A: The time series is stationary.

If the p-value from the test is less than some significance level (e.g. $\alpha = .01$), then we can reject the null hypothesis and conclude that the time series is stationary. The results of both tests on the time series are shown below:

```
Augmented Dickey-Fuller Test
```

data: ts Dickey-Fuller = -3.5736, Lag order = 1, p-value = 0.04496 alternative hypothesis: stationary

```
Phillips-Perron Unit Root Test
```

data: ts Dickey-Fuller Z(alpha) = -19.521, Truncation lag parameter = 3, p-value = 0.05087 alternative hypothesis: stationary

Since the p-values associated with both tests is greater than .01, we fail to reject the null hypothesis and conclude that the time series may be non-stationary.

The KPSS test is set up in the opposite direction to the ADF and PP tests in that the null hypothesis is that the series is trend stationary and the alternative hypothesis is that it is non-stationary.[6]

To confirm our suspicions, the next step is to take a first difference of the time series and perform these stationarity tests on the differenced series:

```
    Augmented Dickey-Fuller Test

data:  dts
Dickey-Fuller = -4.4561, Lag order = 1, p-value = 0.01
alternative hypothesis: stationary


    Phillips-Perron Unit Root Test

data:  dts
Dickey-Fuller Z(alpha) = -40.768, Truncation lag parameter = 3, p-value
= 0.01
alternative hypothesis: stationary
```

The results of both tests for the first-differenced series now show a p-value less than 0.01, indicating stationarity. A plot of the differenced time series is shown at Figure 9.

ARIMA modelling

If successive observations show serial dependence, ‘forecasting methods based on exponential smoothing may be inefficient and sometimes inappropriate because they do not take advantage of the serial dependence in the observations in the most effective way. To formally incorporate this dependent structure’, ARIMA models are used.[7]

We have already concluded from the unit root tests above that the series my be integrated of order 1 and have already differenced the time series to account for that and remove the trend. We have also concluded that the series may be auto-correlated of order 1 from the ACF and PACF plots. This was because the ACF plot showed sinusoidal behaviour, whilst the PACF plot showed a single spike at the first lag, followed by a sudden decline to zero for all subsequent lags.

Using the terminology of Box and Jenkins, our preliminary model, ignoring seasonality for the moment, can therefore be described as ARIMA (1,1,0).

However, we have also concluded that the data series is seasonal of periodicity 12.

[6] Trapletti and Hornik (2020)

[7] Montgomery, Jennings, and Kulahci (2008).

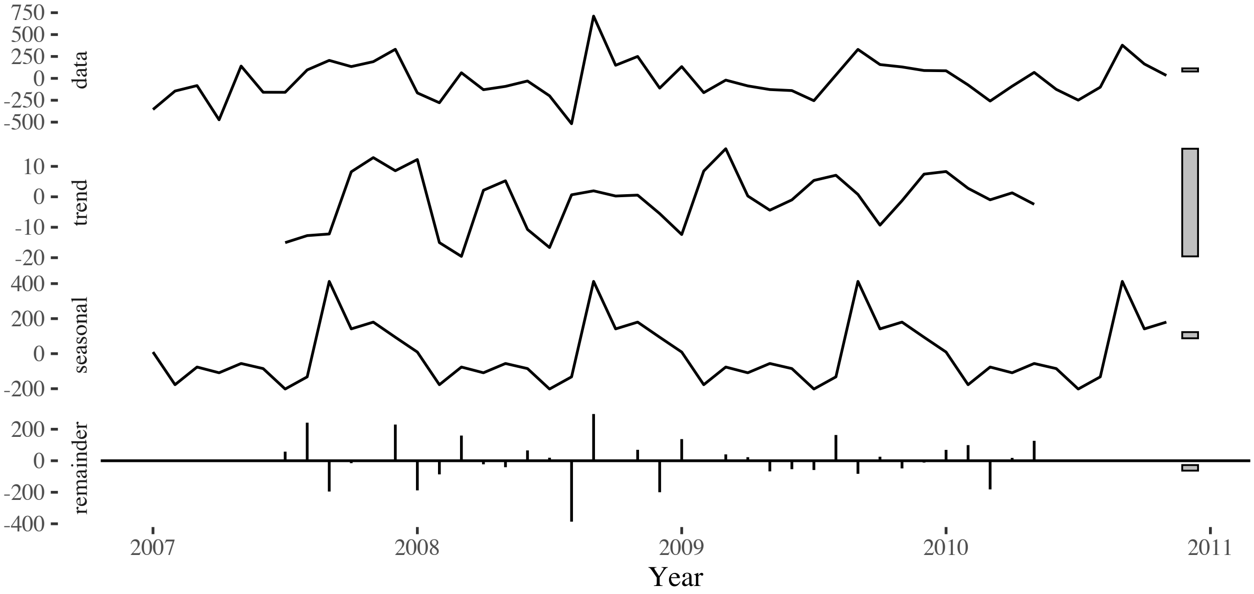


Figure 9: Decomposition of differenced time series.

Rob J. Hyndman and Athanasopoulos (2018) advises that when it is obvious that the series is seasonal it is better to seasonally difference the series before taking the first difference to ascertain whether seasonally differencing the series corrects problems associated with non-stationarity. If so, it may then be unnecessary to also take the first difference, following the principle of model parismony.

As can be seen from Figure 10, doubly differencing the series does not appear to make much difference to the stationarity or other general appearance of the series when compared to the seasonally differenced series.

Lets try more formal tests of stationarity on the series which has only been seasonally differenced:

```

    Augmented Dickey-Fuller Test

data:  sadts
Dickey-Fuller = -5.0883, Lag order = 1, p-value = 0.01
alternative hypothesis: stationary


    Phillips-Perron Unit Root Test

data:  sadts
Dickey-Fuller Z(alpha) = -30.769, Truncation lag parameter = 3, p-value
= 0.01
alternative hypothesis: stationary
```

The results of both tests for the seasonally-differenced series show a p-value less than 0.01, indicating stationarity. Based on these formal tests, it may be unnecessary to take a first difference of the series in addition to seasonally differencing it.

Similarly, the kpss.test indicates that the seasonally differenced series now has little non-stationarity to worry about:

```

    KPSS Test for Level Stationarity
```

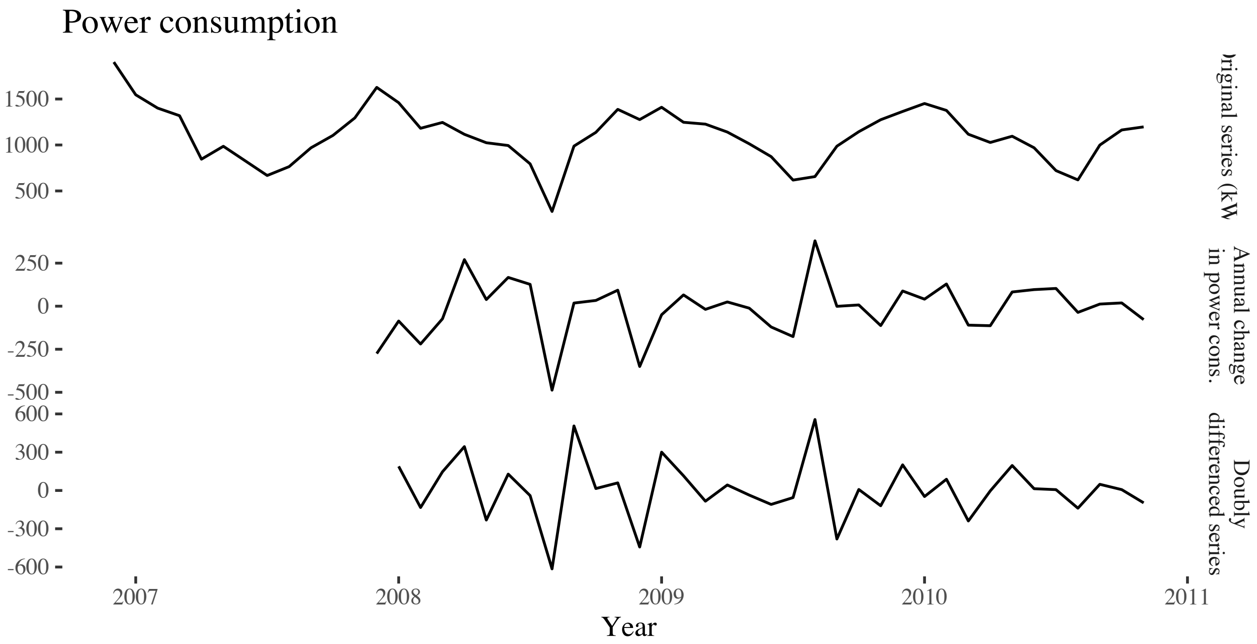


Figure 10: Plots of the original time series, the seasonally differenced series and the doubly differenced series.

```
data:  sadts
KPSS Level = 0.21838, Truncation lag parameter = 3, p-value = 0.1
```

i.e. based on a p-value $>$ .05, we cannot reject the null hypothesis that the series is stationary.

The aim now is to find an appropriate ARIMA model based on the ACF and PACF shown in Figure 11.There are no significant spikes in either the ACF or the PACF plots. Consequently, we begin with an ARIMA(0,0,0)$(0,1,0)_{12}$ model, indicating a seasonal difference only.

Assessing the ARIMA models

Rob J. Hyndman and Athanasopoulos (2018) advise that:

> Good models are obtained by minimising the AIC, AICc or BIC. Our preference is to use the AICc. It is important to note that these information criteria tend not to be good guides to selecting the appropriate order of differencing (d) of a model, but only for selecting the values of p and q. This is because the differencing changes the data on which the likelihood is computed, making the AIC values between models with different orders of differencing not comparable. So we need to use some other approach to choose d, and then we can use the AICc to select p and q.

The Ljung-Box test is applied to the residuals of a fitted ARIMA model and may be defined as:

- H_0: The data are independently distributed (*i.e.* the correlations in the population from which the sample is taken are 0, so that any observed correlations in the data result from randomness of the sampling process);
- H_A: The data are not independently distributed; they exhibit serial correlation.

Looking more closely at the residuals of the ARIMA(0,0,0)$(0,1,0)_{12}$ using the `checkresiduals` function (Figure 12), we can see that Ljung-Box test gives a p-value of 0.9314:

```

    Ljung-Box test

data:  Residuals from ARIMA(0,0,0)(0,1,0)[12]
```

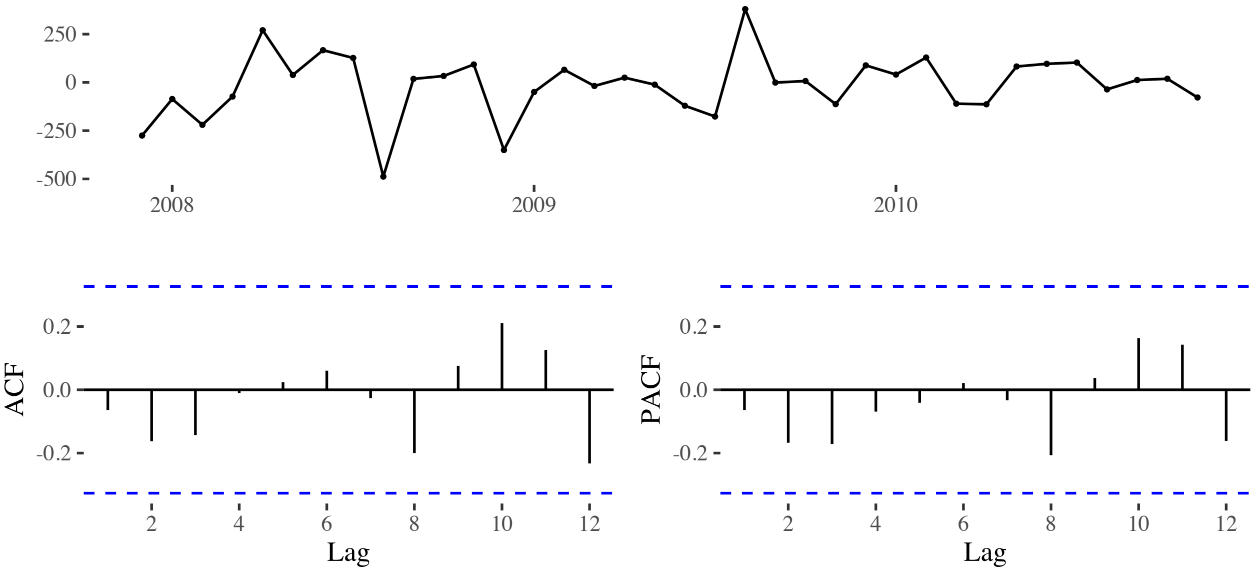


Figure 11: Seasonally differenced power consumption.

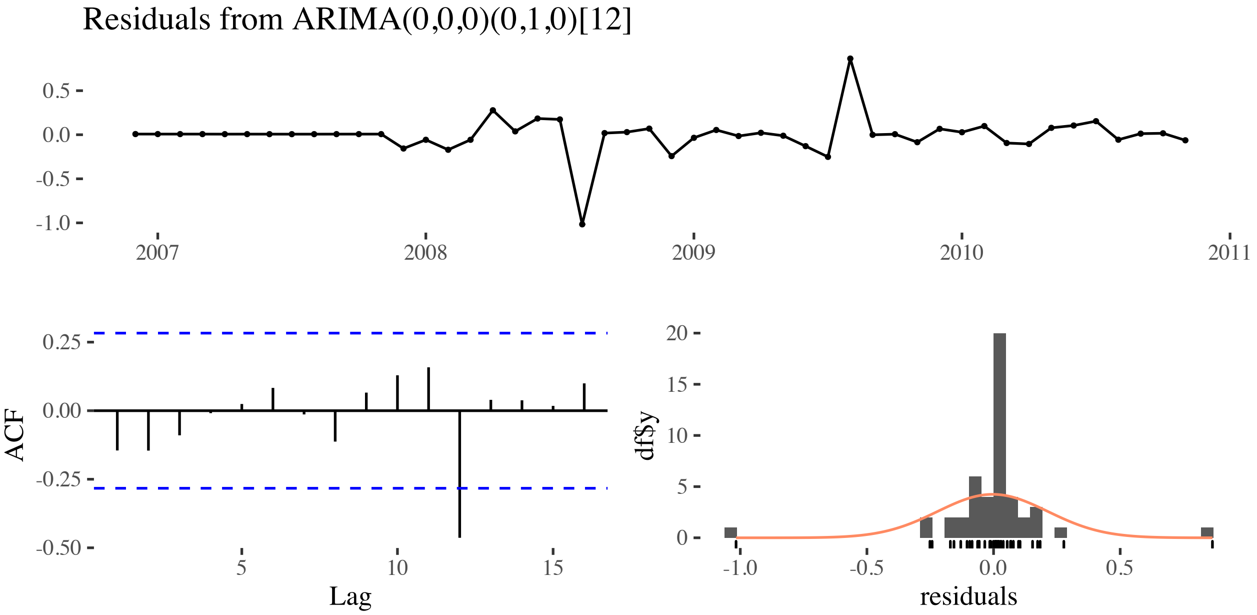


Figure 12: Residuals from the fitted ARIMA(0,0,0)(0,1,0)$_{12}$ model for the power consumption data.

```
Q* = 24.286, df = 36, p-value = 0.9314

Model df: 0.   Total lags used: 36
```

One can also see that most of the autocorrelations are within the threshold limits, apart from the 12th lag. This may suggest a seasonal MA(1) component.

One can therefore try an ARIMA(0,0,0)(0,1,1)$_{12}$ model to see if it fits the data better.

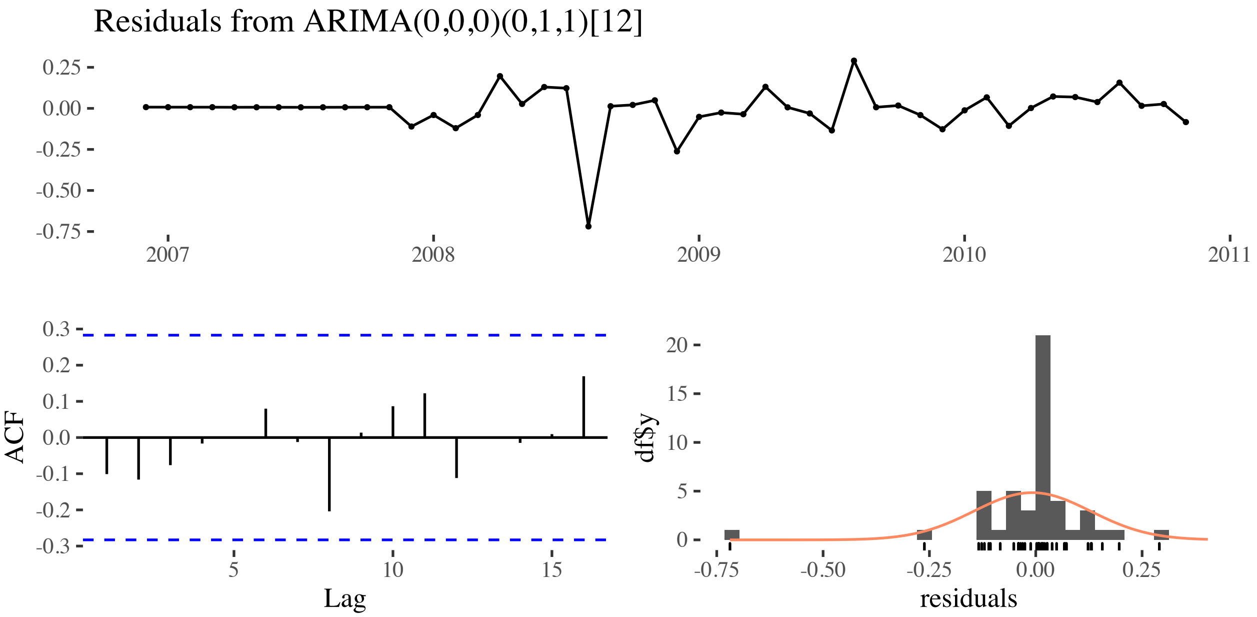


Figure 13: Residuals from the fitted ARIMA(0,0,0)(0,1,1)$_{12}$ model for the power consumption data.

```
    Ljung-Box test

data:  Residuals from ARIMA(0,0,0)(0,1,1)[12]
Q* = 15.096, df = 35, p-value = 0.9987

Model df: 1.   Total lags used: 36
```

Looking at the ACF plot in Figure 13 one can see that all of the autocorrelations are now within the threshold limits. Moreover, the p-value from the Ljung-Box test is slightly higher than for the ARIMA(0,0,0)(0,1,0)$_{12}$ model. Accordingly, the ARIMA(0,0,0)(0,1,0)$_{12}$ model is to be preferred amongst those two ARIMA models.

Finally, for the sake of completeness, it may be interesting to examine an ARIMA(1,1,0) model, given the initial findings before seasonality had been considered above.

```
    Ljung-Box test

data:  Residuals from ARIMA(1,1,0)
Q* = 37.587, df = 35, p-value = 0.3515

Model df: 1.   Total lags used: 36
```

As can be seen from the ACF plot and the p-value associated with the Ljung-Box test, this model is inferior to the two seasonal models considered previously.

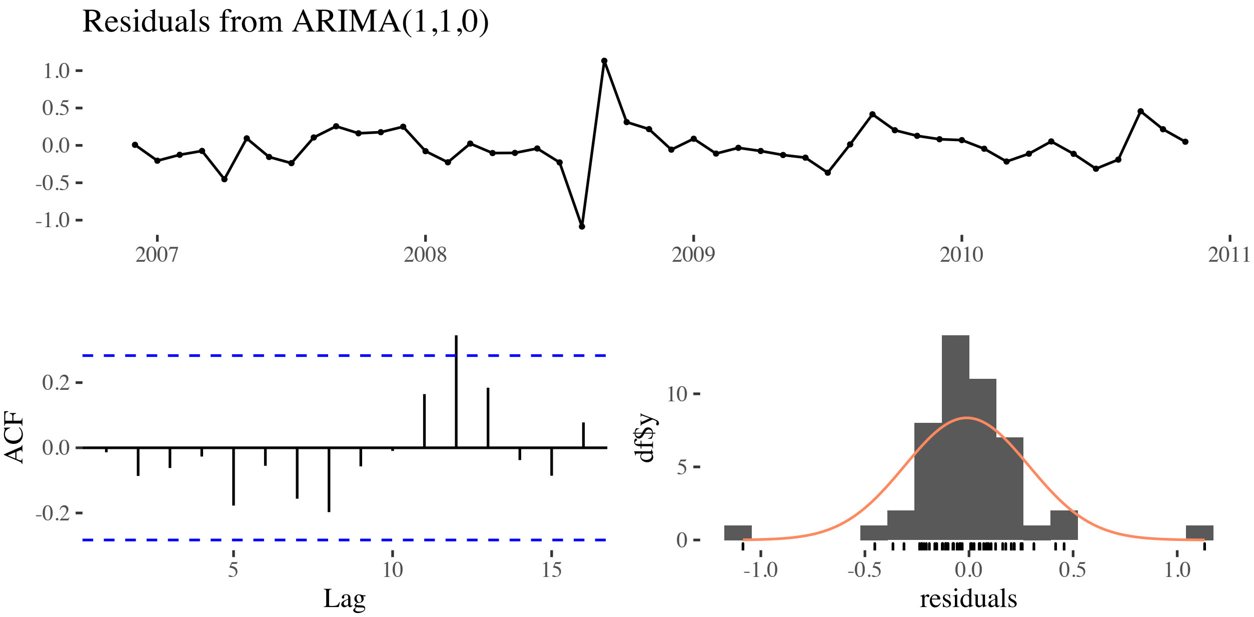


Figure 14: Residuals from the fitted ARIMA(1,1,0) model for the power consumption data.

Using the `auto.arima` function

Now that we have fitted and examined a few ARIMA models, it would be interesting to compare them with the ARIMA model that the `auto.arima` function suggests. The default arguments are designed for rapid estimation of models for many time series. Based on the recommendations of the authors, stepwise=FALSE and approximation=FALSE were set as arguments to the function, given that we are examining only time series.[8]

```
    Ljung-Box test

data:  Residuals from ARIMA(0,0,0)(0,1,1)[12]
Q* = 18.025, df = 35, p-value = 0.9922

Model df: 1.   Total lags used: 36
```

As can be seen, an ARIMA(0,0,0)(0,1,1)$_{12}$, *i.e* a seasonally differenced with a seasonal MA(1) component, model was selected by the algorithm.

Classical or ARIMA model?

Examining the residuals of the classical model created above, one can see that the model performs well.

```
    Ljung-Box test

data:  Residuals from ETS(A,N,A)
Q* = 19.63, df = 22, p-value = 0.6062

Model df: 14.   Total lags used: 36
```

However, the p-value associated with the Ljung-Box test is lower than that associated with the

[8] Rob J. Hyndman and Khandakar (2008).

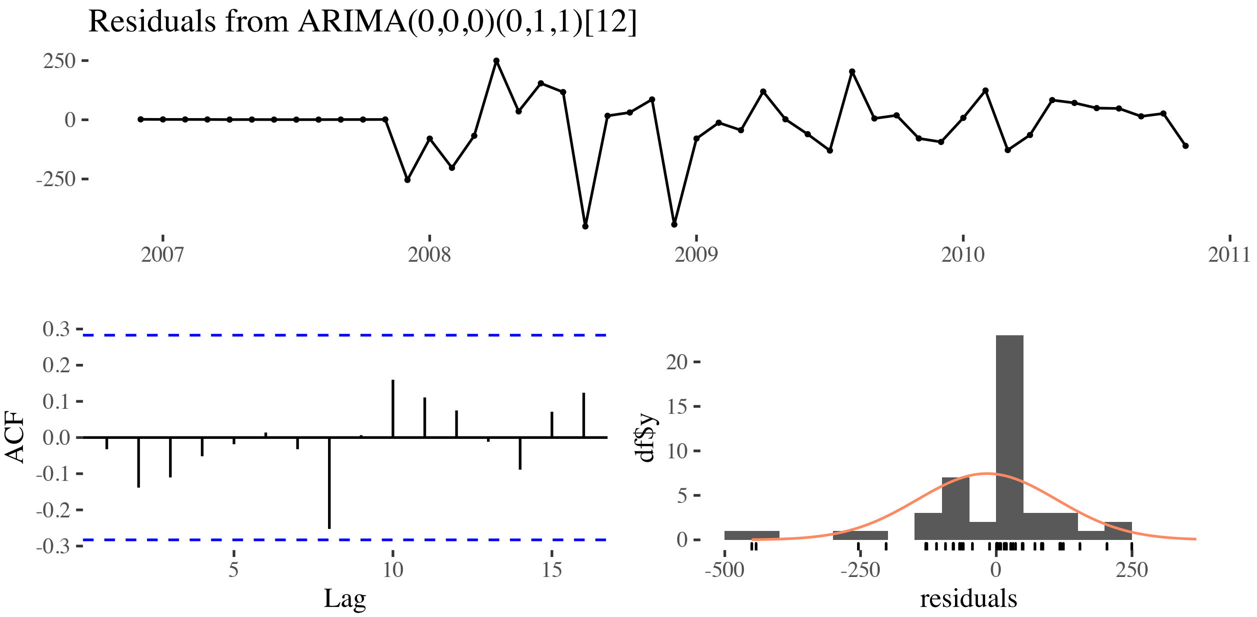


Figure 15: ARIMA model for the power consumption data suggested by `auto.arima`.

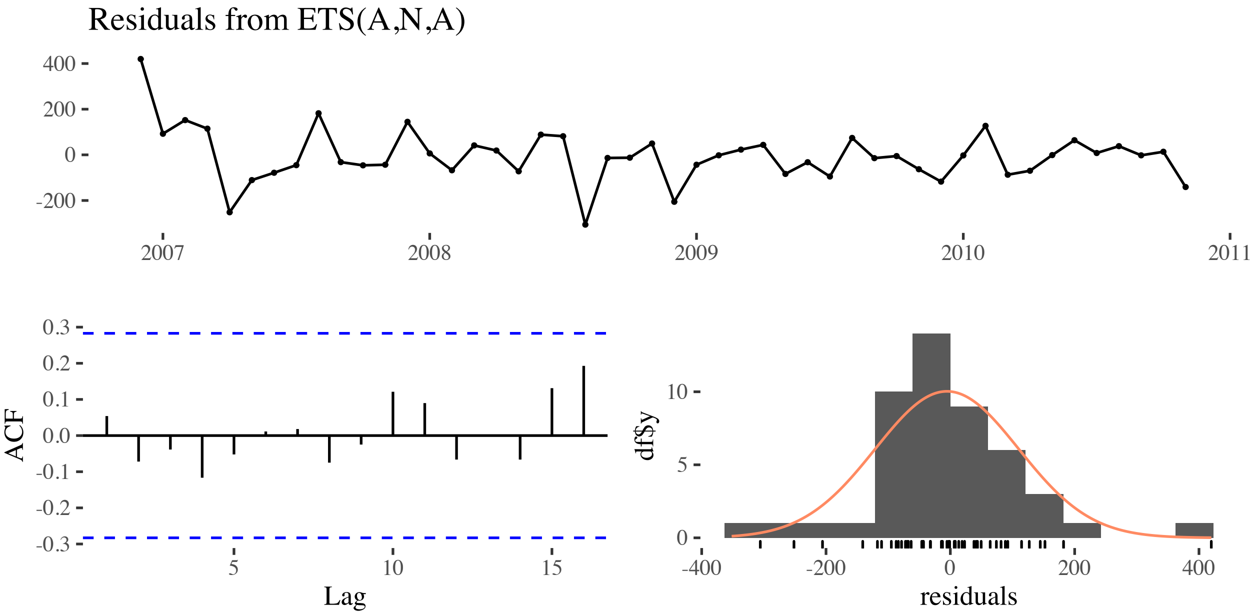


Figure 16: Residuals from the ETAS(A,N,A) classical model.

ARIMA(0,0,0)(0,1,1)$_{12}$ model. One could say, with some justification, that the ARIMA model performs somewhat better than the classical model.

Forecasts

Twelve month forecasts for the next periods are generated next. Figure 17 shows the 12 month forecasts plotted for both the additive and multiplicative classical models.

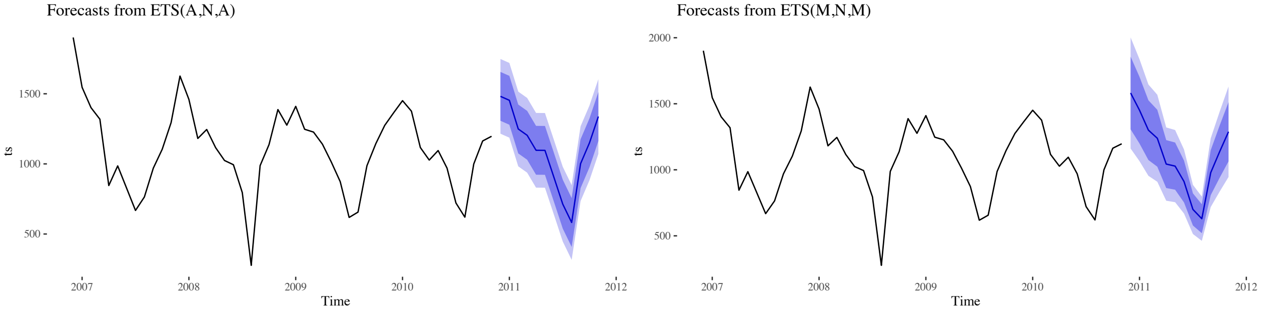


Figure 17: 12 month forecast for (a) classical additive ETS(A,N,A) and (b) multiplicative (M,N,M) models.

Figure 18 shows the 12 month forecasts plotted for both the `auto.arima` and the ARIMA(0,0,0)(0,1,0)$_{12}$ models.

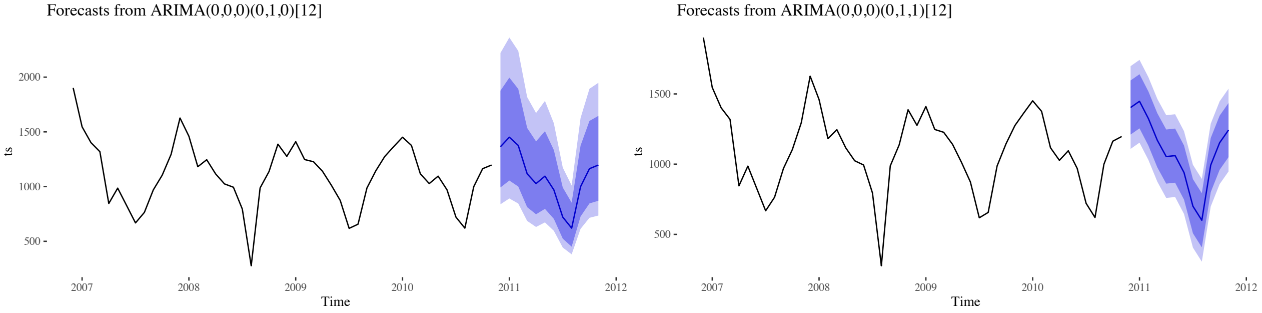


Figure 18: 12 month forecast for (a) auto.arima and (b) ARIMA(0,00)(0,1,0) 12 models.

The power of prediction of the various models, classical and ARIMA, can be checked by forecasting the last periods of the original time series and comparing them with the actual observed values for such period. This was done by sub-setting the original time series so as to exclude the last 11 months and fitting the models on that reduced time series. A plot at Figure 19 was then generated of the full time series, including the last 11 months and overlaid on same were the predictions from the four models: (a) classical additive; (b) classical multiplicative; the `auto.arima` model and; (d) the ARIMA(0,0,0)(0,1,0)$_{12}$ model fitted manually.

Ultimately, the forecasts produced by all four models are very similiar. However, it looks like the auto.arima (ARIMA(0,0,0)(0,1,1)) and the Multiplicative classical model produce the best forecasts.

A table that compares the accuracy measures for all four models is provided at Table 1. It was produced using `forecasts::accuracy`.[9] It is obvious that the forecasts produced by the ARIMA(0,0,0)(0,1,1)$_{12}$ (as produced by `auto.arima`) and the ARIMA(0,0,0)(0,1,0)$_{12}$ model created manually are identical. Based on the criteria used in the `accuracy` function, it appears that the classical multiplicative model produces the best forecasts.

[9] Rob J. Hyndman and Khandakar (2008).

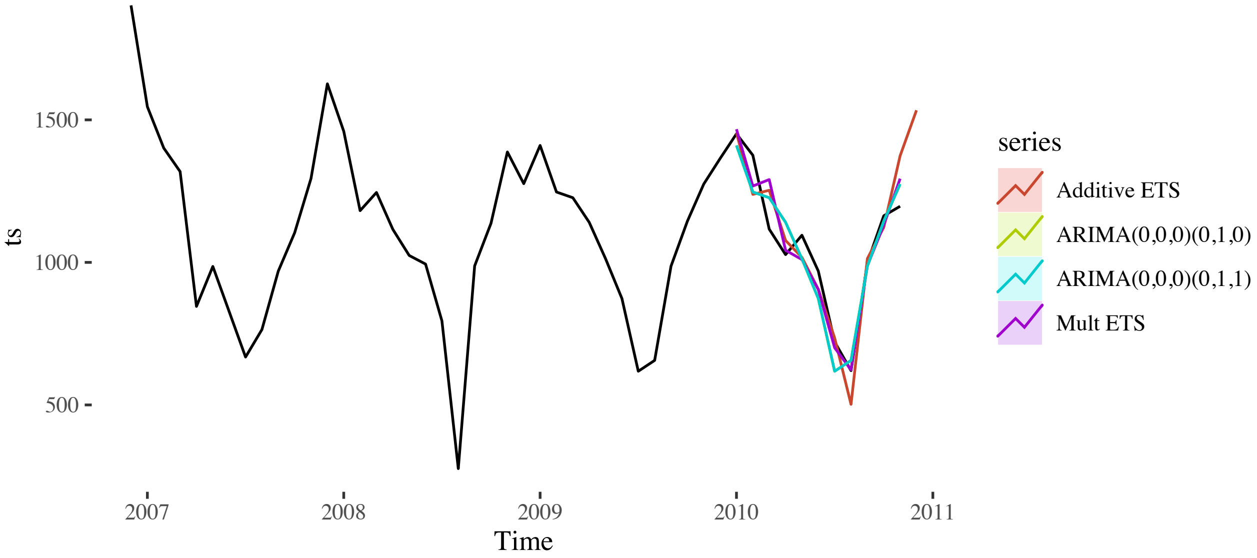


Figure 19: Comparison of the forecasts generated by the four models with the actual time series.

Table 1: Comparison of model accuracies.

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Additive ETS	3.954395	94.07593	75.60231	0.9838583	7.415710	0.5735655	-0.2481229	0.3871664
Multiplicative ETS	1.624117	76.77299	57.03342	0.2189012	5.072834	0.4326905	-0.2389530	0.2728913
ARIMA(0,0,0)(0,1,0)	13.206289	84.11647	74.55999	1.2329235	7.272498	0.5656578	0.0536573	0.3245616
ARIMA(0,0,0)(0,1,1)	13.206289	84.11647	74.55999	1.2329235	7.272498	0.5656578	0.0536573	0.3245616

References

Hyndman, Rob J., and George Athanasopoulos. 2018. *Forecasting: Principles and Practice.* 2nd edition. Lexington, Ky.: Otexts, online, open-access textbook.

Hyndman, Rob J, and Yeasmin Khandakar. 2008. "Automatic Time Series Forecasting: The Forecast Package for R." *Journal of Statistical Software* 26 (3): 1–22. https://www.jstatsoft.org/article/view/v027i03.

Linde, Peter. 2005. "Seasonal Adjustment." Report. Statistics Denmark.

Montgomery, Douglas C., Cheryl L. Jennings, and Murat Kulahci. 2008. *Introduction to Time Series Analysis and Forecasting.* Wiley Series in Probability and Statistics. Hoboken, N.J: Wiley-Interscience.

Moritz, Steffen, and Thomas Bartz-Beielstein. 2017. "imputeTS: Time Series Missing Value Imputation in R." *The R Journal* 9 (1): 207–18. https://doi.org/10.32614/RJ-2017-009.

Trapletti, Adrian, and Kurt Hornik. 2020. *Tseries: Time Series Analysis and Computational Finance.* https://CRAN.R-project.org/package=tseries.

Wongoutong, Chantha. 2020. "Imputation for Consecutive Missing Values in Non-Stationary Time Series Data." *Advances and Applications in Statistics* 64 (October): 87–102. https://doi.org/10.17654/AS064010087.