Assignment 3
Estimating ARMA Processes and Seasonal Processes

Christian Glissov, s146996

April 12, 2018

DTU Compute - Institute for Mathematics and Computer science
Time Series Analysis 02417 April 12, 2018
Lasse Engbo Christiansen

Technical University of Denmark

Contents

Question 3.1

In this assignment the focus will be on predicting the electricity consumption from a district in Malmø. The short data set was chosen due to computational limits. The data given consists of the following:

- Date: Date for observation
- Hour: Hour within day for the observation
- Power: The consumption for that hour in `MWh`

Before plotting the data, the date will be converted into a `POSIX` format as given in the hint of the assignment. Finally two separate data sets will be made, a test data set consisting of excluded observations from 2017-10-15 and onwards for model validation and a training data set from 2017-08-06 to 2017-10-14. The summary statistics can be seen below:

Table 1: Summary Statistics

	Date	Power
Min.	2017-08-06	3.550
1st Qu.	2017-08-24	5.208
Median	2017-09-11	5.977
Mean	2017-09-10	6.100
3rd Qu.	2017-09-29	6.903
Max.	2017-10-17	10.681

Each date consists of a 24 hour period, except the first observation which starts at 8:00:00.

The entire time series plotted gives the following:

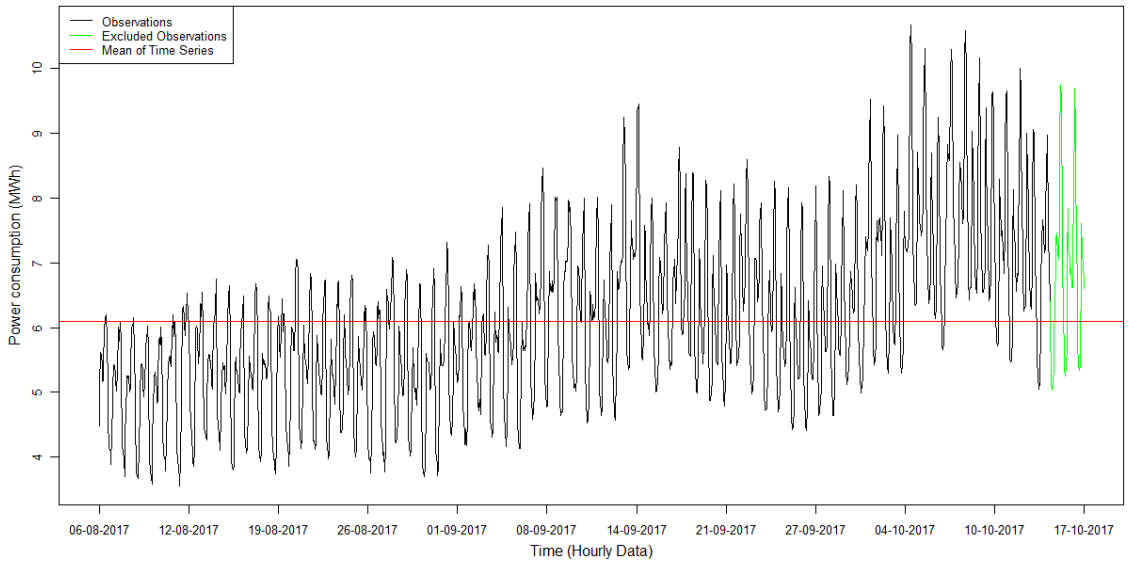

Figure 1: Entire time series, green colour indicates the excluded data.

One can notice the time series is not stationary and the variance seems to be heteroscedastic, with subperiods of increased variance. The trend seems to be increasing, probably because the winter months are arriving, which means people stay indoor more watching television, playing games etc. and electric heating might be a factor too. Finally it can be seen that the time series consists of certain seasons and oscillations. This seasonality will be further explained later.

First the time series will be transformed to make the variance as constant as possible. This is done in two ways. First from chapter 6.7 in [1] the range and the mean will be plotted against each other to find an optimal λ for the Box-Cox transformation in (eq. 1).

$$y_\lambda = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \lambda \neq 0 \\ \log(y) & \lambda = 0 \end{cases} \tag{1}$$

The data has been divided into sequential groups consisting of a 24 period (1 day) based on the daily seasonality of the data.

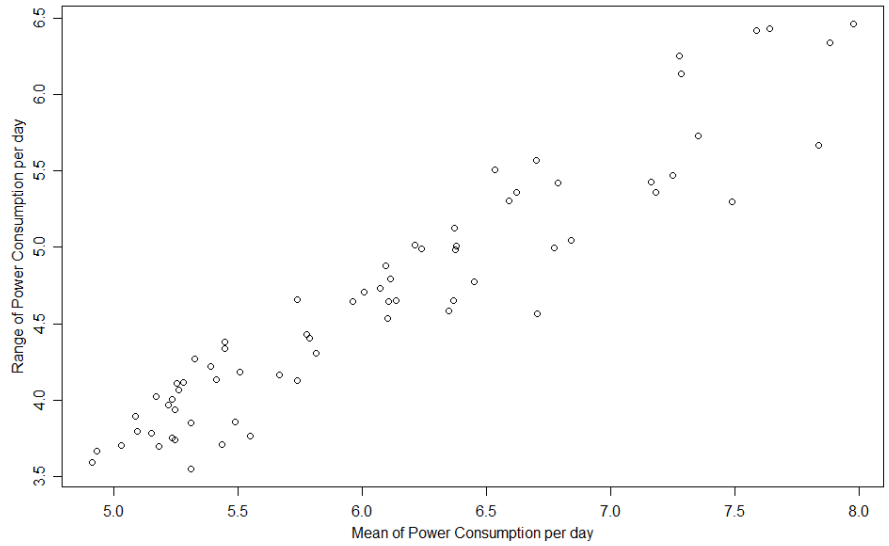

Figure 2: The relationship suggest a log transformation, $\lambda = 0$.

A clear linear dependency between the range and the mean can be seen, based on chapter 6.7 in [1] this suggests a log transformation. However the linear dependency is not perfect (a straight line), therefore based on the ideal of the data being normally distributed, one can optimize λ which maximizes the profile log-likelihood in (eq. 2) found by the log of the normal distribution and the change-of-variable technique.

$$nl_p(\lambda; \mathbf{y}) = -\frac{n}{2}\log(\hat{\sigma}) - \frac{n}{2}\log(2\pi) - \frac{n}{2} + \sum \log(\frac{dy_\lambda}{dy}) \tag{2}$$

The inbuilt R package `MASS` contains a function (`boxcox()`) to do this, which is used. The result can be seen in the plot below:

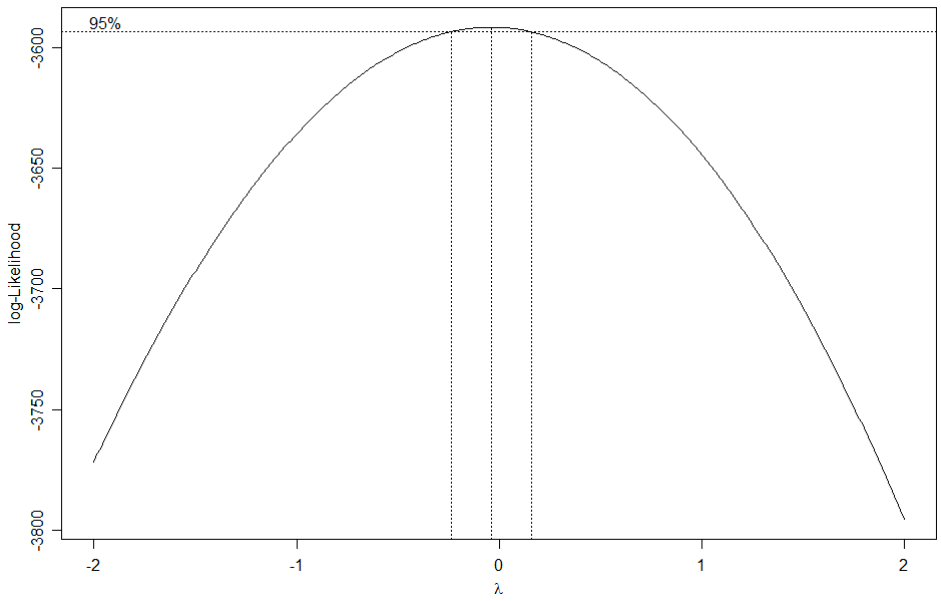

Figure 3: The likelihood is maximised for $\lambda = -0.038$.

The transformation will therefore be a Box-Cox transformation with $\lambda = -0.038$, this will almost give the same values as when using a log-transformation, $\lambda = 0$. However to be exact and maximize the likelihood a transformation with $\lambda = -0.038$ is used. After the transformation another Box-Cox will be made to see if another transformation is needed.

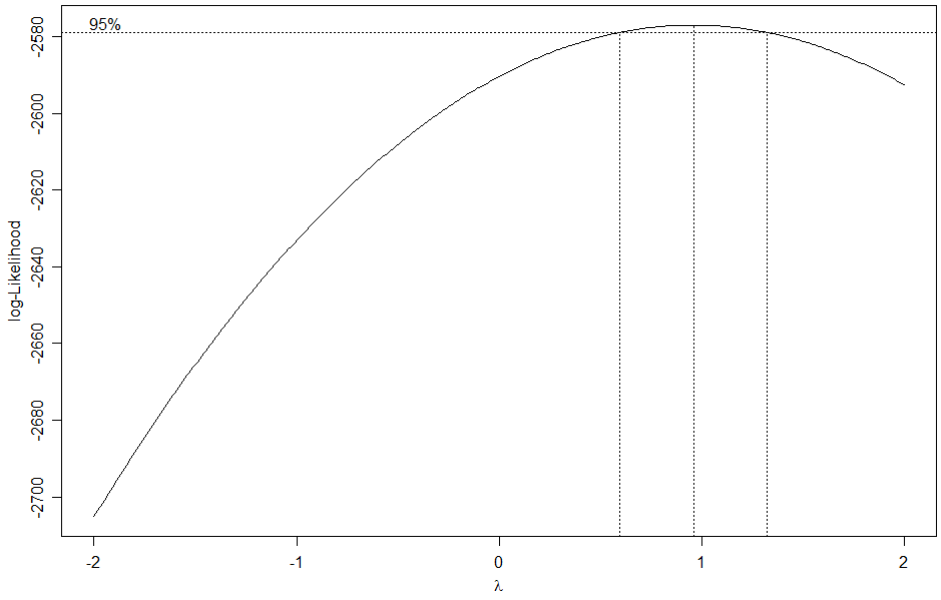

Figure 4: 1 is within the confidence interval of λ.

Since 1 is in the confidence interval of λ another transformation is not necessary. The series is plotted again with the transformation in figure (5)

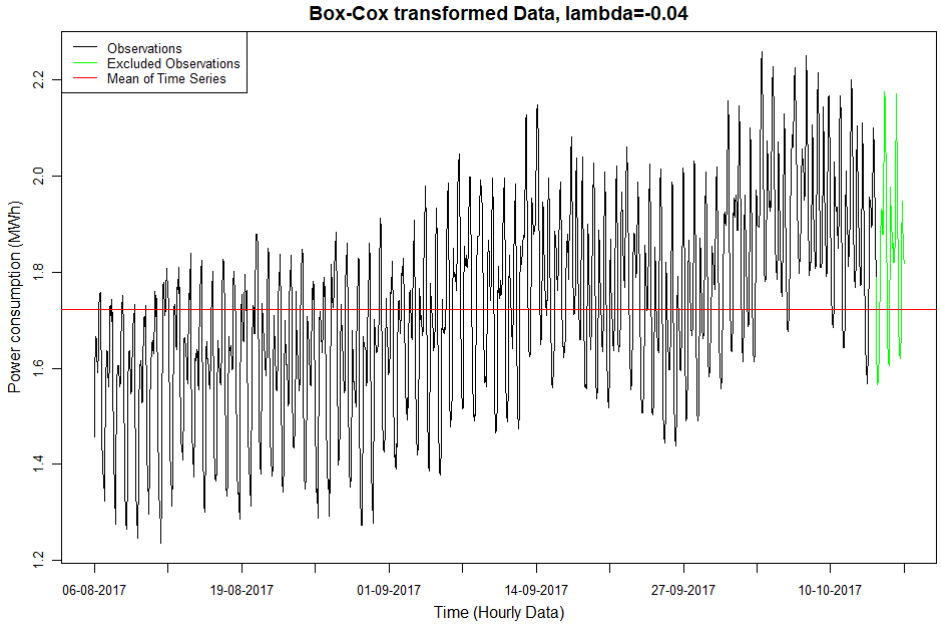

Figure 5: Transformed series.

A slight improvement can be observed. Now the data will be made stationary. First a closer look at the seasonality first for the average daily power consumption:

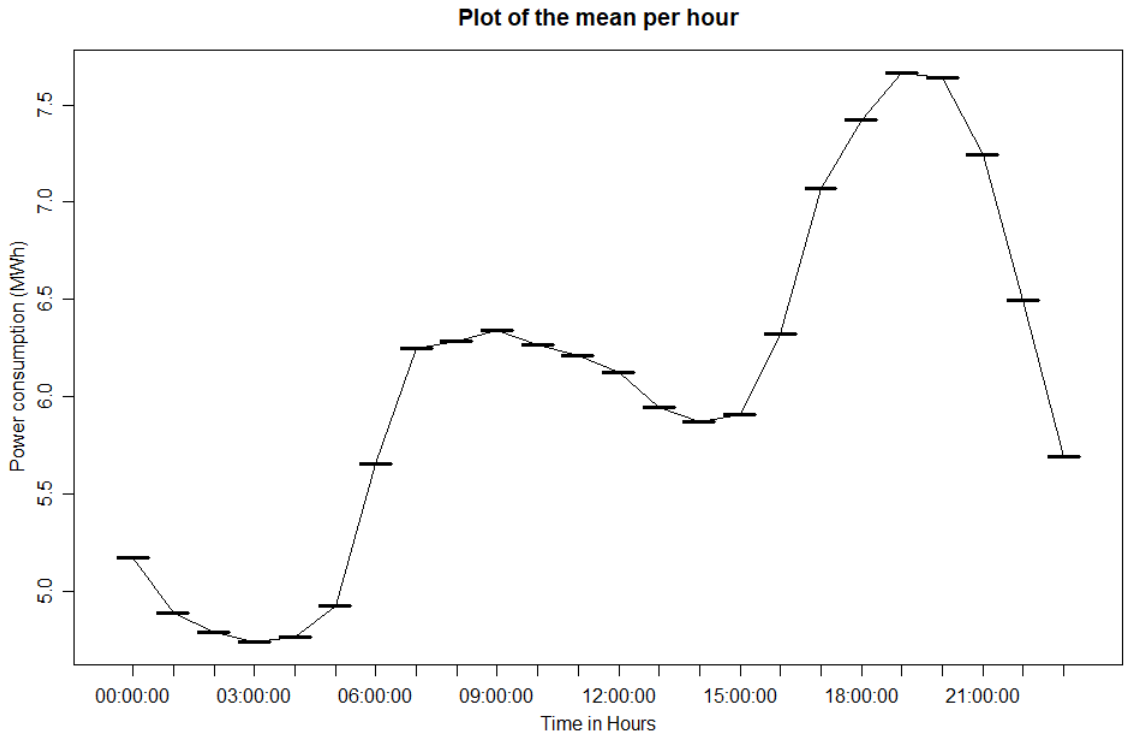

Figure 6: Average daily power consumption for each hour.

Looking at the daily power consumption of each day in each week.

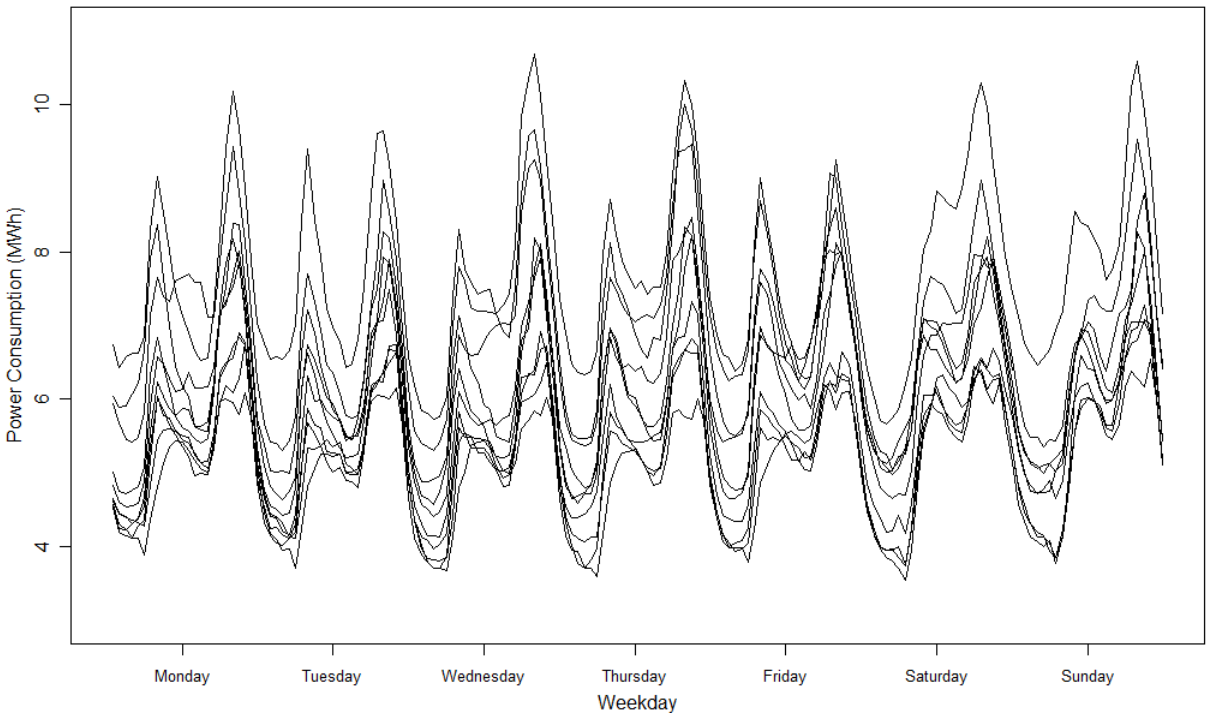

Figure 7: Daily consumption of each day in each week.

A clear seasonality can be seen based on each week day, where the pattern almost identically repeats itself. Due to the daily frequency, see figure (6) and (7), a seasonal differencing of lag 24 will be used.

$$y'_t = y_t - y_{t-24}$$

After the differencing the time series can be seen in the following figure:

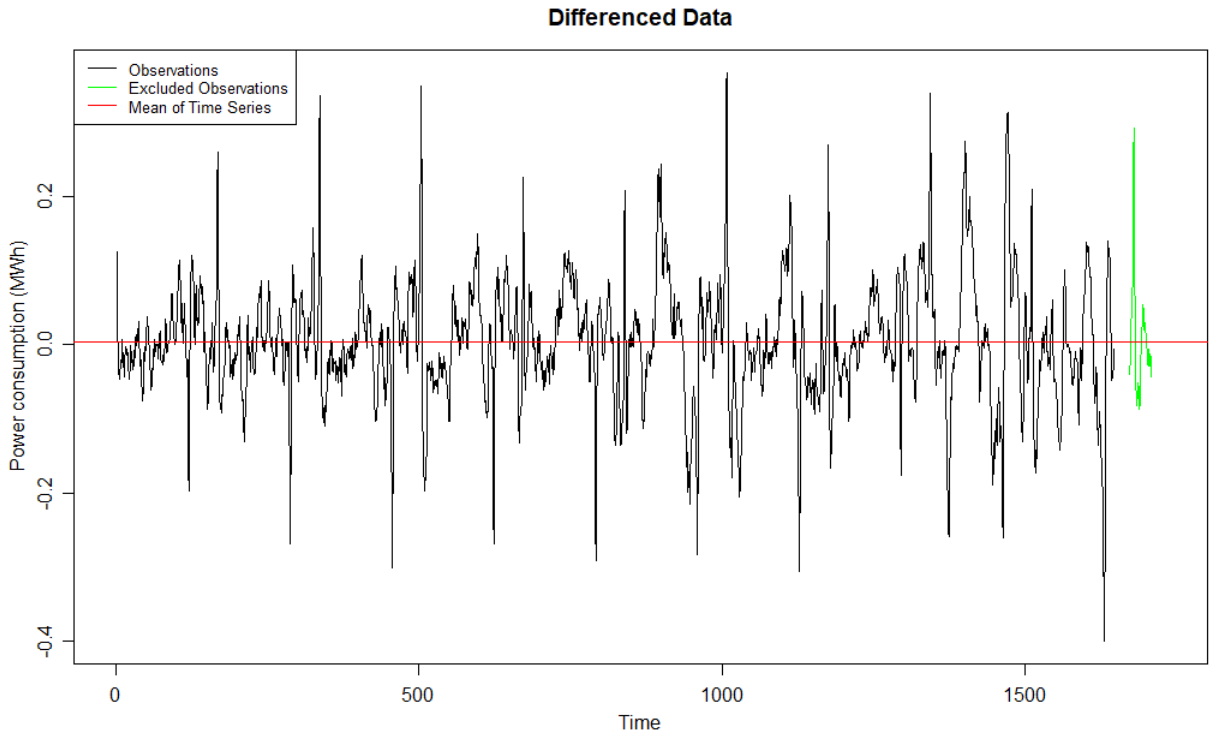

Figure 8: Seasonal differencing with lag 24.

The series looks stationary. To check if it's sufficiently stationary one can look at the ACF plot:

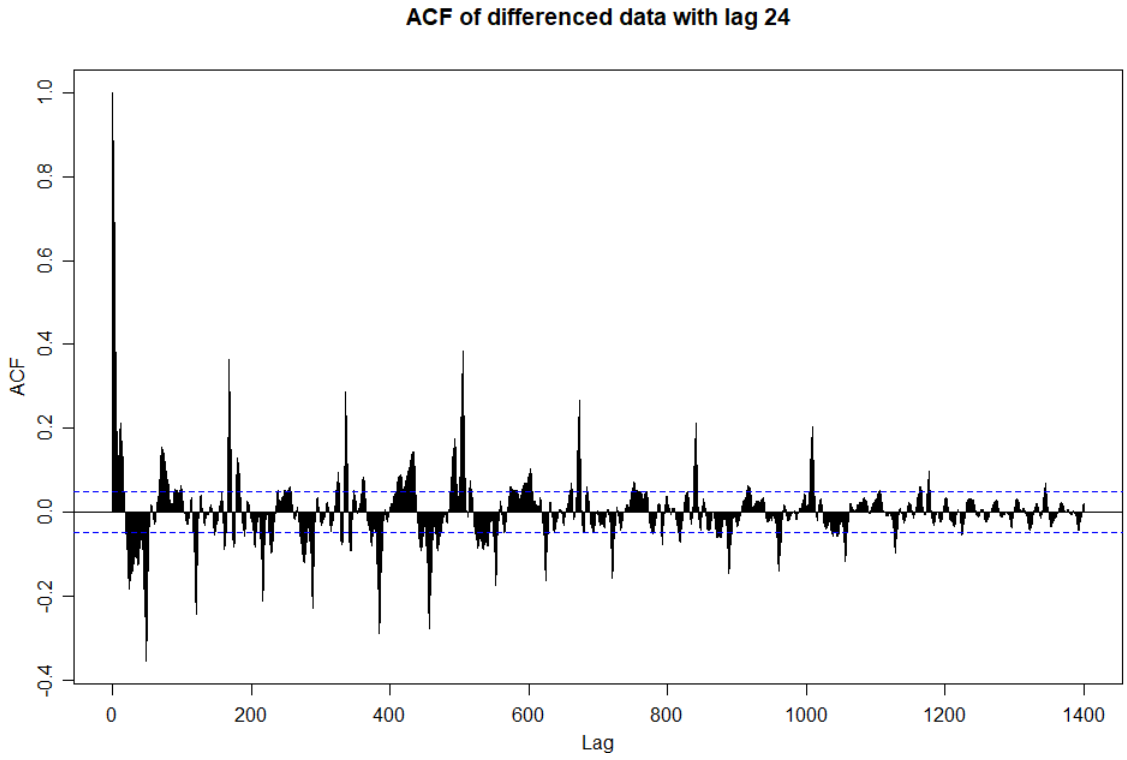

Figure 9: ACF of seasonal differenced data

It can be seen that the ACF slowly converges to 0, a faster convergence would have been preferred, it is an improvement compared to a 1-lag differencing, see appendix figure (27). However there seem to be multiseasonality within the data. For a clearer picture of the seasonality, a weekly seasonality might also be relevant. One can see in figure (10), that some weekdays are expected to have a higher power consumption than others, especially around weekends, this is the case for each week. Finally, figure (7) indicates that a period of increase and decrease in power consumption happens every 12-hour or half a day, this should be taken into account when deciding the model order of the ARIMA.

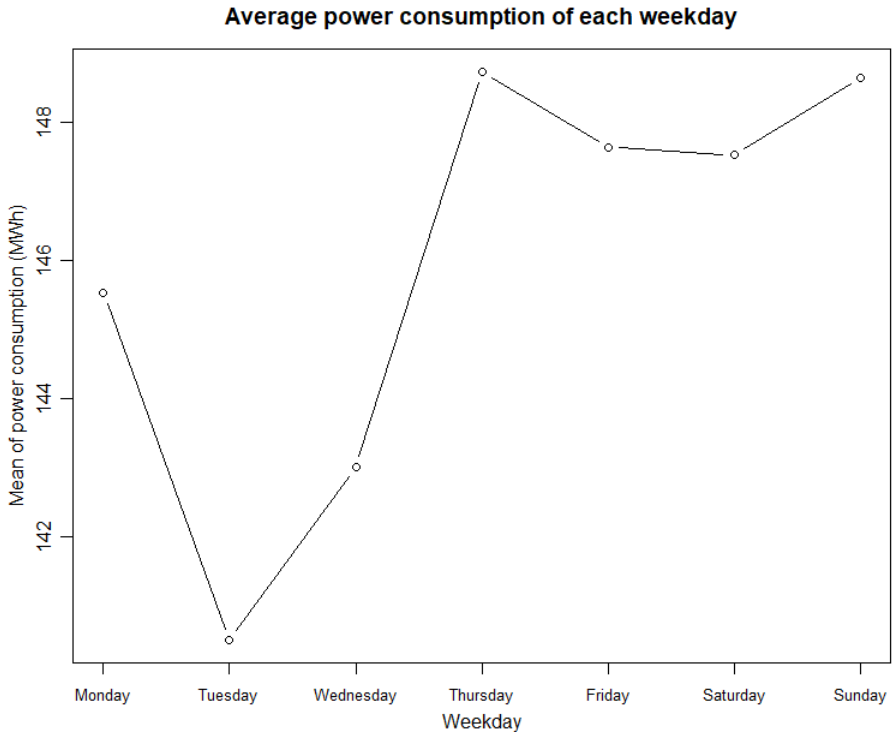

Figure 10: Mean of each week day.

Question 3.2

Looking at the ACF and PACF of the power consumption in figure (11) it is seen that the ACF values don't converge towards 0 sufficiently fast enough, this is due to the non-stationarity.

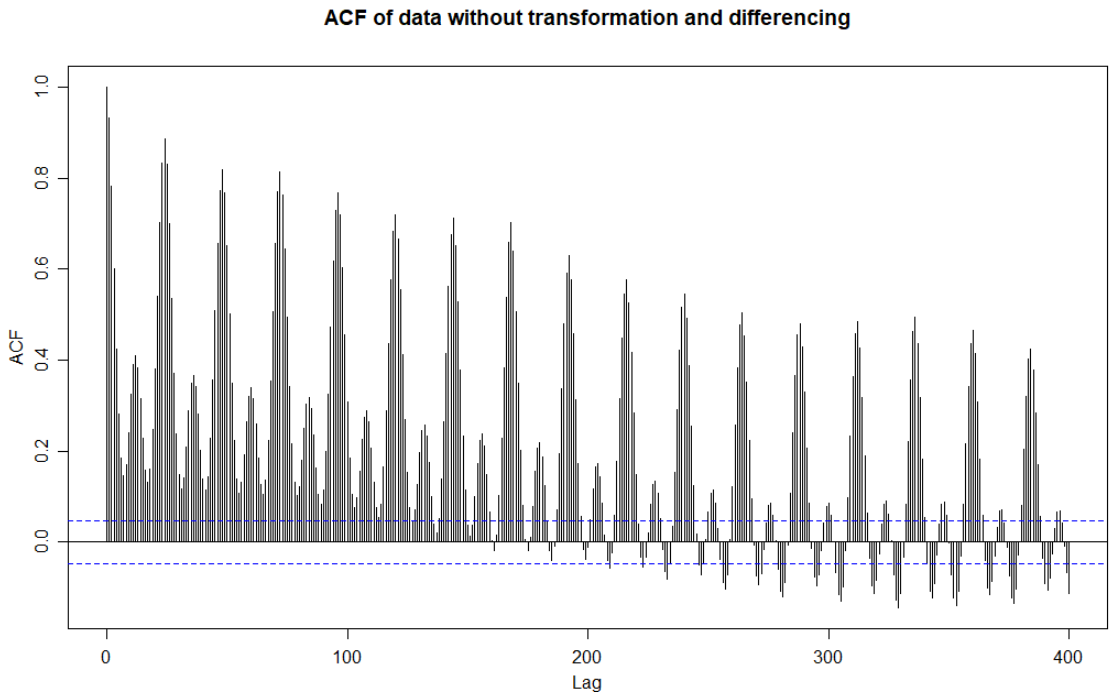

Figure 11: ACF of original data including two weekly seasons

Both the ACF and PACF were made using the inbuilt functions in `R`, `pacf()` and `acf()`. The PACF of the original data can be seen below in 12.

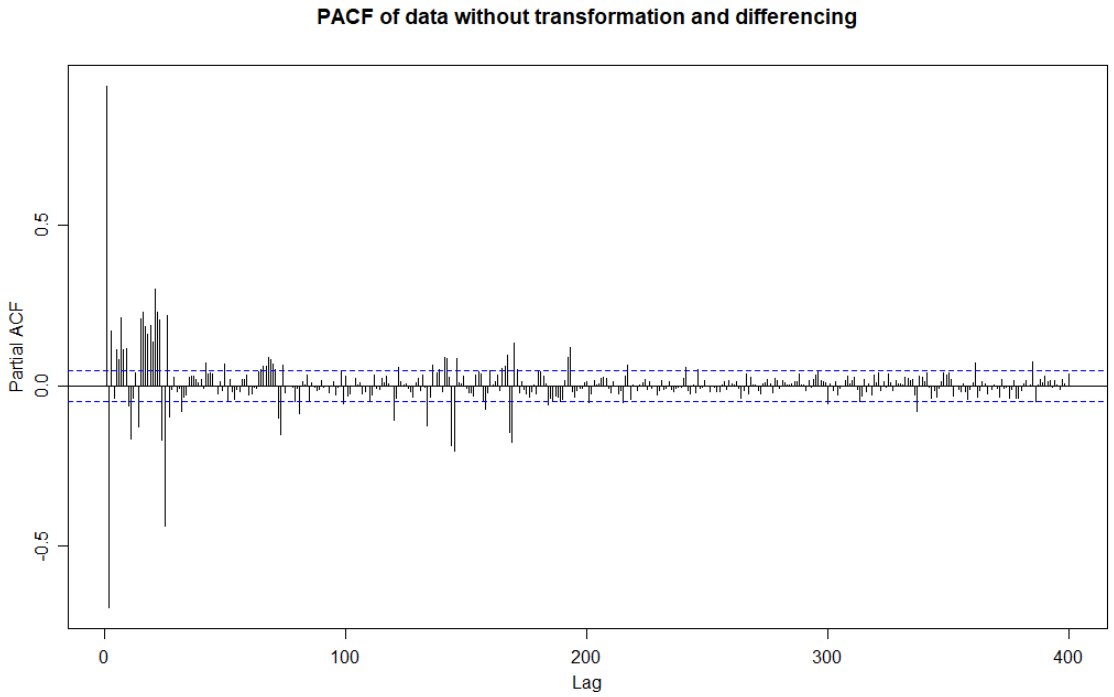

Figure 12: PACF of original data including two weekly seasons

Showing the ACF and PACF for only the transformation, will only scale the ACF and PACF, but not make a change to the structure of the correlation, this figure has been omitted, but can be seen in appendix if necessary. The ACF and PACF of the transformed and differenced data can be seen below.

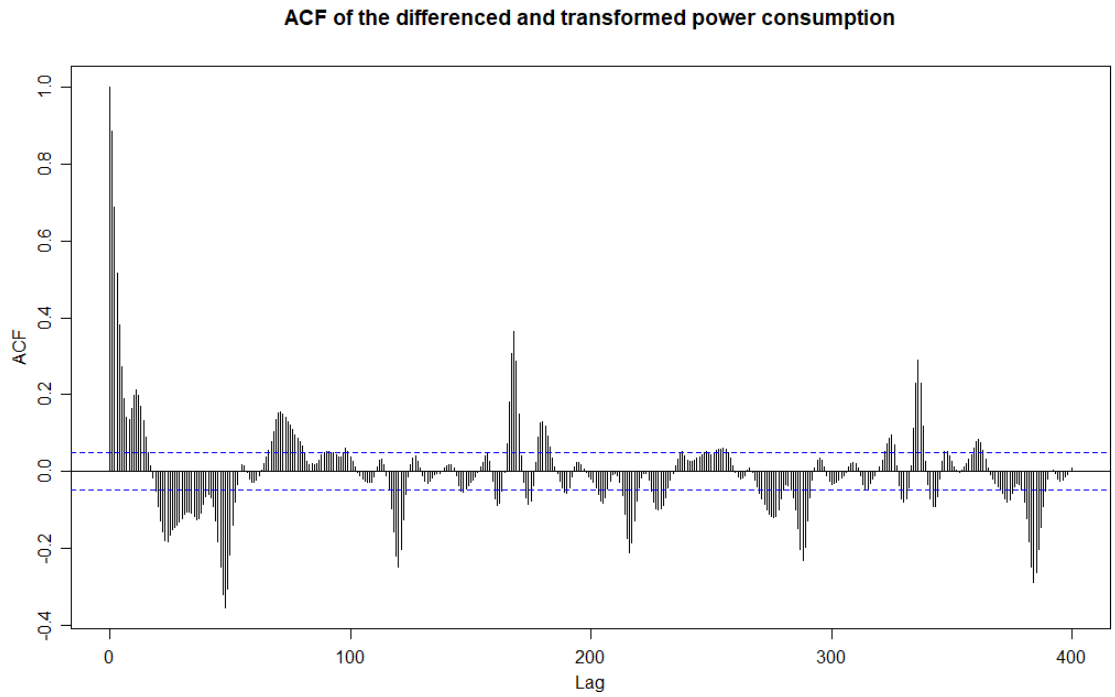

Figure 13: ACF of differenced and transformed data including two weekly seasons

If one wishes to see it for an extended lag, then the same plot up to lag 1400 can be seen in figure 9.

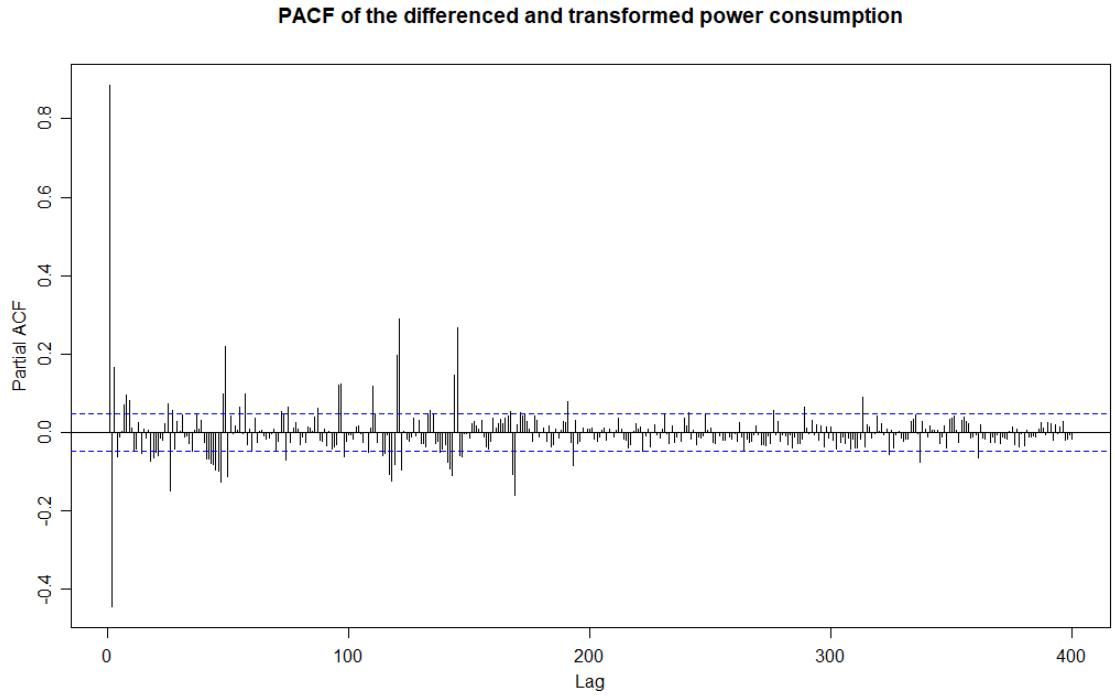

Figure 14: PACF of differenced and transformed data including two weekly seasons

Comparing the ACF and PACF of the original and transformed/differenced data, it is clear that the transformed and differenced dataset is to prefer. The ACF values converge towards 0, which indicate a stable time series. Looking at figure

13 and 14 a number of initial models can be guessed. Based on the table 6.1 in chapter (6.3.2) in [1], also known as "The Golden Table", some qualified guesses can be made.
It can be seen that the ACF is dominated by damped exponential and sine functions, where in the PACF it's difficult to see. Therefore it seems that $\phi_{kk} = 0$ for $k > 4$ in the PACF. To start of slow an AR(2) model with a seasonal differencing, $(2, 0, 0)(0, 1, 0)_{24}$, will be the initial guess.

Question 3.3

The model selection procedure will follow the model building procedure shown in figure 6.1 in chapter 6.1 of [1]. Based on the theory of the ACF, PACF and the data, one is able to identify an initial model order. By continuing to look at the ACF, PACF and other diagnostic tools of the initial model further parameters and differences can be added.

To select a proper model order an iterative approach will be taken. Selection each parameter, which is deemed relevant through the ACF and PACF, will be added to the model. For each new model the PACF and ACF will be checked following table 6.1, "The Golden Table" in [1], to see if further information can be explained.

When an appropriate model order has been specified, the model parameters will be estimated using the maximum likelihood method or a conditional sum of squares if convergence of the optimization of the maximum likelihood can't be achieved. To check if the model is OK, certain criteria have to be met. One of the criteria is that the residuals should be IID, white noise, to check this the following is used:

1. The residuals should be normally distributed. This can be checked using a qqplot.
2. To see if the residuals are random a sign test will be used.
3. The variation of the residuals is supposed to be uniformly distributed on all frequencies, this will be checked with a cumulated periodogram.
4. Lags in ACF and PACF will need to be 0. The Ljung-Box test will tell if the lags in the auto correlation function can be assumed to be 0. The lags of the Ljung-Box test should cover the seasonality of the time series.
5. A residual plot to observe the residual behaviour and to detect patterns.

Given the model residuals are white noise, an attempt of reducing the model complexity further will be made. This means all insignificant parameters will be removed and models will be compared using the information criteria, AIC and BIC, to find the best and least complex model. If models are nested a likelihood ratio test might be performed to see whether one should choose the complex high order model or the simpler lower order model. If the model is adequately simple, its residuals satisfy the white noise properties and it has a good AIC/BIC score compared to other models, then the model will be used for applications, such as forecasting.

Question 3.4

Notation in this question will be based on the multiplicative seasonal ARIMA notation in chapter 5 of [1], ARIMA$(p,d,q)(P,D,Q)_s$. The first approach to find a model will use an iterative forward selection. Look at the ACF in figure 13 it is seen that it's dominated by a damped exponential function and there might even be some sine functions. For the PACF in figure 14 there are significant lags for $k \leq 4$. From table 6.1 in [1] and to start off slowly an AR model with order $p = 2$ with seasonal differencing is chosen, $(2,0,0)(0,1,0)_{24}$ as an initial guess. Table 6.1 will be used as a guidance throughout the whole process, referencing it each time it is used might therefore be omitted. The coefficients can be seen below:

Table 2: Initial Model

Model : $(2,0,0)(0,1,0)_{24}$	**AR(1)**	**AR(2)**
Coefficients	1.2870	-0.4511
S.E	0.0219	0.0219

All the coefficients are significant. The diagnostic plots can be seen below.

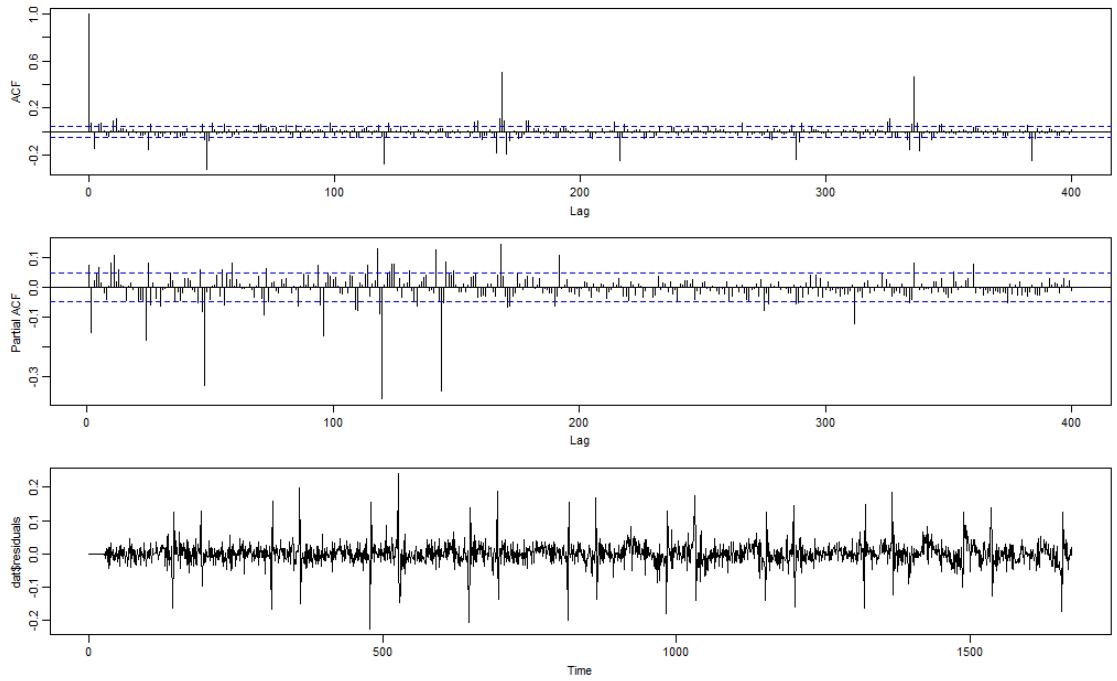

Figure 15: Time series diagnostic plots for $(2,0,0)(0,1,0)_{24}$

For saving time the CSS method might be used for the model selection. Still some significant correlation at second and third lag for the ACF and also some significant lag in the PACF. This time it seems like the PACF fluctuates a bit more, which might indicate an MA parameter is present, but it could also be an increase in the AR order. Both were added, so the model is $(3,0,1)(0,1,0)_{24}$, looking at the standard deviation it is noticed that the additional AR coefficient is less than 2 times the standard deviation and doesn't change the ACF and PACF that much,

so it is insignificant. The chosen model is then $(2,0,1)(0,1,0)_{24}$. Looking if $q=2$ or $p=2$ is better, based on the ACF plot $q=2$ seemed slightly better, so a swap of the q and p order is done. The model is $(2,0,1)(0,1,0)_{24}$

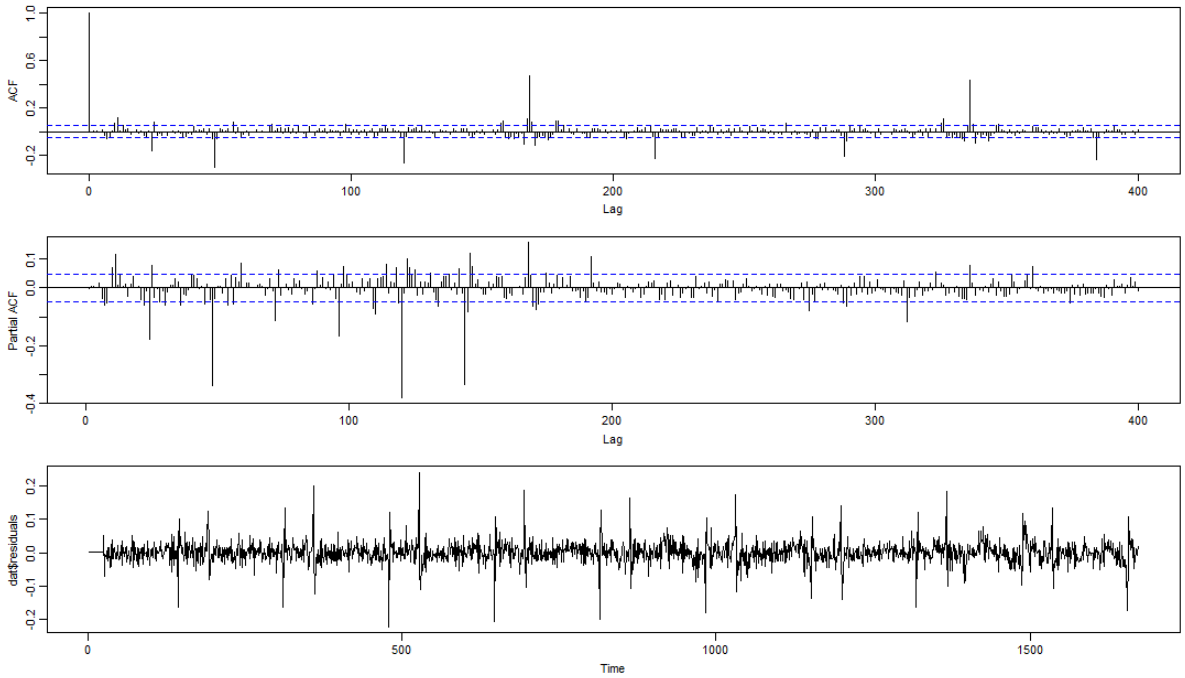

Figure 16: Time series diagnostic plots for $(2,0,1)(0,1,0)_{24}$

Seems to be some very significant seasonality in figure 16. The PACF is dominated with significant lags for the first 200 lags and that ACF seems to have lag spikes every 24 hour and every 168 hour (7 days), which slowly decreases along the lag. First the 24 hour seasonality is explained, it was P=2 and Q=1 removed most of the lag, the diagnostic plots can be seen below.

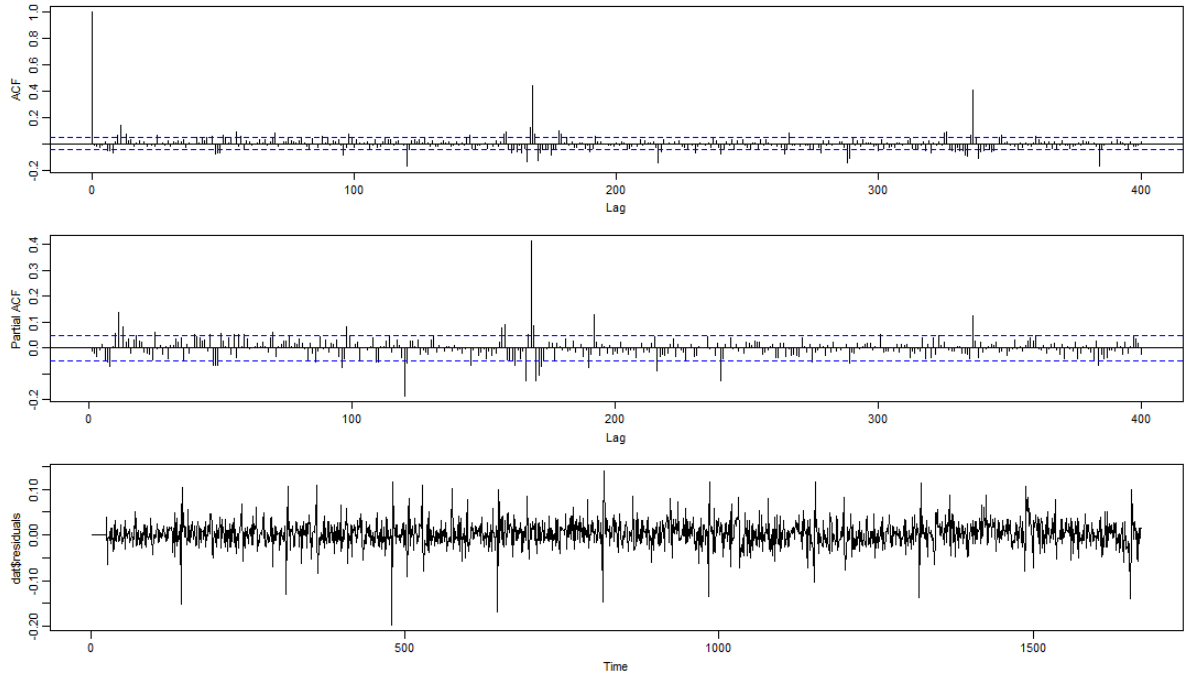

Figure 17: Time series diagnostic plots for $(1,0,2)(2,1,1)_{24}$

Trying to add an AR parameter explaining the seasonal week. The model becomes, $(1,0,2)(7,1,1)_{24}$, where the seasonal AR from 3 to 6 is fixed to 0, so only parameter 1,2 and 7 of the seasonal AR part will be relevant.

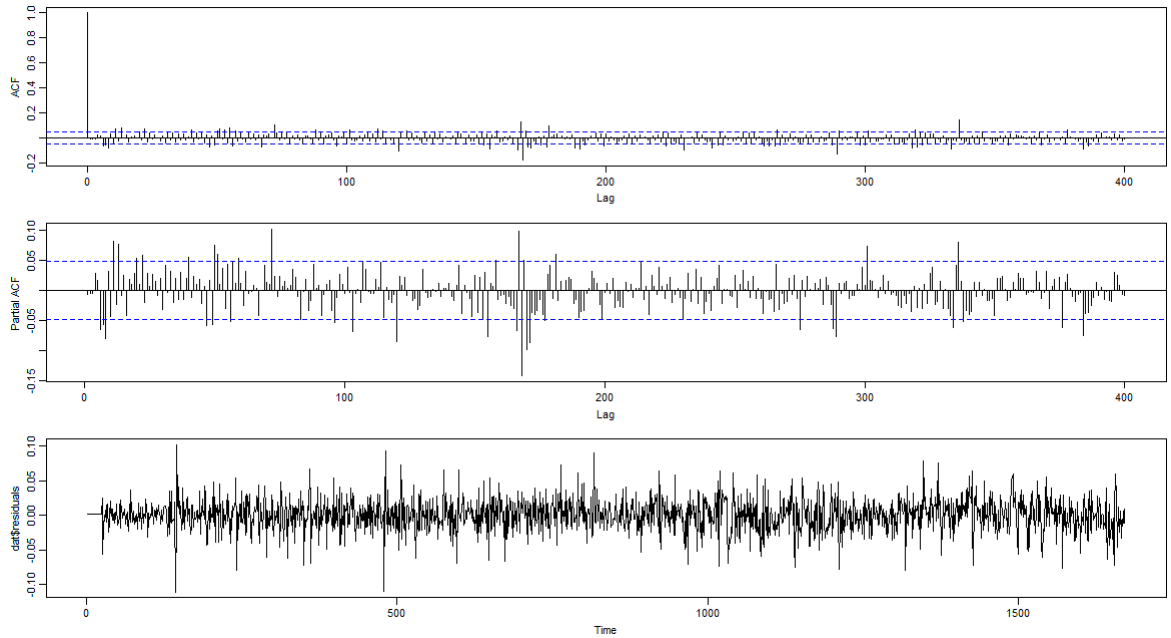

Figure 18: Time series diagnostic plots for $(1,0,2)(7,1,1)_{24}$

It is seen that most of the seasonal lag is now gone, however there are still some significant lag at the start of the PACF and also at lag 168. The seasonality doesn't seem to be very pronounced anymore however. Any further changes didn't seem

to improve the ACF or PACF very much. Every coefficient is significant. A single significant lag still exists in lag 168, although it has been reduced by a lot. There still seem to be a lot of lag in the first 12 hours in the PACF, this corresponds to half a day. A backward selected model will now be made to see if any of the lags at the start can be removed. The initial model will be initialised by modelling half a day with AR parameters. The same was done with MA parameters and both q and p set to 12, however it was not beneficial. To summarise an initial model of $(12, 0, 2)(2, 1, 1)_{24}$ is chosen.

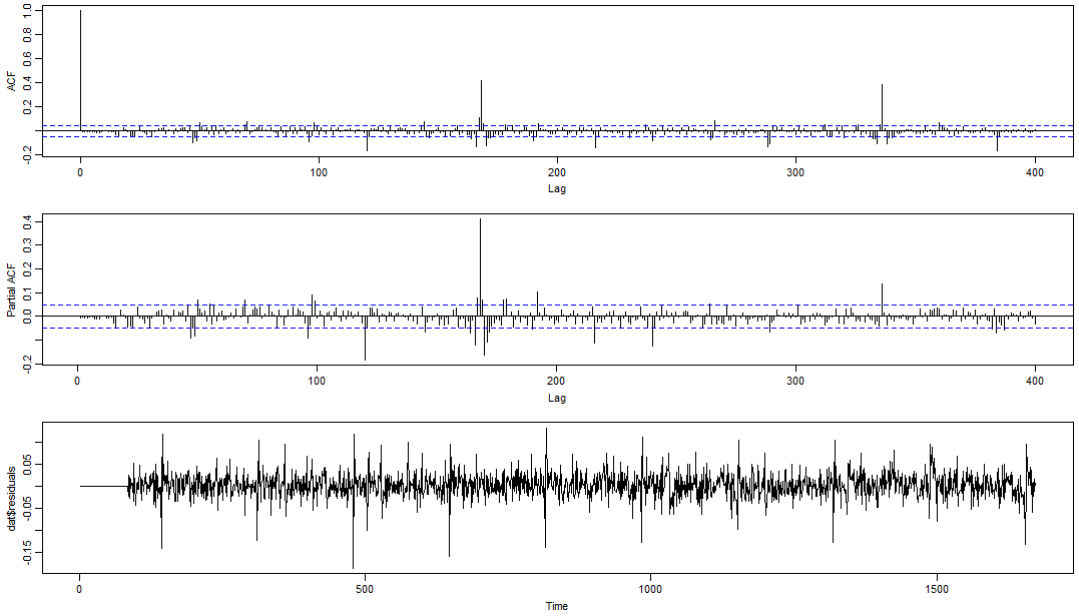

Figure 19: Time series diagnostic plots for reduced $(12, 0, 2)(2, 1, 1)_{24}$ model

It's noticed that the lag in the start of the PACF has been significantly reduced. A clear seasonal tendency is still shown for the weekly lag 168, trying to add a seasonal AR coefficient to lag 168 and the model is re-estimated, $(12, 0, 1)(7, 1, 1)_{24}$.

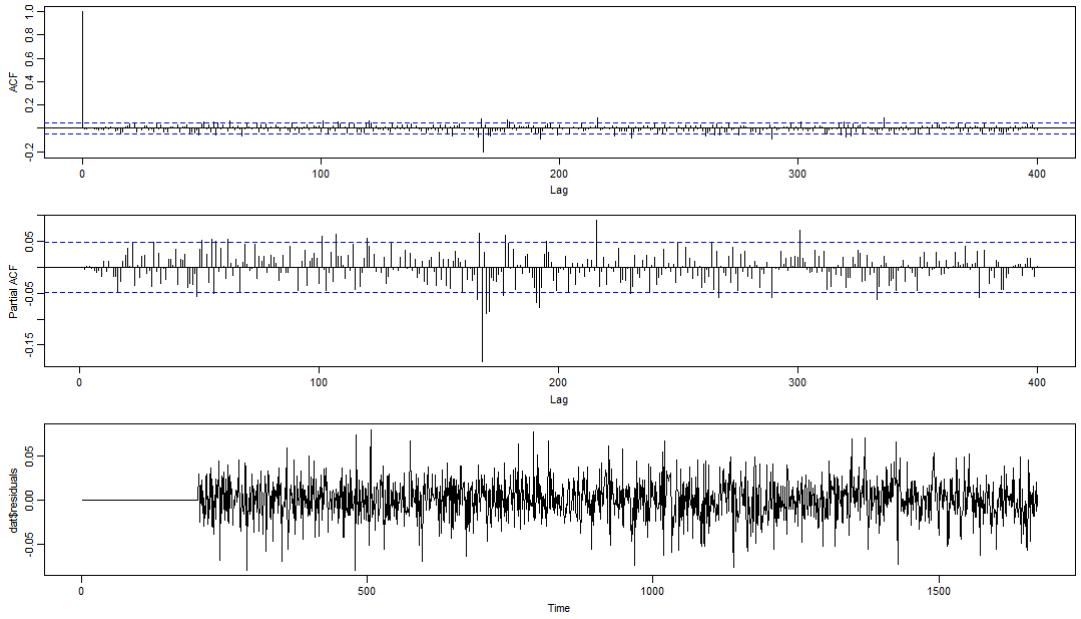

Figure 20: Time series diagnostic plots for reduced $(12,0,1)(7,1,1)_{24}$ model

Still seems to be a fairly large negative lag in PACF for lag 168. However the seasonal tendency has been significantly reduced. The model can only be estimated using conditional sum of squares. It will use 204 conditional observations, by reducing insignificant parameters, the order could be reduced down to $p = 11$, which means 203 conditional observations will be used, approx. 12% of the training data set. A problem with this model is also its complexity. There is a risk of overfitting and the loss of efficiency due to using less observations generally does more harm than the potential bias improvements from defining the weekly lags directly. Therefore $(12,0,1)(7,1,2)_{24}$ is rejected.

The three chosen models will now be reduced and compared, $(1,0,2)(7,1,1)_{24}$, $(1,0,2)(2,1,1)_{24}$ and $(12,0,2)(2,1,1)_{24}$. All estimated using the maximum likelihood in `R`. Only model $(12,0,2)(2,1,1)_{24}$ needs to be reduced, due to insignificant coefficients. A standard normal distribution will be used, because of the large number of observations, which mean the t-distribution is approximately normal, there all coefficients below $2 \cdot S.E$ is insignificant. Now the relevant statistics and coefficients will be shown in tables. For the first model, $(1,0,2)(2,1,1)_{24}$:

Table 3: Model 1 Coefficients

Model 1: $(1,0,2)(2,1,1)_{24}$	AR(1)	MA(1)	MA(2)	SAR(1)	SAR(2)	SMA(1)
Coefficients	0.8290	0.5287	0.0949	0.3923	-0.2697	-0.8360
S.E	0.0182	0.0307	0.0307	0.0271	0.0259	0.0161

Table 4: Relevant statistics for model 1

AIC	BIC	N	Log Likelihood	σ^2	RSS
-6988.88	-6951.031	1672	3501.44	0.0008207	1.352609

For model 2, $(1,0,2)(7,1,1)_{24}$:

Table 5: Model 2 Coefficients

Model 1: $(1,0,2)(7,1,1)_{24}$	AR(1)	MA(1)	MA(2)	SAR(1)	SAR(2)	SAR(7)	SMA(1)
Coefficients	0.9176	0.2708	0.0866	0.2053	-0.1292	0.6019	-0.8561
S.E	0.0115	0.0284	0.0265	0.0221	0.0204	0.0198	0.0221

Table 6: Relevant statistics for model 2

AIC	BIC	N	Log Likelihood	σ^2	RSS
-7601.9	-7558.638	1672	3808.95	0.0005404	0.8906178

Finally for model 3, which has been reduced to $(11,0,2)(2,1,1)_{24}$.

Table 7: Model 3 Coefficients

Model 1: $(11,0,2)(2,1,1)_{24}$	AR(2)	AR(5)	AR(6)	AR(10)	AR(11)
Coefficients	0.6584	0.0517	-0.0934	0.0777	0.1485
S.E	0.0297	0.0256	0.0223	0.0198	0.0198
	MA(1)	MA(2)	SAR(1)	SAR(2)	SMA(1)
Coefficients	1.3211	0.4527	0.3491	-0.2633	-0.8484
S.E	0.0235	0.0272	0.0276	0.0260	0.0156

Table 8: Relevant statistics for model 3

AIC	BIC	N	Log Likelihood	σ^2	RSS
-7056.74	-6997.261	1672	3539.37	0.0007826	1.28971

Now the assumptions will be checked. First a sign test using a binomial distribution, $Binom(N_{sign}-1, \frac{1}{2})$, where N_{sign} is the changes of sign. The test will be made of all three models, the result can be seen in table 9.

Table 9: Sign test of each chosen model

Sign Test	P(Success)	95% Confidence Interval
Model 1	0.513465	[0.4891952, 0.5376874]
Model 2	0.5140634	[0.4897934, 0.5382840]
Model 3	0.5182525	[0.4939816 0.5424592]

The residuals of all three models can be deemed as random according to the sign test on a 0.05 significance level, since 0.5 is within the interval. Now looking at the `Ljung-Box` test, a lag of up to 48 will be used, looking at 2 daily seasons.

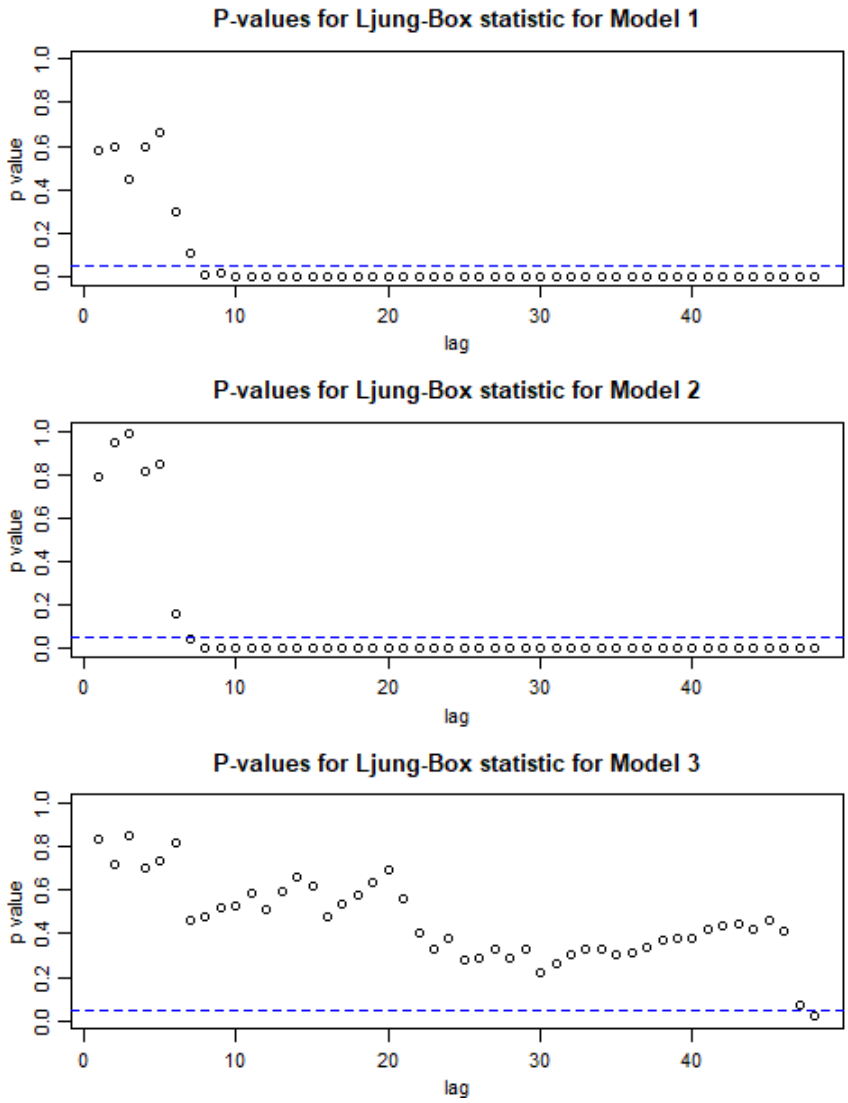

Figure 21: Ljung-Box of all three models.

Looking at the Ljung-Box, it seems to be model 3 that is best, not all information is captured in the time series, as there is still some significant auto-correlation, which can't be assumed to be 0, however it covers the daily seasonality, so it's acceptable. For the two other models, model 2 and 1 seems to be almost equally good. Finally the cumulated periodogram and the qqplot will be made for each model, these can be seen respectively the figures below. For model 1:

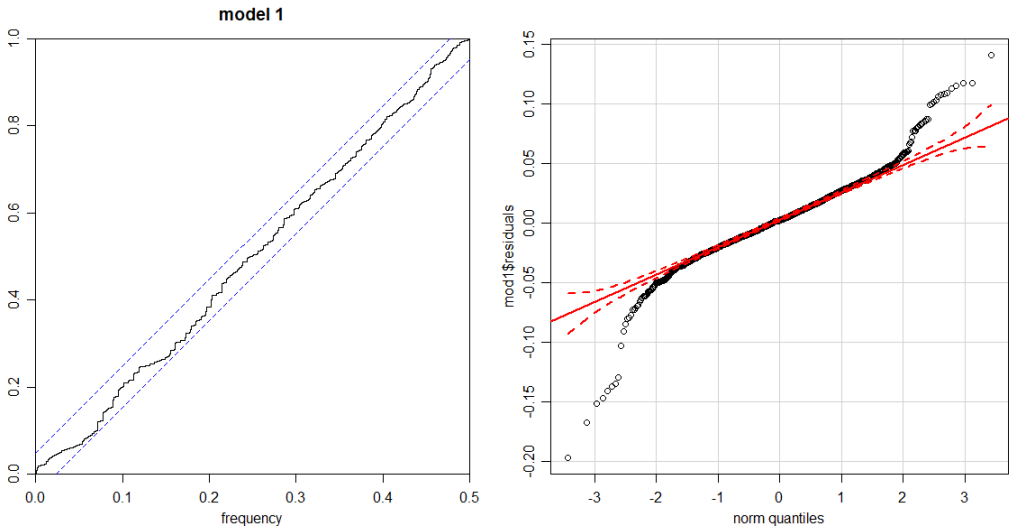

Figure 22: Cumulated periodogram and qqplot for model 1.

For model 2:

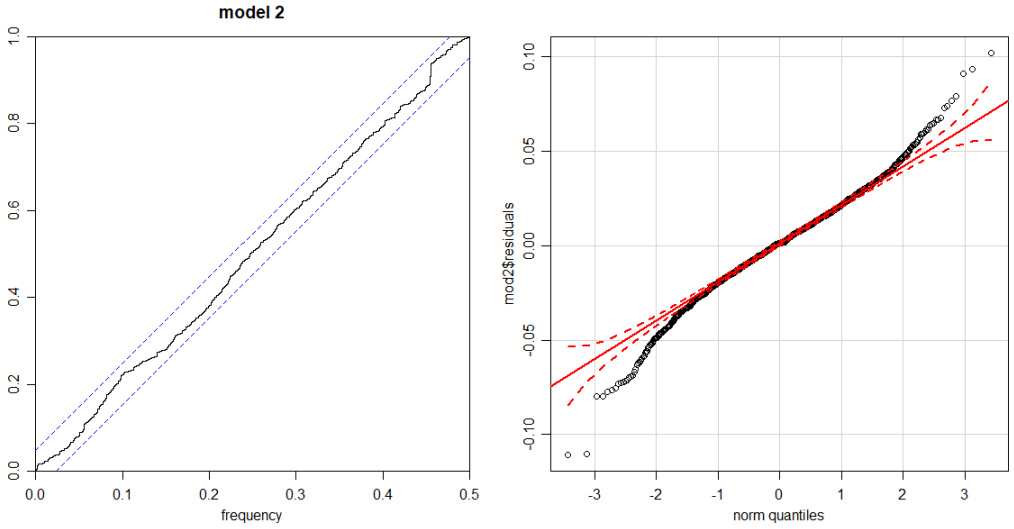

Figure 23: Cumulated periodogram and qqplot for model 2.

For model 3:

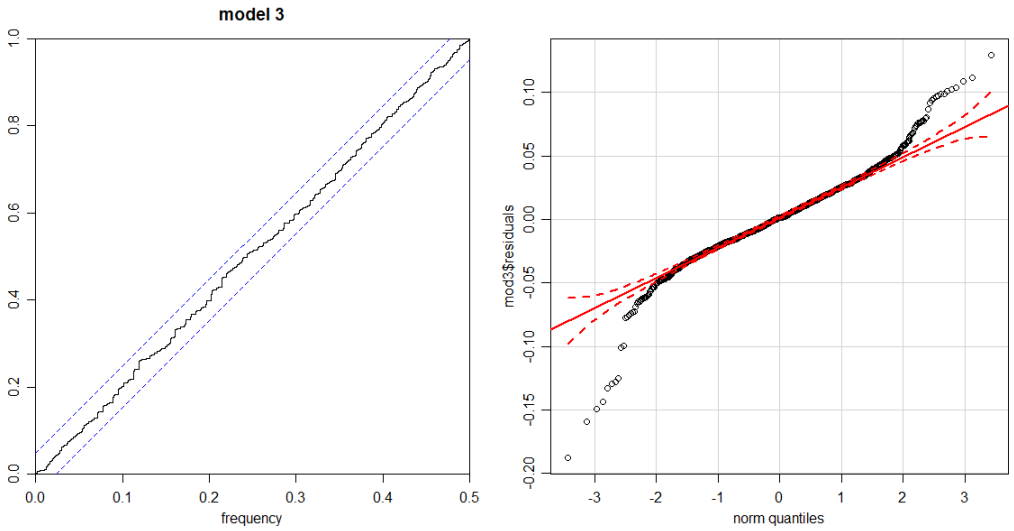

Figure 24: Cumulated periodogram and qqplot for model 3.

All periodograms looks fine and the variance of the residuals can therefore be assumed to be constant at every frequency. The qqplots show that the residuals don't seem to be normally distributed, however model 3 seems to be the best one. The tails in model 2 looks like having most of the data lying close to the line, but when looking closer a lot more of the data lies out of the confidence interval, compared to model 3 with only a few outliers not being in the interval. The periodograms is within the interval for all of the models, with model 3 being once again the best. Finally diagnostic plots can be found of model 1 in 17, for model 2 in 18 and since model 3 was reduced a new diagnostic plots is given in 25

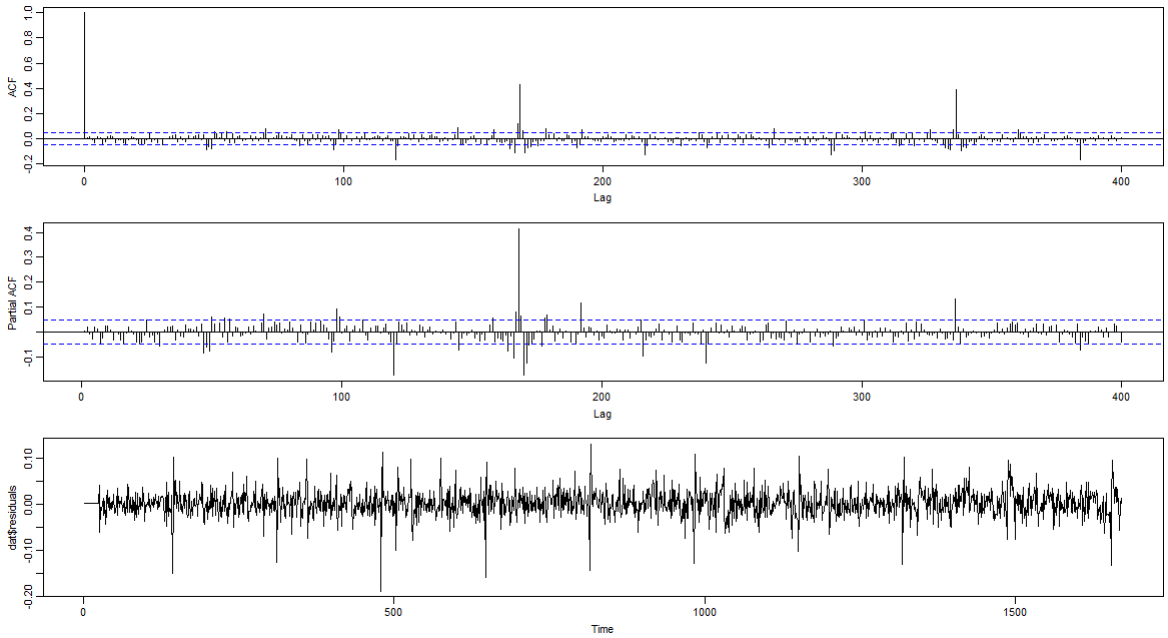

Figure 25: Model 3: $(11, 0, 2)(2, 1, 1)_{24}$

Based on the tests, model 3 seems to fulfil the assumptions the best and having the second highest AIC and BIC. The problem with model 2 is the complexity,

having a seasonal fixed component of order 7, the degree of the polynomial will be up to order 170, which is again very complex. Based on the testing no greater reason was found to choose the complex model 2, and it's therefore rejected.
Since model 1 and 3 have a likelihood close to each other and is nested, a likelihood ratio test is performed:

$$LR = 2log(\frac{L_s(\hat{\theta})}{L_g(\hat{\theta})}) \sim \chi^2(m-n)$$

Where m is number of parameters for the large model $L_g(\hat{\theta})$ and n for the small model $L_s(\hat{\theta})$. In R the value was found to be $1.31e-15$, which is highly significant, therefore the more complex model should be used.
This means model 3 will be used for predictions, not a perfect one, but adequate when looking at the results.

Question 3.5

For the predictions the `forecast` package is used with the function `forecast.Arima()`, using the book [1] one should be using formula (5.148) and (5.155) for the predictions. To find the standard deviation (5.150) can be used and then for the prediction interval formula (3.61) is used where the standard normal distribution of 1.96 for a 0.05 significance level can be used due to the high degrees of freedom of the t-distribution. The predictions transformed back to the original data can be seen in 26.

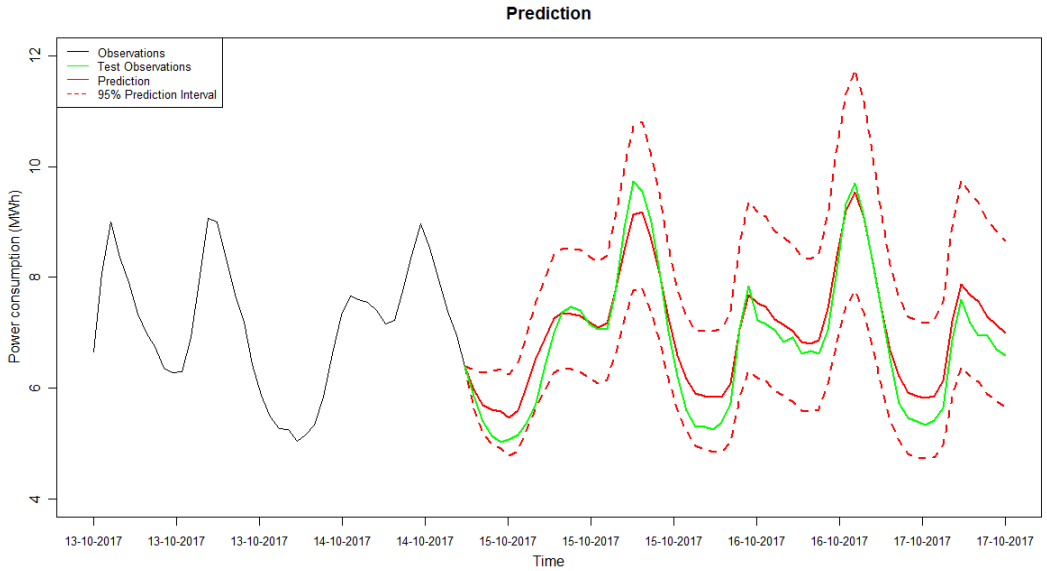

Figure 26: Prediction using model 3, $(11,0,2)(2,1,1)_{24}$, the data is transformed back to its original values.

The model fits very well, one might notice the model doesn't fully capture the valleys of the time series. This might be because the valleys of the previous time steps in the time series has been higher as can be seen in figure 1. However all the

test data is within the 95% prediction interval, which is good. In table 10 all the predictions and their corresponding intervals can be seen. Furthermore the true value and the absolute error.

Table 10: Prediction Table

Prediction	Mean Prediction	Prediction Interval	True Value	Error $\hat{y}_t - y_t$
1 Hour Prediction	5.973315	[5.633246, 6.334743]	5.853	0.1203155
6 Hour Prediction	5.599389	[4.873147, 6.438607]	5.156	0.4433891
12 Hour Prediction	7.348844	[6.342406, 8.522049]	7.467	-0.1181558
24 Hour Prediction	6.596954	[5.608503, 7.767432]	6.215	0.381954
48 Hour Prediction	6.693098	[5.447104, 8.237482]	6.546	0.1470978

Once again from the errors in table 10 it seems the model predicts a bit too high, but the most recent trend was also increasing, this will make the model prediction increase as well. Overall the model seems to predict fairly well within a reasonable limit.

Question 3.6

The model turned out to be decent. The results of the predictions seemed to be good. The prediction interval is not too wide and limits itself within a reasonable range. The assumptions of white noise was fulfilled in most of the cases, except from the weekly seasonality, which can also be seen in the diagnostic plot of the model. A better model taking into account the weekly seasonality would have been an improvement. If a larger dataset is used including yearly seasonality might also be a good idea. Another thing is also to use a less complex model. It would be interesting to use a different method to handle the multiseasonality, such as using an exponential smoothed state space model. This haven't been taught within the limits of this course. Finally it would be interesting to make models based on a current state, where the state could indicate which season is the current season. This would make one able to predict with a greater precision. From knowing the current season, one would expect the power consumption will increase or decrease and should be taken into consideration by the model. This could be further modified so it also takes into account whenever people are having a holiday or days off, such as weekends.

References

[1] Henrik Madsen, *Time Series Analysis*, Chapman & Hall/CRC, 2008, Chapter 6.

Appendix

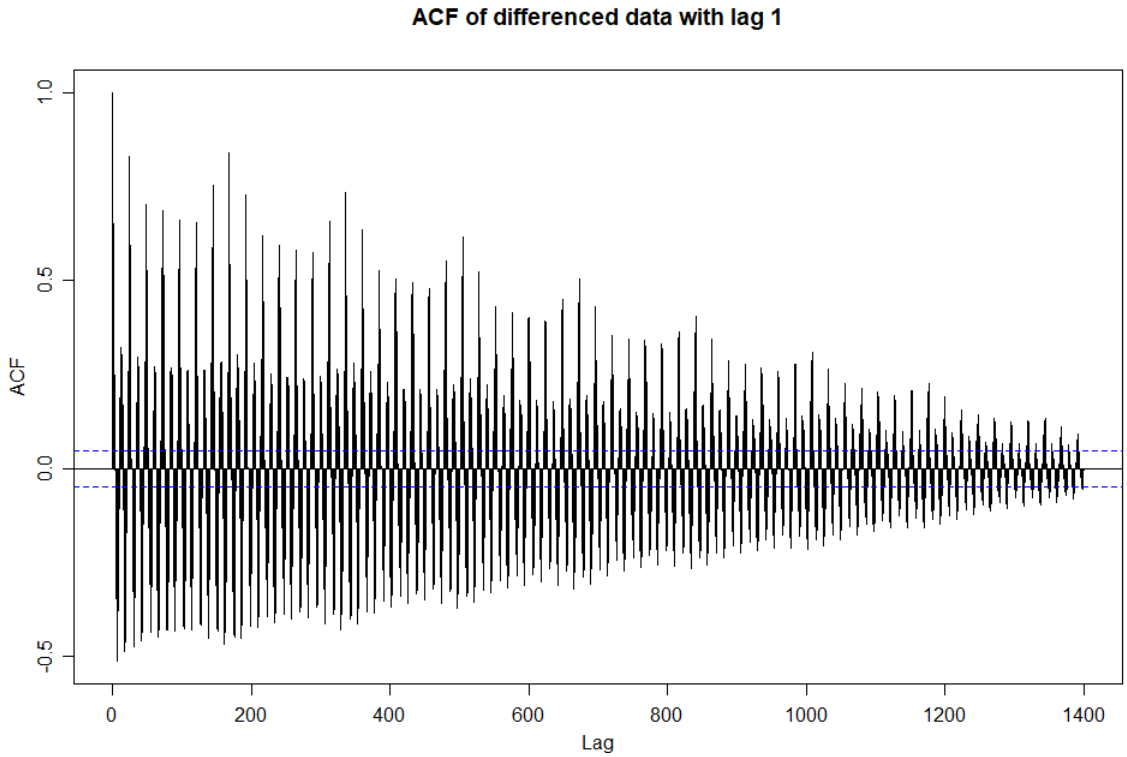

Figure 27: ACF of 1-lagged differenced data

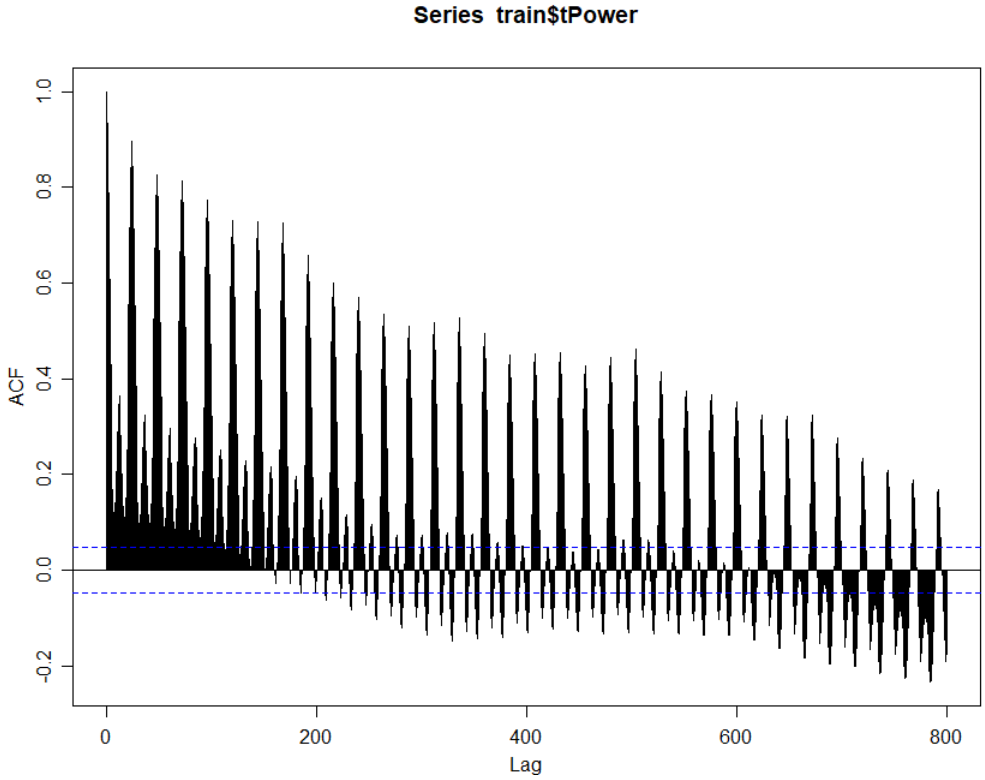

Figure 28: ACF of transformed data

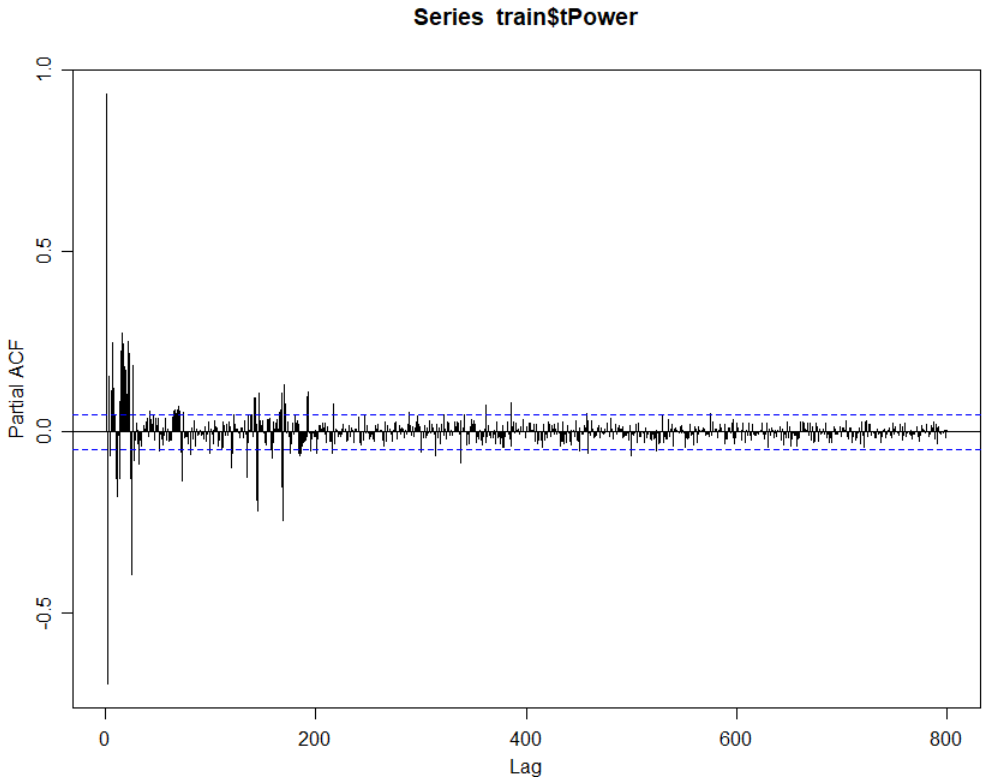

Figure 29: PACF of transformed data

```
############ASSIGNMENT 3############
#Q3.1
library("forecast")
library(MASS)
library(tseries)
library(car)
par(mgp=c(2,0.8,0), mar=c(3,3,2,1),las=0,
    pty="m", xpd=F)
data = read.csv('A3_power_short.txt', header=TRUE,
                sep ="\t",dec = ",")
N = length(data$Power)
data$Date <- as.Date(data$Date, format="%d-%m-%Y")
train = data.frame(Power = data$Power[1:1672],
                   Date=data$Date[1:1672],
                   Hour = data$Hour[1:1672])
test = data.frame(Power = data$Power[1673:N],
                  Date=data$Date[1673:N],
                  Hour = data$Hour[1673:N])

plot(train$Power, type = "l", col="black",
     xlim=c(0,N), xlab = "Time␣(Hourly␣Data)",
     ylab="Power␣consumption␣(MWh)", xaxt='n', cex.axis=.8)
lines(1672:(N), c(train$Power[1672],test$Power),
      col="green")
abline(mean(c(train$Power,test$Power)),0, col="red")
axis(1, c(seq(1,N,length.out=12)),
     format(data$Date[c(seq(1,N,length.out=12))], "%d-%m-%Y"),
     cex.axis = .8)
legend("topleft", legend=c("Observations",
                           "Excluded␣Observations",
                           "Mean␣of␣Time␣Series")
       , col=c(1,"green",2), lty=c(rep(1,3)), cex=0.8)
#make legend

#We see that the series is not stationary and it has
#periods of high variance.
#for the variance, trying to make it more
#homogeneous with a log transform:

lambda <- seq(-2,2,by=0.001)

BC<-boxcox(lm(train$Power~1), lambda=lambda)
BC$x[which(BC$y==max(BC$y))]

bc.trans <- function(lambda,y){
  y.l <- (y^lambda-1)/lambda
  if(lambda == 0){
    y.l<-log(y)
  }
  return(y.l)
```

```
}

train$tPower<-bc.trans(-0.038,train$Power)
test$tPower <- bc.trans(-0.038,test$Power)
BC<-boxcox(lm(train$tPower~1), lambda=lambda)

acf(train$tPower, 800)
pacf(train$tPower, 800)


plot(train$tPower, type = "l", col="black",
     xlim=c(0,N), xlab = "Time (Hourly Data)",
     ylab="Power consumption (MWh)",
     xaxt='n',cex.axis = .8,
     main="Box-Cox transformed Data, lambda=-0.04")
lines(1672:(N), c(train$tPower[1672],test$tPower),
      col="green")
abline(mean(c(train$tPower,test$tPower)),0,col="red")
axis(1, c(seq(1,N,length.out=12)),
     format(data$Date[c(seq(1,N,length.out=12))], "%d-%m-%Y"),
     cex.axis = .8)
legend("topleft", legend=c("Observations",
                           "Excluded Observations",
                           "Mean of Time Series")
       , col=c(1,"green",2), lty=c(rep(1,3)), cex=0.8)

#Want to make the series stationary.
train$dPower <- c(diff(train$tPower), NA)
test$dPower <- c(diff(test$tPower),NA)

plot(train$dPower, type = "l", col="black", xlim=c(0,N),
     xlab = "Time", ylab="Power consumption (MWh)",
     main="Differenced Data")
lines(1672:(N), c(train$dPower[1672],test$dPower),
col="green")
abline(mean(c(train$dPower,test$dPower), na.rm = T),
       0,col="red")
legend("topleft", legend=c("Observations",
                           "Excluded Observations",
                           "Mean of Time Series")
       , col=c(1,"green",2), lty=c(rep(1,3)), cex=0.8)
par(mfrow=c(1,1), mar=c(3,3,3,1))
acf(diff(train$tPower,1), lag=1400,
    main="ACF of differenced data with lag 1")
pacf(diff(train$tPower,1), lag=1400,
     main="ACF of differenced data with lag 1")
#The series looks stationary but ACF decreases too slowly.


#Trying a diff with lag 24
```

```
acf(diff(train$tPower,24), lag=1400,
    main="ACF␣of␣differenced␣data␣with␣lag␣24")
pacf(diff(train$tPower,24), lag=1400,
     main="ACF␣of␣differenced␣data␣with␣lag␣24")
which(acf(diff(train$tPower,24), lag=1400)$acf>0.2)
which(acf(diff(train$tPower,24), lag=1400)$acf<(-0.2))
#Slight improvement however still lag at weekly intervals.
train$dPower <- c(diff(train$tPower,24), rep(NA,24))
test$dPower <- c(diff(test$tPower,24),rep(NA,24))

#plot of new differenced data
plot(train$dPower, type = "l", col="black",
     xlim=c(0,N), xlab = "Time", ylab="Power␣consumption
␣␣␣␣␣(MWh)",
     main="Differenced␣Data")
lines(1672:(N), c(train$dPower[1672],test$dPower),
col="green")
abline(mean(c(train$dPower,test$dPower), na.rm = T),0,
col="red")
legend("topleft", legend=c("Observations",
                           "Excluded␣Observations",
                           "Mean␣of␣Time␣Series")
       , col=c(1,"green",2), lty=c(rep(1,3)), cex=0.8)
par(mfrow=c(1,1), mar=c(3,3,3,1))

#Other interesting things to look at, mean per day:
nDays = length(unique(train$Date))
meanPerDay = rep(0, nDays)
rangePerDay = rep(0, nDays)
#Excluding day 1, since it starts at 08:00:00
meanPerDay[1] = NA
rangePerDay[1]=NA
for(i in 1:(nDays-1)){
  meanPerDay[i+1] = mean(train$Power[(i*24-7):(i*24+16)],
                         na.rm=T)
  rangePerDay[i+1]=range(train$Power[(i*24-7):(i*24+16)],
                         na.rm=T)
}
#Mean range plot
plot(meanPerDay,rangePerDay, type='p',
     ylab="Range␣of␣Power␣Consumption␣per␣day",
     xlab="Mean␣of␣Power␣Consumption␣per␣day")
#Plot of mean per day
plot(meanPerDay, type='l',xlab = "Time␣(Days)",
     ylab="Mean␣daily␣power␣consumption␣(MWh)",
     xaxt='n',main="Daily␣Mean",cex.axis = .8)
axis(1, c(seq(1,nDays,length.out=12)),
     format(train$Date[c(seq(1,N,length.out=12))],
            "%d-%m-%Y"), cex.axis = .8)
points(meanPerDay)
```

```
#Mean per hour
meanPerHour = data.frame(val=rep(0,length(levels(train$Hour))),
                         label = seq(1,24))
meanPerHour$label = factor(
  meanPerHour$label, levels=1:24, labels = levels(train$Hour)
)

for(i in 1:length(meanPerHour$val)){
  meanPerHour$val[i] = mean(
    train$Power[which(train$Hour == levels(train$Hour)[i])])
}
plot(meanPerHour$val~ meanPerHour$label, xlab = "Time in Hours",
     ylab="Power consumption (MWh)",
     main="Plot of the mean per hour")
lines(meanPerHour$val~ as.numeric(meanPerHour$label))

#Daily variation, excluding first observation,
#might be some slight seasonality within each day.
plot(train$Power[(17):(168+16)],
     type='l', ylim = c(3,11), xaxt='n',
     ylab="Power Consumption (MWh)", xlab="Weekday")
nWeeks=nDays/7
meanPerWeek = rep(0, floor(nWeeks))
k=0
for(i in 1:10){
  lines(train$Power[(17+k):(i*168+16)])
  k=k+168
}
axis(1, c(seq(12, 168, by=24)),
     c("Monday", "Tuesday", "Wednesday",
       "Thursday", "Friday", "Saturday","Sunday"),
     cex.axis = .8)

#Excluding day 1, since it starts at 08:00:00
meanPerWeek = c(rep(0, 7))
weekday <- c(rep(0, 10))
for(j in 0:6){
  k=0
  weekday <- c(rep(0, 10))
  for(i in 1:nWeeks){
    weekday[i] <- sum(
      train$Power[(17+j*24+k):(17+24*j+k+23)])
    k=k+168
  }
  print(mean(weekday, na.rm=T))
  j=j+1
  meanPerWeek[j] = mean(weekday,na.rm=T)
}
plot(meanPerWeek, type='b', xaxt='n',
```

```
     ylab="Mean␣of␣power␣consumption␣(MWh)",
     xlab="Weekday",
     main="Average␣power␣consumption␣of␣each␣weekday")
axis(1, c(seq(1, 7, by=1)),
     c("Monday", "Tuesday", "Wednesday", "Thursday",
       "Friday", "Saturday","Sunday"), cex.axis = .8)


###############Q3.2#############
dPower <- train$dPower[1:(1672-24)]
par(mfrow=c(1,1), mar=c(3,3,3,1))


acf(train$Power, lag.max=400,
    main="ACF␣of␣data␣without
␣␣␣␣transformation␣and␣differencing")
pacf(train$Power, lag.max=400,
     main="PACF␣of␣data␣without
␣␣␣␣␣transformation␣and␣differencing")
acf(train$Power, lag.max=1400,
    main="ACF␣of␣data␣without
␣␣␣␣transformation␣and␣differencing")
pacf(train$Power, lag.max=1400,
     main="PACF␣of␣data␣without
␣␣␣␣␣transformation␣and␣differencing")

#Close look
acf(dPower, lag.max = 400,
    main="ACF␣of␣the␣differenced
␣␣␣␣and␣transformed␣power␣consumption")
pacf(dPower, lag.max = 400,
     main="PACF␣of␣the␣differenced
␣␣␣␣␣and␣transformed␣power␣consumption")
#Close look
acf(dPower, lag.max = 1400,
    main="ACF␣of␣the␣differenced
␣␣␣␣and␣transformed␣power␣consumption")
pacf(dPower, lag.max = 1400,
     main="PACF␣of␣the␣differenced
␣␣␣␣␣and␣transformed␣power␣consumption")


#Q3.3
my.tsdiag <- function(x, lag.max, nlag){
  if(class(x)=="Arima"){
    dat<-x
  }
  oldpar <- par(mfrow=c(3,1), mgp=c(2,0.7,0),
                mar=c(3,3,1.5,1))
  on.exit(par(oldpar))
```

```
  acf(dat$residuals, lag=lag.max)
  pacf(dat$residuals, lag=lag.max)
  plot(dat$residuals)
}

my.tssum <- function(x){
  if(class(x)=="Arima"){
    dat<-x
  }
  a<-2*length(dat$mask[dat$mask==TRUE])-2*dat$loglik
  b<-log(dat$nobs-dat$n.cond)*
    length(dat$mask[dat$mask==TRUE])-2*dat$loglik
  print(a)
  print(b)
  print(BIC(dat))
  print(AIC(dat))
  summary(dat)
}

sign.test<-function(x){
  dat <- x
  res <- dat$residuals
  (N.sign.changes <- sum( res[-(dat$n.cond+1)] *
                            res[-length(res)]<0 ))
  print(binom.test(N.sign.changes,
                   (length(dat$residuals)-dat$n.cond)-1))
}

ljung.test<-function(x, nlag, modnr){
  dat <- x
  pval <- sapply(1:nlag, function(i) Box.test(
    dat$residuals, i, type = "Ljung-Box")$p.value)
  plot(1L:nlag, pval, xlab = "lag",
       ylab = "p value", ylim = c(0,1),
       main = paste("P-values for Ljung-Box statistic
for Model",modnr))
  abline(h = 0.05, lty = 2, col = "blue")
}

#if models are nested H0: theta_extra = 0 vs H1: not zero
SSE.test<-function(x1,x2){
  fit1<-x1
  fit2<-x2
  ## F-test
  s1 <- sum(fit1$residuals^2)
  s2 <- sum(fit2$residuals^2)
  n1 <- fit1$mask[fit1$mask==TRUE]
  n2 <- fit2$mask[fit2$mask==TRUE]
  N<-length(fit1$residuals)-fit1$ncond
  ( f.stat <- (s1-s2)/(n2-n1) / (s2/(N-n2)) )
```

```
  print(pf(f.stat , df1 = n2 - n1, df2 = (N-n2), lower.tail = FALSE))
}

#

acf(diff(train$tPower, 24), lag.max = 400,
    main="ACF␣of␣the␣differenced␣and␣transformed
␣␣␣␣power␣consumption")
pacf(diff(train$tPower, 24), lag.max = 400,
     main="PACF␣of␣the␣differenced␣and␣transformed
␣␣␣␣␣power␣consumption")
which(acf(diff(train$tPower,24), lag=1400)$acf>0.2)
which(acf(diff(train$tPower,24), lag=1400)$acf<(-0.2))

#SLOW FORWARD STEPWISE
#trying to go slow with an AR(2) first
mod1<-arima(train$tPower, order=c(2,0,0),
            seasonal = list(order=c(0,1,0),
                            period = 24), include.mean=T,
                            method='CSS-ML'
)
my.tssum(mod1)
my.tsdiag(mod1, 400)

#seems to be both MA and AR noise, ARMA(1,0,1)
mod1<-arima(train$tPower, order=c(2,0,1),
            seasonal = list(order=c(0,1,0),
                            period = 24),
            include.mean=T, method='CSS-ML'
)
my.tsdiag(mod1, 400)
my.tssum(mod1)
ljung.test(mod1, 190, 1)
sign.test(mod1)

#Let's try to swap the AR and MA part, not a big difference
mod1<-arima(train$tPower, order=c(1,0,2),
            seasonal = list(order=c(0,1,0),
                            period = 24),
            include.mean=T, method='CSS-ML'
)
my.tsdiag(mod1, 400)
my.tssum(mod1)
ljung.test(mod1, 190, 1)
sign.test(mod1)

#ARMA(2,2) didn't change anything. We go with the 1,0,2

# Let's look at the 24-hour season first,
#SMA(1) and SAR(1) seems to reduce the ac
```

```
mod1<-arima(train$tPower, order=c(1,0,2),
            seasonal = list(order=c(1,1,1),
                            period = 24),
            include.mean=T, method='CSS',
            fixed=c(p=c(NA), q=c(NA,NA),
                    P=c(NA), Q=c(NA)))
my.tsdiag(mod1, 400)
my.tssum(mod1)
ljung.test(mod1, 190, 1)
sign.test(mod1)

# seems to be 2 significant lags around season, try SAR(2)
# additional SMA(2) or SAR(3) didn't improve model significantly
mod1<-arima(train$tPower, order=c(1,0,2),
            seasonal = list(order=c(2,1,1),
                            period = 24), method='CSS-ML')
my.tsdiag(mod1, 400)
my.tssum(mod1) #everything significant
ljung.test(mod1, 190, 1)
sign.test(mod1)


#Some clear seasonality in the data every week (168 lag)
mod1<-arima(train$tPower, order=c(1,0,2),
            seasonal = list(order=c(7,1,1),
                            period = 24),
            include.mean=T, method='CSS',
            fixed=c(p=c(NA), q=c(NA,NA),
                    P=c(NA,NA,0,0,0,0,NA), Q=c(NA)))
my.tsdiag(mod1, 400)
my.tssum(mod1)
ljung.test(mod1, 190, 1)
sign.test(mod1)
#Still significant lag around 168, but the
#seasons has been removed
# SMA(7) did not improve it.

#Still some noise at the start of the PACF plot and also
#a bit on the ACF.
#Let's try with a backward selection, high order for
#the same model
#lets fit to half a day (12 hours)

mod1<-arima(train$tPower, order=c(12,0,12),
            seasonal = list(order=c(2,1,1),
                            period = 24), method='CSS',
            fixed=c(p=c(NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA),
                    q=c(NA,NA), P=c(NA,NA),
                    Q=c(NA)))
my.tsdiag(mod1, 400)
```

```
my.tssum(mod1)
ljung.test(mod1, 190, 1)
sign.test(mod1)
#Removes a lot of the lag

#Seems to be slightly better
mod1<-arima(train$tPower, order=c(12,0,2),
            seasonal = list(order=c(2,1,1),
                            period = 24), method='CSS',
            fixed=c(p=c(NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA),
                    q=c(NA,NA), P=c(NA,NA),
                    Q=c(NA)))
my.tsdiag(mod1, 400)
my.tssum(mod1)
ljung.test(mod1, 190, 1)
sign.test(mod1)

#Let's try remove lag 168, removes the ACF,
#but is very complex and didn't make test significantly
#better.
mod1<-arima(train$tPower, order=c(12,0,2),
            seasonal = list(order=c(7,1,2),
                            period = 24), method='CSS',
            fixed=c(p=c(NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA),
                    q=c(NA,NA), P=c(NA,NA,NA,NA,NA,NA,NA),
                    Q=c(NA,NA)))
my.tsdiag(mod1, 400)
my.tssum(mod1)
ljung.test(mod1, 190, 1)
sign.test(mod1)


###################MODELS CHOSEN#########################
#Seems to be slightly better
mod1<-arima(train$tPower, order=c(1,0,2),
              seasonal = list(order=c(2,1,1),
                              period = 24), method='CSS-ML')
my.tsdiag(mod1, 400)
my.tssum(mod1)
cpgram(mod1$residuals,main="model␣3")
qqnorm(mod1$residuals)
qqline(mod1$residuals)
sum(mod1$residuals^2)
#########################################################
mod2<-arima(train$tPower, order=c(1,0,2),
            seasonal = list(order=c(7,1,1),
                            period = 24),
            include.mean=T, method='CSS-ML',
            fixed=c(p=c(NA), q=c(NA,NA),
                    P=c(NA,NA,0,0,0,0,NA), Q=c(NA)))
```

```
my.tsdiag(mod2, 400)
my.tssum(mod2)
cpgram(mod2$residuals,main="model␣3")
qqnorm(mod2$residuals)
qqline(mod2$residuals)
sum(mod2$residuals^2)
#########################################################
mod3<-arima(train$tPower, order=c(11,0,2),
        seasonal = list(order=c(2,1,1),
                        period = 24), method='CSS-ML',
        fixed=c(p=c(0,NA,0,0,NA,NA,0,0,0,NA,NA),
                q=c(NA,NA), P=c(NA,NA),
                Q=c(NA)))
my.tsdiag(mod3, 400)
my.tssum(mod3) #everything significant
sum(mod3$residuals^2)
#########################################################

sign.test(mod1)
sign.test(mod2)
sign.test(mod3)

par(mfrow=c(3,1))
ljung.test(mod1, 48, 1)
ljung.test(mod2, 48, 2)
ljung.test(mod3, 48, 3)
par(mfrow=c(1,1))

par(mfrow=c(1,2))
cpgram(mod1$residuals,main="model␣1")
qqPlot(mod1$residuals)
cpgram(mod2$residuals,main="model␣2")
qqPlot(mod2$residuals)
cpgram(mod3$residuals,main="model␣3")
qqPlot(mod3$residuals)


# Alternatively
pchisq(-2 * ( mod1$loglik - mod3$loglik ),
df=4, lower.tail = FALSE)

orig.trans <- function(lambda,ytrans){
  y <- exp(log(lambda*ytrans + 1)/lambda)
  if(lambda == 0){
    y<-exp(ytrans)
  }
  return(y)
}

pred<-forecast.Arima(mod1, h=61, level=0.95)
```

```
pred$mean<-orig.trans(-0.038, pred$mean)
pred$lower<-orig.trans(-0.038, pred$lower)
pred$upper<-orig.trans(-0.038, pred$upper)
err<-sqrt(sum((test$tPower-pred$mean)^2))/(mod1$nobs)
err

pred<-forecast.Arima(mod2, h=61, level=0.95)
pred$mean<-orig.trans(-0.038, pred$mean)
pred$lower<-orig.trans(-0.038, pred$lower)
pred$upper<-orig.trans(-0.038, pred$upper)
err<-sqrt(sum((test$tPower-pred$mean)^2))/(mod2$nobs)
err

pred<-forecast.Arima(mod3, h=61, level=0.95)
pred$mean<-orig.trans(-0.038, pred$mean)
pred$lower<-orig.trans(-0.038, pred$lower)
pred$upper<-orig.trans(-0.038, pred$upper)
err<-sqrt(sum((test$tPower-pred$mean)^2))/(mod3$nobs)
err
par(mfrow=c(1,1))
plot(1630:1672,train$Power[1630:1672], type = "l", col="black",
     xlim=c(1630,N), xlab = "Time",
     ylab="Power␣consumption␣(MWh)",
     main="Prediction", ylim=c(4,12),
     xaxt='n')
lines(1672:(N), c(train$Power[1672],pred$mean), lwd=2,
      col="red")
lines(1672:(N), c(train$Power[1672],pred$lower), lwd=2,
      col="red", lty=2)
lines(1672:(N), c(train$Power[1672],pred$upper), lwd=2,
      col="red", lty=2)
lines(1672:(N), c(train$Power[1672],test$Power),  lwd=2,
      col="green")
legend("topleft", legend=c("Observations",
                           "Test␣Observations", "Prediction",
                           "95%␣Prediction␣Interval")
       , col=c(1,"green",2,2), lty=c(rep(1,3), 2), cex=0.8)
axis(1, c(seq(1630,N,length.out=12)),
     format(data$Date[c(seq(1630,N,length.out=12))],
            "%d-%m-%Y"), cex.axis = .8)

pred$mean[1]
pred$mean[6]
pred$mean[12]
pred$mean[24]
pred$mean[48]

pred$lower[1]
pred$lower[6]
pred$lower[12]
```

```
pred$lower[24]
pred$lower[48]

pred$upper[1]
pred$upper[6]
pred$upper[12]
pred$upper[24]
pred$upper[48]

test$Power[1]
test$Power[6]
test$Power[12]
test$Power[24]
test$Power[48]

pred$mean[1]-test$Power[1]
pred$mean[6]-test$Power[6]
pred$mean[12]-test$Power[12]
pred$mean[24]-test$Power[24]
pred$mean[48]-test$Power[48]
```