DTU - Time Series Analysis - Assignment 3: Estimating ARMA Processes and Seasonal Processes

Emma Demarecaux (s176437)

12/04/2018

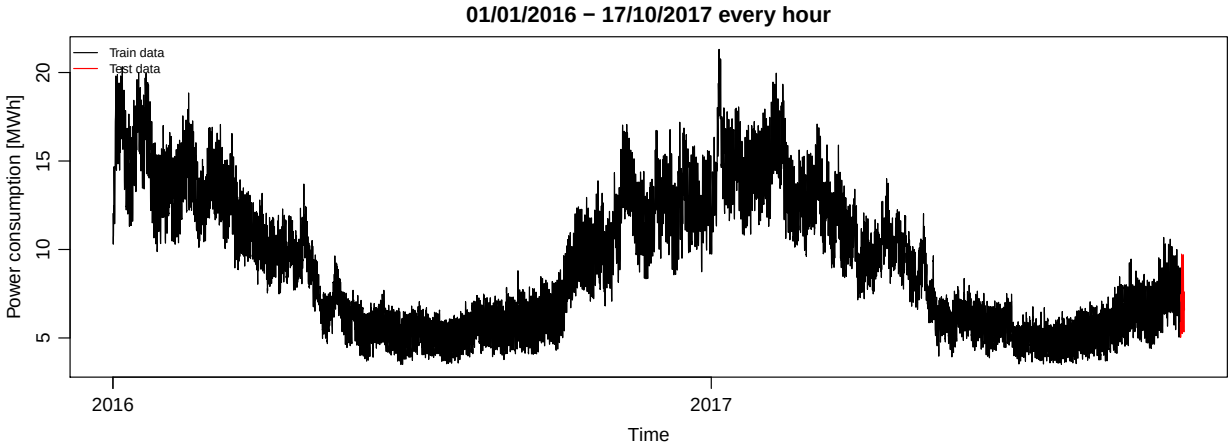
The city of Malmö wants to be carbon neutral by 2030. One step on the way is to get a better understanding of the consumption and production of energy in the city. To balance the electricity grid it is important to make good forecasts covering the planning horizon. In this assignment the focus will be on predicting the electricity consumption from a district in Malmö. The three columns of the dataset are:

- Date: Date for observation (from 06/08/2017 to 17/10/2017);
- Hour: Hour within day for the observation;
- Power: The consumption for that hour in MWh.

The study has been elaborated in colaboration with Adélie Marie Kookai Barre (s170075).

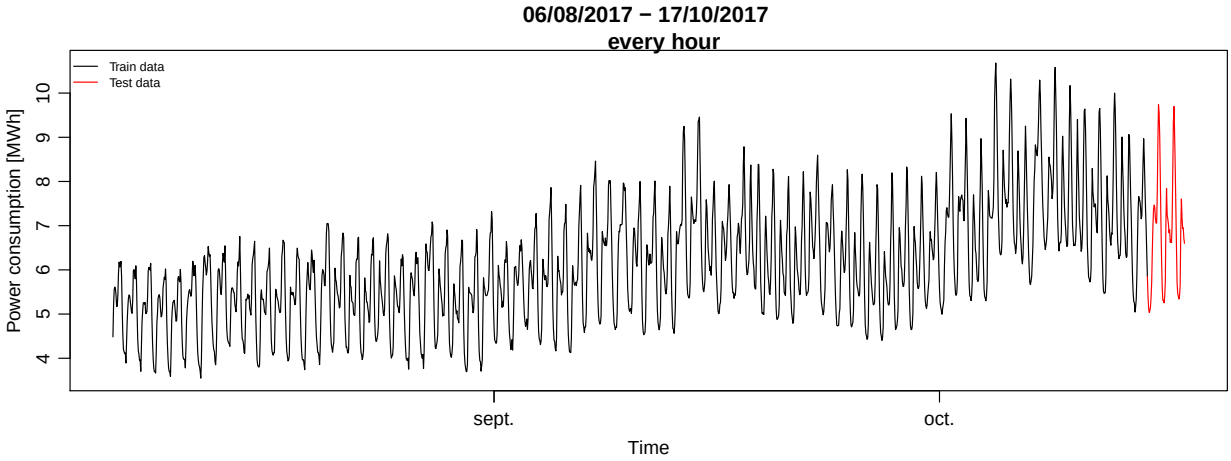
Question 3.1: Presenting the data

In the previous version of assignment 3, the dataset was much larger. Let us first display the large dataset which corresponds to the power consumption from 01/01/2016 to 17/10/2017 every hour. The red line represents the part left as test data that we are going to predict from the 15th of August and onwards.



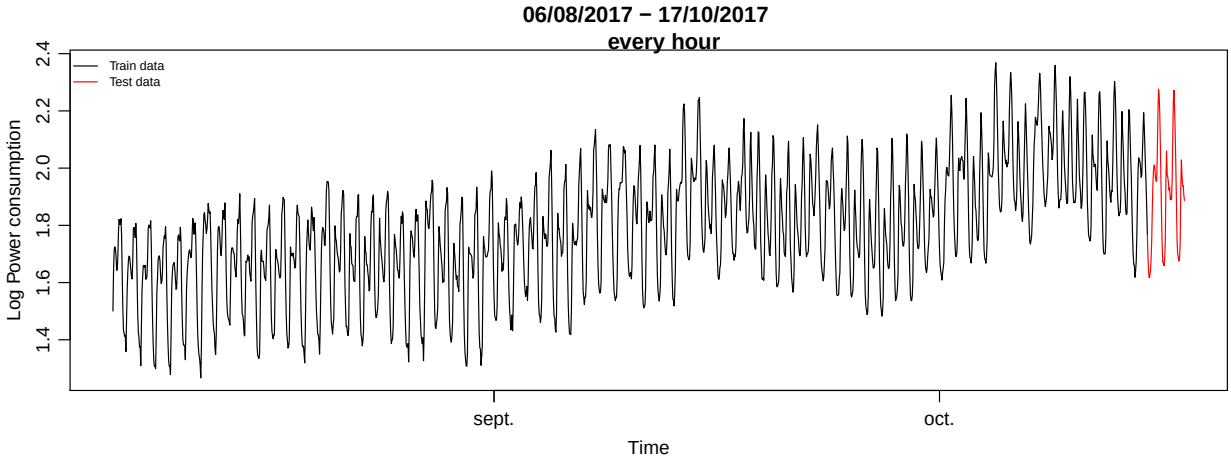
This time series doesn't look stationnary as there is at least a year pattern. We can understand why a smaller part of this full dataset might be simpler than the full one and sufficient to make our predictions for the 61 red observations.

Let us look at the short dataset which is the one we are going to focus on in this report. In the following figure is represented the power consumption from 06/08/2017 to 17/10/2017 every hour.



From this figure, it is obvious that the time series is not stationnary as there is a small increasing trend for this period of time.

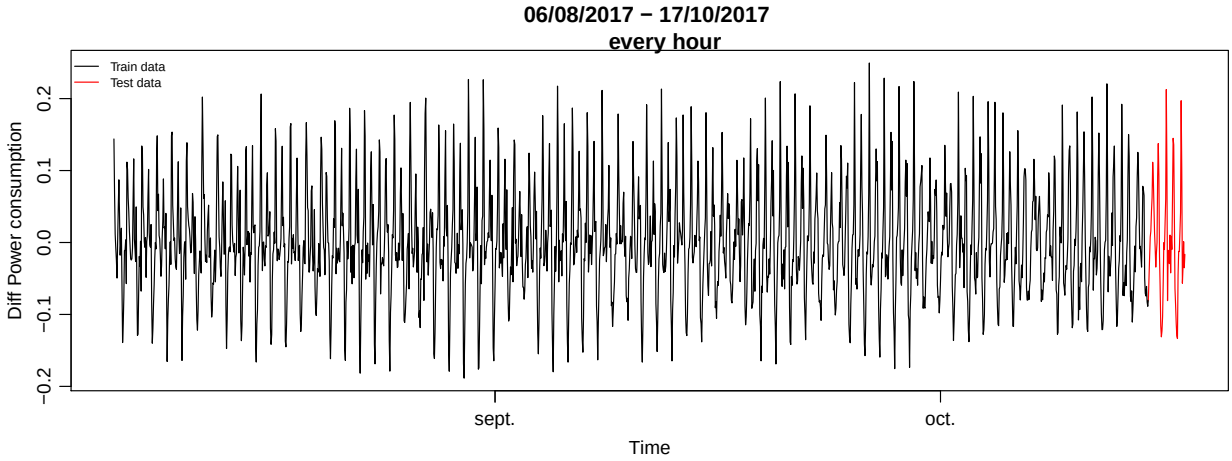
Furthermore, the variance seems to increase a bit with electricity consumption. We will then consider in the following figure the log-transformed dataset:



Even though the two last graphs are very similar, we observe that the variance varies less in the log-transformed data which leads to a more exploitable set of data. Therefore, we will keep working with the log-transformed data starting from now.

There is a clear seasonal tendency in the dataset which corresponds to a daily pattern, suggesting to model the process with a 24-hour-period seasonal model.

Lastly, let us differenciate the log-power consumption in order to remove slight increasing slope:



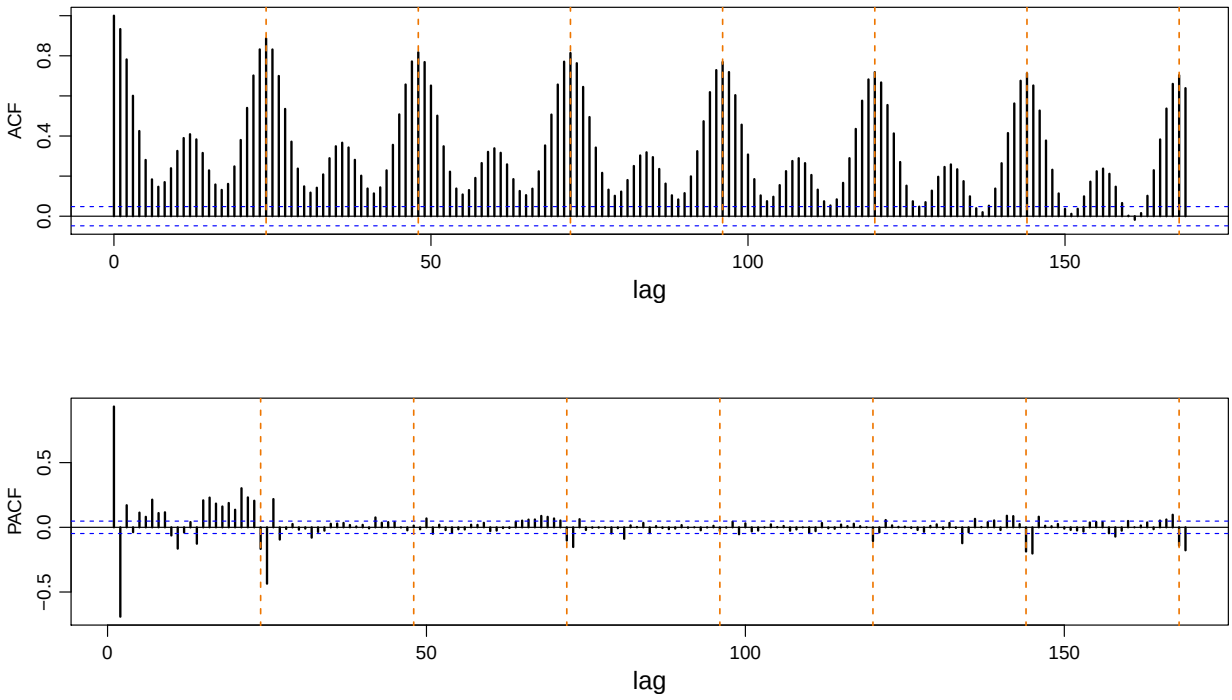
The means finally stays around 0.

In the following questions we will estimate the daily pattern of this time series but it seems that there is also a small weekly behavior that will maybe have an impact at lags $168 * i$ in the ACF and PACF.

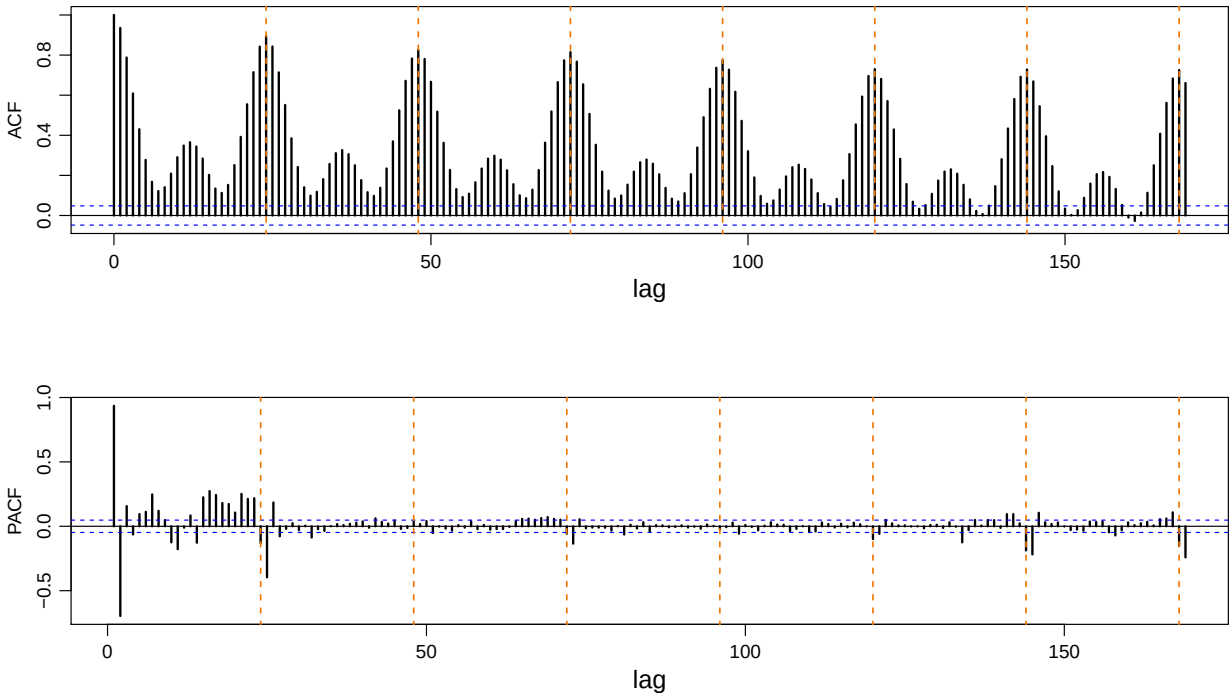
Question 3.2: ACF and PACF

Power consumption not transformed

The following figure represents the autocorrelation function and the partial autocorrelation function of the power consumption. We added orange lines at lags $24 * i$ to see the seasonal behavior:

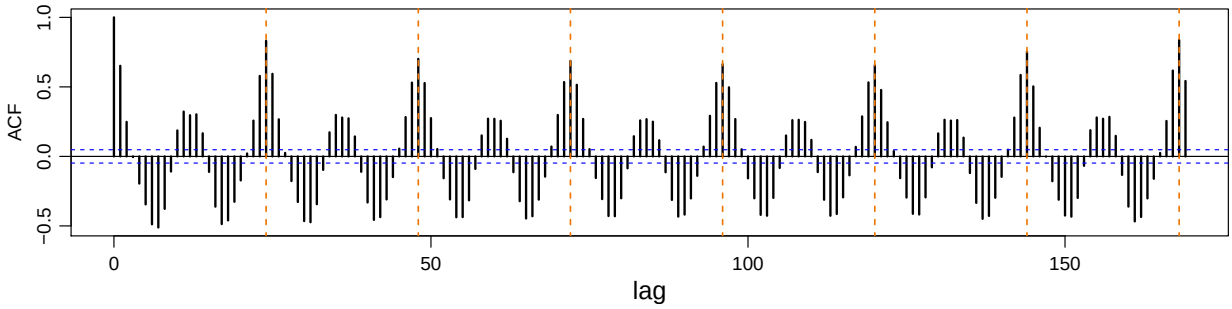


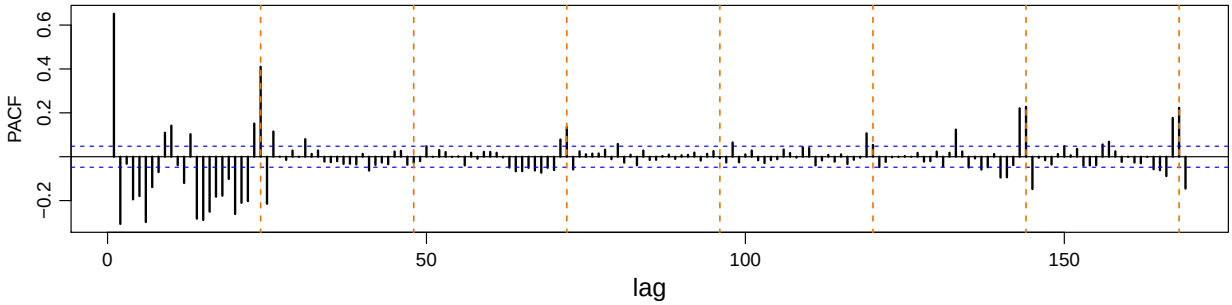
Log-power consumption



Both pairs of graphs are very similar. A seasonal pattern with a period of 24 hours can clearly be identified as well as a 12-hour pattern from the ACF. From the PACF, one can assess the model is probably an AR(2) with the two first main peaks. The ACF doesn't converge to zero within a miningful time point, and therefore, it confirms that the series is not stationnary and will need to be differenciated to model it. Let us then look at the ACF and PACF of the differenciated series so that we can better exploit the ACF and find a good model for this dataset.

Differenciated log-power consumption





From the ACF, no clear decay is detected: this means that we will probably have to differenciate the seasonal behavior of the series. There are often 3 peaks at lags $24 * i$, letting us suppose a MA(3) for the seasonal part of the model.

The PACF shows an exponential decay which leads to think of a global MA model. Peaks at lags $24 * i$ also confirm the daily seasonality that we have to take into account. The high peaks at lag 1 and 2 in the PACF may let us think of an AR(2) model. Finally, a peak at lag 168 is observed, showing a weekly pattern. Here, it is not possible to implement a double seasonality with the *arima* function from R and make a prediction, we will therefore ignore the weekly pattern and still get good short-term predictions.

Question 3.3: Model selection procedure

The aim here is to build up a model to estimate as faithfully as possible the power consumption Y_t as a $(p; d; q) * (P; D; Q)_s$ seasonal process:

$$\phi(B)\Phi(B^s)\nabla^d\nabla^D Y_t = \theta(B)\Theta(B^s)\epsilon_t$$

with ϵ_t a white noise.

In order to find the right model, the observations made previously on the ACF and PACF graphs will be used to come up with an initial model. For each model, the ACF and PACF of the residuals will be plotted in order to see how close the plots are from white noise properties. It will help update the order of the AR and MA from the global or seasonal parts of the model by finding an ARIMA model for the residuals we obtain.

In order to find the best model we are going to look at the ratio between the estimate parameters and the standard deviation. In fact, comparing an ARMA$(p-1,q)$ or an ARMA$(p,q-1)$ with an ARMA(p,q) is the same as looking at the significance of the estimated parameter $\hat{\phi}_p$ (resp. $\hat{\theta}_q$). Let us denote by $\sigma_{\hat{\phi}_p}$ (resp. $\sigma_{\hat{\theta}_q}$) the standard deviation of the parameter $\hat{\phi}_p$ (resp. $\hat{\theta}_q$). Supposing that $n-p$ (resp. $n-q$) is large enough, we accept the ARMA$(p-1,q)$ with a 5% error if $\frac{|\hat{\phi}_p|}{\sigma_{\hat{\phi}_p}} < 1.96$.

This allows us to test the significance of the estimated parameters: they should be at least two times greater than their estimated standard deviation.

Another way of selecting a model is through the information criteria. We should select the model which minimizes some information criterion when model are not nested:

- AIC = -2log(L(Y_N;$\hat{\theta}$,$\hat{\sigma}_\epsilon^2$)) + $2n_{par}$;
- BIC = -2log(L(Y_N;$\hat{\theta}$,$\hat{\sigma}_\epsilon^2$)) + $log(N)n_{par}$.

The AIC is easily availabe when estimating a seasonal process with the function *arima* in R so we are going to focus on this criteria for our study.

Once a model is estimated to be good enough, we can check if it is efficient by doing a residual analysis:

- First and most important we plot the residuals and compare them with random numbers;

- we can also plot the histogram with the theoritical curve and check that the mean value is 0 for example;
- we can look at the QQplot to see if it follows along the expected line;
- we can also look for autocorrelation through the Ljung-Box statistics;
- finally, a statistical testing such as the sign test can be made.

Question 3.4: Model Selection

Choosing the model

Model 1

Accordingly to the remarks made in Question 3.2, it is first needed to identify the order of seasonal differencing D and the order of differencing d. We clearly see that the dataset has to be differenciate as the autocorrelation function of the log-consumption does not decrease sufficiently fast towards 0, so we will try $d = 1$. Furthermore, the ACF of the differenciated log-consumption shows high peaks at lags $24 * i$ that doesn't exponentially decays so we should introduce a seasonal differencing $D = 1$ for our model.

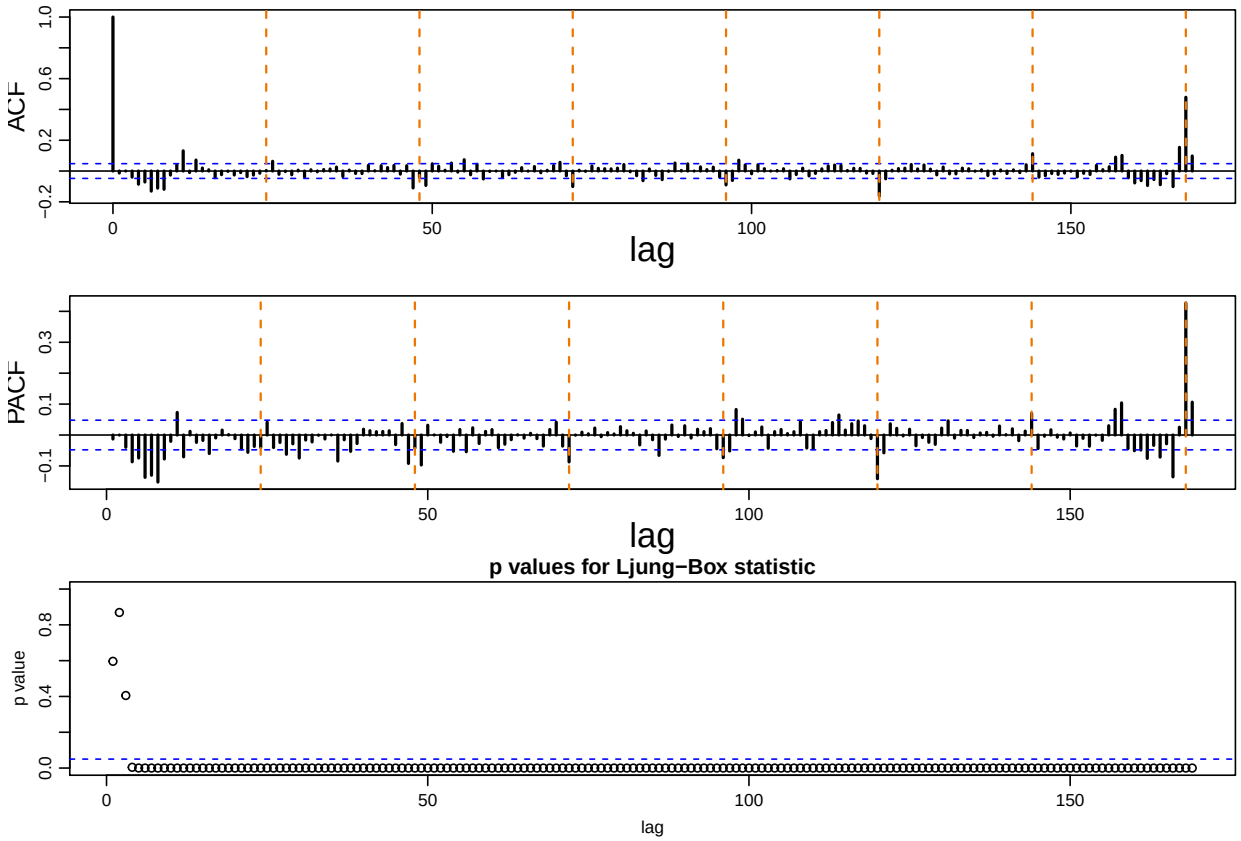
Otherwise, regarding the non-seasonal coefficients and as previously explained in Question 3.2, the PACF let us think of an AR(2) part. For the seasonal coefficients, the ACF leads us to a MA(3) (as explained in the same Question 3.2). We will therefore consider the following initial model: $\text{ARIMA}(2;1;0) * (0;1;3)_{24}$.

The ACF and PACF of the residuals are plotted in the following figures:

```
##
## Call:
## arima(x = data$logPower[1:N], order = c(2, 1, 0), seasonal = list(order = c(0,
##     1, 3), period = 24))
##
## Coefficients:
##          ar1      ar2     sma1     sma2    sma3
##       0.4253  -0.2517  -0.4626  -0.4504  0.0627
## s.e.  0.0243   0.0241   0.0273   0.0251  0.0296
##
## sigma^2 estimated as 0.001002:  log likelihood = 3336.9,  aic = -6661.81
```

The latter model gives already a quite good results looking at the ACF and PACF with peaks within the blue intervals. We can clearly observe the lag 168 that is the weekly seasonality we decided to ignore.

On the ACF, high peaks at lags $24 * i$ are less significant than before which means that the MA(3) component for the seasonal part seems quite reasonnable. However there are still remaining peaks at lags $24 * i$ on both ACF and PACF so we should add one parameter in both the MA and AR components for the seasonal part.

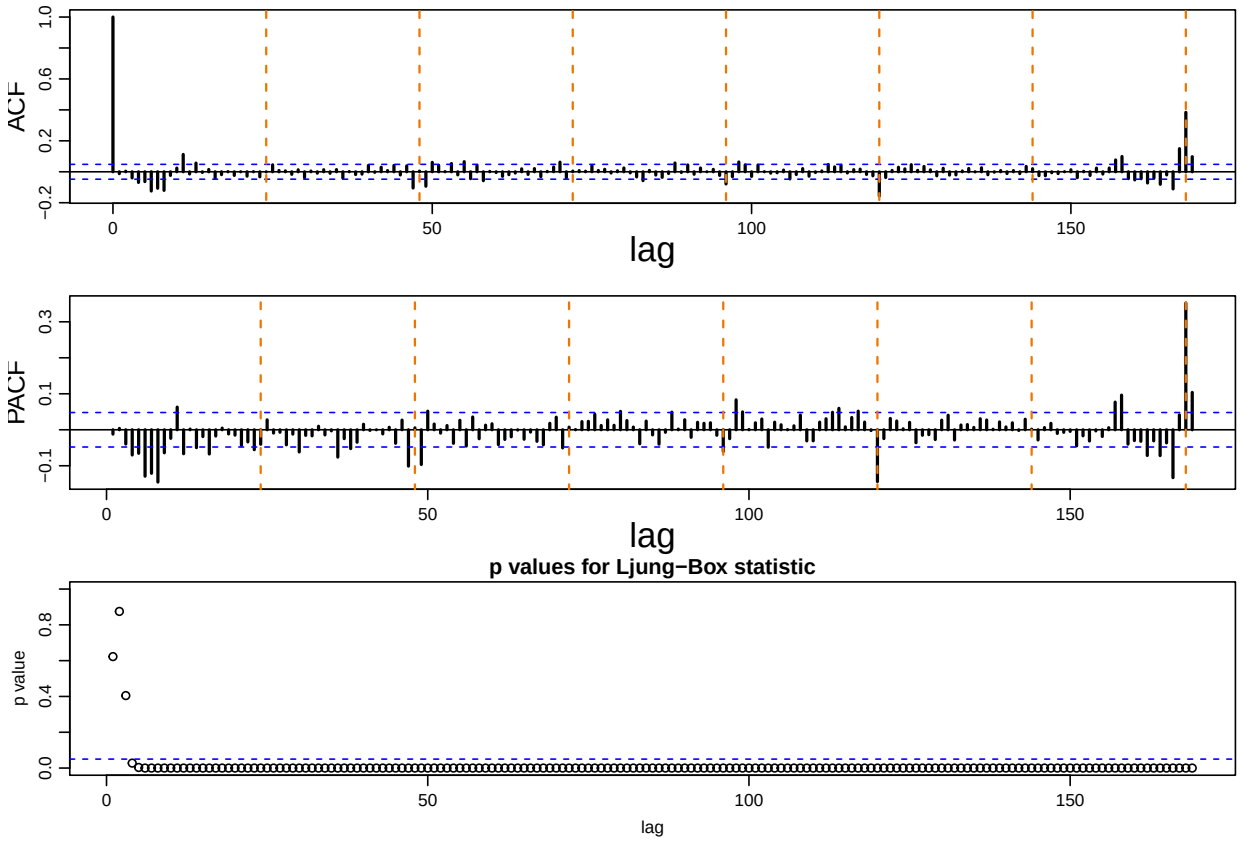


Model 2

The second model to be looked at is an $\text{ARIMA}(2;1;0) * (1;1;4)_{24}$:

```
##
## Call:
## arima(x = data$logPower[1:N], order = c(2, 1, 0), seasonal = list(order = c(1,
##     1, 4), period = 24))
##
## Coefficients:
##          ar1      ar2    sar1     sma1     sma2    sma3    sma4
##       0.4069  -0.2171  0.4804  -0.9964  -0.2823  0.2579  0.1655
## s.e.  0.0246   0.0245  0.0669   0.0687   0.0543  0.0525  0.0381
##
## sigma^2 estimated as 0.0009377:  log likelihood = 3385.19,  aic = -6754.37
```

We can observe some improvements but the model is not perfect. The ACF is actually almost not changing while the PACF has improved. We still have important structures in the first lags along with exponential decays for both ACF and PACF. All our parameters are significant for now. We can try to add one parameter in both MA and AR components.

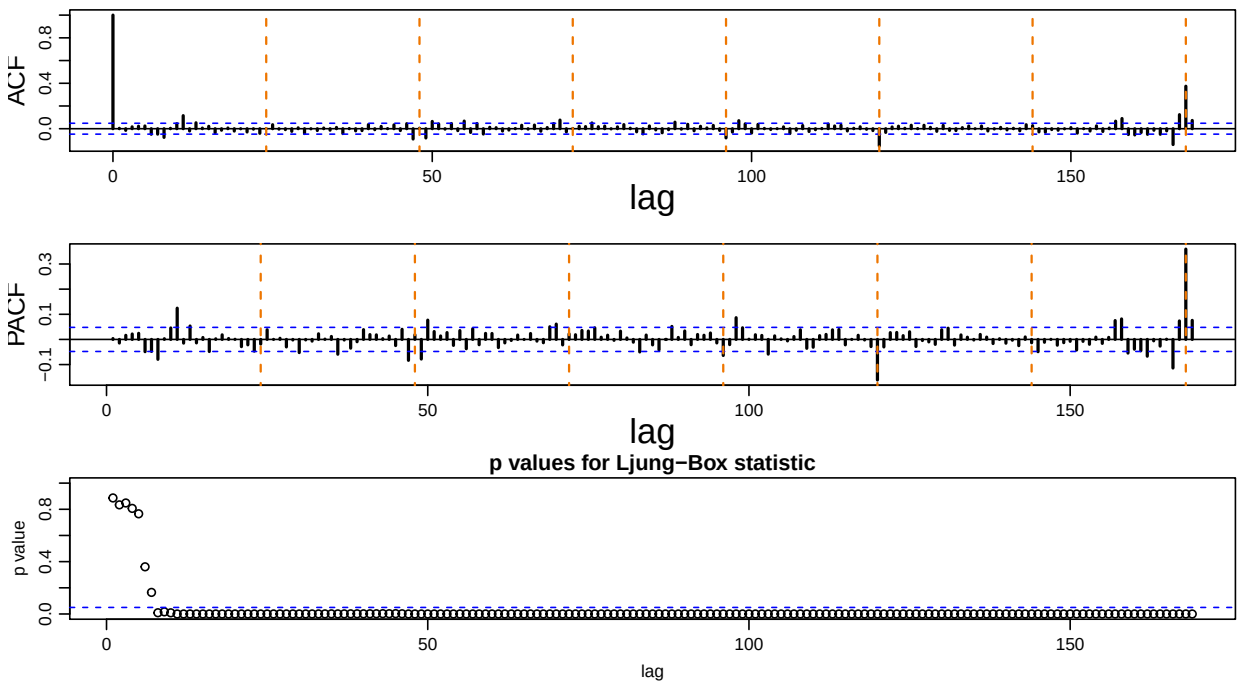


Model 3

The third model is an ARIMA$(3;1;1) * (1;1;4)_{24}$ and the results are plotted thereafter:

```
##
## Call:
## arima(x = data$logPower[1:N], order = c(3, 1, 1), seasonal = list(order = c(1,
##     1, 4), period = 24))
##
## Coefficients:
##          ar1      ar2     ar3      ma1    sar1     sma1     sma2    sma3
##       1.2895  -0.5505  0.1071  -0.9640  0.4718  -0.9950  -0.2715  0.2620
## s.e.  0.0266   0.0391  0.0258   0.0086  0.0707   0.0726   0.0565  0.0523
##         sma4
##       0.1469
## s.e.  0.0376
##
## sigma^2 estimated as 0.0008796:  log likelihood = 3438.42,  aic = -6856.85
```

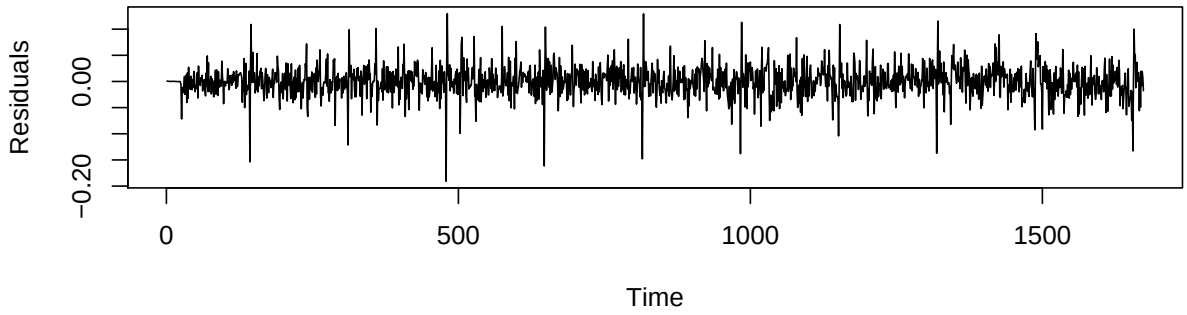
Again, all the parameters are significant so this model seems to fit well the data without overfitting. There are still some seasonal peaks on the PACF that could be removed by adding one last order on the seasonal MA part. However, this last try gives a very slightly higher AIC than the third model (-6853.63), showing no improvement.

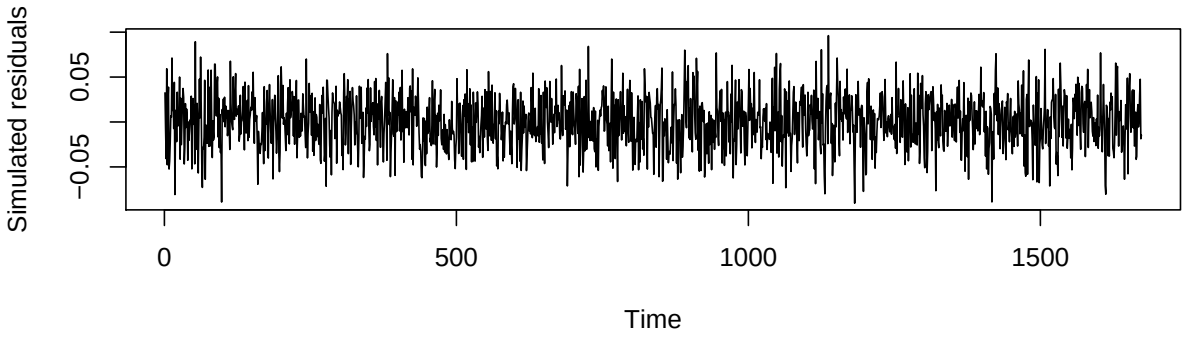


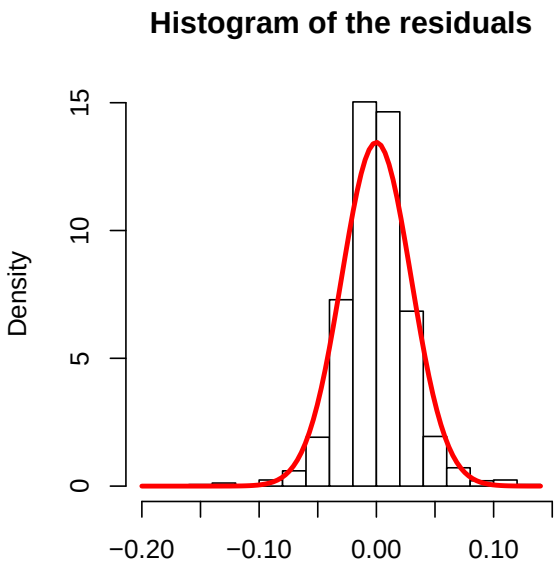
A first study, that has not be explained here led to another model that was an ARIMA$(2;1;3)*(1;1;2)_{24}$, with AIC= -6779.66. As the latter model and the third model are not nested, the AIC criterion is a good way to choose the final model. Therefore, with the lowest AIC value, the third model will be used as the final model although the ACF and PACF plots are not really perfect as all the peaks are not in the confidence interval and as the Ljung-Box statistic show correlation after lag 8. Indeed, the third model was the best that was managed to be found using the function ARIMA in R, that can only take into account one period of seasonality.

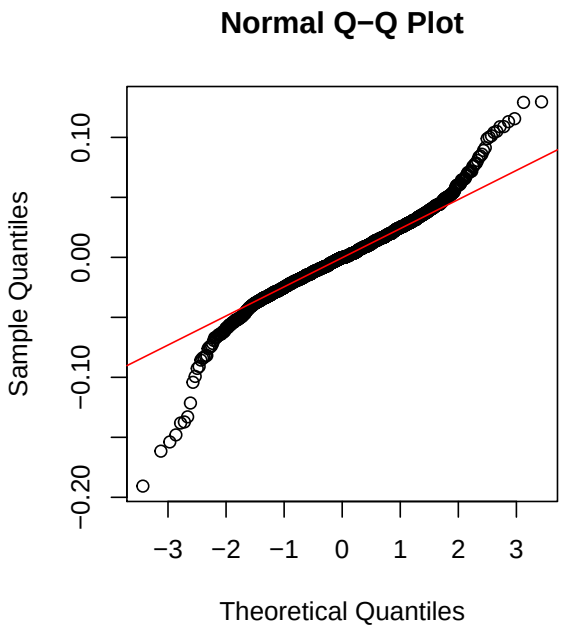
Further analysis of residuals for the model ARIMA$(3;1;1)*(1;1;4)_{24}$

We can first plot the residuals and compare with random numbers along with a histogram, the curve of the normal distribution and the QQplot:









The plot of the residuals shows a clear weekly pattern, as it could have been expected. Otherwise the rest of the series is very close to white noise. The red curve on the histogram fit with the residuals and they seem normally distributed. On the QQplot we can see some dispersion so it just confirms that our model is not perfect.

Finally, let's do the test sign:

```
##  Expected_sign_changes lower_bound upper_bound Len_sign_changes
##                    836    755.8374    916.1626               847
```

The output 847 is between the confidence interval [756;916] so it can't be rejected that these residuals are white noise.

To conclude, the model is convincing enough since most of the tests conclude that they have properties quite close to white noise. The model will now be used for predictions.

Question 3.5: Predictions

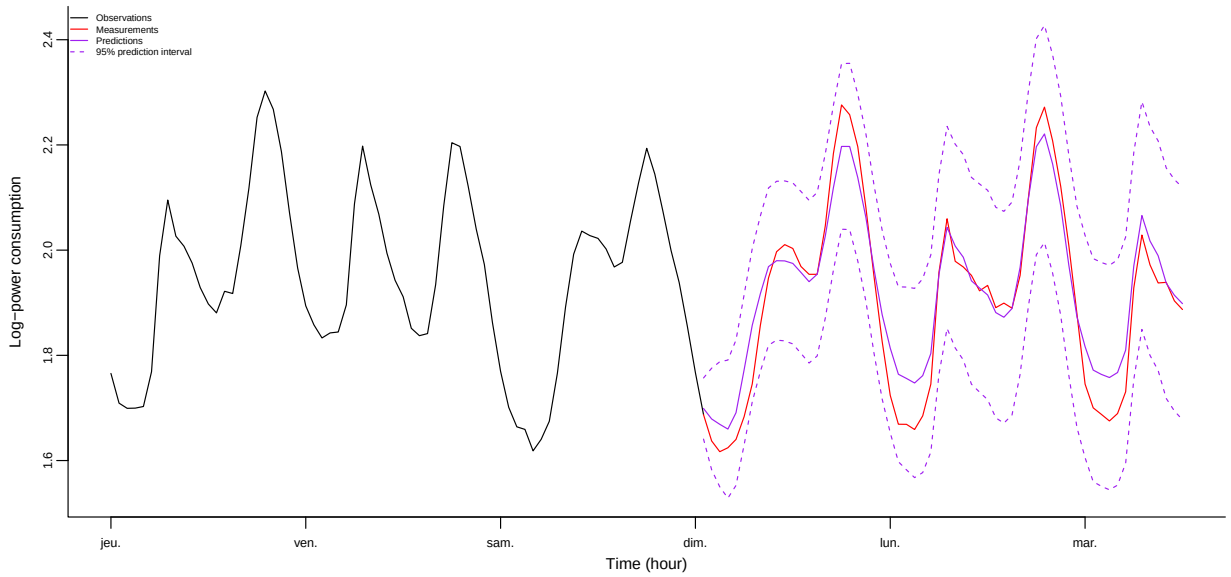
In order to predict from 15-10-2017 and onwards (61 observations) we are going to use the function *predict* in R with the third model as input that will return predictions with corresponding standard errors. Then, the 95% prediction intervals are defined as:

$$\hat{Y}_{t+h} \pm u_{\frac{\alpha}{2}} \sqrt{V[e_{t+h|t}]}$$

with $u_{\frac{\alpha}{2}}$ the $\frac{\alpha}{2}$ quantile of the normal distribution.

Prediction of the log-power consumption

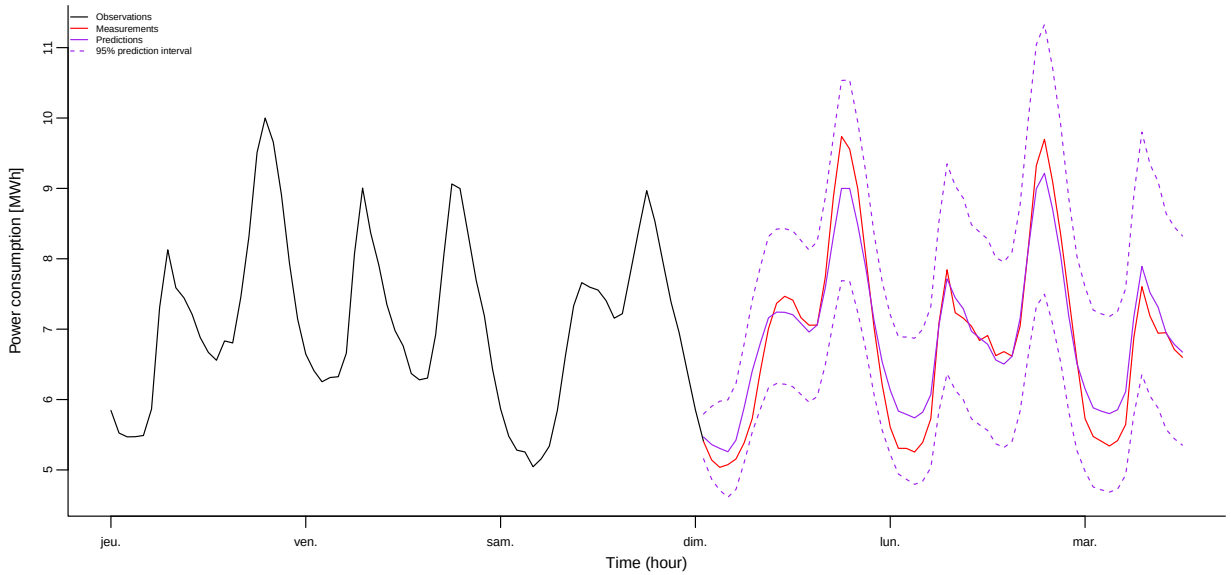
The predicitons for log-transformed data are given by the following plot as well as the confidence intervals in the table:



	2017-10-15 0am	2017-10-15 5am	2017-10-15 11am
Observations	1.687139	1.683432	2.003100
Predictions	1.698909	1.772641	1.974716
Lower bounderies	1.640781	1.629950	1.821694
Upper bounderies	1.757036	1.915333	2.127738
	2017-10-15 23am	2017-10-16 23am	
Observations	1.723837	1.745366	
Predictions	1.813694	1.816661	
Lower bounderies	1.652680	1.605350	
Upper bounderies	1.974709	2.027972	

Prediction of the power consumption

The predicitons for regular data are given by the following plot as well as the confidence intervals in the table:




```
##                   2017-10-15 0am 2017-10-15 5am 2017-10-15 11am
## Observations            5.404000       5.384000        7.412000
## Predictions             5.467978       5.886381        7.204571
## Lower bounderies        5.159200       5.103621        6.182320
## Upper bounderies        5.795236       6.789196        8.395851
##                   2017-10-15 23am 2017-10-16 23am
## Observations             5.606000        5.728000
## Predictions              6.133062        6.151284
## Lower bounderies         5.220952        4.979601
## Upper bounderies         7.204519        7.598662
```

Question 3.6: Comments

One can see that in both cases (log-data and regular data), the predictions are quite faithful to reality and are always within the confidence interval. The predicitons do not have amplitudes as large as the real data but does not display any offset. The trends are well followed in general. Most likely, if the predicitons had to be made for a longer period of time, the lack of weekly seasonality could be more easily seen as well as the fact that weekend and weekday patterns are quite different usually and were not neither taken into account in our model. The latter elements could be considered for a more advanced model. Also, SARIMA models here are well suited for short term analysis but are not suitable for further ahead predicitons.

Appendix

```
## This empties the work space ####
rm(list=ls())
library(ggplot2)
## change directory
## setwd("/path/to/work_dir")
setwd("~/Travail/DTU/Time_series/Ass3")
```

```r
#OLD data !
## reading data,regarding the first lines in the file as names:
data_old <- read.table(file="A3_power.txt", sep="\t", header=TRUE, stringsAsFactors = FALSE)

data_old$Time<-paste(data_old$Date, data_old$Hour, sep=" ")
data_old$Time<-as.POSIXct(data_old$Time, format="%d-%m-%Y %H:%M:%S")
data_old$logPower <- log(data_old$Power)

n_old<-length(data_old$Date)
N_old<-15672
m_old<-15012
M_old<-8785

par(mfrow=c(1,1))
par(mgp=c(2, 0.7,0), mar=c(3,3,2,1))

plot(data_old$Time[1:N_old], data_old$Power[1:N_old],col='black',
     xlim=c(data_old$Time[1],data_old$Time[n_old]),
     ylim=c(min(data_old$Power),max(data_old$Power)),xlab='Time'
     ,ylab="Power consumption [MWh]",
     type='l',main='01/01/2016 - 17/10/2017 every hour')
lines(data_old$Time[N_old:n_old],data_old$Power[N_old:n_old],col='red')
legend('topleft',legend = c("Train data","Test data"),
       col = c('black','red'), pch = c(NA,NA),
       cex=0.7, bty = 'n',lty = c(1,1))

#### New dataset ####
## reading data,regarding the first lines in the file as names:
data <- read.table(file="A3_power_short.txt", sep="\t", header=TRUE, stringsAsFactors = FALSE)

data$Time<-paste(data$Date, data$Hour, sep=" ")
data$Time<-as.POSIXct(data$Time, format="%d-%m-%Y %H:%M:%S")
data$logPower <- log(data$Power)

n<-length(data$Date)
N<-1672
m<-1337
d<-1625

#### Question 1 ####
par(mfrow=c(1,1))
par(mgp=c(2, 0.7,0), mar=c(3,3,2,1))

plot(data$Time[1:N], data$Power[1:N],col='black',xlim=c(data$Time[1],data$Time[n]),
     ylim=c(min(data$Power),max(data$Power)),xlab='Time',ylab="Power consumption [MWh]",
     type='l',main='06/08/2017 - 17/10/2017
     every hour')
lines(data$Time[N:n],data$Power[N:n],col='red')
legend('topleft',legend = c("Train data","Test data"),
       col = c('black','red'), pch = c(NA,NA),
       cex=0.7, bty = 'n',lty = c(1,1))

#Last month
```

```
par(mfrow=c(1,1))
par(mgp=c(2, 0.7,0), mar=c(3,3,2,1))

plot(data$Time[m:N], data$Power[m:N],col='black',xlim=c(data$Time[m],data$Time[n]),
     ylim=c(min(data$Power[m:n]),max(data$Power[m:n])),xlab='Time',ylab="Power consumption [MWh]",
     type='l',main='01/10/2016 - 17/10/2017
     every hour')
lines(data$Time[N:n],data$Power[N:n],col='red')
legend('topleft',legend = c("Train data","Test data"),
       col = c('black','red'), pch = c(NA,NA),
       cex=0.7, bty = 'n',lty = c(1,1))

#log
par(mfrow=c(2,1))
par(mgp=c(2, 0.7,0), mar=c(3,3,2,1))

plot(data$Time[1:N], data$logPower[1:N],col='black',xlim=c(data$Time[1],data$Time[n]),
     ylim=c(min(data$logPower),max(data$logPower)),xlab='Time',ylab="Log Power consumption",
     type='l',main='06/08/2017 - 17/10/2017
     every hour')
lines(data$Time[N:n],data$logPower[N:n],col='red')
legend('topleft',legend = c("Train data","Test data"),
       col = c('black','red'), pch = c(NA,NA),
       cex=0.7, bty = 'n',lty = c(1,1))

#Log Last month
par(mgp=c(2, 0.7,0), mar=c(3,3,2,1))

plot(data$Time[m:N], data$logPower[m:N],col='black',xlim=c(data$Time[m],data$Time[n]),
     ylim=c(min(data$logPower[m:n]),max(data$logPower[m:n])),xlab='Time',ylab="Log Power consumption",
     type='l', main='01/10/2017 - 17/10/2017
     every hour')
lines(data$Time[N:n],log(data$Power[N:n]),col='red')
legend('topleft',legend = c("Train data","Test data"),
       col = c('black','red'), pch = c(NA,NA),
       cex=0.7, bty = 'n',lty = c(1,1))

#Differenciate
diff1<-diff(data$logPower[1:n],1)

par(mfrow=c(1,1))
par(mgp=c(2, 0.7,0), mar=c(3,3,2,1))

plot(data$Time[1:N], diff1[1:N],col='black',xlim=c(data$Time[1],data$Time[n]),
     ylim=c(min(diff1),max(diff1)),xlab='Time',ylab="Diff Power consumption [MWh]",
     type='l',main='06/08/2017 - 17/10/2017
     every hour')
lines(data$Time[N:n],diff1[N:n],col='red')
legend('topleft',legend = c("Train data","Test data"),
       col = c('black','red'), pch = c(NA,NA),
       cex=0.7, bty = 'n',lty = c(1,1))

################################Question 2#####################
```

```r
l<-7
lag<-24*l+1

#Non transform sample
par(mfrow=c(2,1))
par(mgp=c(2, 0.7,0), mar=c(3,3,2,1))

acf(x = data$Power[1:N], lag = lag, type = 'correlation',ann=FALSE, lwd=2,
    main ='')
mtext(side = 1,text = 'lag', line = 2, cex = 1.3)
mtext(side = 2,text = 'ACF', line = 2, cex = 1)
for (i in 1:n)
{lines(x = c(24*i,24*i), y = c(-200,200), type = 'l', lty = 2,lwd=1.4,
       col='darkorange2')}

pacf(x = data$Power[1:N], lag = lag, ann=FALSE, lwd=2, main ='')
mtext(side = 1,text = 'lag', line = 2, cex = 1.3)
mtext(side = 2,text = 'PACF', line = 2, cex = 1)
for (i in 1:n)
{lines(x = c(24*i,24*i), y = c(-200,200), type = 'l', lty = 2,lwd=1.4,
       col='darkorange2')}

my.tsdiag <- function(dat,   nlag = lag, ...){
  if(class(dat) == "Arima")
    dat <- dat$residuals
  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,lag = lag, type = 'correlation',ann=FALSE, lwd=2,
      main ='')
  mtext(side = 1,text = 'lag', line = 2, cex = 1.3)
  mtext(side = 2,text = 'ACF', line = 2, cex = 1)
  for (i in 1:n)
  {lines(x = c(24*i,24*i), y = c(-200,200), type = 'l', lty = 2,lwd=1.4,
         col='darkorange2')}
  pacf(dat,lag = lag, ann=FALSE, lwd=2,main ='')
  mtext(side = 1,text = 'lag', line = 2, cex = 1.3)
  mtext(side = 2,text = 'PACF', line = 2, cex = 1)
  for (i in 1:n)
  {lines(x = c(24*i,24*i), y = c(-200,200), type = 'l', lty = 2,lwd=1.4,
         col='darkorange2')}

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

#log transformed Sample
par(mfrow=c(2,1))
par(mgp=c(2, 0.7,0), mar=c(3,3,2,1))

acf(x = data$logPower[1:N], lag = lag, type = 'correlation',ann=FALSE, lwd=2,
    main ='')
mtext(side = 1,text = 'lag', line = 2, cex = 1.3)
```

```r
mtext(side = 2,text = 'ACF', line = 2, cex = 1)
for (i in 1:n)
{lines(x = c(24*i,24*i), y = c(-200,200), type = 'l', lty = 2,lwd=1.4,
       col='darkorange2')}

pacf(x = data$logPower[1:N], lag = lag, ann=FALSE, lwd=2, main ='')
mtext(side = 1,text = 'lag', line = 2, cex = 1.3)
mtext(side = 2,text = 'PACF', line = 2, cex = 1)
for (i in 1:n)
{lines(x = c(24*i,24*i), y = c(-200,200), type = 'l', lty = 2,lwd=1.4,
       col='darkorange2')}

#Diff
par(mfrow=c(2,1))
par(mgp=c(2, 0.7,0), mar=c(3,3,2,1))

acf(x = diff1[1:N], lag = lag, type = 'correlation',ann=FALSE, lwd=2,
    main ='')
mtext(side = 1,text = 'lag', line = 2, cex = 1.3)
mtext(side = 2,text = 'ACF', line = 2, cex = 1)
for (i in 1:n)
{lines(x = c(24*i,24*i), y = c(-200,200), type = 'l', lty = 2,lwd=1.4,
       col='darkorange2')}

pacf(x = diff1[1:N], lag = lag, ann=FALSE, lwd=2, main ='')
mtext(side = 1,text = 'lag', line = 2, cex = 1.3)
mtext(side = 2,text = 'PACF', line = 2, cex = 1)
for (i in 1:n)
{lines(x = c(24*i,24*i), y = c(-200,200), type = 'l', lty = 2,lwd=1.4,
       col='darkorange2')}

###### question 4

Model1<-arima(x = data$logPower[1:N], order = c(2, 1, 0),
              seasonal = list(order = c(0, 1, 3), period = 24))
my.tsdiag(Model1)
Model1

Model2<-arima(x = data$logPower[1:N], order = c(2, 1, 0),
              seasonal = list(order = c(1, 1, 4), period = 24))
my.tsdiag(Model2)
Model2

Model3<-arima(x = data$logPower[1:N], order = c(3, 1, 1),
              seasonal = list(order = c(1, 1, 4), period = 24))
Model3
#aic = -6856.85
my.tsdiag(Model3)

model1<-arima(x = data$logPower[1:N], order = c(2, 1, 3),
             seasonal = list(order = c(1, 1, 2), period = 24))
my.tsdiag(model1)
```

```r
## Comparing with random numbers ...
par(mfrow=c(2,1))
ts.plot(Model3$residuals, col='black',ylab='Residuals')
ts.plot(ts(rnorm(length(Model3$residuals))*sqrt(Model3$sigma2)),
        ylab='Simulated residuals')

## Looking at distributional assumption
par(mfrow=c(1,2))
hist(Model3$residuals,probability=T,col='white',main='Histogram of the residuals', xlab='')
curve(dnorm(x,sd=sqrt(Model3$sigma2)), col=2, lwd=3, add = TRUE)

qqnorm(Model3$residuals)
qqline(Model3$residuals,col=2)


# sign test mean and sd:
Test <- data.frame(row.names=c(''))

Len <- length(Model3$residuals)
Test$Expected_sign_changes <- (Len-1)/2 #expected normal of sign changes
### sd: sqrt((Len-1)/4)
### 95% interval:
Test$lower_bound <- (Len-1)/2 - 1.96 * sqrt(Len-1/4)
Test$upper_bound <- (Len-1)/2 + 1.96 * sqrt(Len-1/4)
### test:
res <- Model3$residuals
Test$Len_sign_changes <- sum( res[-1] * res[-length(res)]<0 )

### or:
binom.test(Len.sign.changes, length(Model3$residuals)-1)

######## Question 5: Prediction
Pred<-predict(Model3, n.ahead = n-N)

#95% confidence interval
sd<-Pred$se
tlower95<-qnorm(0.025)*sd
tupper95<-qnorm(0.975)*sd

Predupper<-Pred$pred+tupper95
Predlower<-Pred$pred+tlower95

par(mfrow=c(1,1))
par(mgp=c(2, 0.7,0), mar=c(3,3,2,1))
plot(data$Time[d:n],c(data$logPower[d:(N+1)],rep(NA,n-N-1)), type = 'l',lwd=1,lty=1,ann=FALSE,
     bty='l',pch=4,cex=0.6,cex.axis = 0.7, mgp=c(3,0.5,0),
     ylim=c(min(Predlower),max(Predupper)))
lines(data$Time[(N+1):n],data$logPower[(N+1):n], type = 'l',
      lwd=1,lty=1,ann=FALSE,bty='l',pch=4,cex=0.6, cex.axis = 0.7, mgp=c(3,0.5,0), col ='red')
lines(data$Time[(N+1):n], Pred$pred, type = 'l',  lwd=1,lty=1,ann=FALSE,bty='l',pch=4,cex=0.6,
      cex.axis = 0.7, mgp=c(3,0.5,0), col ='purple')
lines(data$Time[(N+1):n],Predupper, type = 'l',  lwd=1,lty=2,ann=FALSE,bty='l',pch=4,cex=0.6,
      cex.axis = 0.7, mgp=c(3,0.5,0), col ='purple')
```

```
lines(data$Time[(N+1):n],Predlower, type = 'l',  lwd=1,lty=2,ann=FALSE,bty='l',pch=4,cex=0.6,
      cex.axis = 0.7, mgp=c(3,0.5,0), col ='purple')

mtext(side = 1,text = 'Time (hour)', line = 1.5, cex = 0.85)
mtext(side = 2,text = 'Log-power consumption', line = 1.9, cex = 0.85)

legend('topleft',legend = c("Observations", "Measurements",'Predictions','95% prediction interval'),
       col = c("black","red",'purple','purple'),lty = c(1,1,1,2), pch = c(NA,NA,NA,NA),cex=0.5,
       bty = 'n')

table <- matrix(cbind(data$logPower[c(N+1,N+6,N+12,N+24,N+48)],Pred$pred[c(1,6,12,24,48)],Predlower[c(1
colnames(table) <- c("2017-10-15 0am","2017-10-15 5am","2017-10-15 11am","2017-10-15 23am"
                     ,"2017-10-16 23am")
rownames(table) <- c("Observations","Predictions","Lower bounderies","Upper bounderies")
table <- as.table(table)
table

#Original domain
#Prediction
Ini_Pred<-exp(Pred$pred)

#95% confidence interval
Ini_Predupper<-exp(Predupper)
Ini_Predlower<-exp(Predlower)

#PLot in the initial domain

par(mfrow=c(1,1))
par(mgp=c(2, 0.7,0), mar=c(3,3,2,1))
plot(data$Time[d:n],c(data$Power[d:(N+1)],rep(NA,n-N-1)), type = 'l',lwd=1,lty=1,ann=FALSE,
     bty='l',pch=4,cex=0.6,cex.axis = 0.7, mgp=c(3,0.5,0),
     ylim=c(min(Ini_Predlower),max(Ini_Predupper)))
lines(data$Time[(N+1):n],data$Power[(N+1):n], type = 'l',
      lwd=1,lty=1,ann=FALSE,bty='l',pch=4,cex=0.6, cex.axis = 0.7, mgp=c(3,0.5,0), col ='red')
lines(data$Time[(N+1):n], Ini_Pred, type = 'l',  lwd=1,lty=1,ann=FALSE,bty='l',pch=4,cex=0.6,
      cex.axis = 0.7, mgp=c(3,0.5,0), col ='purple')
lines(data$Time[(N+1):n],Ini_Predupper, type = 'l',  lwd=1,lty=2,ann=FALSE,bty='l',pch=4,cex=0.6,
      cex.axis = 0.7, mgp=c(3,0.5,0), col ='purple')
lines(data$Time[(N+1):n],Ini_Predlower, type = 'l',  lwd=1,lty=2,ann=FALSE,bty='l',pch=4,cex=0.6,
      cex.axis = 0.7, mgp=c(3,0.5,0), col ='purple')

mtext(side = 1,text = 'Time (hour)', line = 1.5, cex = 0.85)
mtext(side = 2,text = 'Power consumption', line = 1.9, cex = 0.85)

legend('topleft',legend = c("Observations", "Measurements",'Predictions','95% prediction interval'),
       col = c("black","red",'purple','purple'),lty = c(1,1,1,2), pch = c(NA,NA,NA,NA),cex=0.5,
       bty = 'n')

table <- matrix(cbind(data$Power[c(N+1,N+6,N+12,N+24,N+48)],Ini_Pred[c(1,6,12,24,48)],Ini_Predlower[c(1
colnames(table) <- c("2017-10-15 0am","2017-10-15 5am","2017-10-15 11am","2017-10-15 23am"
                     ,"2017-10-16 23am")
rownames(table) <- c("Observations","Predictions","Lower bounderies","Upper bounderies")
table <- as.table(table)
```

```
table
```