Ramiro Mata (s161601)**Assignment 3: Time Series Analysis:Estimating ARMA Processes and Seasonal Processes**

Ramiro Mata (s161601)

1 Presenting the data

Figure 1 shows the time series data. Note that the order of the data was reversed to go forward in time. In addition, the last 4 values were excluded as they contain NA values for the NOx column. This resulted in a time series starting on January 7th 2017 at 00:00 and ending on February 6th 2017 at 15:00. The last 48 hours, however, are not used for training the models as that is our test set for our predictions. That means that we will predict the time period beginning on Saturday February 4th at 15:00 and ending Monday February 6th at 15:00. We implement the reversing and the frequency of 24 hours in R as follows:

```
data <- read.table("hcoe17.csv",header = T,sep = ";")
xts  <- ts(rev(data[1:735,3]), frequency = 24)  #reversing data!
lnxts <- log(xts) #log transformation of data to subdue increase in variance
```

1.1 Comment on Behavior, Stationarity and Transformations

As seen in Figure 1 there is a daily **periodicity** in the data. Because of this, notice that sampling for the mean will result in different values depending on where within that periodicity one samples the data. For instance, if we sample during a peak versus a trough, noticeably different means will be sampled. Therefore, we don't expect the mean to be invariant with time, and thus we can say that this time series is **not stationary**. In addition, there are noticeable higher peaks towards the end. One potential Box-Cox transformation to stabilize the variance is the **log transformation**, which we implement as well. Further, we stack weekly data in Figure 3 to visualize any potential weekly patterns. As can be seen, Mondays, Thursdays and Fridays seem particularly traffic-heavy days. The weekdays' periodicity is more accentuated compared to the weekends, which can reflect strict driving times invariant to weather (i.e. people have to get to their jobs). The weekends, however, seem more sporadic and could be influenced by citizens' more ample flexibility, and whose driving activities could be influenced more by other factors such as local weather. Therefore, we would expect weekdays to be adequately predicted with the given data while weekend predictions could be supplemented by further data (other covariates), such as weather forecasts, holiday schedules, and city social events.

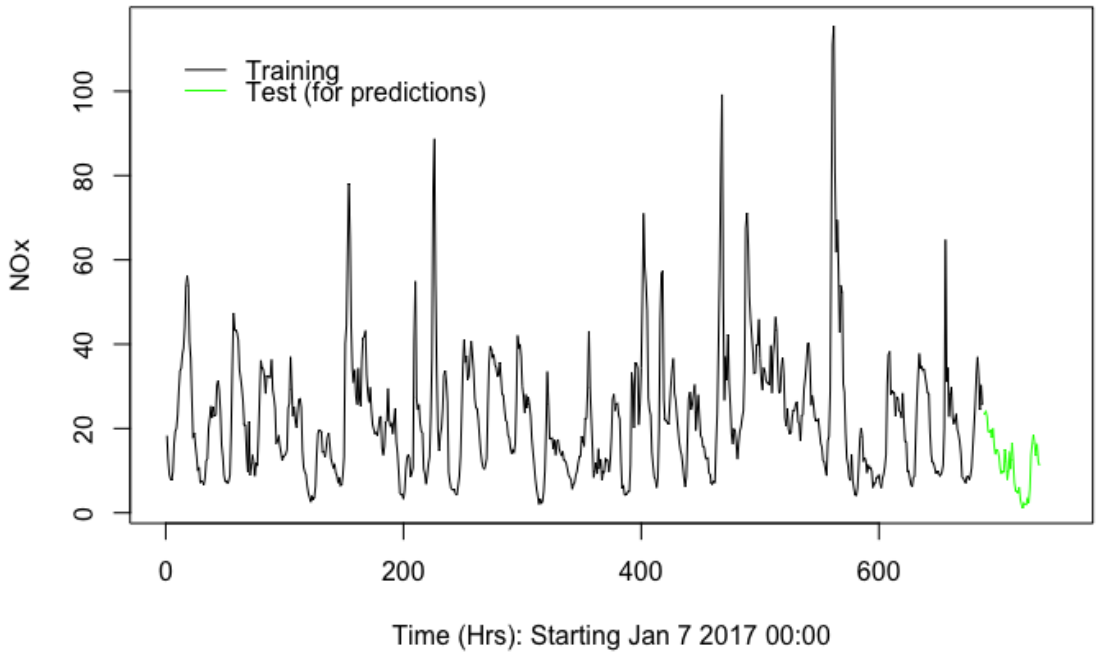

Figure 1: **The time series data is plotted beginning on January 7th 2017 at 00:00 and ending February 6th 2017 at 15:00. The numbers on the x axis represent the hours starting from January 7. Clearly there is a daily pattern in NO_x emissions. Note that the last 2 days (48 hrs) is what we seek to predict (in green).**

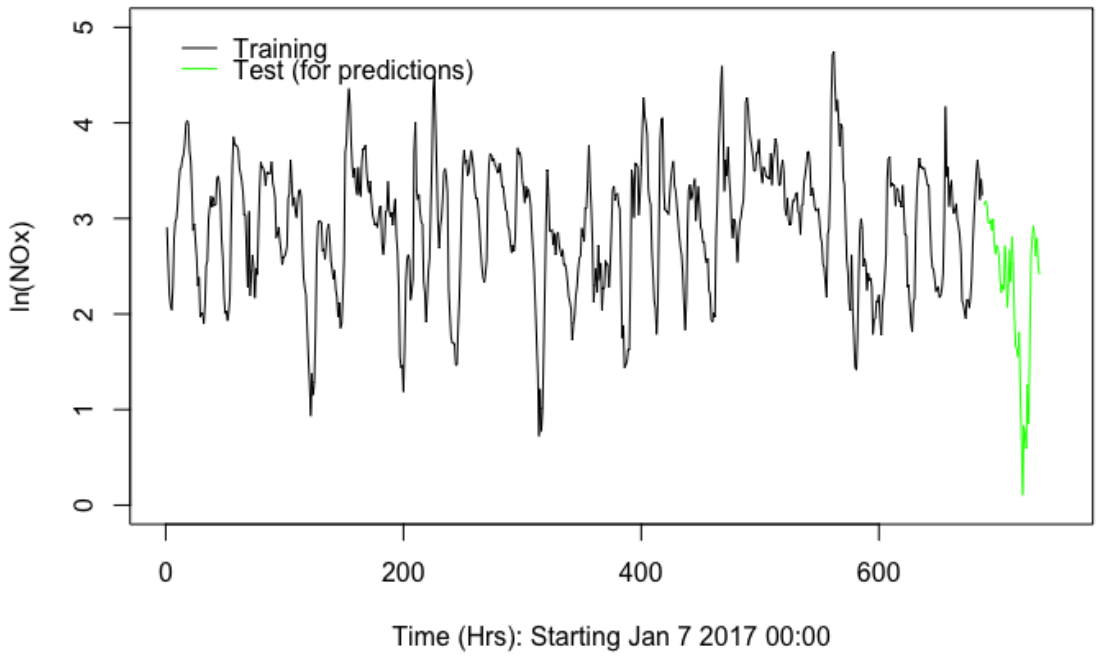

Figure 2: **The time series data with a log-transform (to stabalize variance) is plotted beginning on January 7th 2017 at 00:00 and ending February 6th 2017 at 15:00. The numbers on the x axis represent the hours starting from January 7. Clearly there is a daily pattern in NO_x emissions. Note that the last 2 days (48 hrs) is what we seek to predict (in green).**

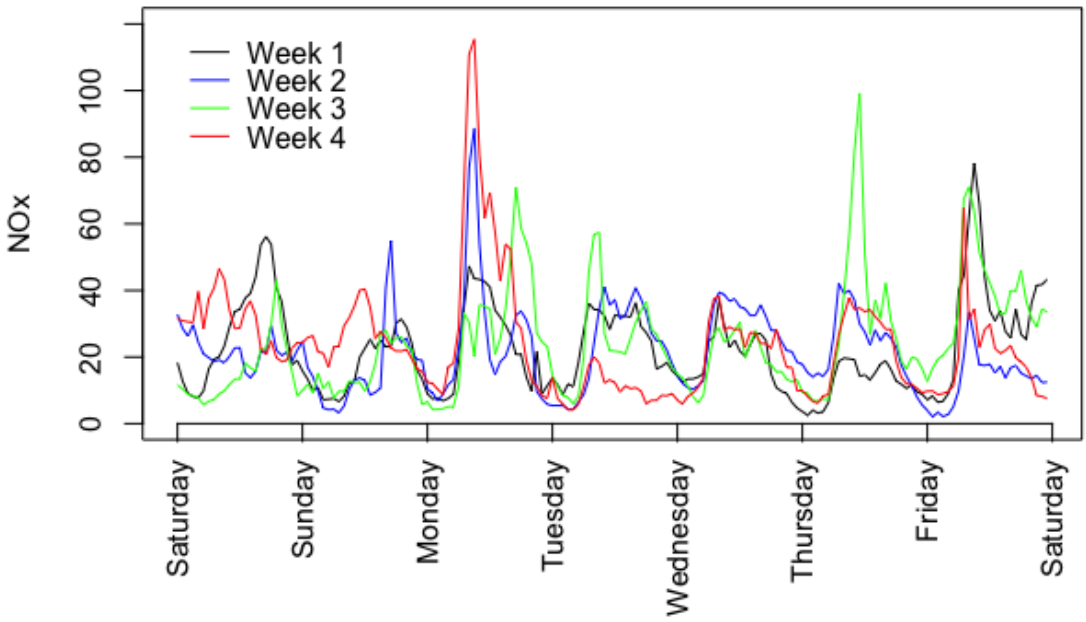

Figure 3: **Same data as above, but the 4 weeks have been stacked to visualize weekly patterns. Mondays, Thursdays and Fridays seem particularly traffic-heavy days. The weekdays' periodicity is more accentuated compared to the weekends, which can reflect strict driving times invariant to weather (i.e. people have to get to their jobs). The weekends, however, seem more sporadic and could be influenced by citizens' more ample flexibility, and whose driving activities could be influenced more by other factors such as local weather. Therefore, we would expect weekdays to be adequately predicted with the given data while weekend predictions could be supplemented by further data (other covariates), such as weather forecasts, holiday schedules, and city social events.**

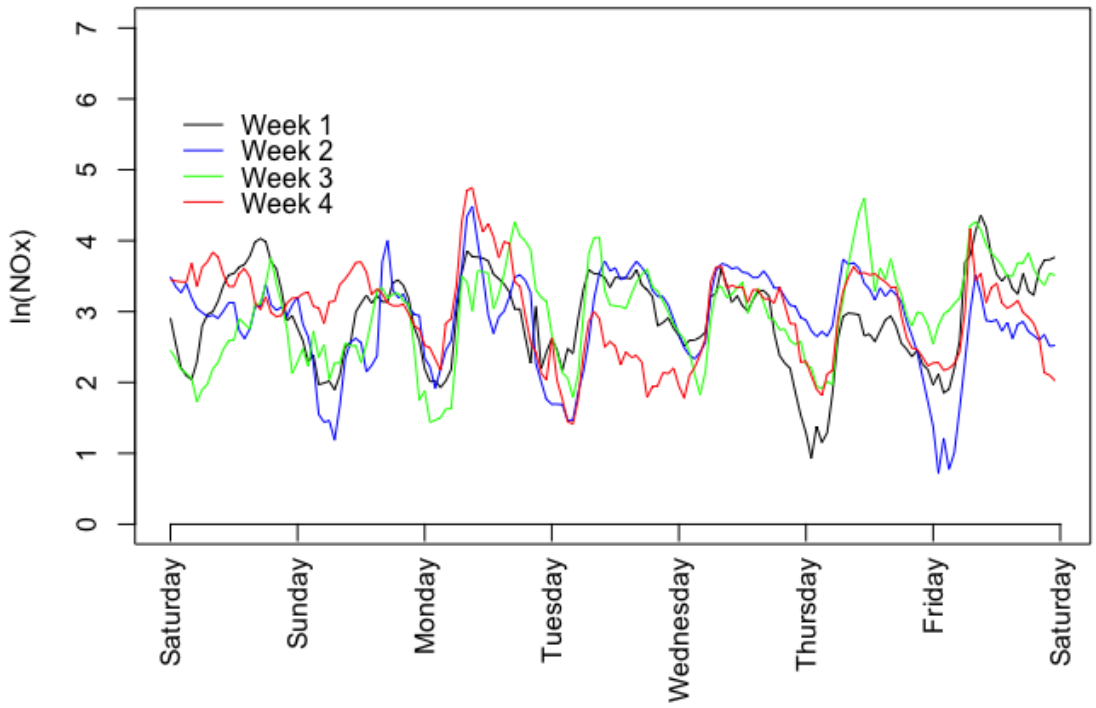

Figure 4: **Same as in Figure 3 but with log transformation. Applying the transformation subdues the peaks exhibited before and therefore stabilizes the variance to some extent. In addition, the weekday periodicity is more clearly illustrated than before. As without the transformation, the weekend pattern is less accentuated than the weekdays.**

2 ACF and PACF

The ACF for both (log transform and original) doesn't exhibit the classic exponential drop to 0 displayed by clear AR processes. However, we do see a sinuosidal pattern that tends somewhat towards zero, but which may not be as clear-cut due to the daily and potentially weekly seasonality we observe. In other words, the seasonality may be masking the exponential drop towards zero since it shows up periodically in the ACF plot. Nonetheless, we see that the PACF drops under the 95% CI after $k = 2$ and $k = 3$ for the original and the log transformed time series, respectively, which suggests an AR(2) and AR(3) model order for the non-seasonal component. In addition, we can see in both PACFs that lags after the dashed lines (24 multiples) exceed the confidence bounds. This is in line with our hypothesized daily seasonality. Note that immediately before the dashed lines, we also see some lags that exceed the confidence bounds and slowly fade after 100 lags. This corresponds to the last hours of the day (i.e. the evening) and potentially shows that there might be some correlation (cyclical behavior) during those hours of the day across the board. For simplicity, however, we can start with our initial hypothesis of a 24 hour seasonality and focus on the log-transformed data from now on, so we will add an AR(3) for our initial model. Further, to start with a model as simple as possible, we can for now ignore any MA components in the initial model.

2.1 Seeking Stationarity

Apart from the seasonal AR(3) initial model mentioned above, we can start a new branch of models with a different strategy - seeking stationarity first! Basically, our time series does not look stationary and so we can take a seasonal difference and then a regular difference in our ARIMA model and see what our time series, ACF and PACF look like. We do this in R as follows:

```
tsdisplay(diff(lnxts,24), lag.max = 200,
          main="Seasonally differenced: (0,0,0)x(0,1,0)_24", xlab="Time(hrs)")

tsdisplay(diff(diff(lnxts,24)), lag.max = 200,
          main="Non-seasonal & Seasonally differenced: (0,1,0)x(0,1,0)_24", xlab="
              Time (hrs)")
```

This is visualized in Figures 7 and 8. We can see that with a seasonal difference (24) the ACF still does not behave as we would expect. When (in addition) we take a difference in the non-seasonal component, we see that the time series looks much more stationary and the ACF and PACF now behave as something that we can recognize from the "Golden Table." In particular, notice that the ACF has a main spike at k=24, and the PACF shows exponential decay in the seasonal lags (i.e. in the lags of multiples of 24). This behavior in the ACF and PACF suggest a seasonal MA component leading to an $(0, 1, 0)x(0, 1, 1)_{24}$ ARIMA model. Now with this double differenced seasonal MA(1) and the above mentioned non-differenced AR(3) models, we will use them as our initial building blocks to explore variations of these and seek the best model by looking at AIC, the models' coefficient p-values, RMSE to get an idea of their associated error, and finally do likelihood ratio tests to obtain a simple as possible model in an attempt to reduce overfitting and to also to embrace parsimony.

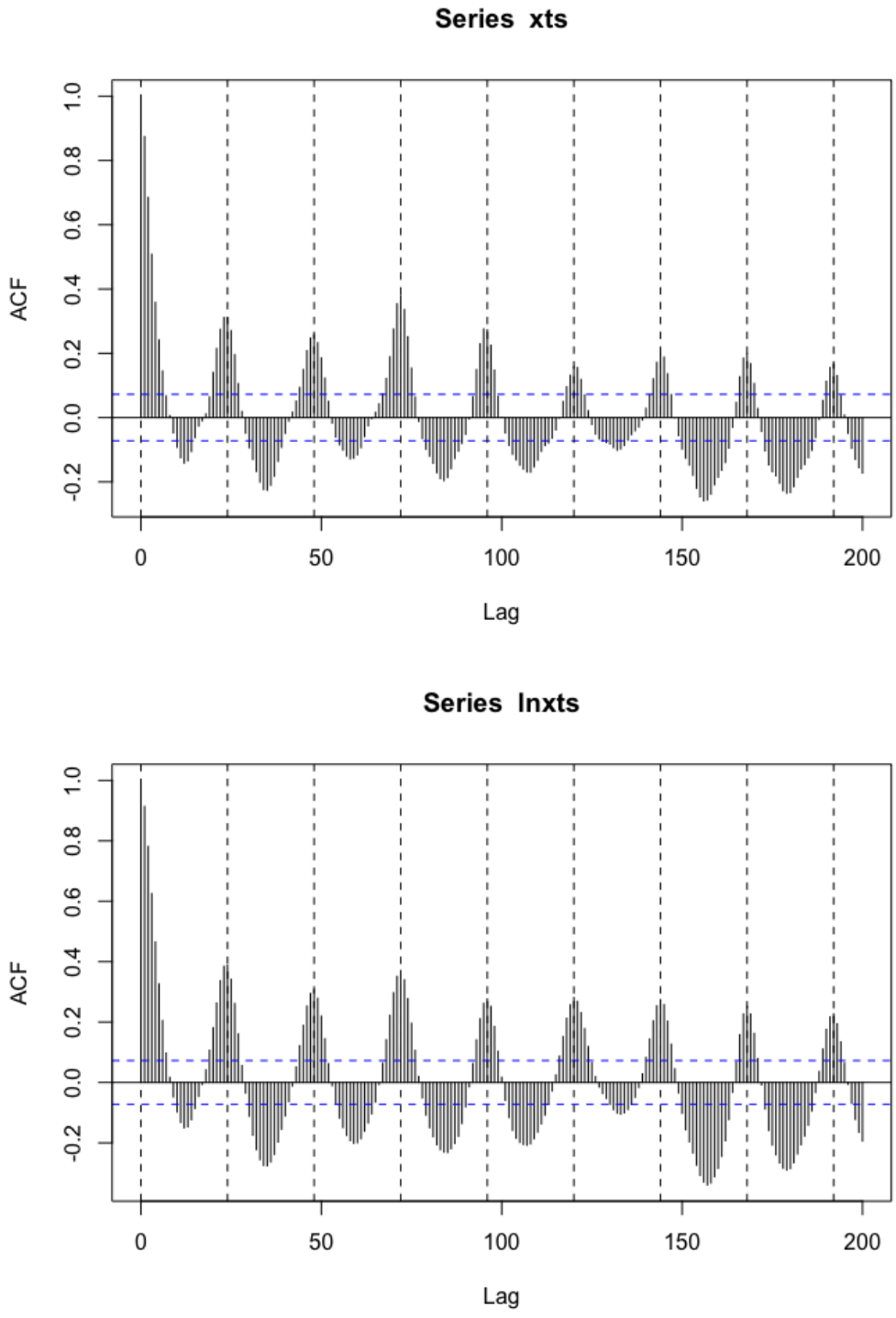

Figure 5: The upper plot shows the ACF of the original time series while the lower shows the ACF when the log transform is applied. Evidently both plots look very similar. Note that we show more than 168 (24x7) lags expose weekly patterns. Dashed lines have been placed at multiples of 24.

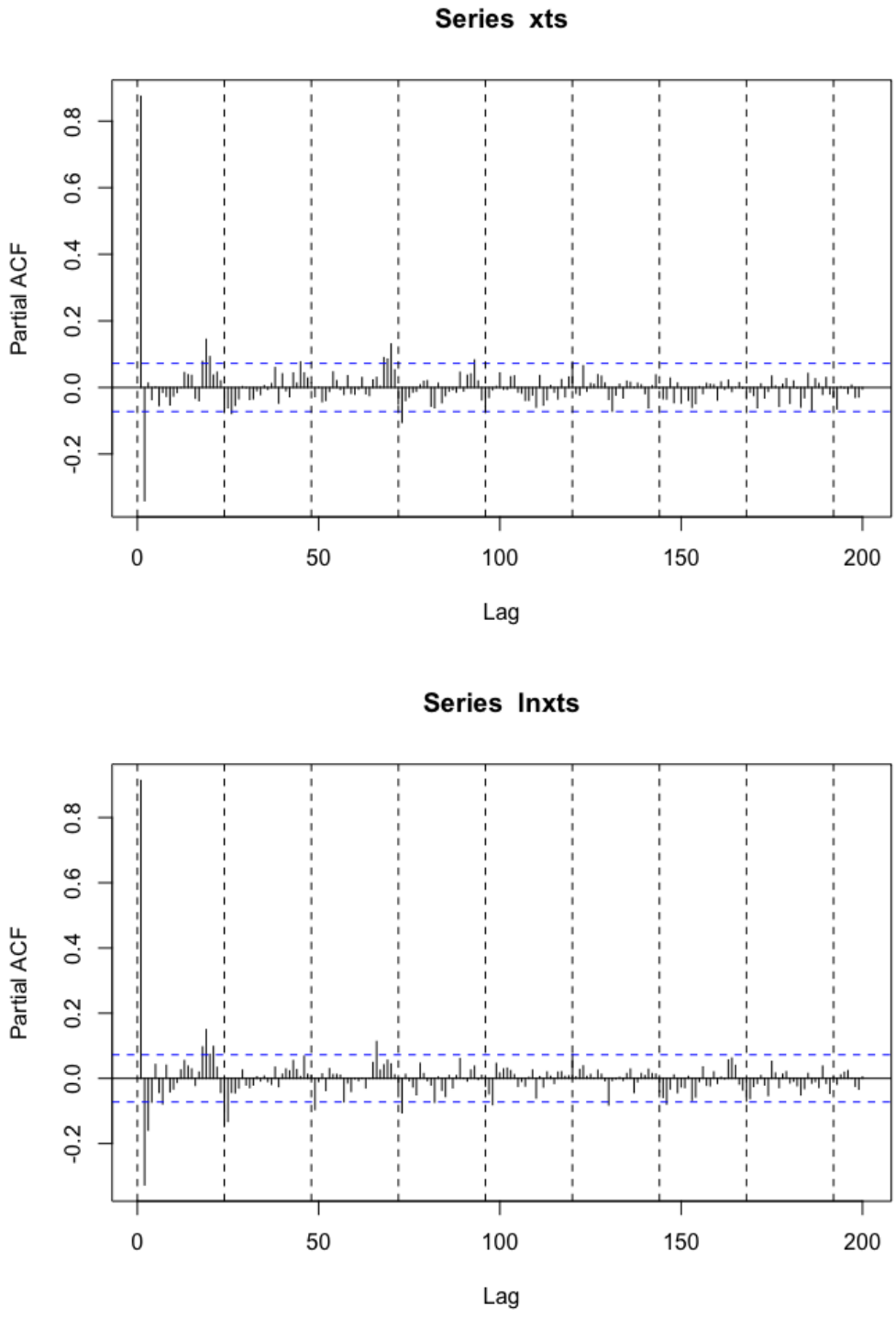

Figure 6: The upper plot shows the PACF of the original time series while the lower shows the PACF when the log transform is applied. Evidently both plots look very similar. Note that we show more than 168 (24x7) lags expose weekly patterns. Dashed lines have been placed at multiples of 24.

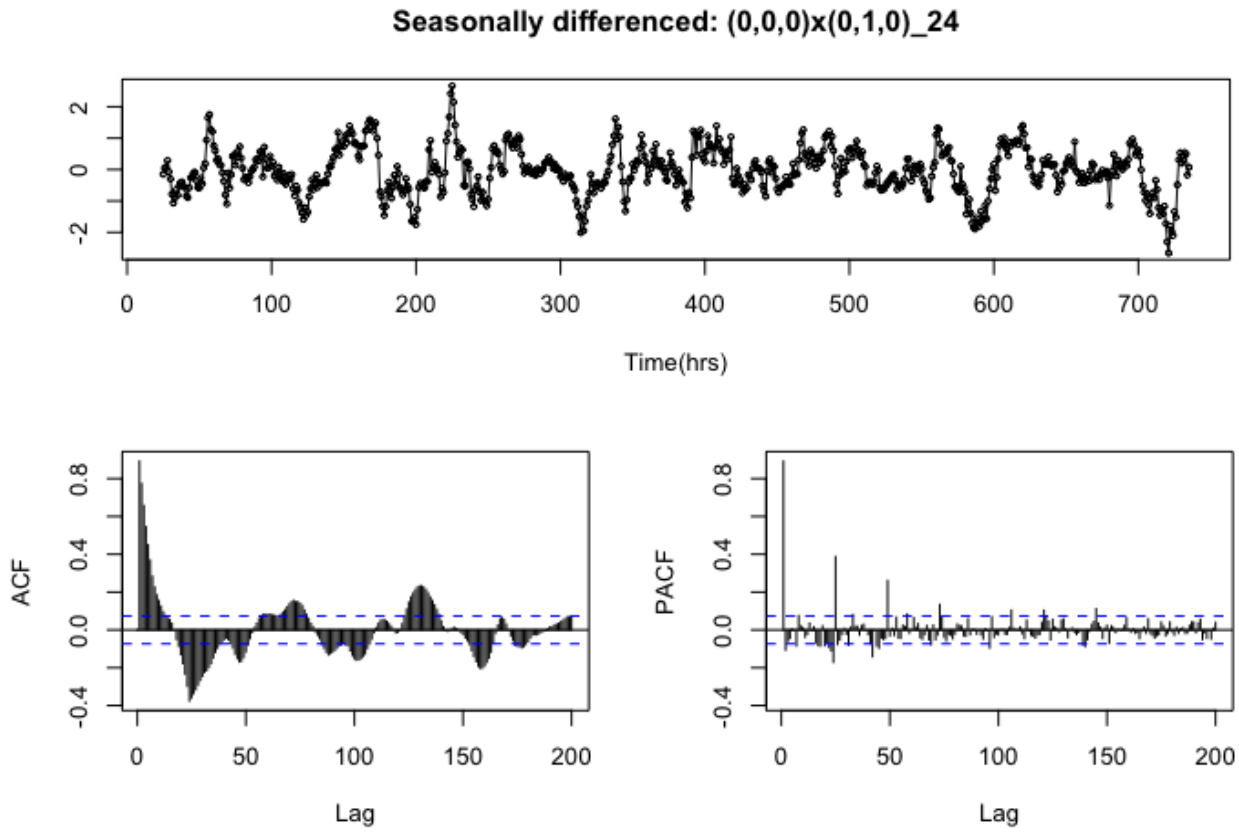

Figure 7: **SEEKING STATIONARITY: The figure shows the log-transformed data after a seasonal difference. Notice that the data now seem more stationary, however, the ACF's behavior is still not clear cut. The PACF on the other hand, does show relevant spikes at multiples of our seasonality (24).**

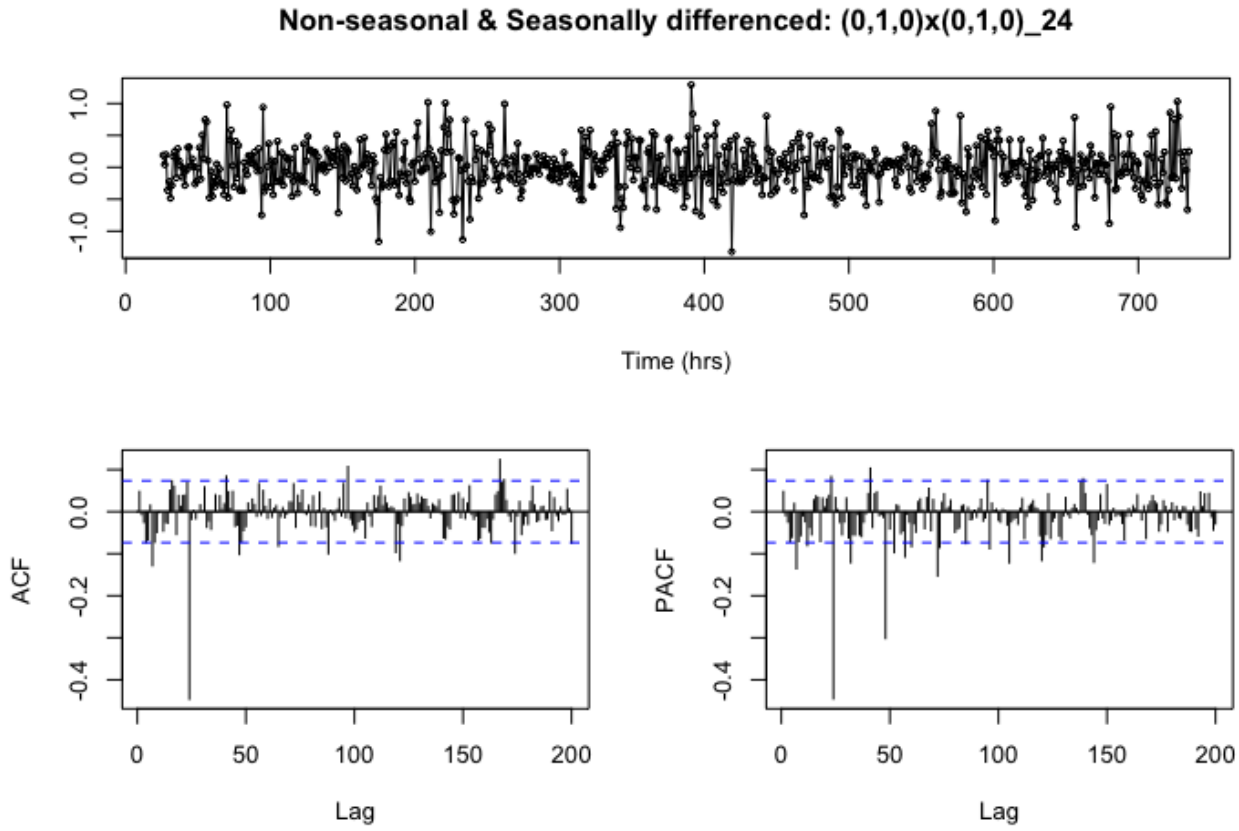

Figure 8: **SEEKING STATIONARITY: The figure above shows the log-transformed data after a seasonal and a non-seasonal difference. Notice that the time series seems stationary. Furthermore, the ACF and the PACF now behave as something that we can interpret using the "Golden Table." Specifically, the ACF shows a seasonal spike at lag 24 (as expected), and the PACF shows seasonal spikes at multiples of 24, which suggest a seasonal MA component.**

3 Model selection

3.1 Initial Models

As summarized in the Time Series Analysis book by Henrik Madsen in Section 6.3, one can move on to estimating the ARMA model order after safely assuming ergodicity and stationarity in the time series. Given the daily patterns, stationarity is still suspect in one of our initial models (the non-differenced AR(3) model).

For now however, we start with that initial model reflecting the ACF and PACF plots as well as our strong belief in daily seasonality. To recap, we see that the PACF drops under the 95% CI after $k = 3$ in the log transformed time series, which suggests an AR(3) model order. In addition, we can see in both PACFs that lags after the dashed lines (24 multiples) exceed the confidence bounds. This is in line with our hypothesized daily seasonality. Eventhough the ACF doesn't drop to 0 sufficiently fast as classical examples of AR processes, we suspect the daily periodicity maybe masking this and hence we move forward towards formulating our initial model as an AR(3) model with seasonality of 24 using the log-transformed data. The second initial model we pursue is the one that first seeks stationarity. Basically, as figures 7 and 8 show, by taking a seasonal and a non-seasonal difference, our time series looks much more stationary. Further, the ACF and PACF now look like something we can interpret using the "Golden Table." To recap, the ACF shows a seasonal spike at lag 24 (as expected), and the PACF shows seasonal spikes at multiples of 24, which suggest a seasonal MA component. We formulate the two initial models as follows:

$$InitialModel1 : ARIMA(p,d,q)x(P,D,Q) = ARIMA(3,0,0) \times (1,0,0)_{24} \tag{1}$$

$$InitialModel2 : ARIMA(p,d,q)x(P,D,Q) = ARIMA(0,1,0) \times (0,1,1)_{24} \tag{2}$$

3.2 Iterating through Different Models

We take the initial two models above and try different variations of them and compare them (when relevant) using the AIC measure. However, we can only use AIC within models of the same seasonal difference order. Therefore, we also compute the RMSE of each model to do cross-comparison across models of different seasonal difference orders. Further, when necessary, we discard certain AR or MA coefficients if their p-values show them to be insignificant. For instance, in our Model 6 the 2nd AR coefficient was insignificant, so we reformulated the model everything else the same but without the coefficient term shown to be insignificant. We do this in R as follows:

```
model6 <- arima(x = lnxts.train, order = c(3, 0, 0), seasonal = list(order = c(0, 1,
    1), period = 24))
model6; p_values(model6);  # aic = 15.75, ar2 coeff. insig

model6s <- arima(x=lnxts.train, order=c(3,0,0), seasonal=list(order=c(0,1,1),period
   =24),transform.pars = FALSE, fixed= c(NA,0,NA,NA))
model6s;                    # aic = 14.69   # REMOVING INSIGNIFICANT AR2 coefficient
```

Finally, in order to prevent overfitting and embrace parsimony (i.e. "the simpler the better"), we use a likelihood ratio test to determine between models of different parameter complexity and performance. The models are listed in Table 1. Note that we only list the best models found. For all models tested, we refer the reader to the Appendix in the CODE section.

Model	Formulation	AIC	RMSE
4	$(3,0,1)x(1,0,1)_{24}$	-9.15	0.9664464
5	$(2,0,1)x(1,0,1)_{24}$	-10.51	0.9654734
14	$(3,0,0)x(1,0,2)_{24}$	-8.18	0.9674714
14s	$(3,0,0)x(1,0,2)_{24}$	-10.74	0.9674714
6	$(3,0,0)x(0,1,1)_{24}$	15.75	0.9678455
6s	$(3,0,0)x(0,1,1)_{24}$	14.69	0.9679578
10	$(2,0,0)x(0,2,1)_{24}$	80.38	0.8925458
11	$(2,0,1)x(0,2,1)_{24}$	482.96	0.888723
Auto D	$(2,0,4)x(0,1,0)_{24}$	-1160.05	0.9739474
13	$(0,1,1)x(0,1,1)_{24}$	61.95	0.9710542
Auto DD	$(5,1,0)x(0,1,0)_{24}$	-1096.57	1.314824

TABLE 1: The table shows the best models tested. Note that they are listed in order of differencing (not model number). When an s is placed after the model number, it signifies that a coefficient was insignificant and thus removed from the original model. In both cases, the AIC went down (meaning the model improved!) The Auto models are those as suggested by the auto.arima function. Note that we cannot compare all the above models with AIC due to differences in seasonal differencing order. However, the RMSE allows us to compare them all.

3.3 Model Diagnostics

After testing a variety of models, we chose to select the best (according to AIC (lower AIC = better fit) among 3 nested groups: no differencing, seasonal differencing, seasonal differencing plus non-seasonal differencing. These are listed in Table 1 in order of differencing order. When two nested models had very similar performance, we implemented a **likelihood ratio test** to determine whether the more complex model is significanlty better than the simpler model. We did this for the following 2 nested model groups in R as follows:

```
# Ratio Likelihood Tests: Are the more complex models significantly better? A: Nope!
ratiotest(model5, model4, 1)    #not sig. p =  0.4206363
ratiotest(model10, model11, 1)  #not sig. p =  0.07512814
```

Next, we used **residual analysis (looking at ACF, QQ plot, cumulative periodogram, Ljung-Box test) to determine whether the residuals looked like white noise (not auto-correlated) and in general determine model fitness**. The figures below show the residual plots of the most relevant models. Further, all these models passed the **signs test**.

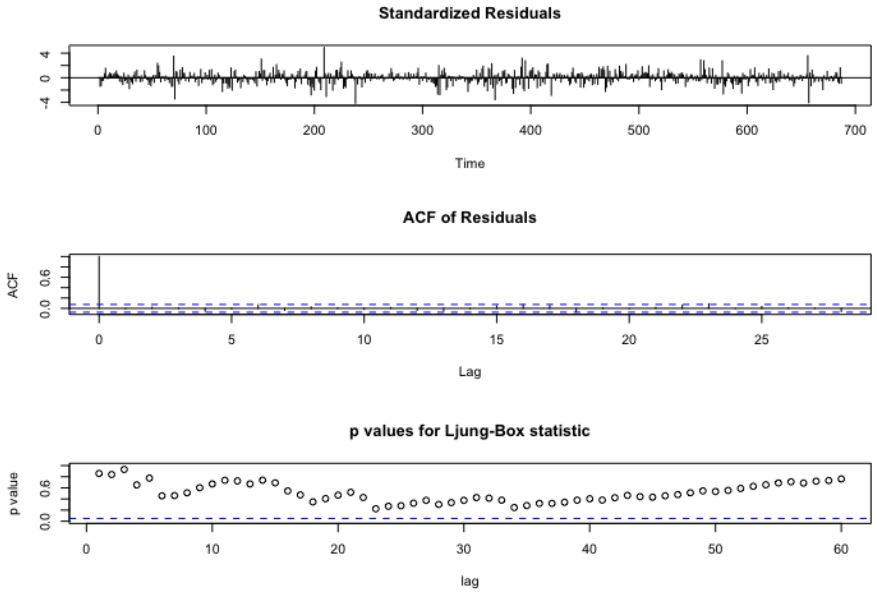

Figure 9: The diagnostics above are for Model 5. Note however that models 4 and 6 are remarkably similar. The ACF shows that there are no spikes except for the initial one, which is a good sign and tells us that the residuals are not auto-correlated. Further, all the p-values in the Ljung-Box test are outside the bounds, which suggests the residuals are not autocorrelated.

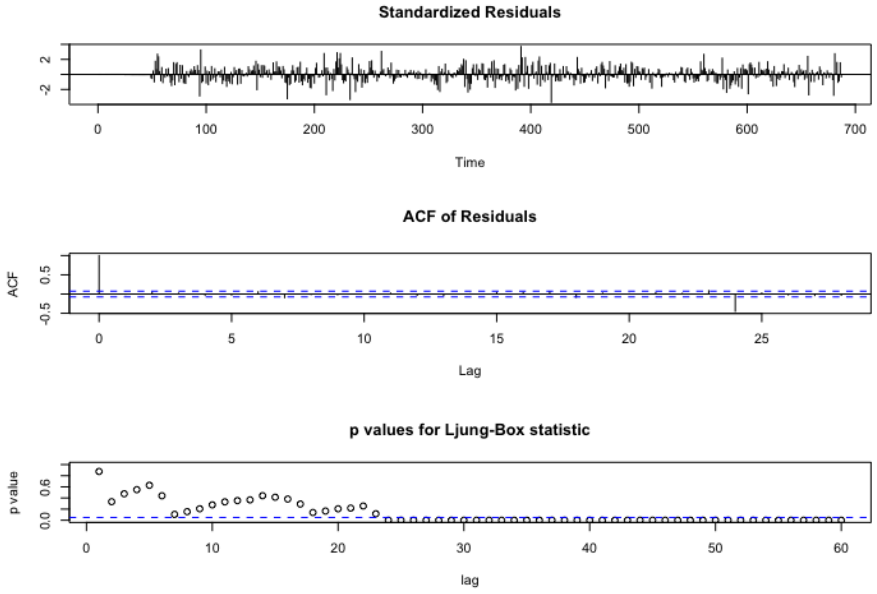

Figure 10: The diagnostics above are for Model 10. Note however that model 11 is remarkably similar. The ACF shows that there is a spike in lag 24, which tells us that we haven't gotten rid of some of that seasonality within the residuals. Further, the p-value at lag 24 in the Ljung-Box test is inside the bounds, which leads us to opt for model 5 instead.

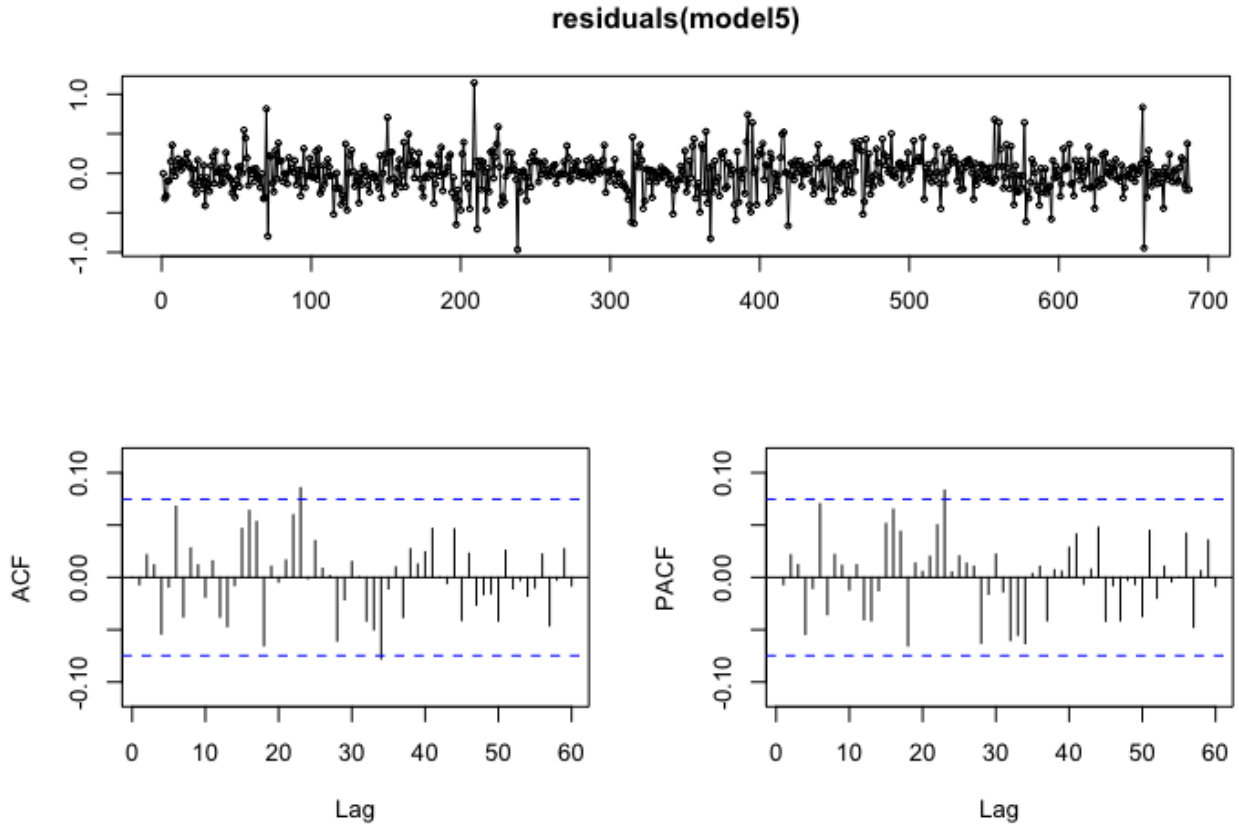

Figure 11: **OPTIMAL MODEL**: The diagnostics above are for Model 5. We plot the residuals' ACF and PACF larger for closer nspection (of our optimal model). Note that almost all the lags are within the boundaries. The only spike outside the boundary is lag 24, but it is almost inside. This tells us that the residuals are behaving like white noise.

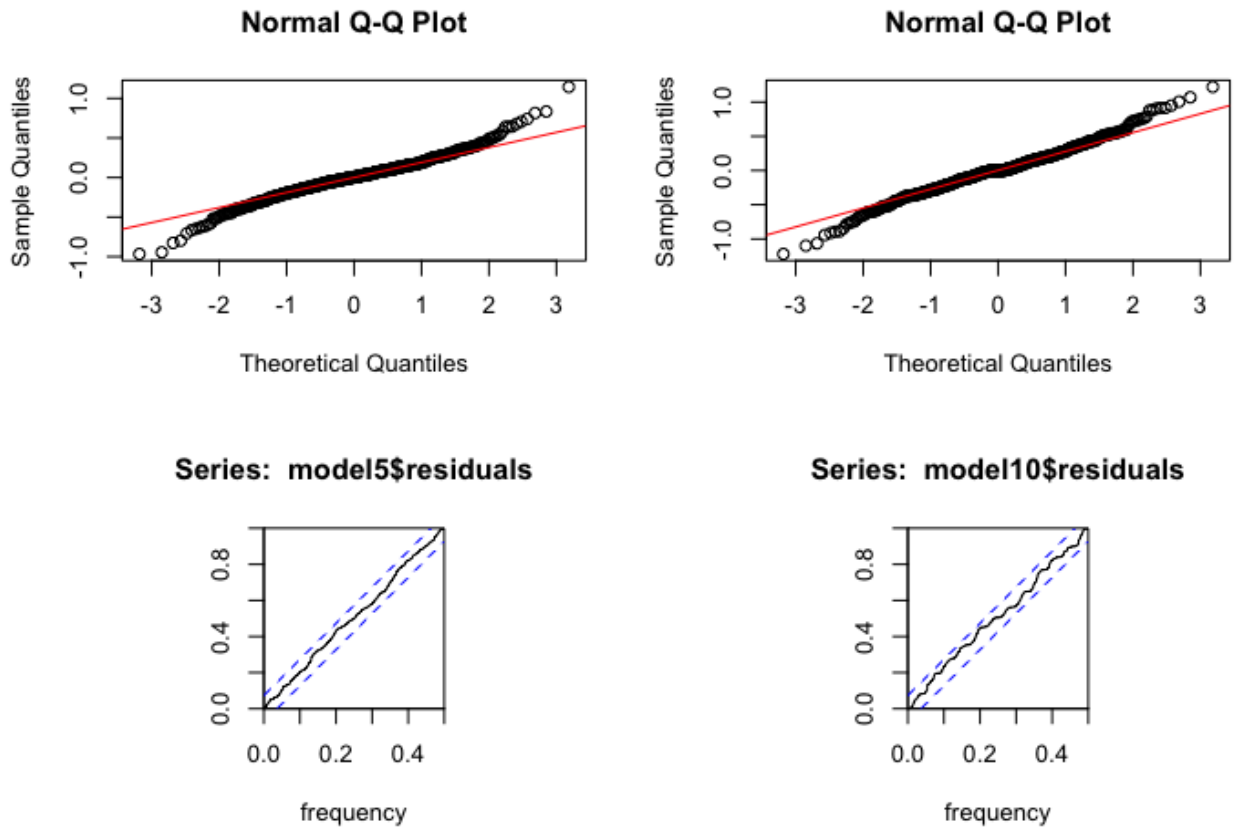

Figure 12: **QQ-Plot & CP:(Left Column corresponds to Model5; Right Column corresponds to Model 10) On the top row we can see that the residuals don't fit perfectly the theoretical quantiles of a white noise gaussian process for either model. The residuals follow a good pattern along the line in the center but over disperse from the outer theoretical quantiles. The cumuulative periodogram (CP: bottom row), however, shows us that the residuals do behave like white noise to a certain extent. As seen above, the cumulative periodogram shows a linearly increasing frequency, which is characteristic of white noise.**

3.4 Optimal Model and Parameter Estimation

Given the performed diagnostic tests, we chose Model 5 as our optimal model. Model 5 had the lowest AIC compared to its peer nested models. Further, although it had a higher RMSE than models 10 and 11, Model 5 performed well on the Ljung-Box statistic while models 10 and 11 didn't. In addition, all of its parameters are significant (shown at the very bottom), and when compared to more complex models of similar performance, the likelihood ratio test pointed us towards model 5. Finally, it also passed the signs test. We show a summary of Model 5 below:

```
> model5; p_values(model5)  # lowest AIC=-10.51 of non-differenced models, all
    signfinicant, sar almost 1, which suggests seasonal differencing

Call:
arima(x = lnxts.train, order = c(2, 0, 1), seasonal = list(order = c(1, 0, 1),
    period = 24))

Coefficients:
         ar1      ar2      ma1     sar1     sma1  intercept
      1.5223  -0.5971  -0.4775  0.9934  -0.9391     2.9147
s.e.  0.1135   0.1003   0.1275  0.0077   0.0357     0.1802

sigma^2 estimated as 0.05433:  log likelihood = 12.25,  aic = -10.51

P-VALUES FOR COEFFICIENTS:
         ar1          ar2          ma1          sar1          sma1     intercept
0.000000e+00 2.627876e-09 1.801697e-04 0.000000e+00 0.000000e+00 0.000000e+00
```

4 Predictions and Model Improvement

4.1 Predictions

Now we have chosen our optimal model (which passed a series of tests) and can visualize our forecast (Figure 13). As described in the figure, we can see that the model performs adequately. Eventhough the lower 95% confidence interval doesn't capture the down-spike in the observations, it nonetheless does capture the daily seasonality pattern exhibited by the observations. One reason why our model did not perform as well as we might have expected is that the prediction interval is between a weekday and the weekend. That is, we have to predict the behavior starting Friday afternoon and ending Sunday afternoon. As we showed in the first section in Figure 3, there seems to be a **weekly pattern** and also a **distinct behavior between weekdays and weekends**. Therefore, our model might have done better predicting intervals either in weekday or weekend frames, but the fact that we have to predict a transition period can be problematic we have not introduced additional factors for our model to account for that.

4.2 Model Improvement

Towards this end, we introduce a covariate. In the beginning we mentioned that weather forecasts might be a good explaining variable as to how traffic responds during the weekends. However, we were not able to find any weather data for Copenhagen during that time period at that time resoultion. Instead, we simply fit a dummy variable and introduced it in the **xreg** argument in the arima model fitting function. This is in hope that by introducing a dummy variable the model has extra flexibility and can therefore improve prediction performance. Unfortunately, we were not able to improve performance in the model. This might have been do to specifying xreg incorrectly or lack of a better design matrix. Nonetheless, it is fruitful to note that this model could be improved by adding covariates that can explain the weekly patter/behavior and disentagle the dichotomy between weekdays and weekends.

Time	Prediction	Lower	Upper
1hr	25.79307	16.3253	40.7516
24hr	23.68745	8.186426	68.5397
48hr	23.64792	8.161493	68.51982

Table 2: Predictions of Optimal Model 5 : $(2,0,1)x(1,0,1)_{24}$ with 95% CI

5 Code

```
# TSA Assignment 3 #

#------------------ Script Setup & Data ------------------------------#

rm(list=ls())
setwd("~/Desktop/TSA/Assignment3")
library(expm)
library(timeSeries)
library(tseries)
library(forecast)
library(lmtest)
library(PBSmodelling)
```

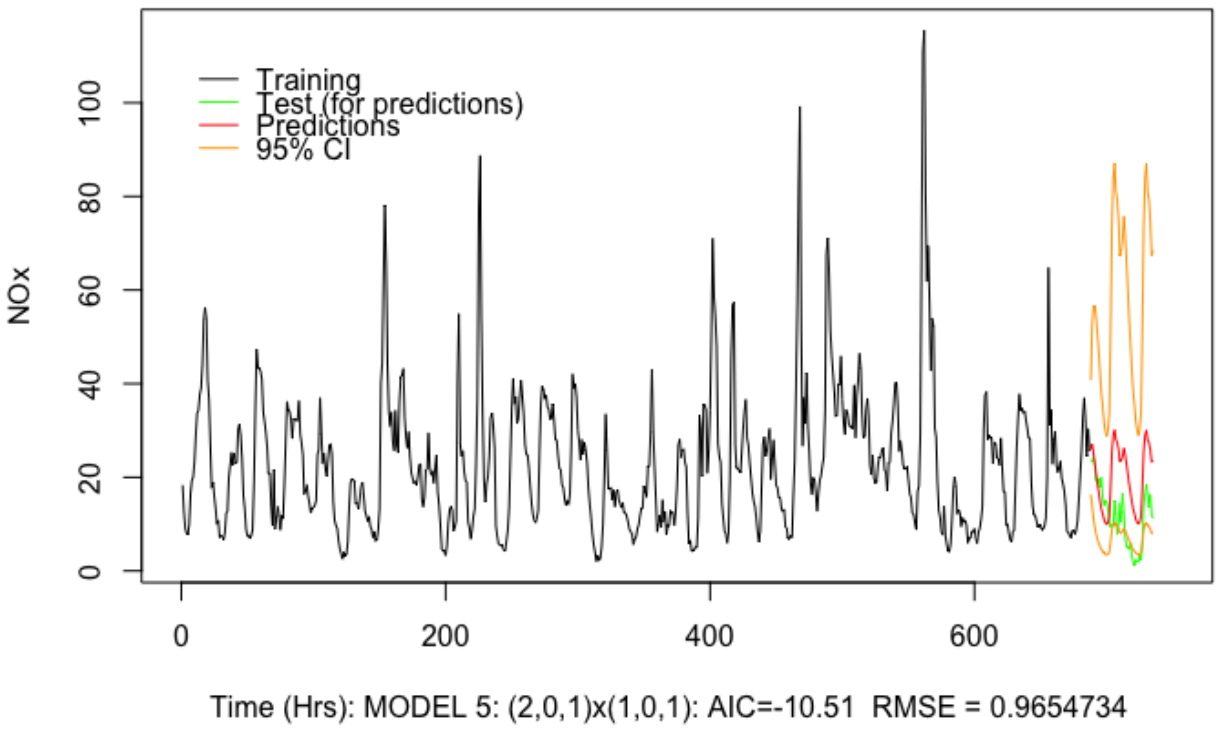

Figure 13: **OPTIMAL MODEL FORECAST**: The predictions are shown above in red compared to the observations (green). Note that our model is not able to capture the intricacies of the observations as well, however, it does capture the daily periodicity. Given that our optimal model is an AR(2) MA(1) model with seasonal components, we can see how the predictions resemble previous daily periodicities. Unfortunately, our model underestimates the down-trend in the observations and they even go outside of the lower 95% confidence interval.

```
data <- read.table("hcoe17.csv",header = T,sep = ";")
xts  <- ts(rev(data[1:735,3]))  #reversing data!  #, frequency = 24
lnxts <- log(xts) #log transformation of data to subdue increase in variance

# Subsetting Training Data (ie excluding last 48 hr observations for testing)
lnxts.train <- lnxts[1:687]


# ------------------ Visualizing the Data ----------------------------#

# Plot entire original series
plot(xts[1:687], ylab='NOx', xlab ="Time (Hrs): Starting Jan 7 2017 00:00", type = "
    l", xlim=c(0,750))
lines(688:735, xts[688:735], col = "green")
legend(0,110, c("Training", "Test (for predictions)"), col = c("black", "green"),
    lty = c(1,1), bty ="n")

# Plot entire log series
plot(lnxts[1:687], ylab='ln(NOx)', xlab ="Time (Hrs): Starting Jan 7 2017 00:00",
    type = "l", xlim=c(0,750), ylim=c(0,5))
lines(688:735, lnxts[688:735], col = "green")
legend(0,5, c("Training", "Test (for predictions)"), col = c("black", "green"), lty
    = c(1,1), bty ="n")


# Plot weekly series, starting on Saturday morning 12:01am (aka Friday after
    midnight)
plot(lnxts[1:168], type ="l", col = "black", #1st week
xaxt = "n", ylab = "ln(NOx)", xlab="", ylim = c(0,7)) #
axis(side=1, at=seq(1,169,24), labels=c("Saturday", "Sunday", "Monday", "Tuesday",
                                        "Wednesday", "Thursday", "Friday", "Saturday
                                            "),
     pos=0, lty=c(1,1,1,1,1,1,1,1), las=2)
lines(lnxts[169:336], col = "blue")          #2nd week
lines(lnxts[337:504], col = "green")         #3rd week
lines(lnxts[505:672], col = "red")           #4th week
#lines(xts[673:735], col = 'yellow')
legend(0,6, c("Week 1", "Week 2", "Week 3", "Week 4"), col = c("black", "blue", "
    green", "red"), lty = c(1,1,1,1), bty ="n")

#------------------- ACF and PACF -----------------------------#

acf(xts, lag.max = 200, plot = TRUE)
abline(v=seq(0,200,24), lty = c(2,2,2,2,2,2,2,2,2))

acf(lnxts, lag.max = 200, plot = TRUE)
abline(v=seq(0,200,24), lty = c(2,2,2,2,2,2,2,2,2))

pacf(xts,lag.max = 200, plot = TRUE)
abline(v=seq(0,200,24), lty = c(2,2,2,2,2,2,2,2,2))

pacf(lnxts,lag.max = 200, plot = TRUE)
abline(v=seq(0,200,24), lty = c(2,2,2,2,2,2,2,2,2))



#---------- USEFUL FUNCTIONS (Tests and Plots) -------------#
```

```r
# Useful Plotting Function:
diagtool <- function(residuals){
  par(mfrow=c(3,1), mar=c(3,3,1,1), mgp=c(2,0.7,0))
  plot(residuals, type="l")
  acf(residuals)
  pacf(residuals)
}


# Likelihood Ratio Test Function:
#     To test whether the more complex model is significanlty better than the
    simpler model
ratiotest <- function(fit1, fit2, df){
  pchisq(-2* ( fit1$loglik - fit2$loglik ), df=df, lower.tail = FALSE)
}


# F-test Function:
Ftest <- function(fit1, fit2){
  s1 <- sum(fit1$residuals^2)
  s2 <- sum(fit2$residuals^2)
  n1 <- 3
  n2 <- 4

  pf( (s1-s2)/(n2-n1) / (s2/(length(fit1$residuals)-n2)), df1 = n2 - n1, df2 = (
      length(fit1$residuals)-n2), lower.tail = FALSE)
}

# Signs test using Binomial Model:
signs <- function(analyse0){
  # sign test, mean, and sd
  res <- analyse0$residuals
  n.res <- length(res)
  (n.res-1)/2 # mean:
  sqrt((n.res-1)/4)  ### sd:
  (n.res-1)/2 + 1.96 * sqrt((n.res-1)/4) * c(-1,1) ### 95% interval:
  ### test:
  (N.sign.changes <- sum( res[-1] * res[-n.res]<0 ))
  binom.test(N.sign.changes, n.res-1)
}

# p-values of coefficients function
# gives the p-values of the hypothesis test: H0 (Null Hypothesis) coeff = 0 -vs- H1
    coeff != 0
p_values <- function(aa){
  (1-pnorm(abs(aa$coef)/sqrt(diag(aa$var.coef))))*2
}



#___________INITIAL MODEL EXPLORATION: SEEKING STATIONARITY_____________#

tsdisplay(xts, lag.max = 170,
          main="Original Time Series Data - No Transform - No ARIMA applied", xlab="
              Time (hrs)")

tsdisplay(lnxts, lag.max = 200,
          main="Log Transformation - No ARIMA components applied", xlab="Time (hrs)"
              )
```

```r
tsdisplay(diff(lnxts,24), lag.max = 200,
          main="Seasonally differenced: (0,0,0)x(0,1,0)_24", xlab="Time(hrs)")

tsdisplay(diff(diff(lnxts,24)), lag.max = 200,
          main="Non-seasonal & Seasonally differenced: (0,1,0)x(0,1,0)_24", xlab="
              Time (hrs)")


# COMMENTS: The 2 differenced ACF and PACF suggest adding an ARIMA(0,0,0)(0,0,1)24
    component
# on top of the differences; resulting in  an ARIMA(0,1,0)(0,1,1)24.


#__________ MODEL EXPLORATION _____________#
#       (Trying different setups)

#.......Non-Differenced Models...........#     AIC WINNERS: Models 4,5 and 14s
model0 <- arima(x = lnxts.train, order = c(3, 0, 0))
model0; p_values(model0)  # AIC=80.38, all coeff. significant

model1 <- arima(x = lnxts.train, order = c(3, 0, 0), seasonal = list(order = c(1, 0,
     0), period = 24))
model1; p_values(model1)   # AIC=62, all coeff. significant

model2 <- arima(x = lnxts.train, order = c(2, 0, 0), seasonal = list(order = c(1, 0,
     0), period = 24))
model2; p_values(model2)  # AIC=76.86, all coeff. significant

model3 <- arima(x = lnxts.train, order = c(3, 0, 1), seasonal = list(order = c(1, 0,
     0), period = 24))
model3; p_values(model3)  # AIC=61.79, 3 coeff. not significant (remove?)

model4 <- arima(x = lnxts.train, order = c(3, 0, 1), seasonal = list(order = c(1, 0,
     1), period = 24))
model4; p_values(model4)  # AIC=-9.15, 3 coeff. not significant (remove?), sar
    almost 1

model5 <- arima(x = lnxts.train, order = c(2, 0, 1), seasonal = list(order = c(1, 0,
     1), period = 24))
model5; p_values(model5)  # lowest AIC=-10.51 of non-differenced models, all
    signfinicant, sar almost 1, which suggests seasonal differencing

model14 <- arima(x=lnxts.train, order=c(3,0,0), seasonal=list(order=c(1,0,2),period
    =24))
model14; p_values(model14) # lowest AIC=-8.18, ar2 insignificant

model14s <- arima(x=lnxts.train, order=c(3,0,0), seasonal=list(order=c(1,0,2),period
    =24),transform.pars = FALSE, fixed= c(NA,0,NA,NA,NA,0,NA))
model14s;                  # lowest AIC=-10.74,

# Model Suggested by auto.arima: (2,0,1)x(0,0,0)
auto_model <- auto.arima(lnxts.train, stepwise=FALSE, approximation=FALSE)
auto_model; p_values(auto_model)   # AIC=78.22, all coeff. significant


#.......Seasonal-Differenced Models...........#     AIC WINNERS: Models 6, 6s and sD

model6 <- arima(x = lnxts.train, order = c(3, 0, 0), seasonal = list(order = c(0, 1,
     1), period = 24))
model6; p_values(model6);  # aic = 15.75, ar2 coeff. insig
```

```
model6s <- arima(x=lnxts.train, order=c(3,0,0), seasonal=list(order=c(0,1,1),period
    =24),transform.pars = FALSE, fixed= c(NA,0,NA,NA))
model6s;                    # aic = 14.69   # REMOVING INSIGNIFICANT AR2 coefficient

model7 <- arima(x = lnxts.train, order = c(2, 0, 0), seasonal = list(order = c(0, 1,
     1), period = 24))
model7; p_values(model7)  # aic = 19.42, all sig,

model8 <- arima(x = lnxts.train, order = c(2, 0, 1), seasonal = list(order = c(1, 1,
     1), period = 24))
model8; p_values(model8)  # aic = 17.81, 1 coeff. insig,

model10 <- arima(x = lnxts.train, order = c(2, 0, 0), seasonal = list(order = c(0,
    2, 1), period = 24))
model0; p_values(model10) # aic = 80.38, all sig

model11 <- arima(x = lnxts.train, order = c(2, 0, 1), seasonal = list(order = c(0,
    2, 1), period = 24))
model11; p_values(model11) # aic = 482.96, all sig

# Model Suggested by auto.arima: (2,0,4)x(0,1,0)
auto_model_sD <- auto.arima(lnxts.train, lambda=0, d=0, D=1, max.order=9, stepwise=
    FALSE, approximation=FALSE)
auto_model_sD; p_values(auto_model_sD)   # AIC= -1160.05!!! , 1 coeff. insignificant
     (ma3)


#.......Seasonal-Differenced + Non-Seasonal Differenced Models..........#     AIC
    WINNERS: Models 13 and sDd

model9 <- arima(x = lnxts.train, order = c(2, 1, 1), seasonal = list(order = c(1, 1,
     1), period = 24))
model9; p_values(model9)  # aic = 67.39, ar2 coeff insignificant

model13 <- arima(x = lnxts.train, order = c(0, 1, 1), seasonal = list(order = c(0,
    1, 1), period = 24))
model13; p_values(model13) # aic = 61.95, all sig

model12 <- arima(x = lnxts.train, order = c(0, 1, 0), seasonal = list(order = c(0,
    1, 1), period = 24))
model12; p_values(model12) # aic = 64.99, all sig

# Model Suggested by auto.arima: (5,1,0)x(0,1,0)
auto_model_sDd <- auto.arima(lnxts.train, lambda=0, d=1, D=1, max.order=9, stepwise=
    FALSE, approximation=FALSE)
auto_model_sDd; p_values(auto_model_sDd)   # AIC=-1096.57 !!! , 1 coeff.
    insignificant (ar3)



#_________________________ RESIDUAL ANALYSIS ___________________________#
#          (We filter out models that dont pass this test)
#    (Then we compute further detailed tests in the next section)

tsdisplay(residuals(model0))
tsdisplay(residuals(model1))
tsdisplay(residuals(model2))
tsdisplay(residuals(model3))
tsdisplay(residuals(model4), lag.max = 60) # Good! Looks almost like white noise
```

```
    tsdiag(model4, gof.lag = 60)
tsdisplay(residuals(model5), lag.max = 60) # Looks the same as above
    tsdiag(model5, gof.lag = 60)
tsdisplay(residuals(model6), lag.max = 60) # Great! Nothing outside the bounds
    tsdiag(model6, gof.lag = 60)
tsdisplay(residuals(model7), lag.max = 60)
tsdisplay(residuals(model8), lag.max = 60)
tsdisplay(residuals(model9))
tsdisplay(residuals(model10), lag.max = 60) # 24 lag out of bounds,
    tsdiag(model10, gof.lag = 60)
tsdisplay(residuals(model11), lag.max = 60)
    tsdiag(model11, gof.lag = 60)
tsdisplay(residuals(model12))
tsdisplay(residuals(auto_model))
tsdisplay(residuals(auto_model_sD))
tsdisplay(residuals(auto_model_sDd))
    tsdiag(auto_model_sDd, gof.lag = 60)


#_______________ Model Diagnostics & Tests ____________________#
#      (on models that passed initial inspection)


# QQ Plot
qqnorm(model4$residuals)
qqline(model4$residuals,col=2)

qqnorm(model5$residuals)
qqline(model5$residuals,col=2)

qqnorm(model6$residuals)
qqline(model6$residuals,col=2)

qqnorm(model10$residuals)
qqline(model10$residuals,col=2)

qqnorm(model11$residuals)
qqline(model11$residuals,col=2)

# Signs (+ -) Autocorrelation Test
signs(model4)
signs(model5)
signs(model6)
signs(model10)
signs(model11)

# Cumulative Periodogram  PLot
#     - do residuals look like white noise?
cpgram(lnxts.train)
cpgram(model4$residuals) # yes!
cpgram(model5$residuals) # yes!
cpgram(model6$residuals) # yes!
cpgram(model10$residuals) # yes!
cpgram(model11$residuals) # yes!

par(mfrow=c(2,2))
qqnorm(model5$residuals)
qqline(model5$residuals,col=2)

qqnorm(model10$residuals)
```

```r
qqline(model10$residuals,col=2)

cpgram(model5$residuals) # yes!
cpgram(model10$residuals) # yes!
dev.off()

# Ratio Likelihood Tests: Is the more complex models significantly better? A: Nope!
ratiotest(model5, model4, 1)    #not sig. p =  0.4206363
ratiotest(model10, model11, 1)  #not sig. p =  0.07512814

#_____________________RMSE: COMPARING ALL MODELS_____________________#

getrmse <- function(x,h,fit)
{
  train.end <- time(x)[length(x)-h]
  test.start <- time(x)[length(x)-h+1]
  train <- window(x,end=train.end)
  test <- window(x,start=test.start)
  fc <- forecast(fit,h=h)
  return(accuracy(fc,test)[2,"RMSE"]) #the smaller the RMSE, the better prediction
      performance
}


#..... Non-differenced Models.........#
getrmse(lnxts,h=48, model1)   #model1: 1.026942
getrmse(lnxts,h=48, model2)   #model2: 1.028274
getrmse(lnxts,h=48, model3)   #model3: 1.02536
getrmse(lnxts,h=48, model4)   #model4: 0.9664464  Great!
getrmse(lnxts,h=48, model5)   #model5: 0.9654734  Great as well!   BEST!
getrmse(lnxts,h=48, auto_model)  #model_auto: 1.031515
getrmse(lnxts,h=48, model14)  #model_sigs:  0.9674714
getrmse(lnxts,h=48, model14s) # 0.9674714

#.....Seasonal Differenced Models.......#
getrmse(lnxts,h=48, model6)   #model6: 0.9678455
getrmse(lnxts,h=48, model6s)  #model6s 0.9679578
getrmse(lnxts,h=48, model7)   #model7: 0.967771
getrmse(lnxts,h=48, model8)   #model8: 0.9673714
getrmse(lnxts,h=48, model10)  #model10: 0.8925458                  BEST!
getrmse(lnxts,h=48, model11)  #model11: 0.888723                   BESTEST EVER!
getrmse(lnxts,h=48, auto_model_sD)  #model_sD: 0.9739474

#.....Seasonal & Non-seasonal Differenced Models.......#
getrmse(lnxts,h=48, model9)   #model9: 0.9735253
getrmse(lnxts,h=48, model13)  #model13: 0.9710542
getrmse(lnxts,h=48, model12)  #model12: 0.9866931
getrmse(lnxts,h=48, auto_model_sDd)  #model_sDd: 1.314824  worst!

#_____________________________________________________________***  RMSE WINNERS:
    Models 5, 10 and 11 ***_____


#_____________Forecasts_____________#
pred_model4 <-forecast.Arima(model4, h=48, level = 95)
pred_model11 <- forecast.Arima(model11, h=48, level = 95)
pred_model6 <-forecast.Arima(model6, h=48, level = 95)
pred_model14 <- forecast.Arima(model14, h=48, level = 95)
pred_model14s <-forecast.Arima(model14s, h=48, level = 95)
pred_model5 <- forecast.Arima(model5, h=48, level = 95)
```

```
pred_model10 <-forecast.Arima(model10, h=48, level = 95)
pred_model11 <- forecast.Arima(model11, h=48, level = 95)

plot(pred_model4, ylab="NOx", xlab="Time(hrs) Model 4")       #model4
lines(688:735, lnxts[688:735], col = "green")

plot(forecast(model6, h = 48), ylab="NOx", xlab="Time(hrs) Model 6")  #model6
lines(688:735, lnxts[688:735], col = "green")

plot(forecast(model14), ylab="NOx", xlab="Time(hrs) Model 14") #model14
lines(688:735, lnxts[688:735], col = "green")

plot(forecast(model14s), ylab="NOx", xlab="Time(hrs) Model 14s") #model14s
lines(688:735, lnxts[688:735], col = "green")

plot(forecast(model5), ylab="NOx", xlab="Time(hrs) Model 5")  #model5
lines(688:735, lnxts[688:735], col = "green")

plot(forecast(model10), ylab="NOx", xlab="Time(hrs) Model 10") #model10    #BEST!
lines(688:735, lnxts[688:735], col = "green")

plot(forecast(model11), ylab="NOx", xlab="Time(hrs) Model 11") #model11    #AND BEST
    !
lines(688:735, lnxts[688:735], col = "green")


# Transform forecasts back to original scale
mean4 = exp(pred_model4$mean)
lower4 = exp(pred_model4$lower)
upper4 = exp(pred_model4$upper)

mean5 = exp(pred_model5$mean)
lower5 = exp(pred_model5$lower)
upper5 = exp(pred_model5$upper)

mean6 = exp(pred_model6$mean)
lower6 = exp(pred_model6$lower)
upper6 = exp(pred_model6$upper)

mean10 = exp(pred_model10$mean)
lower10 = exp(pred_model10$lower)
upper10 = exp(pred_model10$upper)

mean11 = exp(pred_model11$mean)
lower11 = exp(pred_model11$lower)
upper11 = exp(pred_model11$upper)

mean14s = exp(pred_model14s$mean)
lower14s = exp(pred_model14s$lower)
upper14s = exp(pred_model14s$upper)


# PLOT IN ORIGINAL SCALE

# Plot in original scale: MOdel 4
plot(xts[1:687], ylab='NOx', xlab ="Time (Hrs): MODEL 4: (3,0,1)x(1,0,1): AIC=-9.15
     RMSE = 0.9664464 ", type = "l", xlim=c(0,750))
lines(688:735, xts[688:735], col = "green")
lines(688:735, mean4, col ='red')
lines(688:735, lower4, col ='orange')
```

```
lines(688:735, upper4, col ='orange')
legend(0,110, c("Training", "Test (for predictions)", "Predictions", "95% CI"), col
    = c("black", "green", "red", "orange"), lty = c(1,1,1,1), bty ="n")


# Plot in original scale: MOdel 5
plot(xts[1:687], ylab='NOx', xlab ="Time (Hrs): MODEL 5: (2,0,1)x(1,0,1): AIC=-10.51
      RMSE = 0.9654734  ", type = "l", xlim=c(0,750))
lines(688:735, xts[688:735], col = "green")
lines(688:735, mean5, col ='red')
lines(688:735, lower5, col ='orange')
lines(688:735, upper5, col ='orange')
legend(0,110, c("Training", "Test (for predictions)", "Predictions", "95% CI"), col
    = c("black", "green", "red", "orange"), lty = c(1,1,1,1), bty ="n")

# Plot in original scale: MOdel 6
plot(xts[1:687], ylab='NOx', xlab ="Time (Hrs): MODEL 6: (3,0,1)x(0,1,1): AIC=14.69
     RMSE = 0.9678455 ", type = "l", xlim=c(0,750))
lines(688:735, xts[688:735], col = "green")
lines(688:735, mean6, col ='red')
lines(688:735, lower6, col ='orange')
lines(688:735, upper6, col ='orange')
legend(0,110, c("Training", "Test (for predictions)", "Predictions", "95% CI"), col
    = c("black", "green", "red", "orange"), lty = c(1,1,1,1), bty ="n")

# Plot in original scale: Model 10
plot(xts[1:687], ylab='NOx', xlab ="Time (Hrs): MODEL 10: (2,0,0)x(0,2,1): AIC=80.38
     RMSE = 0.8925458  ", type = "l", xlim=c(0,750))
lines(688:735, xts[688:735], col = "green")
lines(688:735, mean10, col ='red')
lines(688:735, lower10, col ='orange')
lines(688:735, upper10, col ='orange')
legend(0,110, c("Training", "Test (for predictions)", "Predictions", "95% CI"), col
    = c("black", "green", "red", "orange"), lty = c(1,1,1,1), bty ="n")


# Plot in original scale: Model 11
plot(xts[1:687], ylab='NOx', xlab ="Time (Hrs): MODEL 11: (2,0,1)x(0,2,1): AIC
    =482.96 RMSE = 0.888723   ", type = "l", xlim=c(0,750))
lines(688:735, xts[688:735], col = "green")
lines(688:735, mean11, col ='red')
lines(688:735, lower11, col ='orange')
lines(688:735, upper11, col ='orange')
legend(0,110, c("Training", "Test (for predictions)", "Predictions", "95% CI"), col
    = c("black", "green", "red", "orange"), lty = c(1,1,1,1), bty ="n")


# Plot in original scale: MOdel 14s
plot(xts[1:687], ylab='NOx', xlab ="Time (Hrs): MODEL 14s: (3,0,0)x(1,0,2): AIC
    =-8.18 RMSE = 0.9674714  ", type = "l", xlim=c(0,750))
lines(688:735, xts[688:735], col = "green")
lines(688:735, mean14s, col ='red')
lines(688:735, lower14s, col ='orange')
lines(688:735, upper14s, col ='orange')
legend(0,110, c("Training", "Test (for predictions)", "Predictions", "95% CI"), col
    = c("black", "green", "red", "orange"), lty = c(1,1,1,1), bty ="n")

# POINT PREDICTIONS (1,24,48 hr)

mean5[1]; mean5[24]; mean5[48]
```

```
lower5[1]; lower5[24]; lower5[48]
upper5[1]; upper5[24]; upper5[48];


# MODEL IMPROVEMENT

xreg <- rnorm(687,0, 1)  #dummy variable
model5xreg <- arima(x = lnxts.train, order = c(2, 0, 1), seasonal = list(order = c
    (1, 0, 1), period = 24), xreg = xreg)
model5xreg; p_values(model5)   # aic = -8.7
SSRxreg <- sum(sqrt(abs(model5xreg$residuals)))   # 254.7958
SSR <- sum(sqrt(abs(model5$residuals)))   #254.3872
```