ElectricityProductionTimeSeries

Subham Moda

2024-06-12

```
library(TSA)
```

```
##
## Attaching package: 'TSA'
```

```
## The following objects are masked from 'package:stats':
##
##     acf, arima
```

```
## The following object is masked from 'package:utils':
##
##     tar
```

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.2.3
```

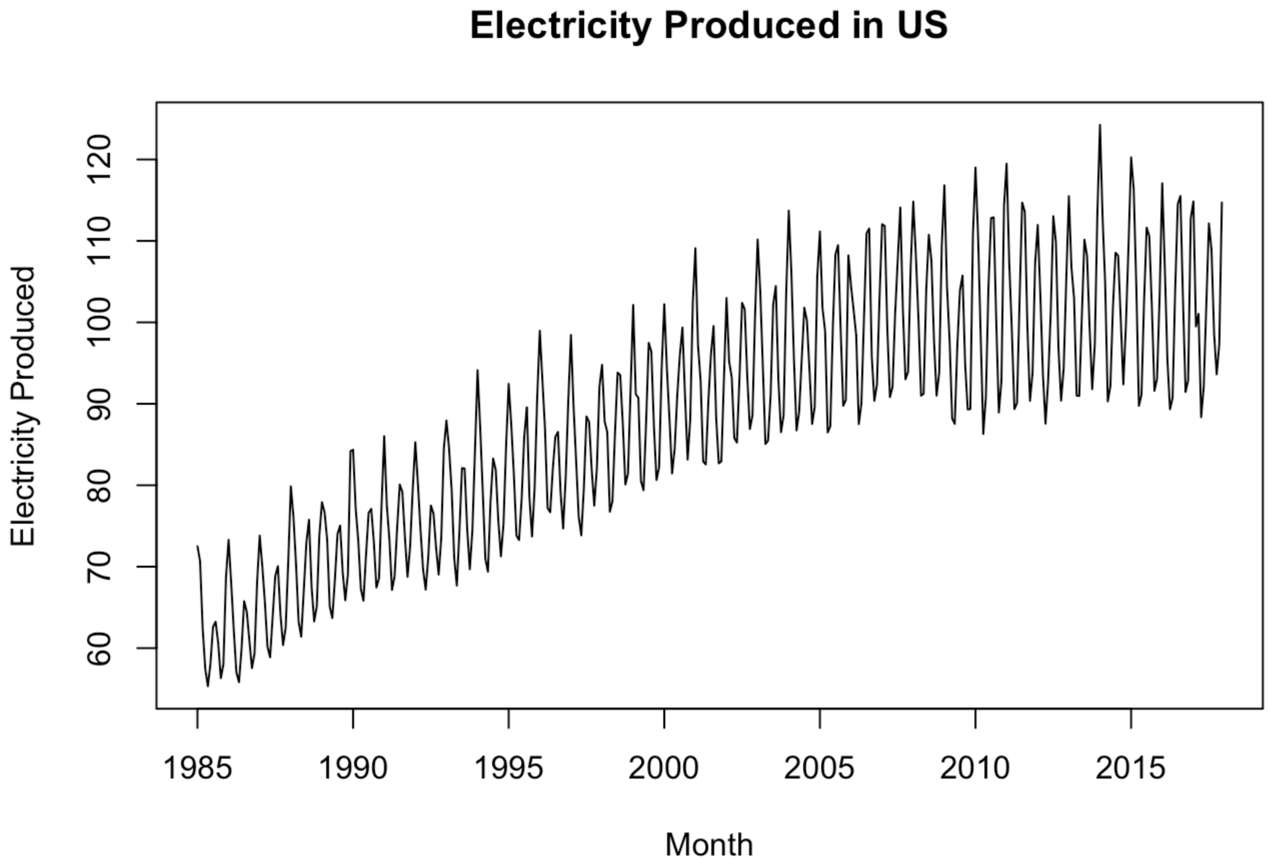
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

I've taken Electricity Production data for United States, which has monthly data for the amount of electricity produced in the US from 1985 to 2018. I will fit the data to a time series model and lastly predict the electric production for future years.

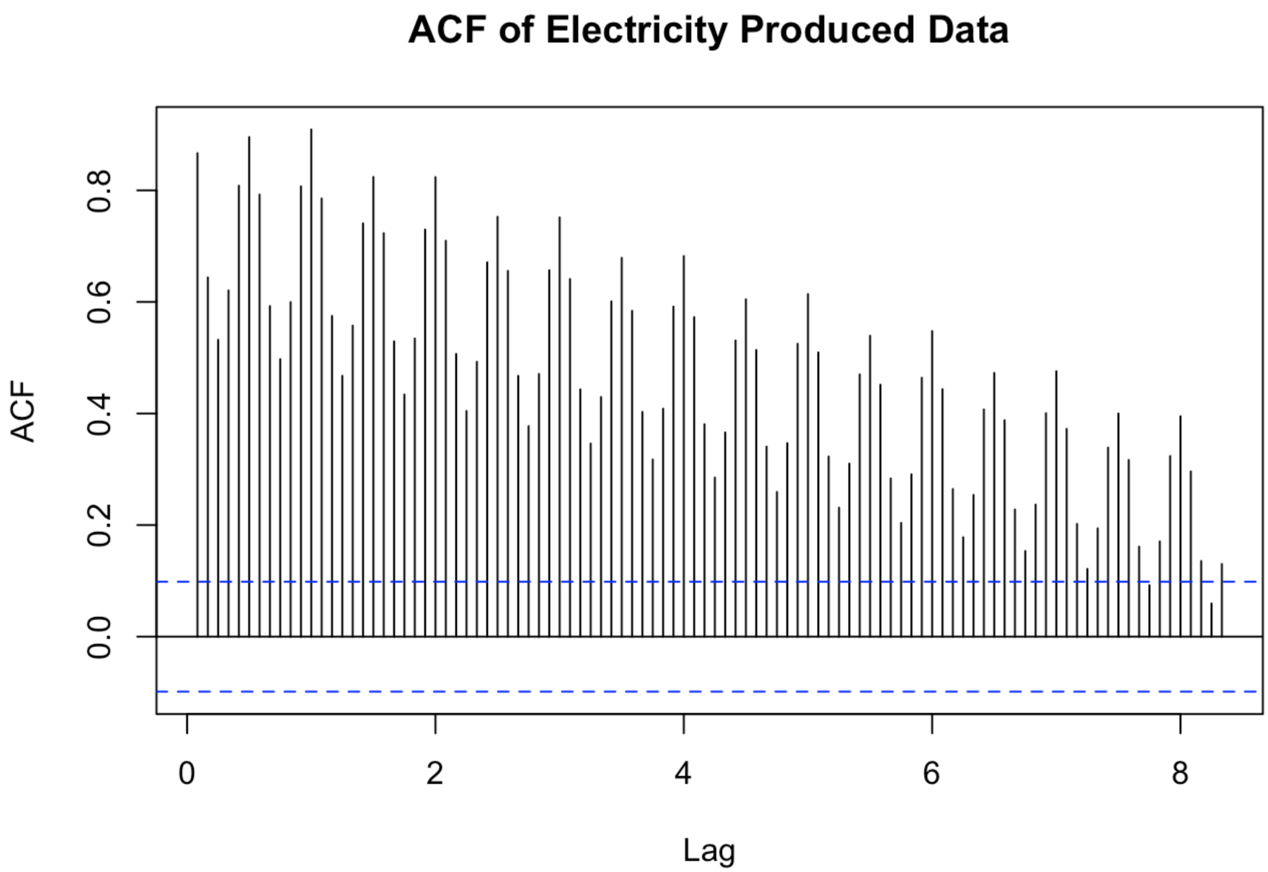
```
s_data <- read.csv('/Users/subhammoda/Documents/Projects/MA641_Project/Electric_Producti
on.csv')
s_data$Value = as.numeric(s_data$Value)
s_data$DATE = as.Date(s_data$DATE, "%m-%d-%Y")

sf_data <- head(s_data, 396)
sf_data <- ts(sf_data$Value, frequency = 12, start = c(1985, 1))
head(sf_data)
```

```
## [1] 72.5052 70.6720 62.4502 57.4714 55.3151 58.0904
```

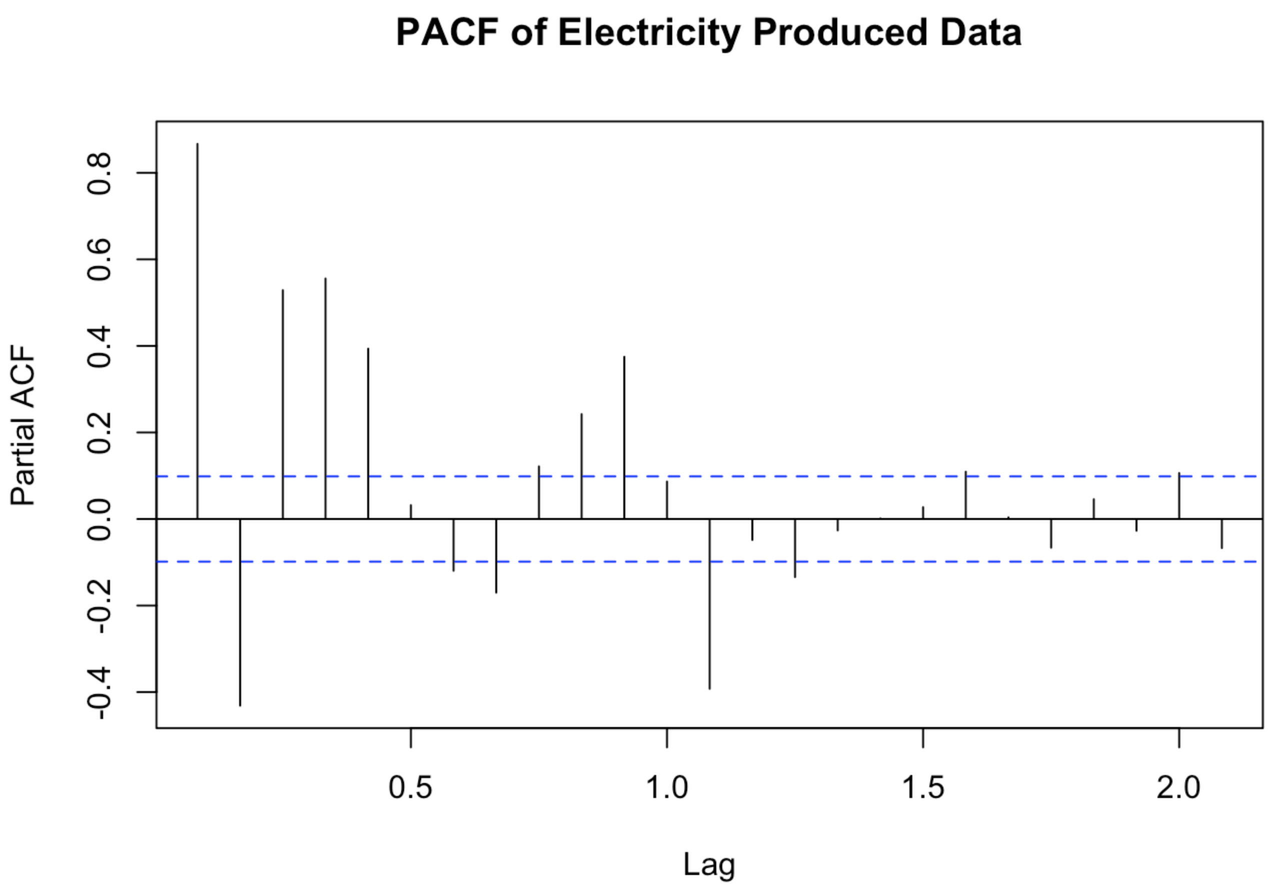
```
plot.ts(sf_data, type = 'l', ylab = 'Electricity Produced', xlab = 'Month', main = "Elec
tricity Produced in US")
```




```
acf(sf_data, main = "ACF of Electricity Produced Data", lag.max = 100)
```




```
pacf(sf_data, main = "PACF of Electricity Produced Data")
```



Check for stationarity using Dicky-Fuller Test.

H0: The time series is non-stationary.

H1: The time series is stationary.

```
adf.test(sf_data)
```

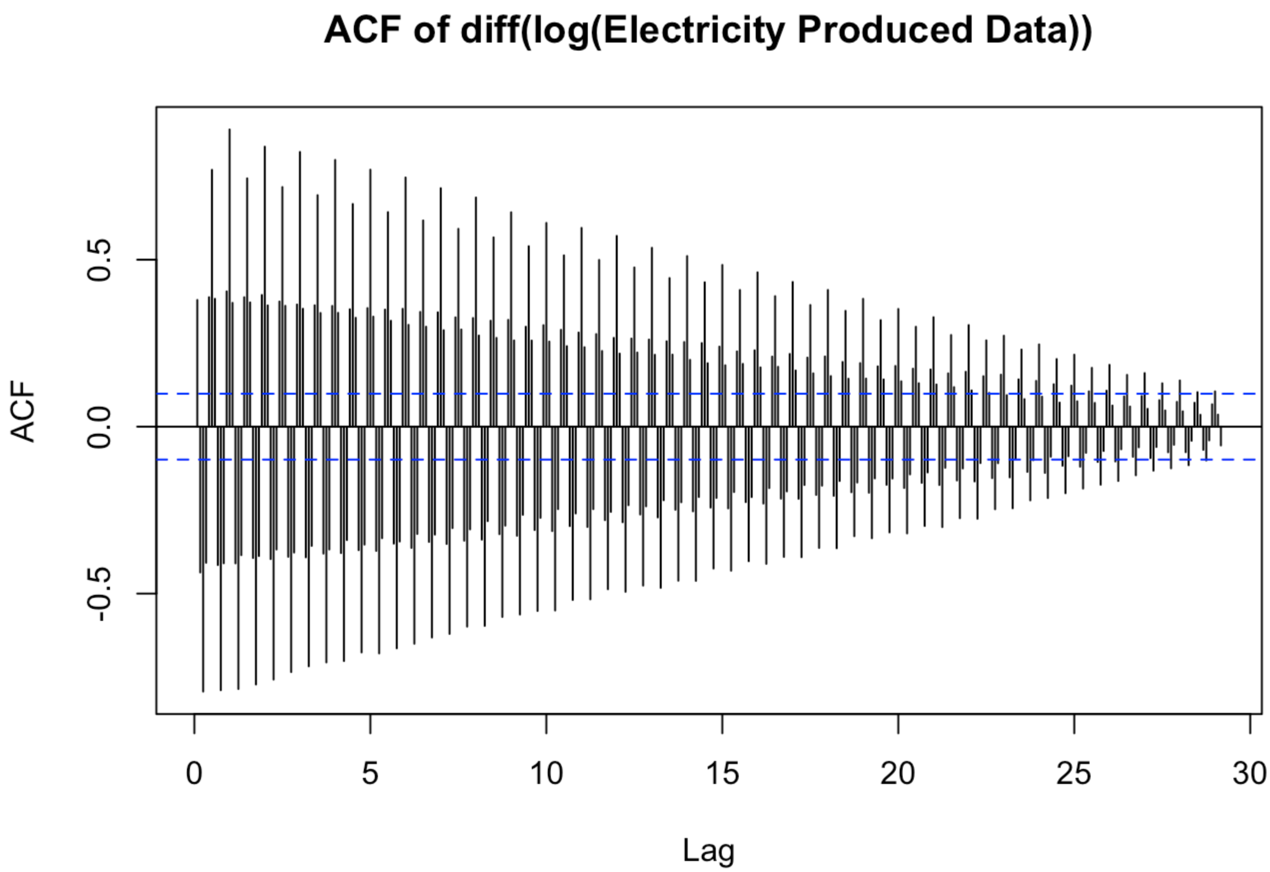
```
## Warning in adf.test(sf_data): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  sf_data
## Dickey-Fuller = -4.9331, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

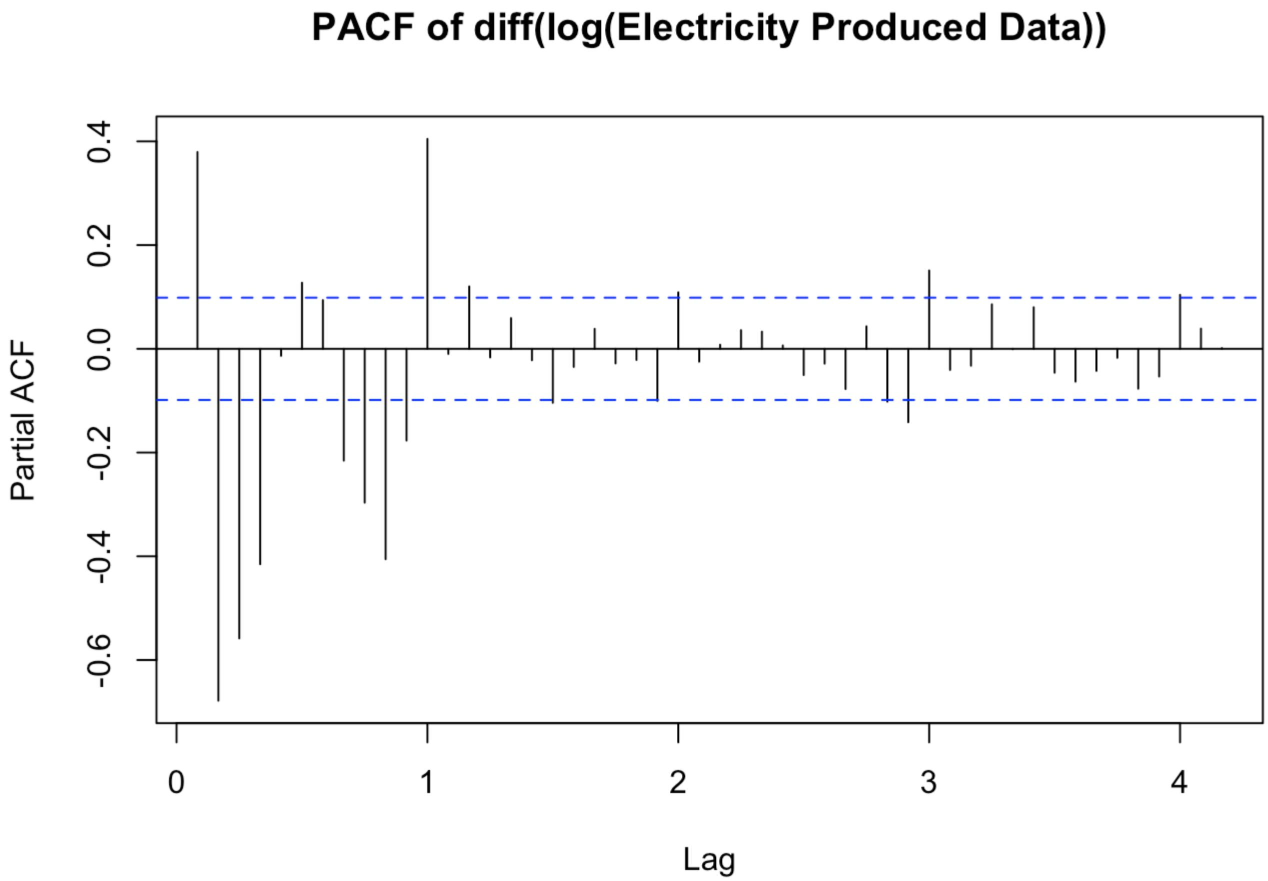
Since p-value is 0.01 < 0.05, we reject H0, the data is stationary.

Since, we are unable to directly capture the seasonality in the data, we try to modify the data by taking difference of log of data.

```
sf_data_l <- log(sf_data)
acf(diff(sf_data_l), lag.max = 350, main = "ACF of diff(log(Electricity Produced Dat
a))")
```



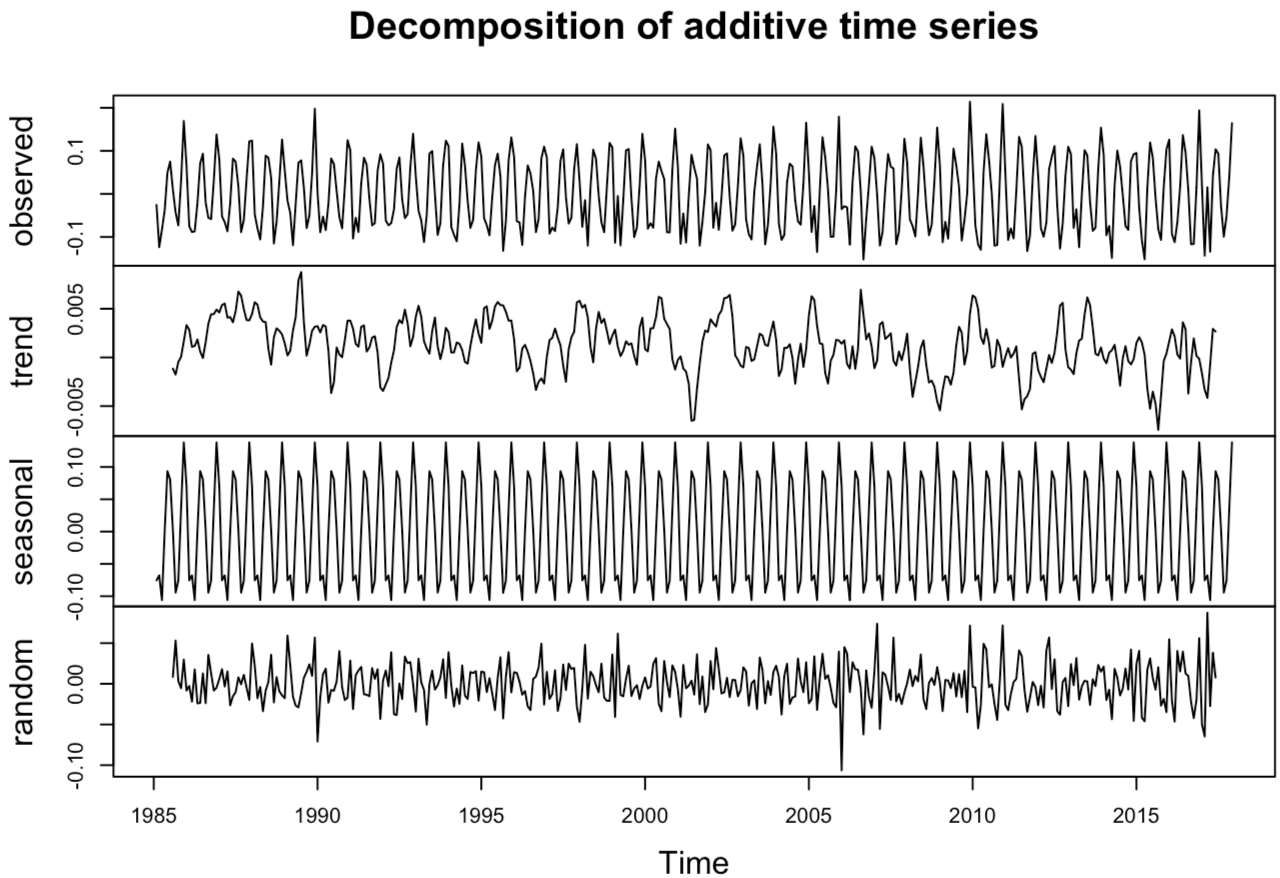

```
pacf(diff(sf_data_l),lag.max = 50, main = "PACF of diff(log(Electricity Produced Dat
a))")
```




```
eacf(diff(sf_data_l))
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x x x  x  x  x
## 1 x x x x x x x x x x x  x  x  x
## 2 x x x x o o o o o o x  x  x  x
## 3 x x x x o x o o o o o  x  x  x
## 4 x x o x o o o o o o o  x  o  o
## 5 x x o x x o o o o o o  x  o  o
## 6 x x o x x o o o o o o  x  x  o
## 7 x x x x o x o o o o o  x  x  o
```

```
plot(decompose(diff(sf_data_l)))
```



Based on the ACF, PACF and EACF, we test for the following 4 models:-

1. ARIMA(1,1,1)x(4,1,4)6
2. ARIMA(1,1,1)x(2,1,2)12
3. ARIMA(1,1,1)x(2,1,1)12
4. ARIMA(1,1,1)x(2,1,0)12

Model1 - ARIMA(1,1,1)x(4,1,4)6

```
model1 <- arima(sf_data_l, order= c(1,1,1), seasonal=list(order=c(4,1,4), period= 6))
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg = xreg,
## : possible convergence problem: optim gave code = 1
```

```
model1
```

```
##
## Call:
## arima(x = sf_data_l, order = c(1, 1, 1), seasonal = list(order = c(4, 1, 4),
##     period = 6))
##
## Coefficients:
##          ar1      ma1     sar1    sar2     sar3     sar4     sma1     sma2
##       0.5228  -0.9335  -0.0895  0.4815  -0.5688  -0.1470  -0.8868  -0.3402
## s.e.  0.0528   0.0228   0.3096  0.1960   0.2689   0.0674   0.3118   0.2752
##         sma3     sma4
##       0.8018  -0.4280
## s.e.  0.2600   0.2262
##
## sigma^2 estimated as 0.0005734:  log likelihood = 887.75,  aic = -1755.5
```

```
AIC(model1)
```

```
## [1] -1753.499
```

```
BIC(model1)
```

```
## [1] -1709.9
```

Model2 - ARIMA(1,1,1)x(2,1,2)12

```
model2 <- arima(sf_data_l, order= c(1,1,1), seasonal=list(order=c(2,1,2), period= 12))
model2
```

```
##
## Call:
## arima(x = sf_data_l, order = c(1, 1, 1), seasonal = list(order = c(2, 1, 2),
##     period = 12))
##
## Coefficients:
##          ar1      ma1    sar1     sar2     sma1    sma2
##       0.4926  -0.9235  0.3463  -0.2534  -1.1082  0.3022
## s.e.  0.0542   0.0237  0.2830   0.0650   0.2868  0.2476
##
## sigma^2 estimated as 0.0005633:  log likelihood = 881.13,  aic = -1750.26
```

```
AIC(model2)
```

```
## [1] -1748.265
```

```
BIC(model2)
```

```
## [1] -1720.628
```

Model3 - ARIMA(1,1,1)x(2,1,1)12

```
model3 <- arima(sf_data_l, order= c(1,1,1), seasonal=list(order=c(2,1,1), period= 12))
model3
```

```
##
## Call:
## arima(x = sf_data_l, order = c(1, 1, 1), seasonal = list(order = c(2, 1, 1),
##     period = 12))
##
## Coefficients:
##          ar1      ma1    sar1     sar2     sma1
##       0.4981  -0.9227  0.0126  -0.2182  -0.7659
## s.e.  0.0536   0.0233  0.0650   0.0603   0.0491
##
## sigma^2 estimated as 0.0005652:  log likelihood = 880.56,  aic = -1751.12
```

```
AIC(model3)
```

```
## [1] -1749.125
```

```
BIC(model3)
```

```
## [1] -1725.437
```

Model4 - ARIMA(1,1,1)x(2,1,0)12

```
model4 <- arima(sf_data_l, order= c(1,1,1), seasonal=list(order=c(2,1,0), period= 12))
model4
```

```
##
## Call:
## arima(x = sf_data_l, order = c(1, 1, 1), seasonal = list(order = c(2, 1, 0),
##     period = 12))
##
## Coefficients:
##          ar1      ma1     sar1     sar2
##       0.5339  -0.9647  -0.5131  -0.3909
## s.e.  0.0477   0.0143   0.0499   0.0500
##
## sigma^2 estimated as 0.0006672:  log likelihood = 852.83,  aic = -1697.66
```

```
AIC(model4)
```

```
## [1] -1695.663
```

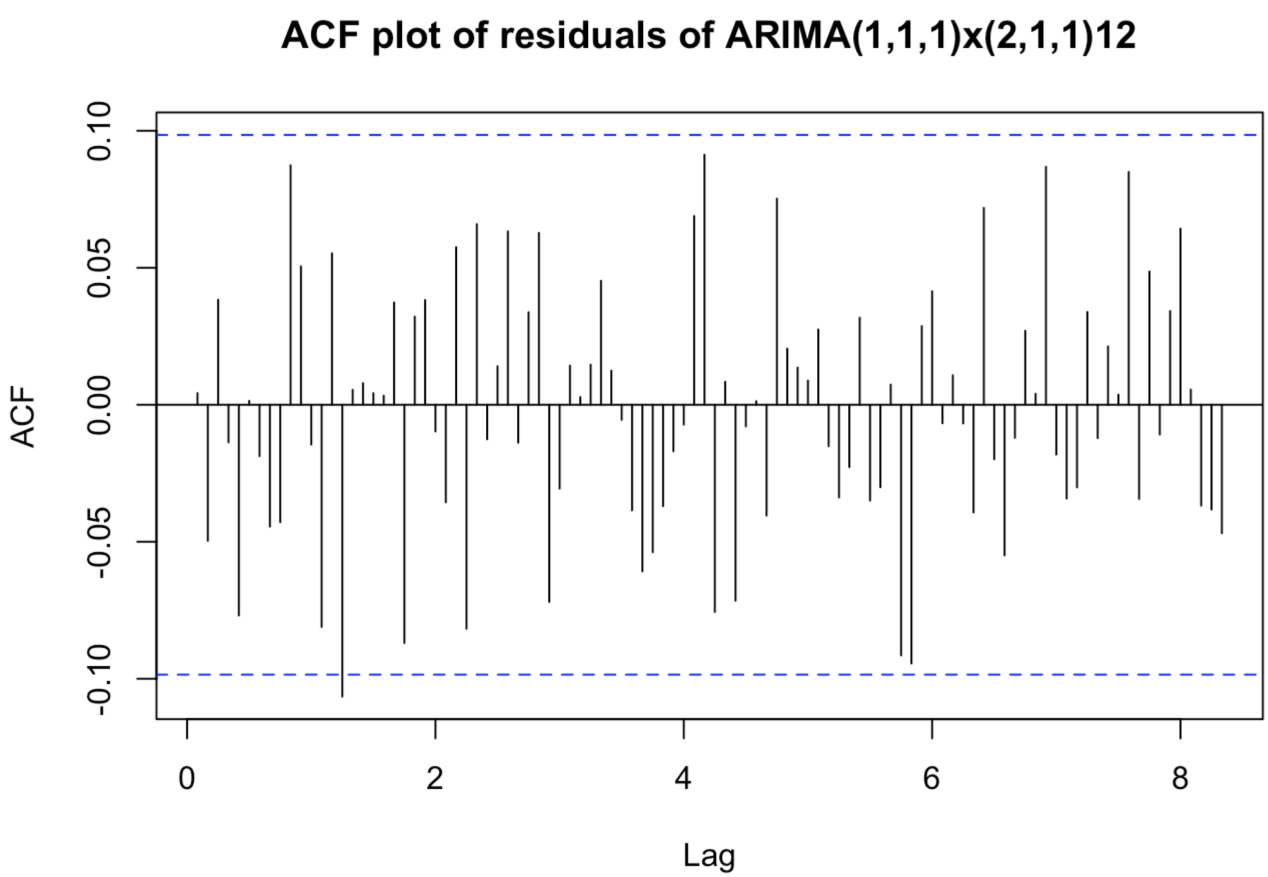
```
BIC(model4)
```

```
## [1] -1675.923
```

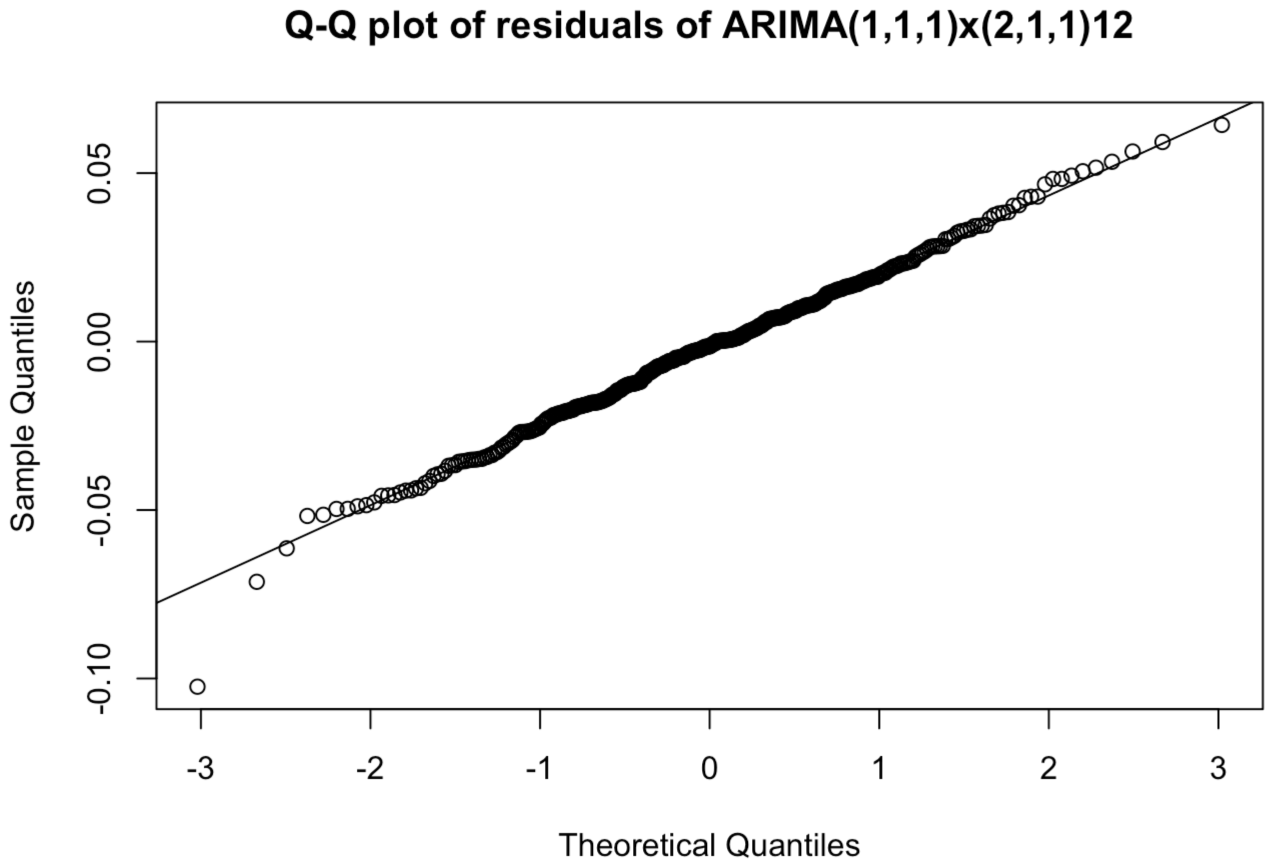
The best model for the above seasonal data is ARIMA(1,1,1)x(2,1,1)12 based on AIC and BIC values.

Residual Analysis

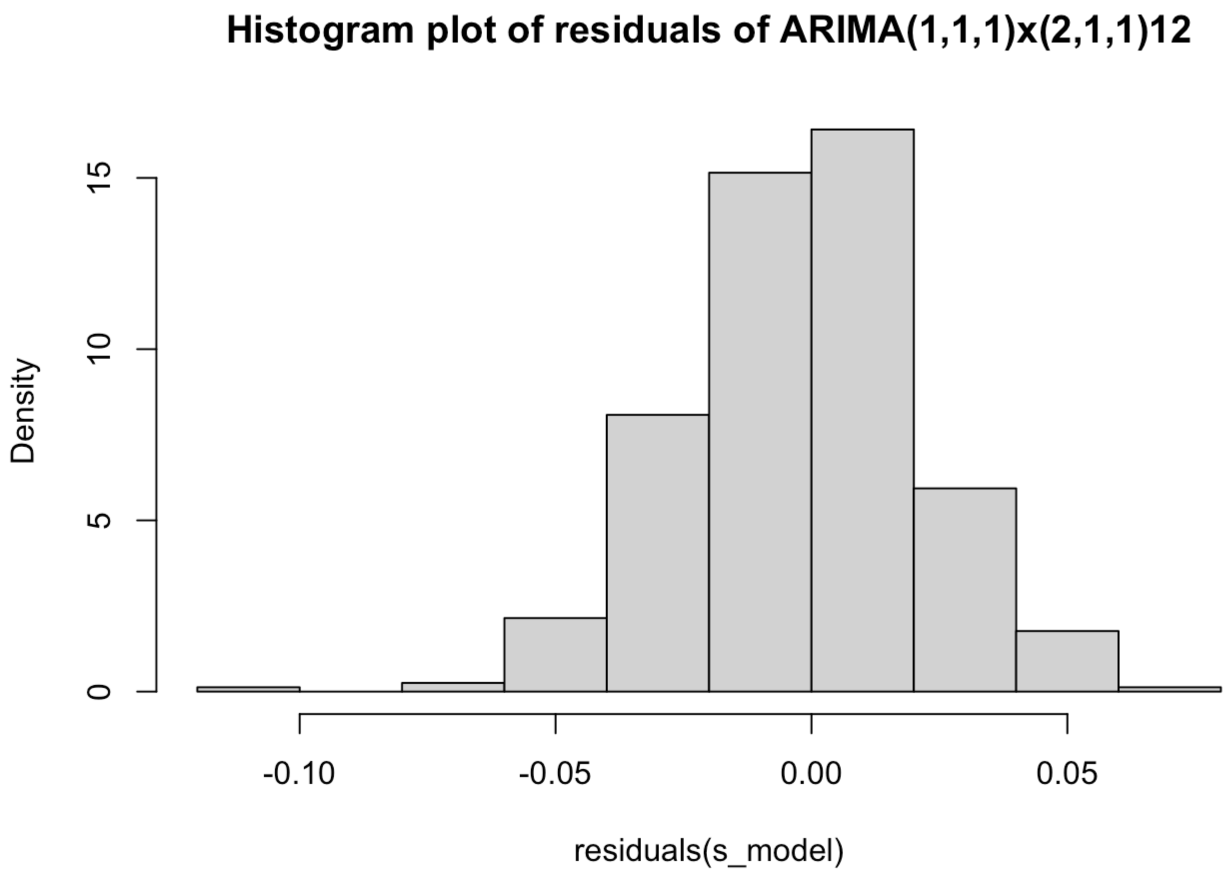
```
s_model <- arima(sf_data_l, order= c(1,1,1), seasonal=list(order=c(2,1,1), period= 12))
acf(residuals(s_model), lag.max = 100, main = "ACF plot of residuals of ARIMA(1,1,1)x(2,
1,1)12")
```




```
qqnorm(residuals(s_model), main = "Q-Q plot of residuals of ARIMA(1,1,1)x(2,1,1)12"); qq
line(residuals(s_model))
```




```
hist(residuals(s_model), freq = FALSE, main = "Histogram plot of residuals of ARIMA(1,1,
1)x(2,1,1)12")
```




```
shapiro.test(residuals(s_model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(s_model)
## W = 0.9944, p-value = 0.1569
```

From the Shapiro-Wilk test, the p-value of 0.1569 > 0.05, shows that the residual is normal.

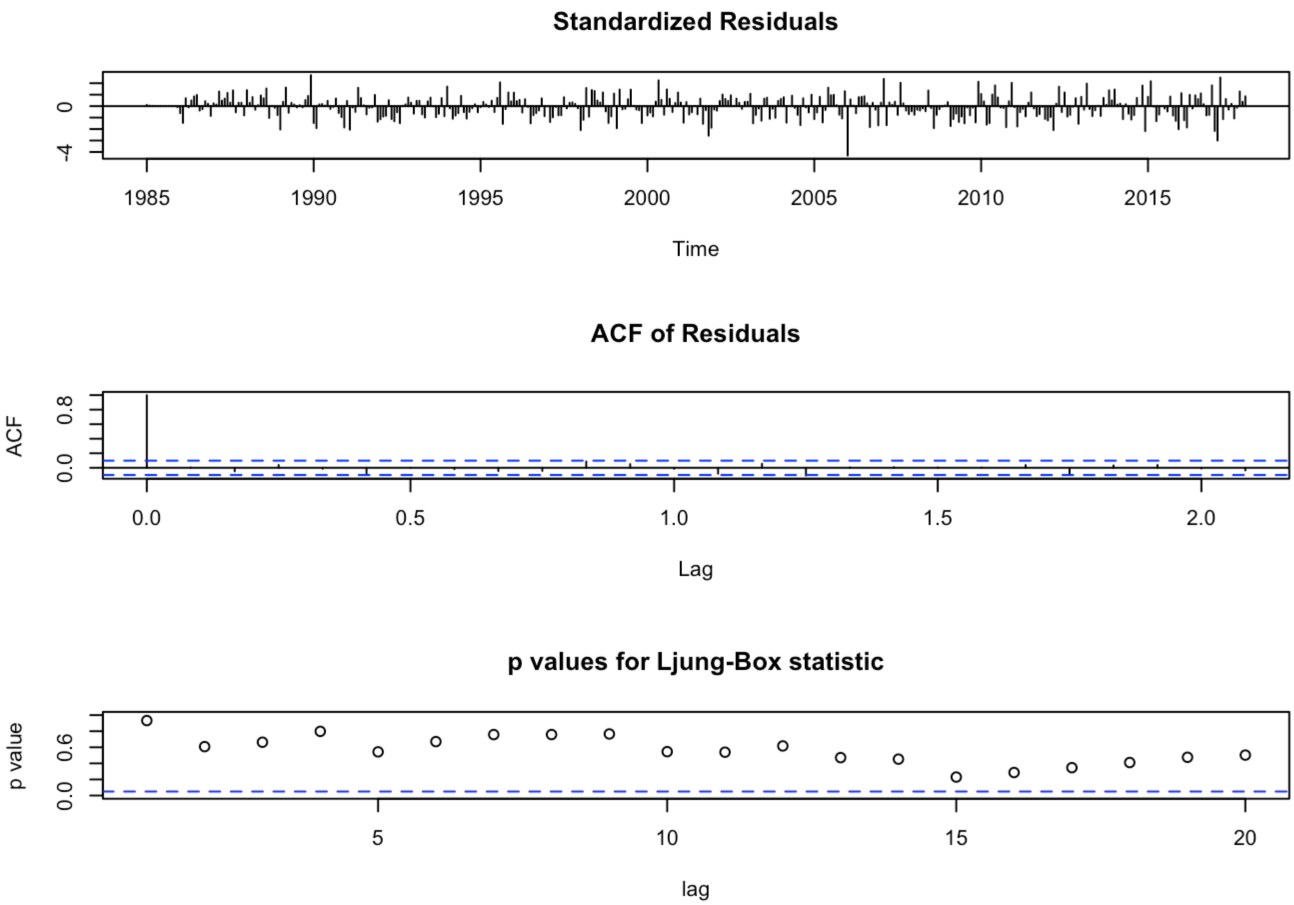
```
Box.test(residuals(s_model), lag = 10, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  residuals(s_model)
## X-squared = 8.8598, df = 10, p-value = 0.5455
```

The Box-Ljung test, having p-value 0.5455 > 0.05, shows that the residuals are independent and identically distributed.

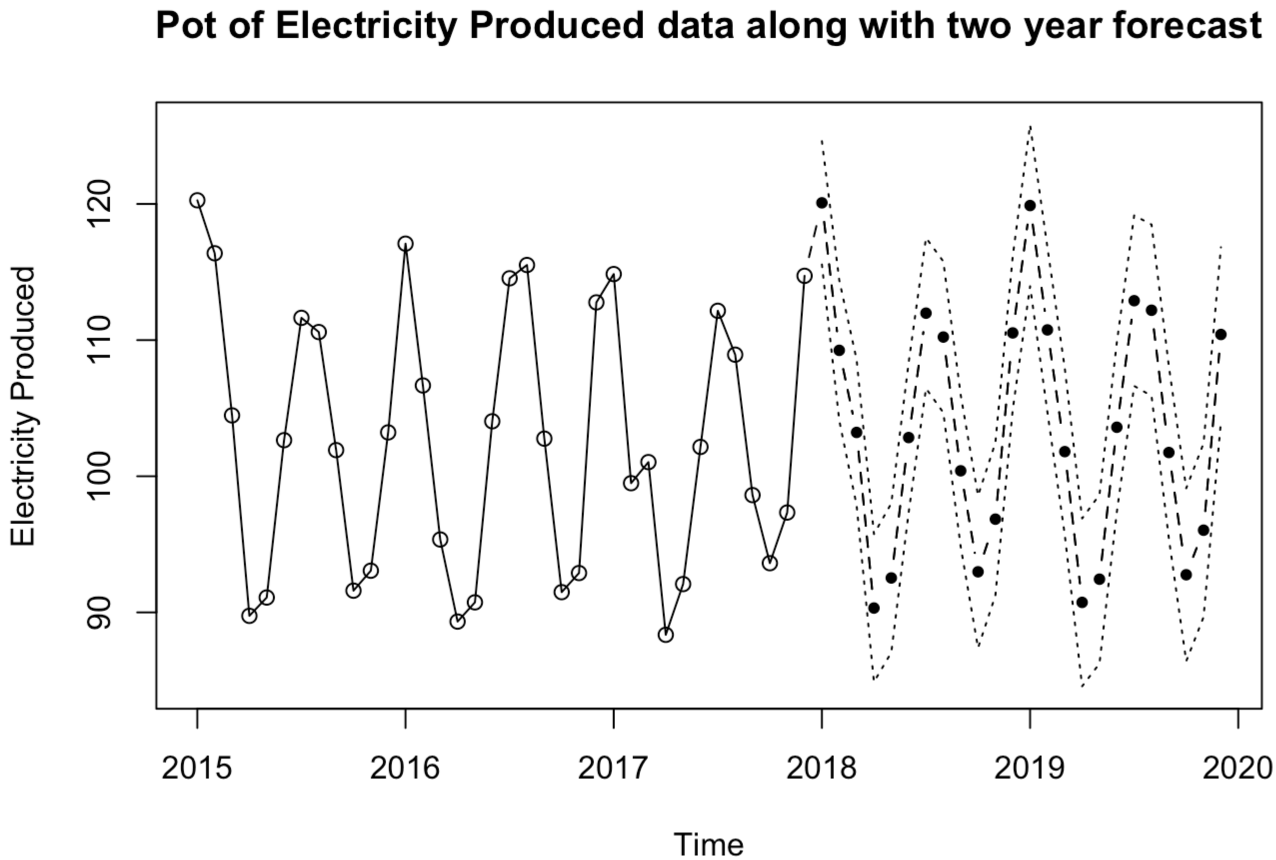
Diagnostic plot of ARIMA(1,1,1)x(2,1,1)12

```
tsdiag(s_model, gof.lag = 20)
```



Forecast

```
model_s <- arima(sf_data, order= c(1,1,1), seasonal=list(order=c(2,1,1), period= 12))
plot(model_s, n1=c(2015,1), n.ahead=24,ylab='Electricity Produced',pch=20, main = "Pot o
f Electricity Produced data along with two year forecast")
```



Conclusion

We can see that SARIMA(1,1,1)x(2,1,1)[12] is a great fit to the data, and is able to forecast the Electricity Production by capturing the seasonality trends.