Electric_Production_Forecasting

Bharat Sharma_23MDT1051

2024-02-25

```r
## Electric Production Analysis and Forecasting

#Loading Electric Production Dataset
dataset=read.csv("Electric_Production.csv")

#Getting Quick overview of Data Set Structure and contents
head(dataset)
```

```
##       DATE IPG2211A2N
## 1 1/1/1985    72.5052
## 2 2/1/1985    70.6720
## 3 3/1/1985    62.4502
## 4 4/1/1985    57.4714
## 5 5/1/1985    55.3151
## 6 6/1/1985    58.0904
```

```r
tail(dataset)
```

```
##           DATE IPG2211A2N
## 392  8/1/2017   108.9312
## 393  9/1/2017    98.6154
## 394 10/1/2017    93.6137
## 395 11/1/2017    97.3359
## 396 12/1/2017   114.7212
## 397  1/1/2018   129.4048
```

```r
#total Entries
print(paste("Total Entries :-",nrow(dataset)))
```

```
## [1] "Total Entries :- 397"
```

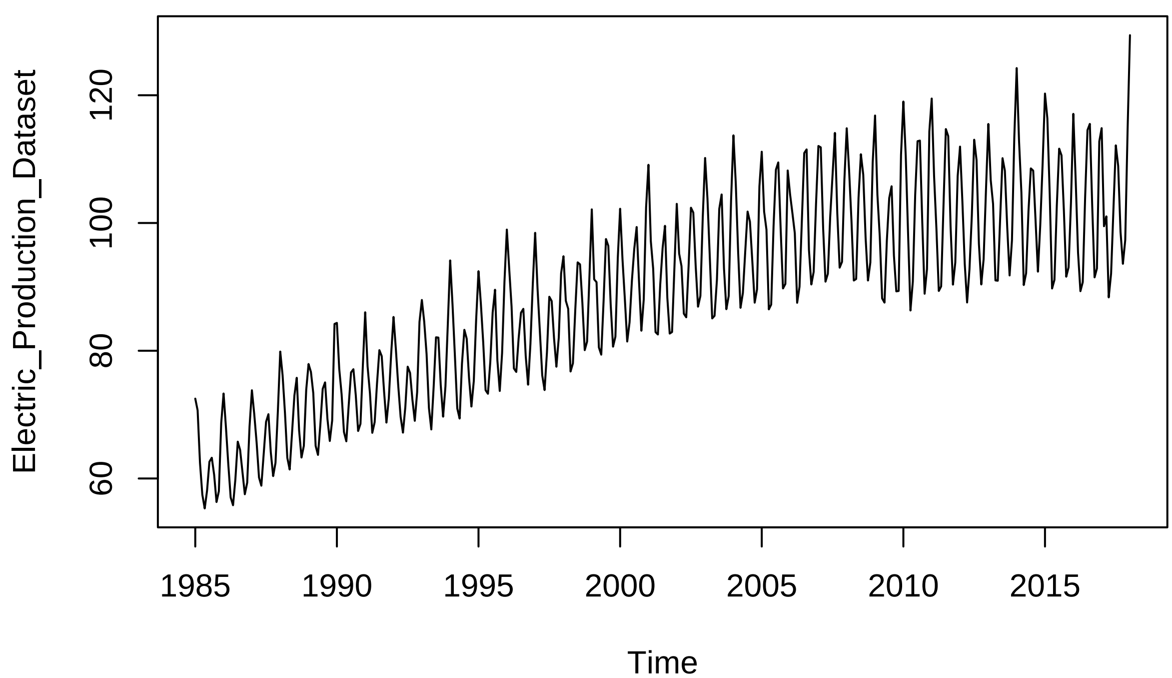
```r
#Time Series Indexing
temp<-dataset["IPG2211A2N"]
rownames(temp)<-dataset$DATE
dataset=temp

#Counting Missing Values
sum(is.na(dataset))
```

```
## [1] 0
```

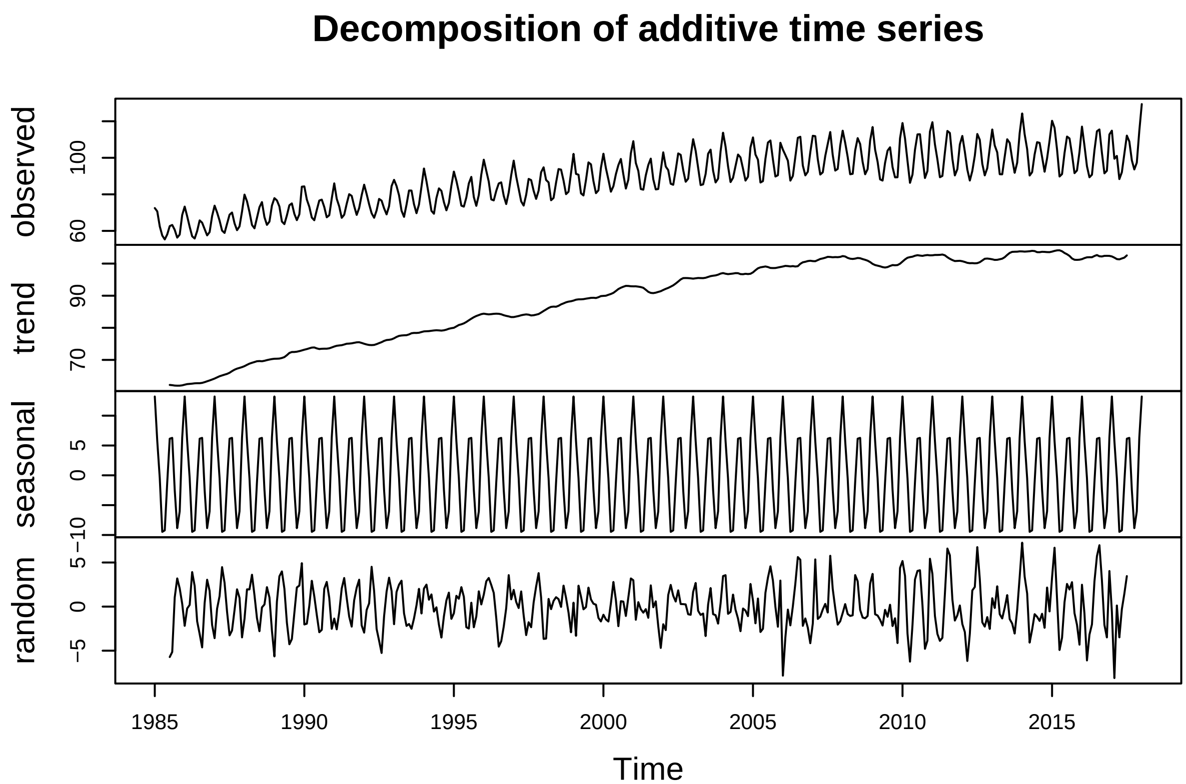
```r
library(timeSeries)
```

```
## Loading required package: timeDate

##
## Attaching package: 'timeSeries'

## The following objects are masked from 'package:graphics':
##
##     lines, points
```

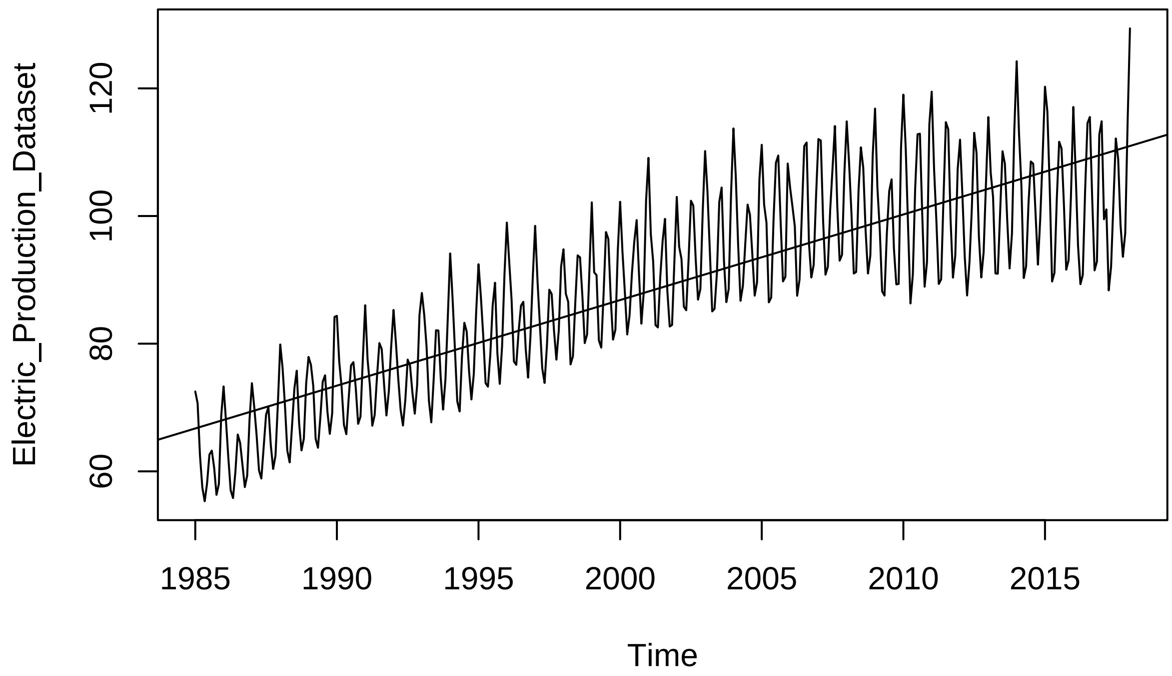
```r
#data set to Time Series Object
Electric_Production_Dataset=ts(dataset$IPG2211A2N,start=c(1985,1),frequency=12)
plot(Electric_Production_Dataset)
```




```r
#Since as from the Graph Plotted above we can say that the Electric_Production_Dataset is Additive in n
decompose_data=decompose(Electric_Production_Dataset,"additive")
plot(decompose_data)
```

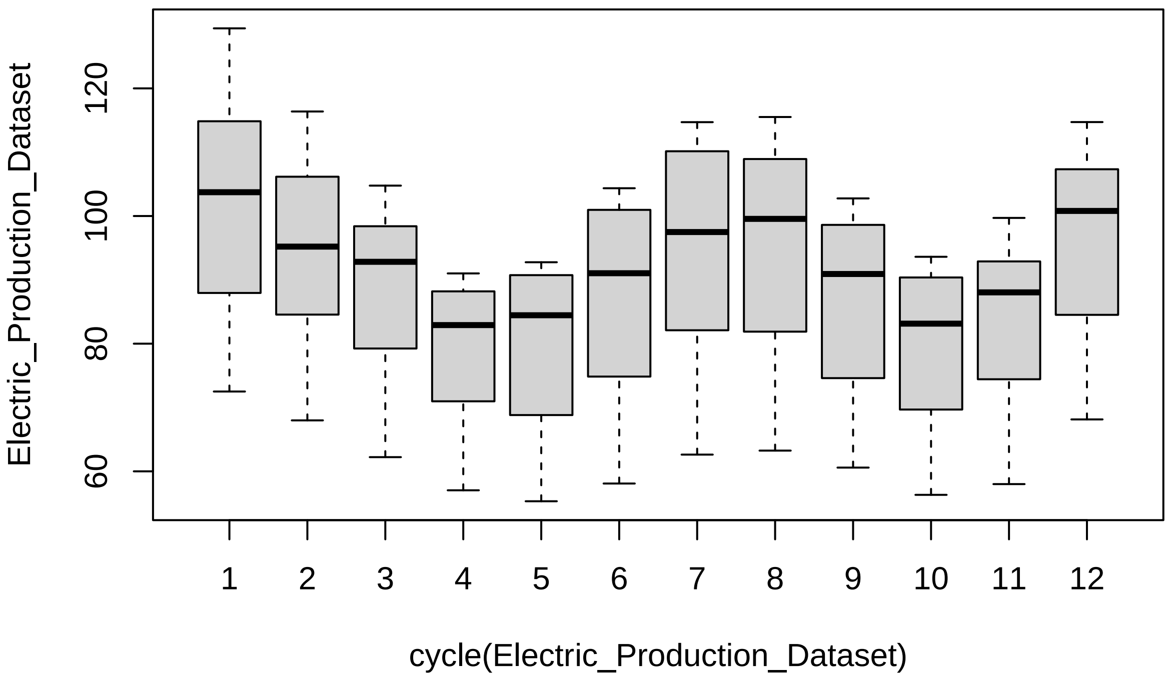



```
#Plotting Trend Line
plot(Electric_Production_Dataset)
abline(reg=lm(Electric_Production_Dataset~time(Electric_Production_Dataset)))
```

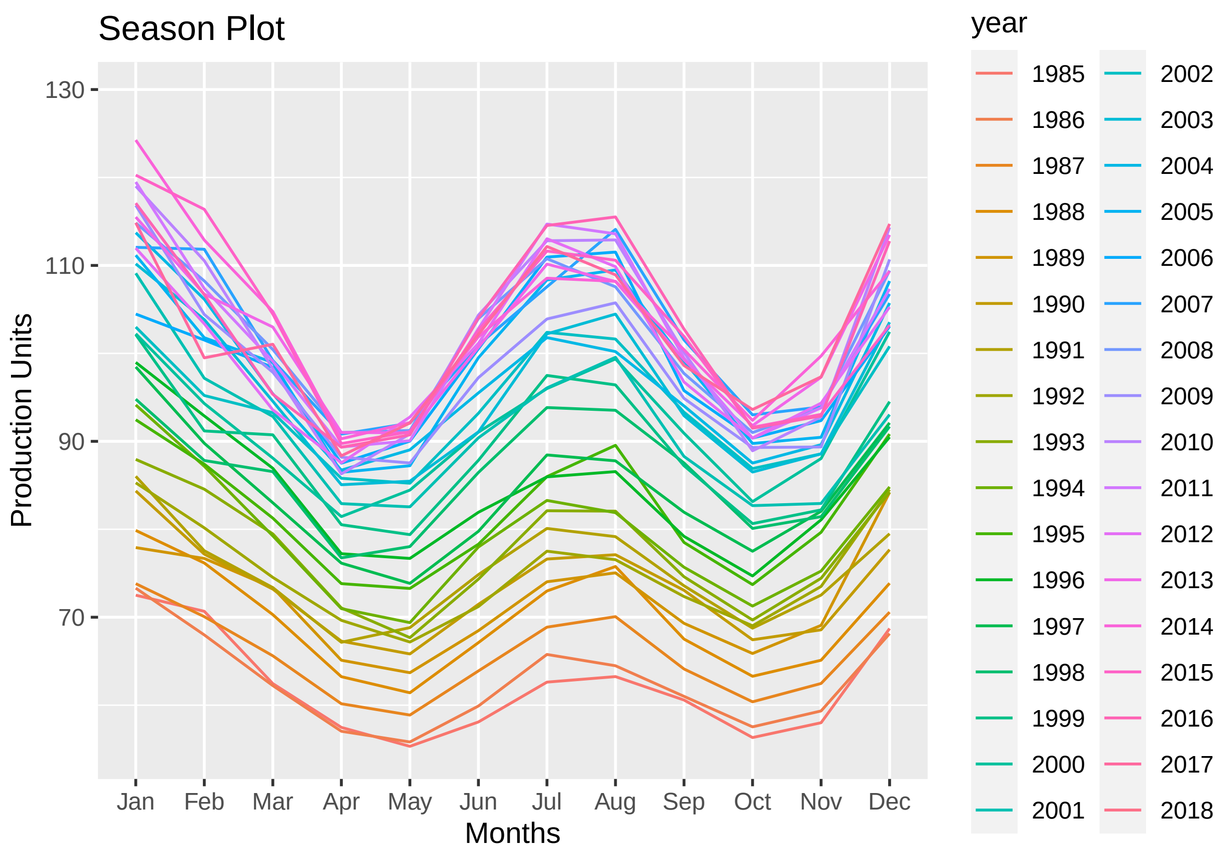



```
#Creating boxplot with cycle()
boxplot(Electric_Production_Dataset~cycle(Electric_Production_Dataset))

#Plotting Seasonl Plot

library(forecast)
```

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




```r
ggseasonplot(Electric_Production_Dataset,ylab="Production Units",xlab="Months",main="Season Plot")
```

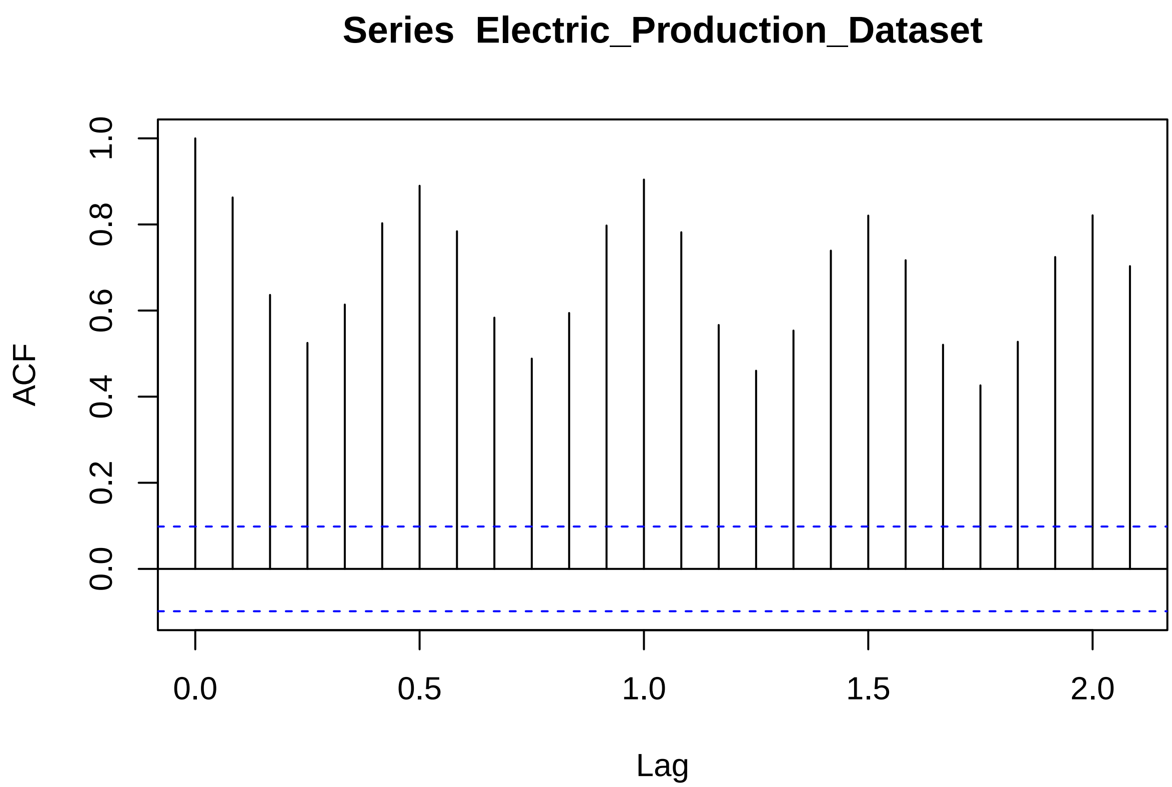



```r
#Checking Satationarity
library(tseries)
adf.test(Electric_Production_Dataset)
```

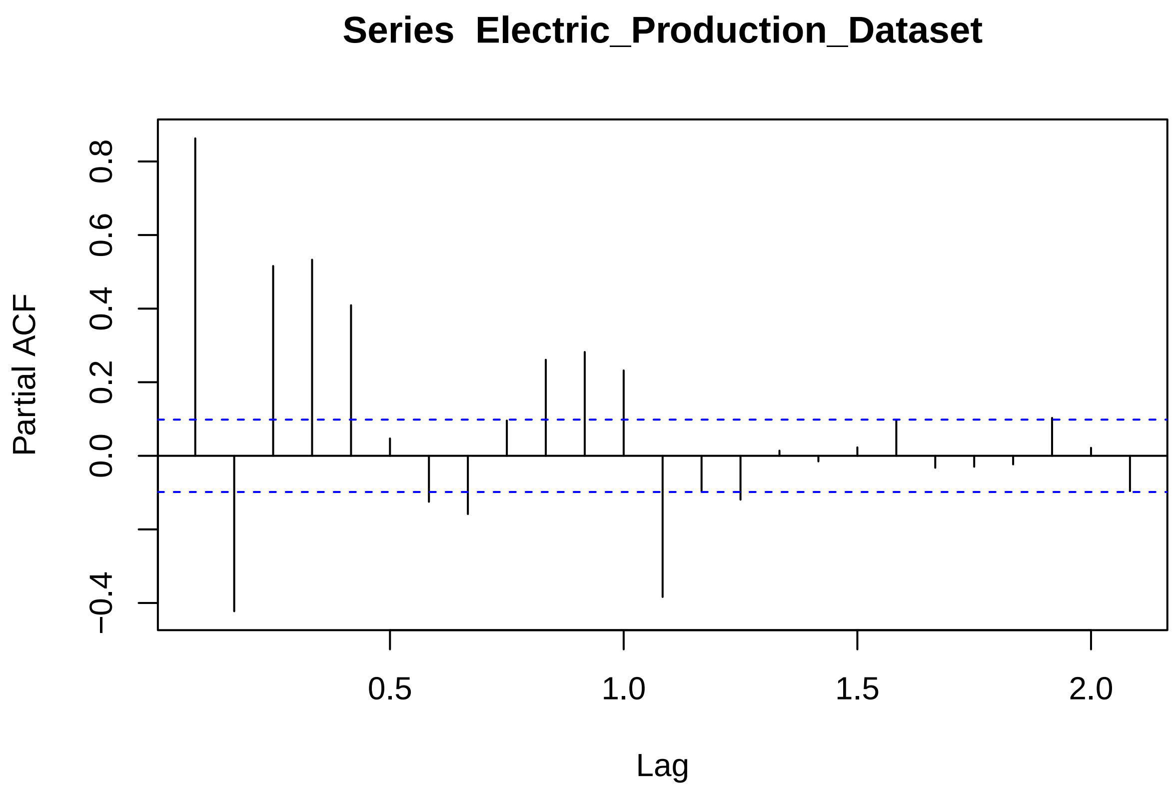
```
## Warning in adf.test(Electric_Production_Dataset): p-value smaller than printed
## p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  Electric_Production_Dataset
## Dickey-Fuller = -5.139, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

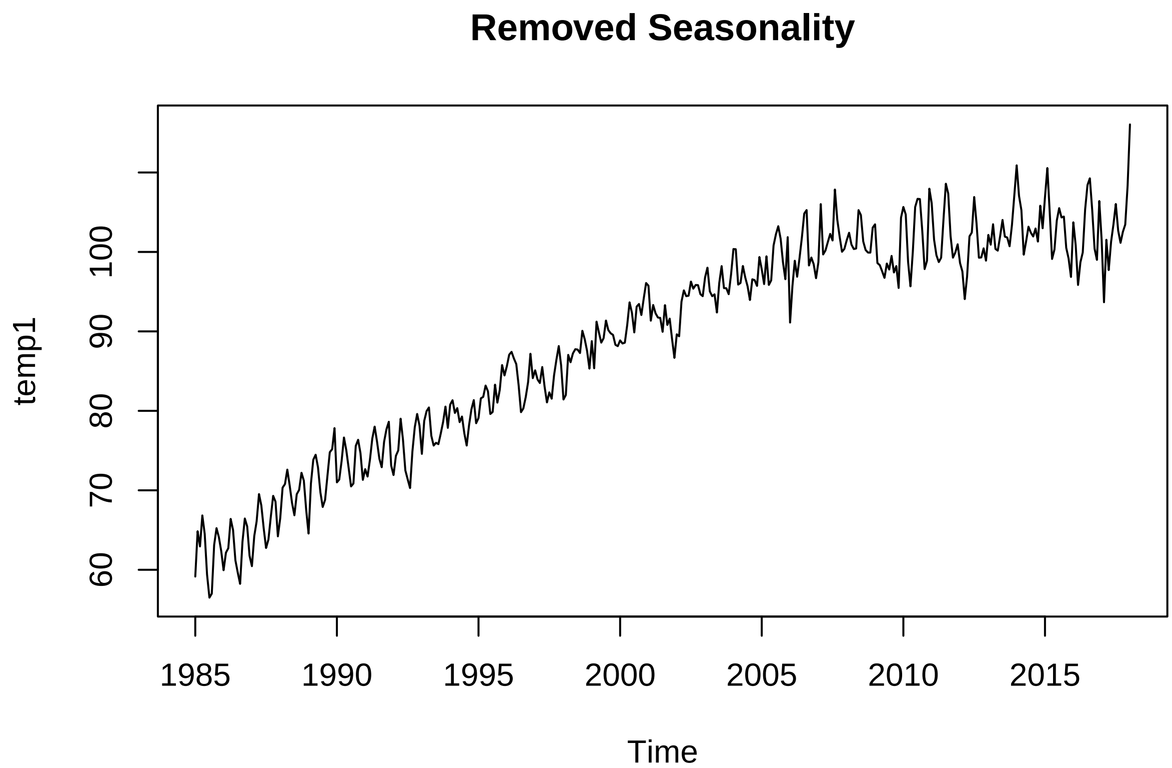
```r
#adf test says that the Series is Stationary so Plotting ACF

acf(Electric_Production_Dataset)
```




```
pacf(Electric_Production_Dataset)
```



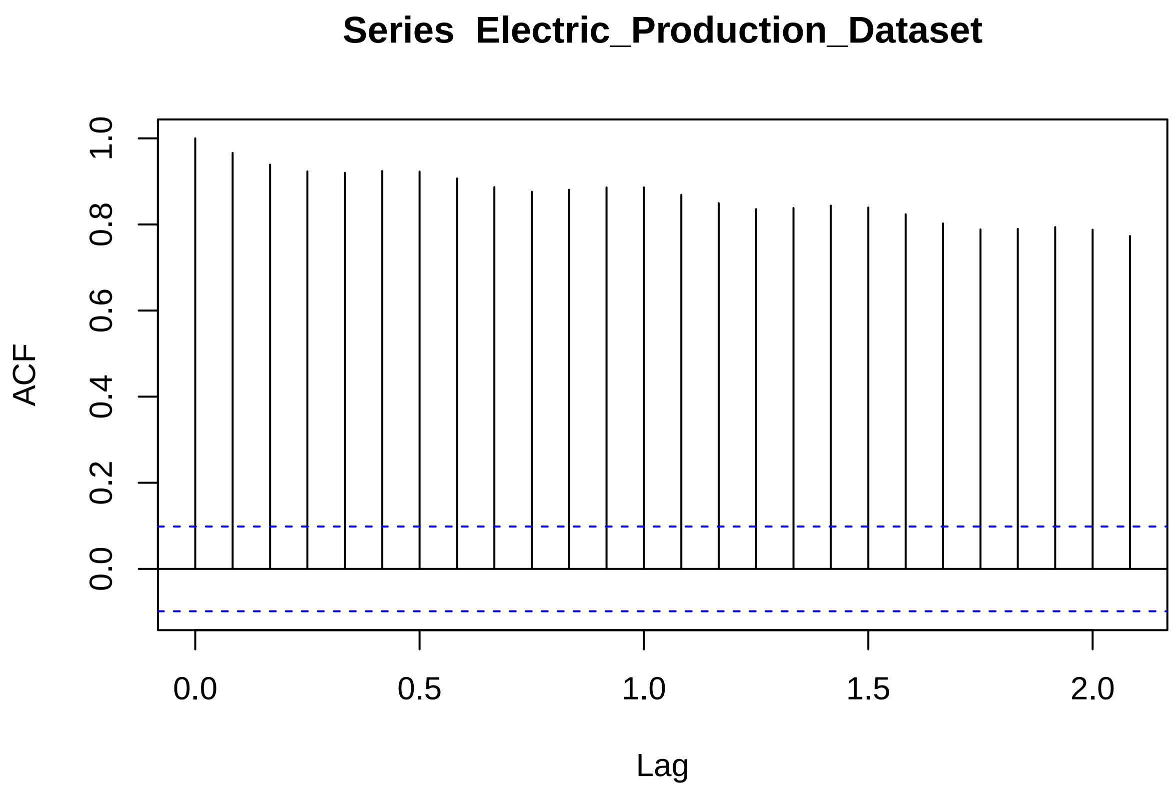

```r
#acf and pacf showing non Stationarity, So from above We concluded that the Electric_Production_Dataset

#Removing Seasonality for making dataset stationary
temp=stl(Electric_Production_Dataset,'per')
temp1=seasadj(temp)
plot(temp1,main="Removed Seasonality")
```



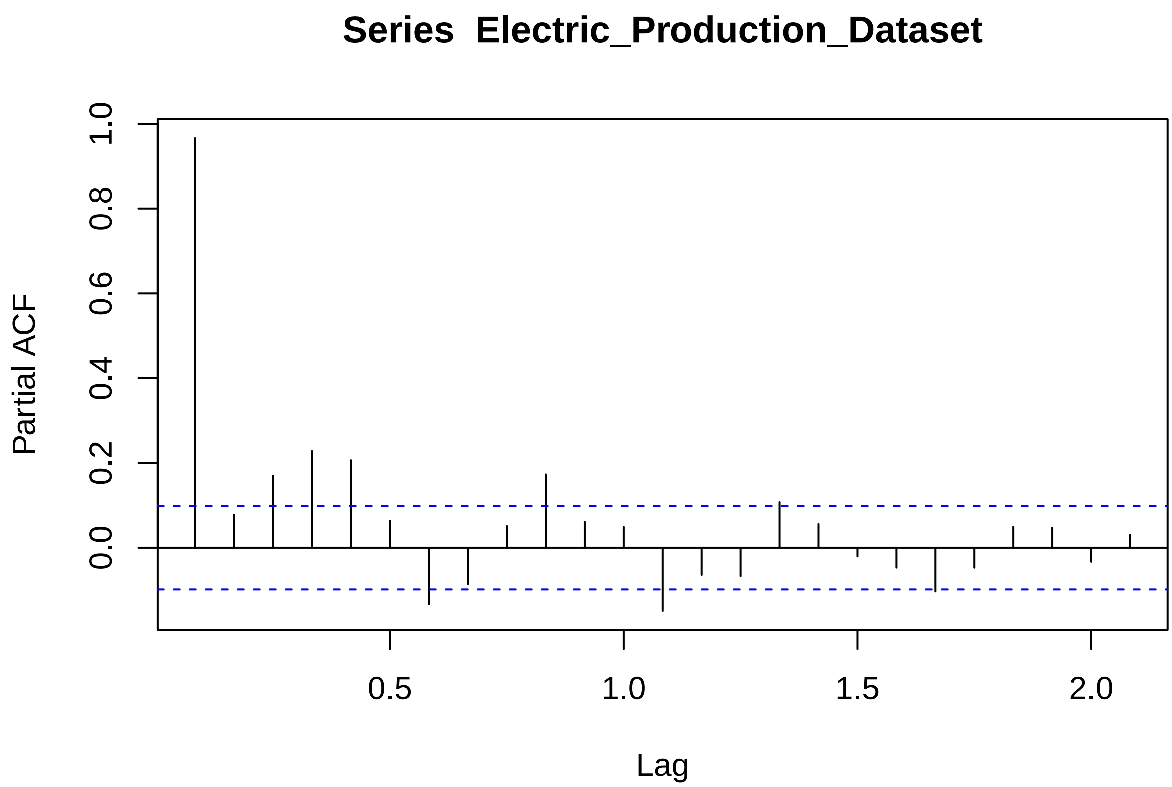

```r
Electric_Production_Dataset=temp1


adf.test(Electric_Production_Dataset)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  Electric_Production_Dataset
## Dickey-Fuller = -3.1667, Lag order = 7, p-value = 0.09376
## alternative hypothesis: stationary
```

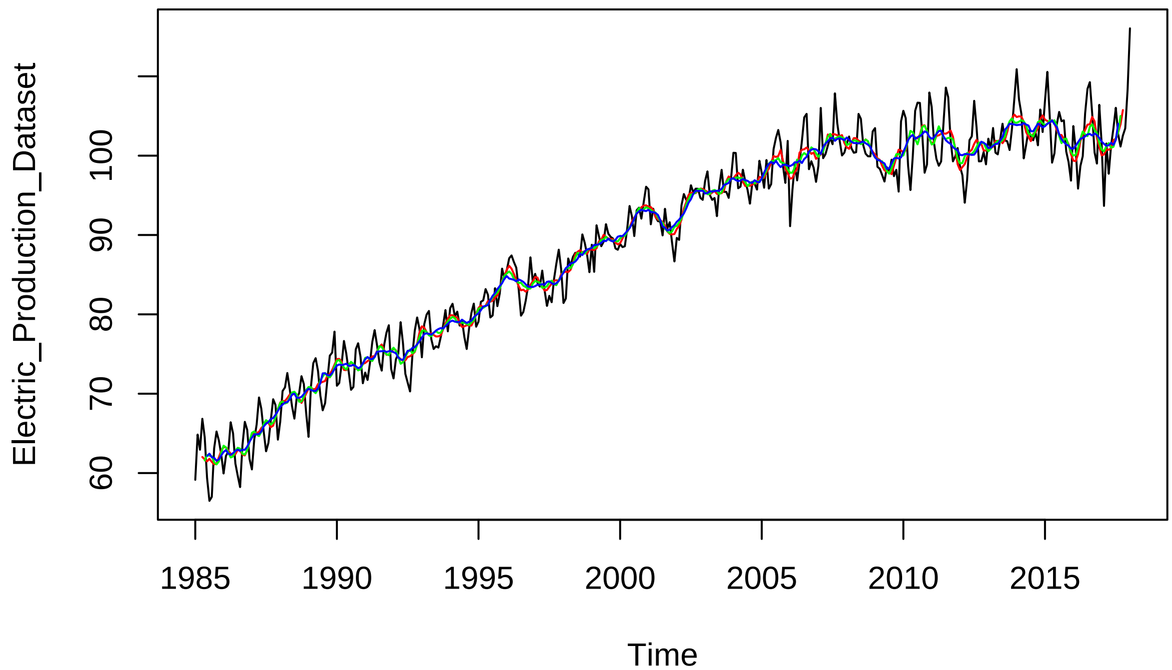
```r
#adf test says that the Series is Stationary so Plotting ACF and PACF
acf(Electric_Production_Dataset)
```



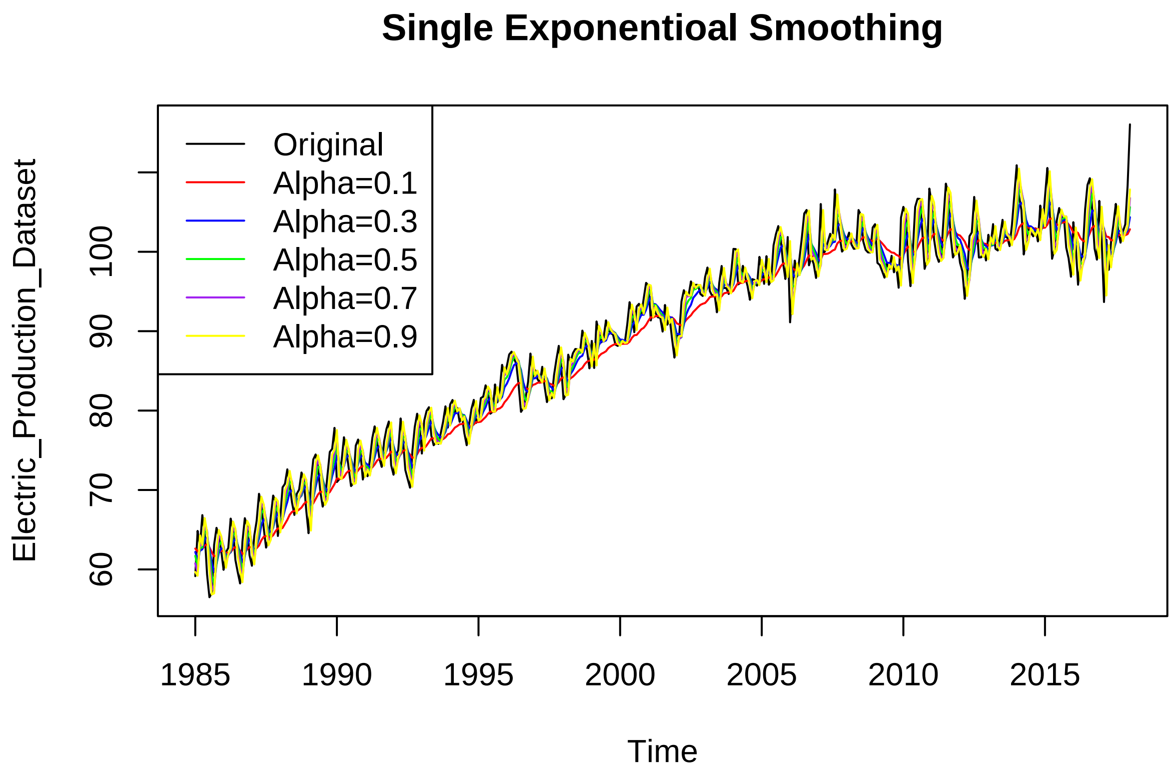

```
pacf(Electric_Production_Dataset)
```




```
#Dataset is Stationarized

# Plotting Simple Moving Average with 7,9,11 intervals
plot(Electric_Production_Dataset)
sm1<-ma(Electric_Production_Dataset,order=7)
lines(sm1,col="red")
sm2<-ma(Electric_Production_Dataset,order=9)
lines(sm2,col="green")
sm3<-ma(Electric_Production_Dataset,order=11)
lines(sm3,col="blue")
```



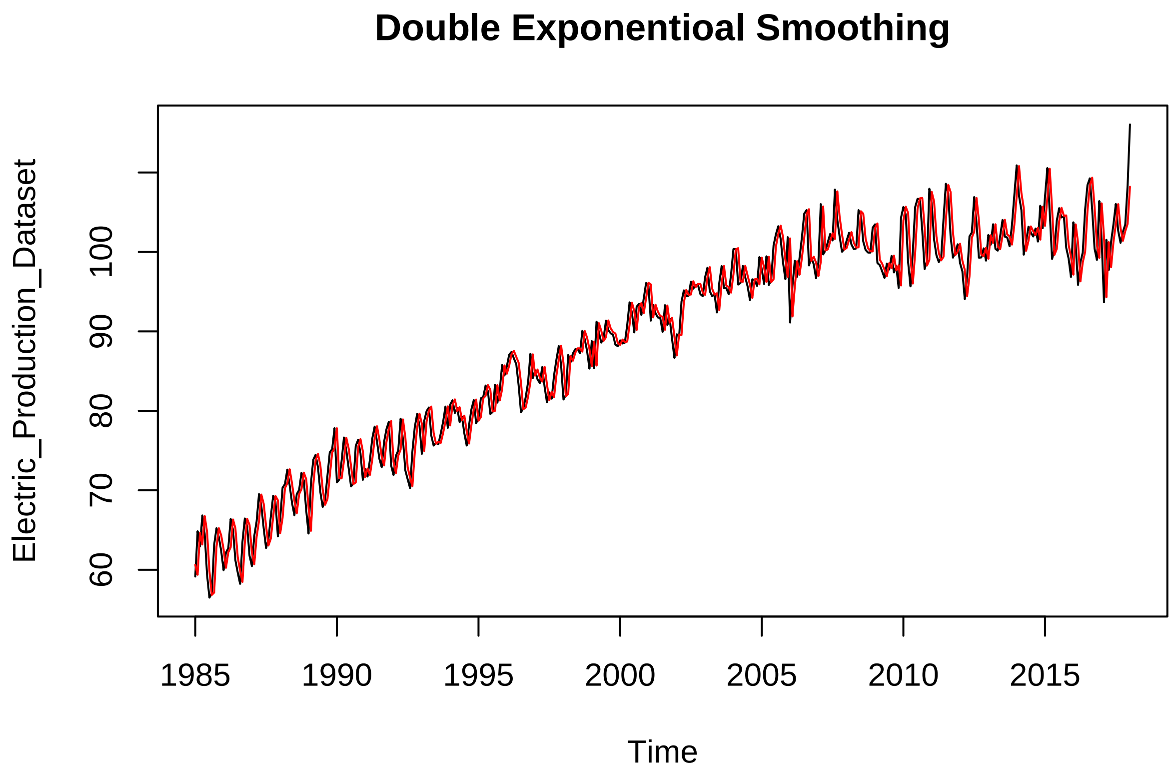

```
#applying single exponential smothing with Different values of alpha
plot(Electric_Production_Dataset,main="Single Exponentioal Smoothing")
legend("topleft",c("Original","Alpha=0.1","Alpha=0.3","Alpha=0.5","Alpha=0.7","Alpha=0.9"),lty=c(1,1,1,
model1=ets(Electric_Production_Dataset,model="ANN",alpha = 0.1)
lines(model1$fitted,col="red")
model2=ets(Electric_Production_Dataset,model="ANN",alpha = 0.3)
lines(model2$fitted,col="blue")
model3=ets(Electric_Production_Dataset,model="ANN",alpha = 0.5)
lines(model3$fitted,col="green")
model4=ets(Electric_Production_Dataset,model="ANN",alpha = 0.7)
lines(model4$fitted,col="purple")
model5=ets(Electric_Production_Dataset,model="ANN",alpha = 0.9)
lines(model5$fitted,col="yellow")
```




```
forecast(model5,11)
```

```
##          Point Forecast    Lo 80    Hi 80     Lo 95    Hi 95
## Feb 2018       115.2265 111.5321 118.9208 109.57645 120.8765
## Mar 2018       115.2265 110.2562 120.1967 107.62515 122.8278
## Apr 2018       115.2265 109.2466 121.2063 106.08112 124.3718
## May 2018       115.2265 108.3844 122.0685 104.76250 125.6904
## Jun 2018       115.2265 107.6193 122.8336 103.59238 126.8605
## Jul 2018       115.2265 106.9245 123.5284 102.52965 127.9233
## Aug 2018       115.2265 106.2834 124.1695 101.54924 128.9037
## Sep 2018       115.2265 105.6853 124.7676 100.63456 129.8183
## Oct 2018       115.2265 105.1226 125.3303  99.77393 130.6790
## Nov 2018       115.2265 104.5896 125.8633  98.95876 131.4941
## Dec 2018       115.2265 104.0820 126.3709  98.18254 132.2704
```

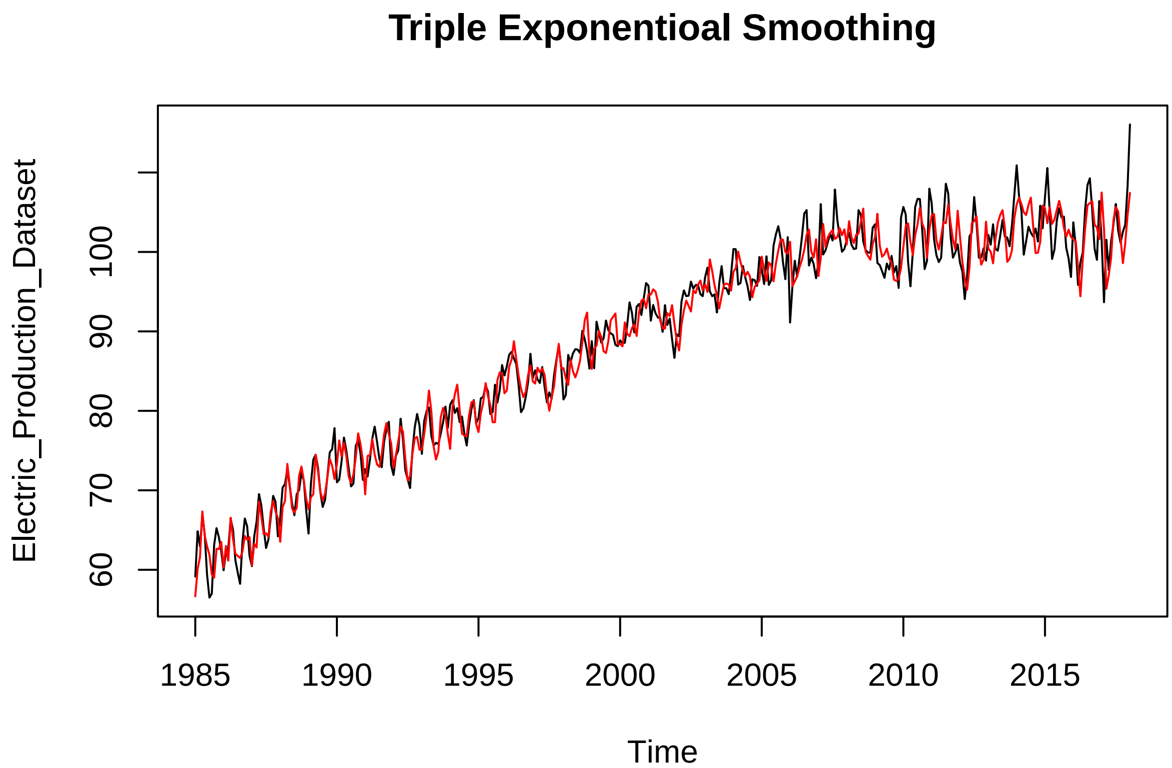
```
#applying Double exponential smoothing with default values
plot(Electric_Production_Dataset,main="Double Exponentioal Smoothing")

model=ets(Electric_Production_Dataset,model = "AAN")
lines(model$fitted,col="red")
```




```r
forecast(model,11)
```

```
##          Point Forecast    Lo 80    Hi 80     Lo 95    Hi 95
## Feb 2018       115.7194 112.0214 119.4173 110.06385 121.3749
## Mar 2018       115.8586 110.7815 120.9356 108.09390 123.6232
## Apr 2018       115.9977 109.8430 122.1524 106.58494 125.4105
## May 2018       116.1369 109.0668 123.2071 105.32407 126.9498
## Jun 2018       116.2761 108.3960 124.1562 104.22451 128.3277
## Jul 2018       116.4153 107.8009 125.0297 103.24069 129.5899
## Aug 2018       116.5545 107.2635 125.8454 102.34514 130.7638
## Sep 2018       116.6936 106.7720 126.6153 101.51979 131.8675
## Oct 2018       116.8328 106.3181 127.3475 100.75199 132.9136
## Nov 2018       116.9720 105.8959 128.0481 100.03252 133.9115
## Dec 2018       117.1112 105.5006 128.7217  99.35434 134.8680
```

```r
#applying Triple exponential smothing with default values
plot(Electric_Production_Dataset,main="Triple Exponentioal Smoothing")

model=ets(Electric_Production_Dataset,model = "AAA")
lines(model$fitted,col="red")
```




```r
forecast(model,11)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Feb 2018       108.7064 105.3132 112.0996 103.5170 113.8958
## Mar 2018       109.0455 105.3226 112.7683 103.3519 114.7391
## Apr 2018       106.6135 102.5877 110.6392 100.4566 112.7703
## May 2018       108.8240 104.5165 113.1316 102.2362 115.4118
## Jun 2018       112.0553 107.4832 116.6274 105.0628 119.0477
## Jul 2018       114.1676 109.3453 118.9899 106.7926 121.5426
## Aug 2018       112.7656 107.7054 117.8258 105.0267 120.5045
## Sep 2018       110.9931 105.7055 116.2806 102.9064 119.0797
## Oct 2018       109.1904 103.6848 114.6960 100.7703 117.6105
## Nov 2018       109.5343 103.8189 115.2497 100.7933 118.2753
## Dec 2018       112.3817 106.4637 118.2996 103.3310 121.4323
```