IEE 579

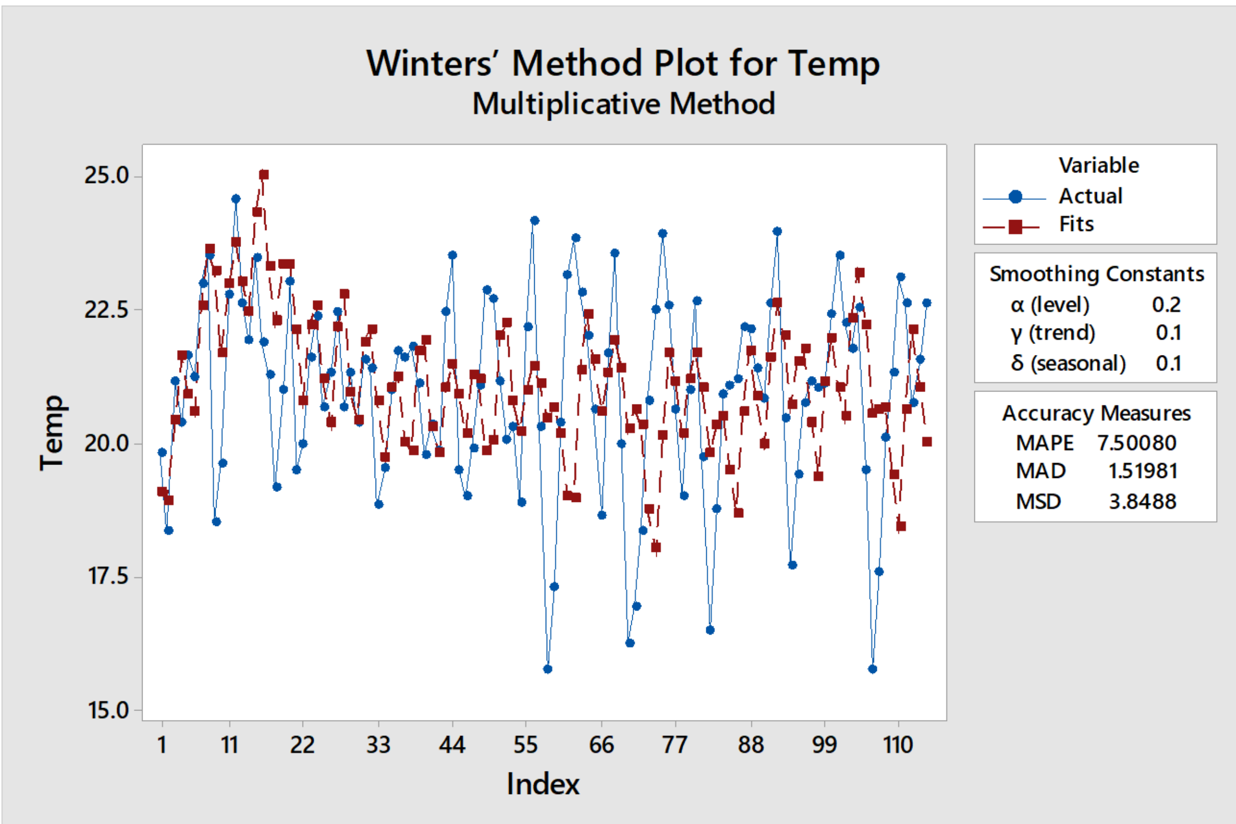
Time Series Analysis

Kushal Kapadia

1213714799

Case Study – 2

1] The output from Minitab for the Holt-winter forecasting method is below:



Output:

Winters' Method for Temp

Method

Model type	Multiplicative Method
Data	Temp
Length	114

Smoothing Constants

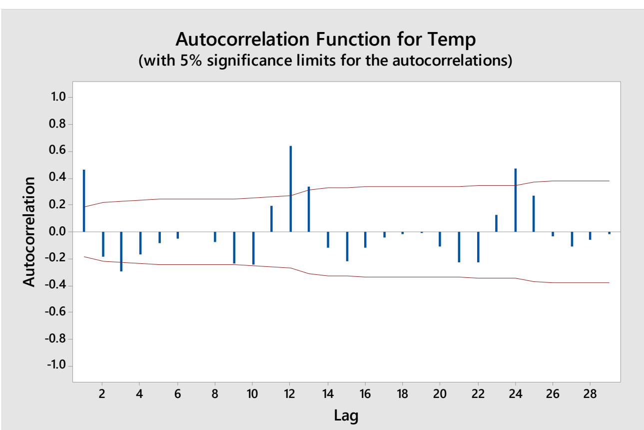
α (level)	0.2
γ (trend)	0.1
δ (seasonal)	0.1

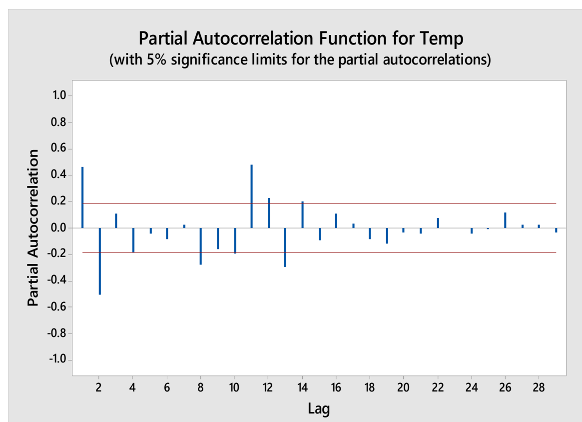
Accuracy Measures

MAPE	7.50080
MAD	1.51981
MSD	3.84880

Here, I have used Holt-Winter Multiplicative method for forecasting. After trying out a couple combinations, I'm using the weights for the level, trend and seasonal as 0.2, 0.1 and 0.1 respectively as also shown in the output. Also, the seasonal length is 4. I have used these specific weights because I tried a bunch of combinations with multiplicative as well as additive model and this one gave me the lowest performance statistics MAPE, MAD and MSD.

2]





Here, I'm trying to see how the Sample Autocorrelation and partial autocorrelation look like. From both the plots, it is very clear that they show us that we should probably use AR(2) model here because the SACF is an exponential decay and the PACF cuts after 2 lags.

Also, one more thing to note is that if we look just at the ACF, the spikes at lags 1,12,24. This gives us an idea that that this model has S=12 i.e. the time span of repeating seasonal pattern. As, from the time series plot, it's very clear that it has almost constant mean, I'm not gonna use seasonal and non-seasonal differences. Thus, D=d=0.

I have tried working on a bunch of different models including AR and MA components together, excluding them with different values but what worked the best is below:

ARIMA(2,0,0) x $(1,0,0)^{12}$

The output from R:

```
Call:
arima(x = temperature, order = c(2, 0, 0), seasonal = list(order = c(1, 0, 0)
,
    period = 12))

Coefficients:
         ar1      ar2    sar1  intercept
      0.8658  -0.5401  0.8227    21.0587
s.e.  0.0822   0.0827  0.0509     0.4784

sigma^2 estimated as 0.734:  log likelihood = -151.45,  aic = 312.91
```

From the output, now we have a complete clarity that the MSE of seasonal ARIMA model is even lower than the Holt-Winter Multiplicative model that we used in the first question. Thus, seasonal ARIMA model works better than Holt-Winter model in our case.

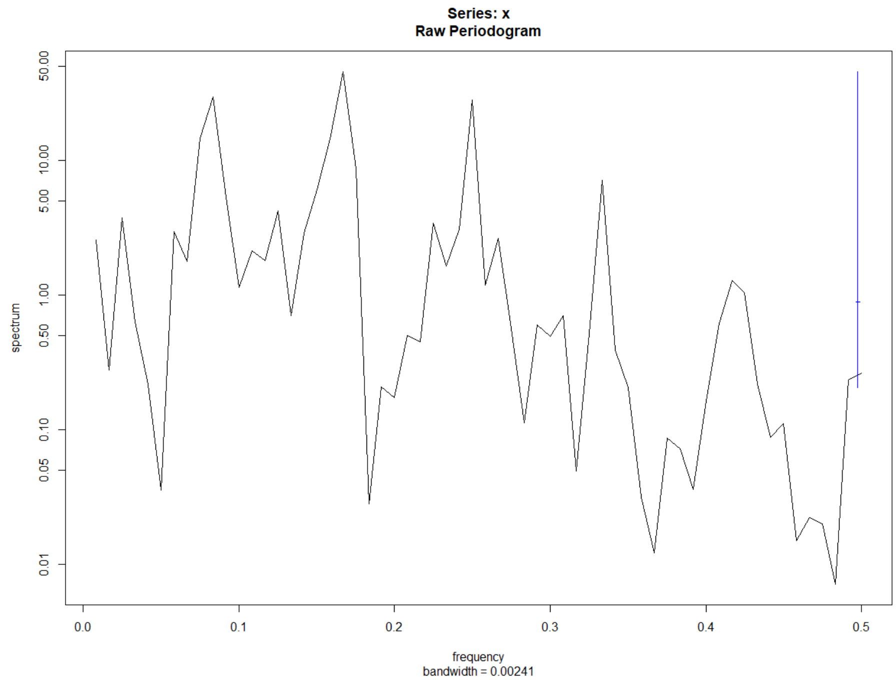
3] Here, we're interested in doing the spectral analysis in order to find the significant frequency:

I'm attaching the code from R here:

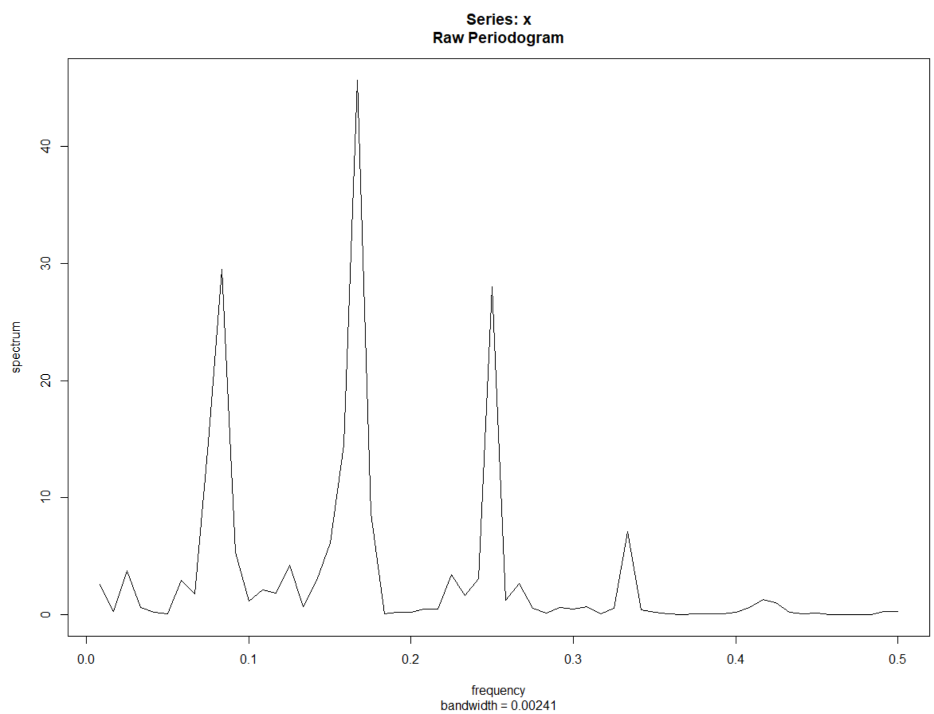
```
tem_train = read.csv("D:\\ASU Stuff\\SEM-1,2,3\\IEE 579\\train(1).csv")

install.packages("spectral")

ss.log = spectrum(tem_train$Temp) #contains log values on the y-axis

ss = spectrum(tem_train$Temp, log = 'no')
```

Output:

The output when log values were considered on the y-axis:



The output when log values were **NOT** considered on the y-axis:



R-code with the output:

```
> max(ss$spec)
[1] 45.67795
```

```
> ss$freq[which.max(ss$spec)]
[1] 0.1666667
1/ss$freq[which.max(ss$spec)]
[1] 6
```

Here, from the raw Periodogram with no log values on y-axis, we can see that there are indeed 4 significant frequencies. But the most significant frequency from the code and the output occurs at frequency 0.167 and spectrum value 45.678. Also, the pattern shows a 6-month cycle.

4] Here, the building of transfer function model starts by first building up model for temperature. I'm using the model in part 2 i.e. ARIMA(2,0,0) x $(1,0,0)^{12}$ .The code and output from R:

Code:

```
require(forecast)
data1 = read.csv("D:\\ASU Stuff\\SEM-1,2,3\\IEE 579\\train(1).csv")
temperature = data1$Temp
electricity = data1$Elec

par(mfrow=c(2,1), oma=c(0,0,0,0))
plot(temperature, type="o", pch=16, cex=.5, xlab="time", ylab="Temperature")
plot(electricity, type="o", pch=16, cex=.5, xlab="time", ylab="Electricity Consumption")

ccf(electricity, temperature, main="CCF of Temperature and Electricity", ylab="CCF")
abline(v=0, col="blue")

par(mfrow=c(1,2), oma=c(0,0,0,0))
acf(temperature, lag.max=25, type="correlation", main="ACF for Temperature")
acf(temperature, lag.max=25, type="partial", main="PACF for Temperature", ylab="PACF")

temperature.arima = arima(temperature,order=c(2,0,0),seasonal=list(order=c(1,0,0),period=12))
temperature.arima

res.temperature = as.vector(residuals(temperature.arima))

library(forecast)
fit.temperature = as.vector(fitted(temperature.arima)) #fitted values for temperature
```

Output:

```
Call:
arima(x = temperature, order = c(2, 0, 0), seasonal = list(order = c(1, 0, 0)
,
    period = 12))

Coefficients:
         ar1      ar2    sar1  intercept
      0.8658  -0.5401  0.8227    21.0587
s.e.  0.0822   0.0827  0.0509     0.4784
```

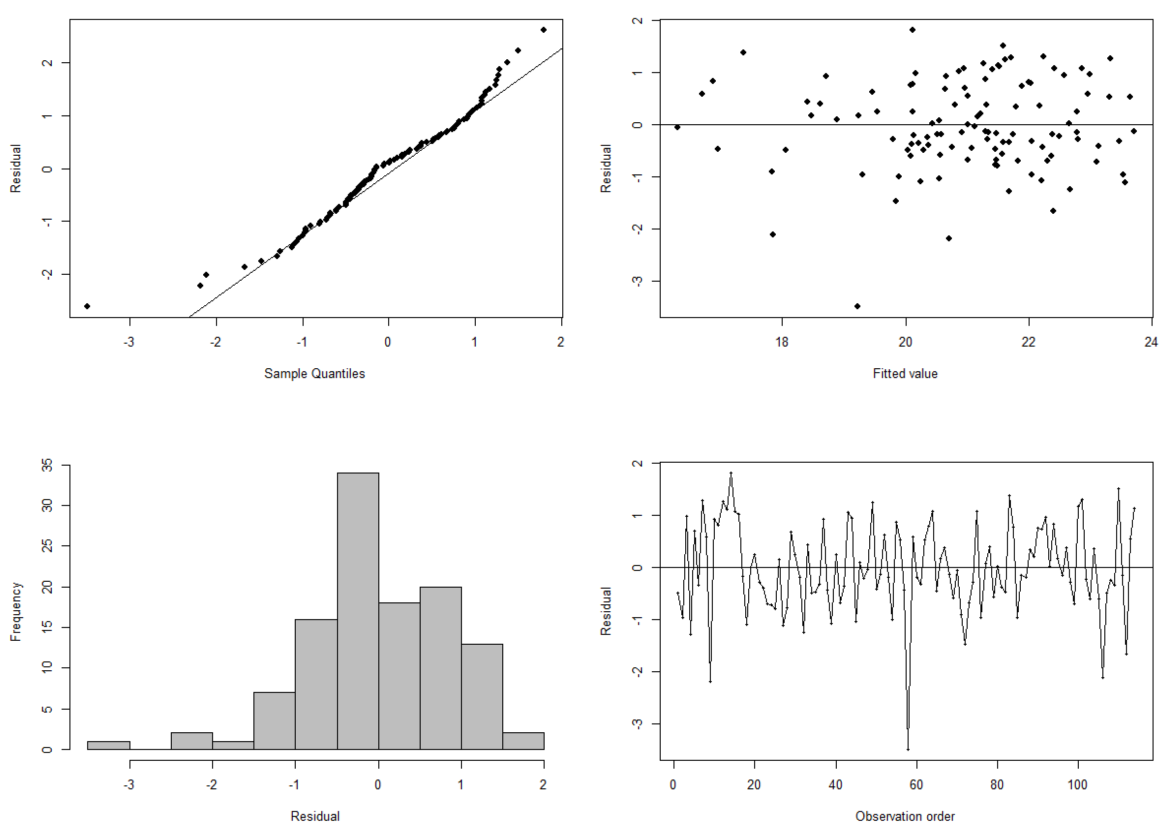
```
sigma^2 estimated as 0.734:  log likelihood = -151.45,  aic = 312.91
```

So, from this model, we can easily get the coefficients in order to use it to find α_t and β_t by just substituting in their respective equations. But we'll do the model adequacy checks:

Code:

```
par(mfrow=c(2,2), oma=c(0,0,0,0))
qqnorm(res.temperature, datax=TRUE, pch=16, xlab="Residual", main="")
qqline(res.temperature, datax=TRUE)
plot(fit.temperature, res.temperature, pch=16, xlab="Fitted value", ylab="Residual")
abline(h=0)
hist(res.temperature, col="gray", xlab="Residual", main="")
plot(res.temperature, type="l", xlab="Observation order", ylab="Residual")
points(res.temperature, pch=16, cex=0.5)
abline(h=0)
```

Output:



Code:

```
T<-length(temperature)
alphat<- 21.0587+temperature[3:T]-1.6885*temperature[2:(T-1)]-1.2524*temperature[1:(T-2)]+0.4443*temperature[1:(T-2)]
```

```
betat<- 21.0587+temperature[3:T]-1.6885*temperature[2:(T-1)]-1.2524*temperature[1:(T-2)]+0.4443*temperature[1:(T-2)]
par(mfrow=c(1,1))
ralbe<-ccf(betat, alphat, main="CCF of alpha(t) and beta(t)", ylab="CCF")
abline(v=0, col="blue")

vhat<-sqrt(var(betat)/var(alphat))*ralbe$acf
nl<-length(vhat)
plot(seq(-(nl-1)/2, (nl-1)/2, 1), vhat, type="h", xlab="Lag", ylab=expression(italic(hat(v)[italic(j)])))
abline(v=0, col="blue")
abline(h=0)
v1<-vhat[19]
v2<-vhat[20]
v3<-vhat[21]
```

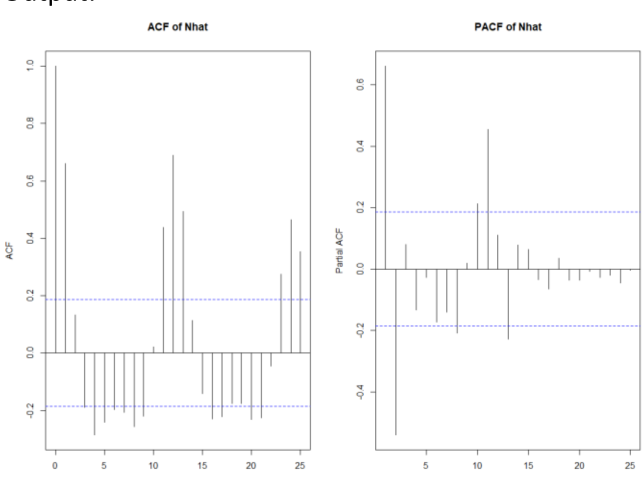
The output for the 6 months after June 2006 for temperature from Minitab is below:

Interpretation: This code helps us to find the α_t and β_t and also which in turns helps us to calculate to initial estimates of $\hat{v}_j$.

Now, we're in a position to model the noise. The code:

```
Nhat<-array(0, dim=c(1,T))
for (i in 4:T){
  Nhat[i]<-electricity[i]-(v3*temperature[i-3]+v2*temperature[i-2]+v1*temperature[i-1])
}
Nhat<-Nhat[4:T]
plot(Nhat, type="o", pch=16, cex=.5, xlab="Time", ylab=expression(italic(hat(N)[italic(t)])))
par(mfrow=c(1,2), oma=c(0,0,0,0))
acf(Nhat, lag.max=25, type="correlation", main="ACF of Nhat")
acf(Nhat, lag.max=25, type="partial", main="PACF of Nhat")
```

Output:



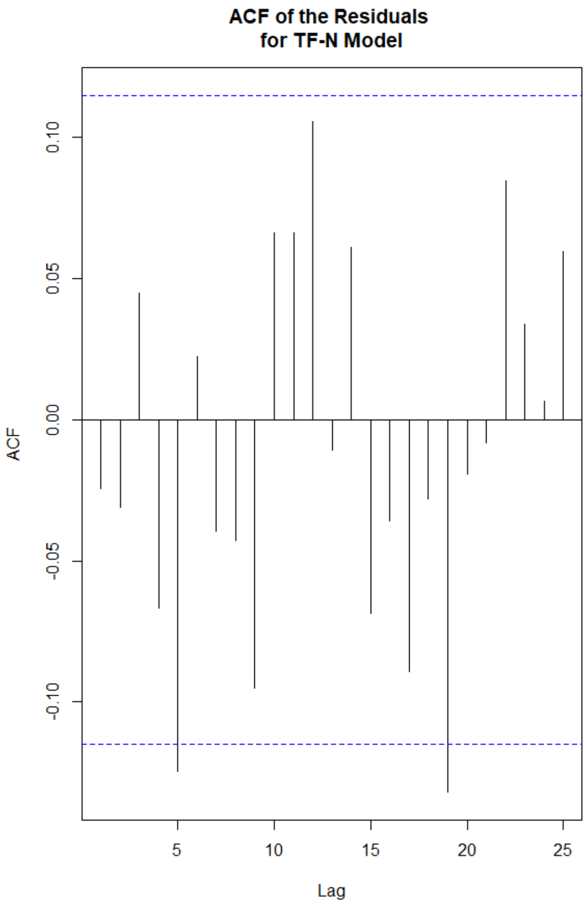
Now, that we have even the noise model, we can create the transfer-function model with the following code:

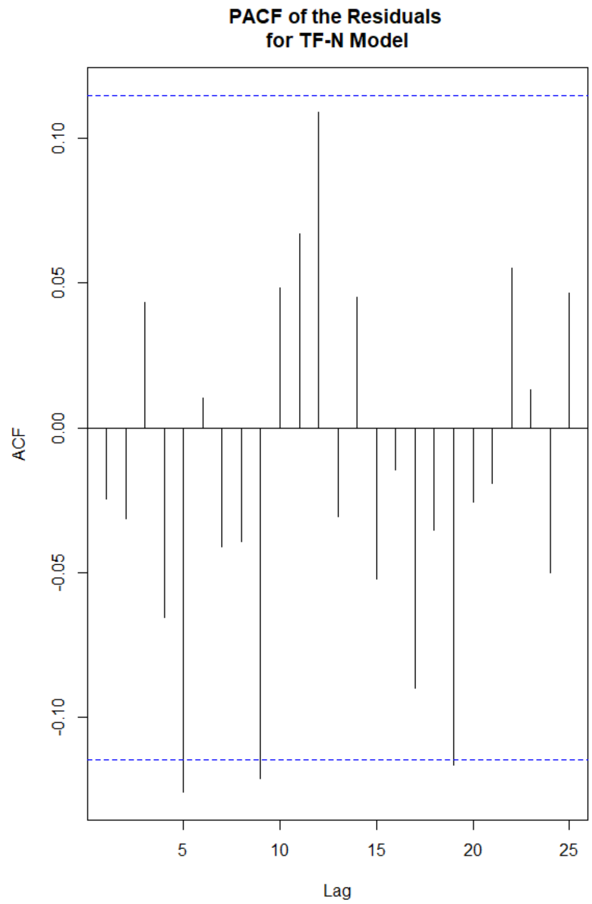
```
library(TSA)
ts.temperature<-ts(temperature)
lag3.x<-lag(ts.temperature, -3)
ts.electricity<-ts(electricity)
dat3<-cbind(ts.temperature, lag3.x, ts.electricity)
dimnames(dat3)[[2]]<-c("temperature","lag3x", "electricity")
data2<-na.omit(as.data.frame(dat3))
elec.tf1<-arimax(data2$electricity, order=c(2,0,0), xtransf = data.frame(data2$lag3x), transfer =
list(c(1,2)), include.mean = FALSE)
elec.tf1

res.elec.tf<-na.omit(as.vector(residuals(elec.tf1)))
par(mfrow=c(1,2), oma=c(0,0,0,0))
acf(res.elec.tf, lag.max = 25, type="correlation", main="ACF of the Residuals \nfor TF-N Model")
acf(res.elec.tf, lag.max = 25, type="partial", main="PACF of the Residuals \nfor TF-N Model")

par(mfrow=c(1,1))
T<-length(res.elec.tf)
Ta<-length(alphat)
ccf(res.elec.tf, alphat[(Ta-T+1):Ta], main="CCF of alpha(t) and \n Residuals of TF-N Model", ylab="CCF")
abline(v=0, col="blue")
```

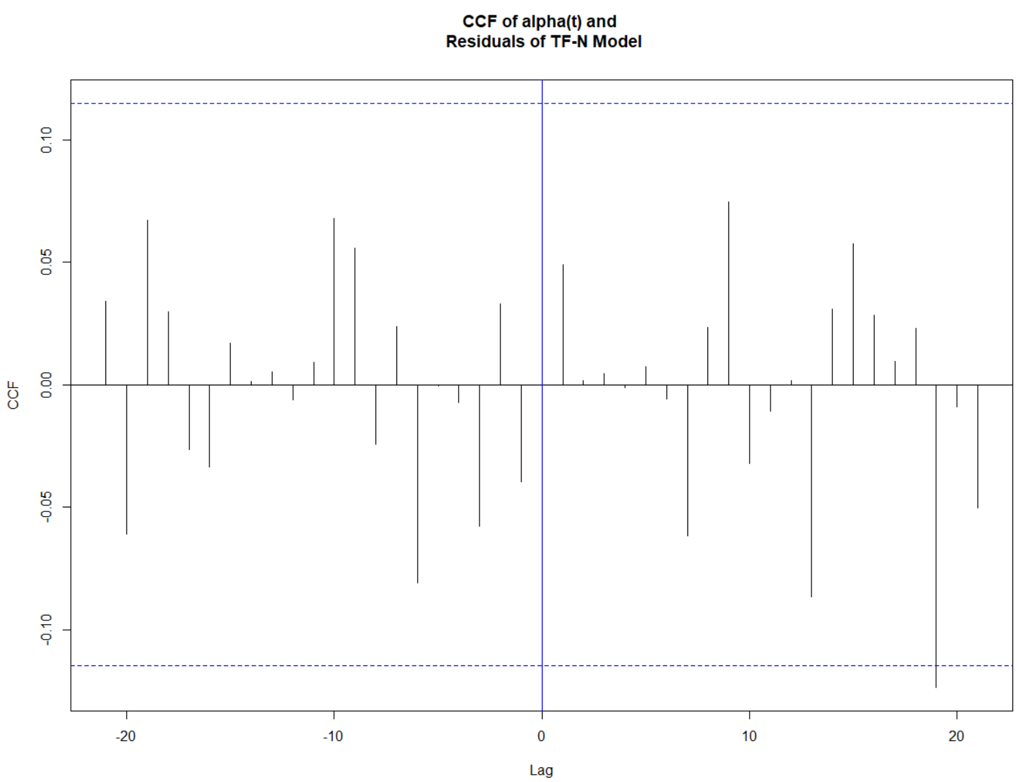
Output:

```
Call:
arimax(x = data2$electricity, order = c(2, 0, 0), include.mean = FALSE, xtran
sf = data.frame(data2$lag3x),
    transfer = list(c(1, 2)))

Coefficients:
         ar1      ar2  data2.lag3x-AR1  data2.lag3x-MA0  data2.lag3x-MA1  dat
a2.lag3x-MA2
      1.2864  -0.5426           0.3962           1.3815           0.4494
1.6563
s.e.  0.0903   0.0885           0.0641           0.2194           0.2574
0.2609

sigma^2 estimated as 5.14:  log likelihood = -319.25,  aic = 650.51
```

Model adequacy checks:


ACF of the Residuals
for TF-N Model
ACF
Lag
0.10
0.05
0.00
-0.05
-0.10
5
10
15
20
25


PACF of the Residuals
for TF-N Model
ACF
Lag
0.10
0.05
0.00
-0.05
-0.10
5
10
15
20
25

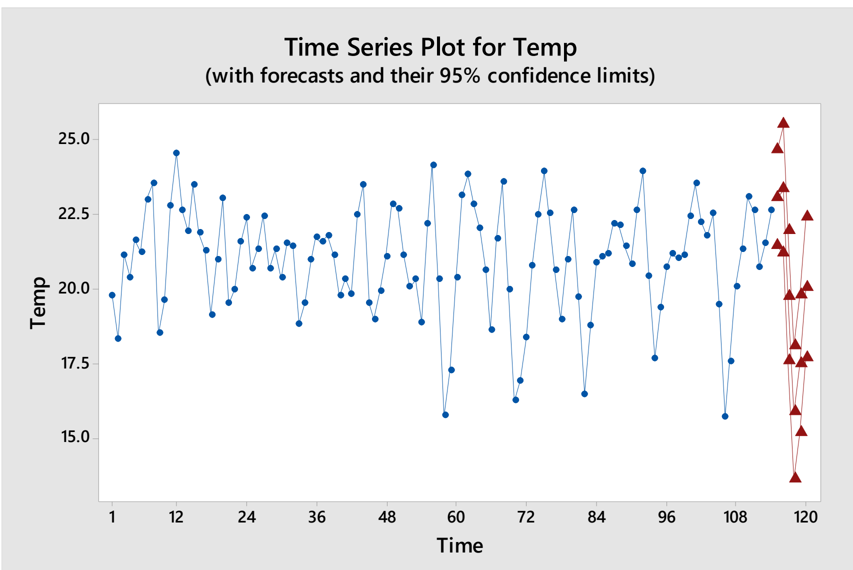

CCF of alpha(t) and
Residuals of TF-N Model
CCF
Lag
0.10
0.05
0.00
-0.05
-0.10
-20
-10
0
10
20

5] Here, we have to forecast the next 6 months after June 2006 for both temperature and electricity:

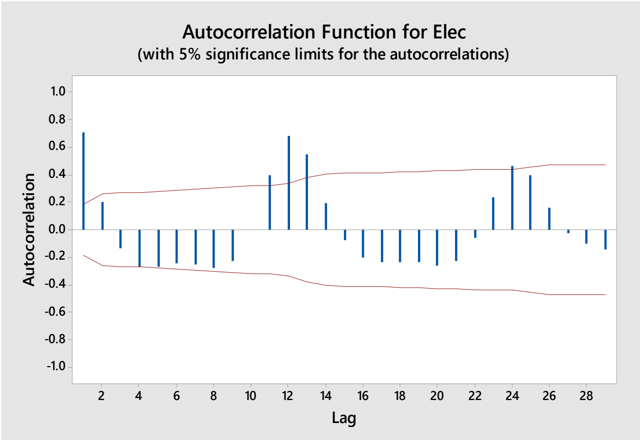
For electricity, we have already got a seasonal ARIMA model i.e. ARIMA(2,0,0) x $(1,0,0)^{12}$.

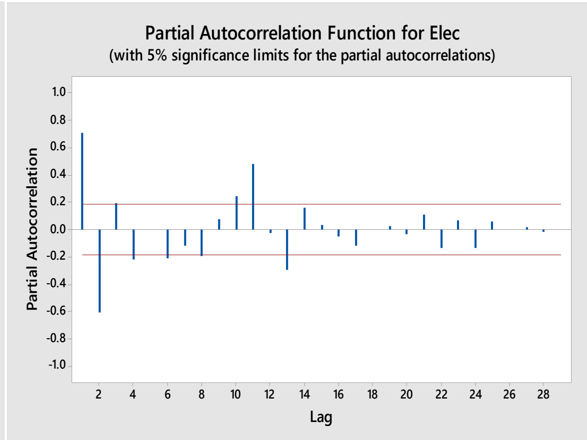
Forecasts from period 114

		95% Limits		
Period	Forecast	Lower	Upper	Actual
115	23.0685	21.4553	24.6817	
116	23.3383	21.1719	25.5047	
117	19.7502	17.5445	21.9560	
118	15.8479	13.6028	18.0929	
119	17.4797	15.1555	19.8038	
120	20.0271	17.6822	22.3719	



Now, we're concerned about finding the model for electricity.





Judging from the ACF and PACF, we can see that it should have AR(2) component surely. I've tried a bunch of different combinations and ARIMA(2,0,1) x $(1,1,2)^{12}$ worked best and the output of the forecasts from minitab is also attached:

Forecasts from period 114

		95% Limits		
Period	Forecast	Lower	Upper	Actual
115	123.240	117.904	128.576	
116	123.562	115.620	131.504	
117	123.511	114.786	132.236	
118	126.906	118.130	135.682	
119	129.219	120.410	138.027	
120	129.466	120.568	138.365