Time Series Analysis & Forecasting

18BCE1104 - Ankita Duraphe

11/02/2021

### Forecasting on sample data

highest Petrol price in every month from 2018 to yesterday

data\_vector = c(75.63, 76.12, 76.29, 77.47, 81.47, 81.32, 79.8, 81.63, 86.8, 87.39, 82.51, 75.34,74.03, 74.43, 75.65, 75.96, 75.9, 74.39, 76.29, 75.65, 77.34, 77.56, 77.763, 78.1, 79.04, 76.01, 74.79, 72.37, 75.78, 83.63, 83.67, 85.05, 85.09, 84.19, 85.31, 86.51, 88.8, 89.68)

converting vector to dataset

data\_forecasting = ts(data\_vector, start=c(2018,1), end=c(2021,2), frequency=12)

importance of splitting automatically

start(data\_forecasting)

## [1] 2018 1

end(data\_forecasting)

## [1] 2021 2

frequency(data\_forecasting)

## [1] 12

cycle(data\_forecasting)

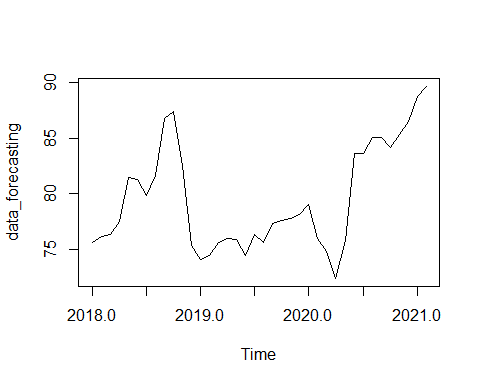
## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec  
## 2018 1 2 3 4 5 6 7 8 9 10 11 12  
## 2019 1 2 3 4 5 6 7 8 9 10 11 12  
## 2020 1 2 3 4 5 6 7 8 9 10 11 12  
## 2021 1 2

summary(data\_forecasting)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 72.37 75.81 77.66 79.60 83.66 89.68

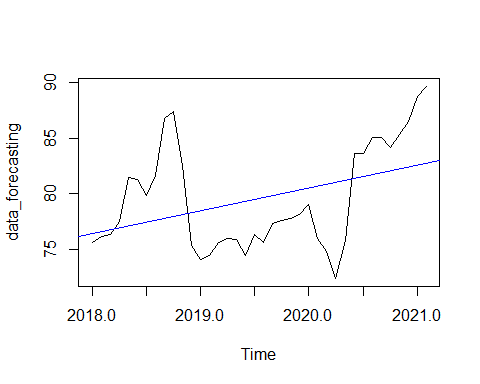
Analysis PART: plotting the series

plot(data\_forecasting)



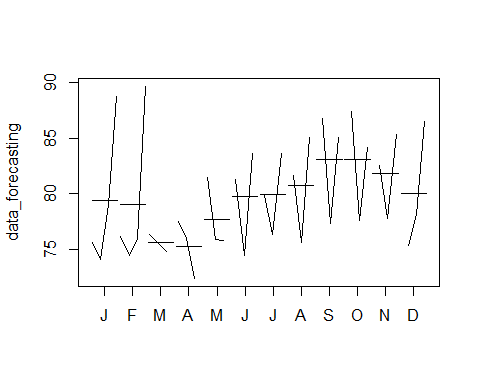
regression

plot(data\_forecasting)  
abline(reg=lm(data\_forecasting~time(data\_forecasting)), col="blue")



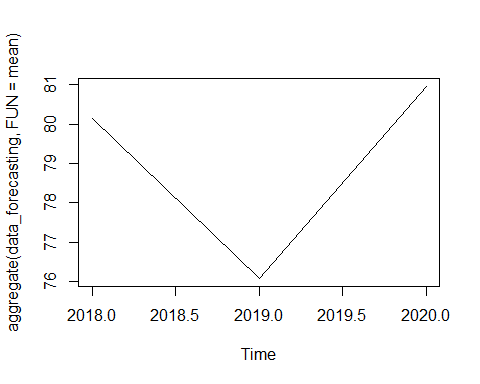
month-wise (range and mean) plot

monthplot(data\_forecasting)



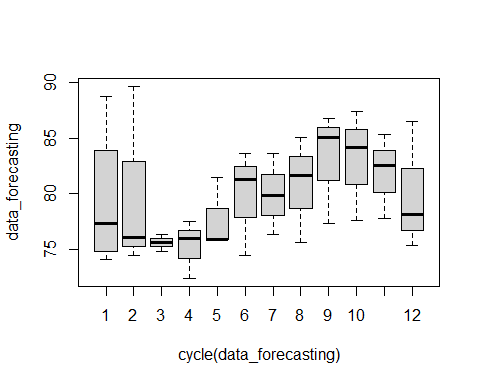
year-wise mean plot

plot(aggregate(data\_forecasting,FUN=mean))



month-wise box plot

boxplot(data\_forecasting~cycle(data\_forecasting))

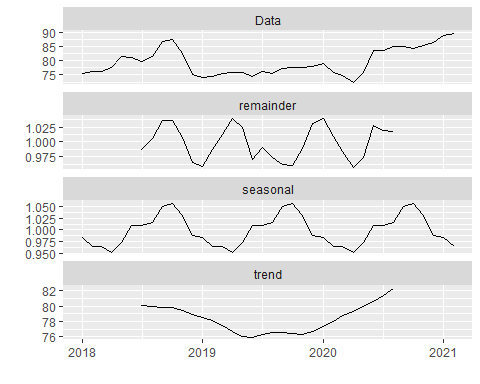


Load the required packages

library(fabletools)  
library(ggplot2)  
library(forecast)  
library(ggfortify)

visualisation

decompose\_fc = decompose(data\_forecasting,"multiplicative")  
autoplot(decompose\_fc)

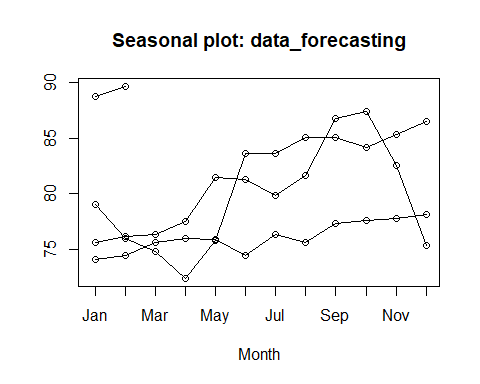


FORECASTING / PREDICTING

library(forecast)

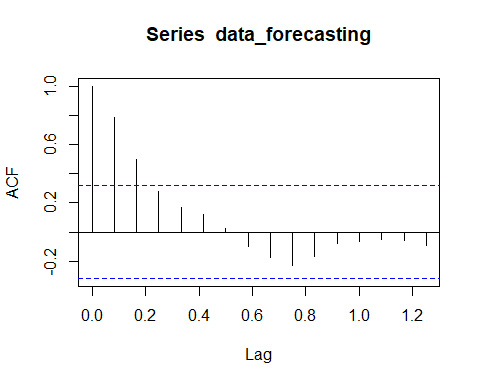
Monthwise predicting

seasonplot(data\_forecasting)

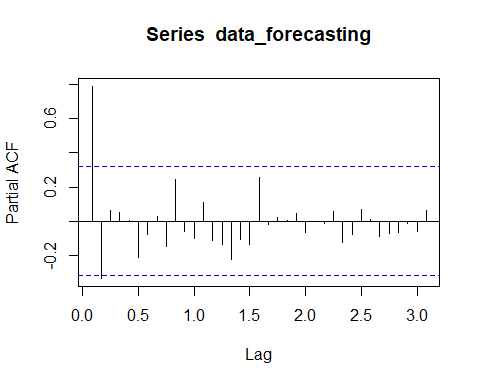


Autocorrelation and Partial Autocorrelation plots

acf(data\_forecasting) #for MA - Q



pacf(data\_forecasting, lag=length(data\_forecasting),pl=TRUE) # for AR - P



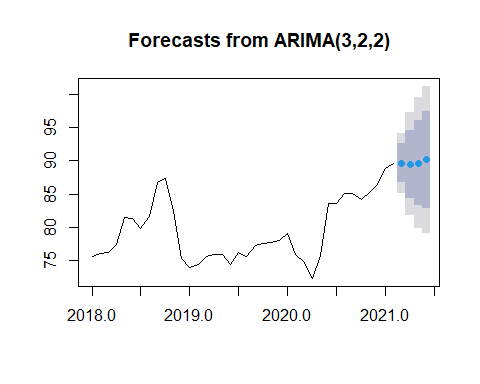
ARIMA

AR (P - autoregressive lags) + I (d - order of differentiation) + MA (Q - moving average)  
correlation between previous time period with current  
noise | averaging / smoothing

fit\_arima = arima(data\_forecasting, order=c(3, 2, 2))  
accuracy(fit\_arima)

## ME RMSE MAE MPE MAPE MASE  
## Training set 0.01807052 2.197007 1.632346 -0.01379927 2.039507 0.9485911  
## ACF1  
## Training set -0.006140107

newdata\_arima = forecast(fit\_arima, 4)  
plot(newdata\_arima)



**MFE: Mean Forecast Error or ME: Mean Error (for forecast)**

From accuracy(fit\_arima), MFE or ME = 0.01807052

**MAD: Mean Absolute Deviation or MAE: Mean Absolute Error**

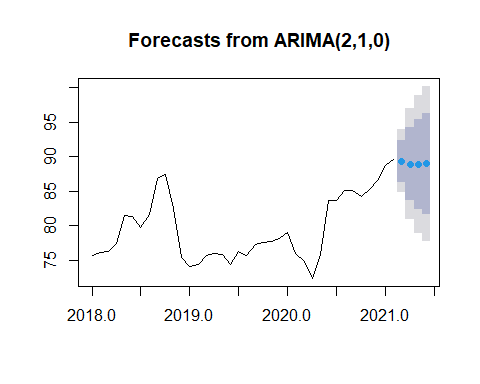
From accuracy(fit\_arima), MAD or MAE = 1.632346

Auto ARIMA

fit\_autoarima = auto.arima(data\_forecasting)  
accuracy(fit\_autoarima)

## ME RMSE MAE MPE MAPE MASE  
## Training set 0.3113201 2.251578 1.679884 0.3445943 2.103429 0.3008596  
## ACF1  
## Training set -0.05613608

newdata\_autoarima = forecast(fit\_autoarima, 4)  
plot(newdata\_autoarima)



**MFE: Mean Forecast Error or ME: Mean Error (for forecast)**

From accuracy(fit\_autoarima), MFE or ME = 0.3113201

**MAD: Mean Absolute Deviation or MAE: Mean Absolute Error**

From accuracy(fit\_autoarima), MAD or MAE = 1.679884

### Forecasting for Weather History dataset

Load the required packages

library(ggplot2)  
library(dplyr)

Read the dataset

data <- read.csv("weatherHistory.csv")

View the dataset

View(data)

Converting To Time Series object using ts() function: For Weekly data

weather\_forecasting = ts(data$Humidity, start=c(2006,1,4), end=c(2016,9,9), frequency=7)

importance of spliting automatically

start(weather\_forecasting)

## [1] 2006 1

end(weather\_forecasting)

## [1] 2017 2

frequency(weather\_forecasting)

## [1] 7

cycle(weather\_forecasting)

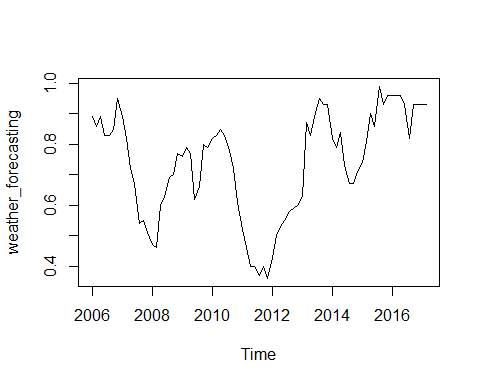
## Time Series:  
## Start = c(2006, 1)   
## End = c(2017, 2)   
## Frequency = 7   
## [1] 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3  
## [39] 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6  
## [77] 7 1 2

summary(weather\_forecasting)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.3600 0.6050 0.7900 0.7372 0.8800 0.9900

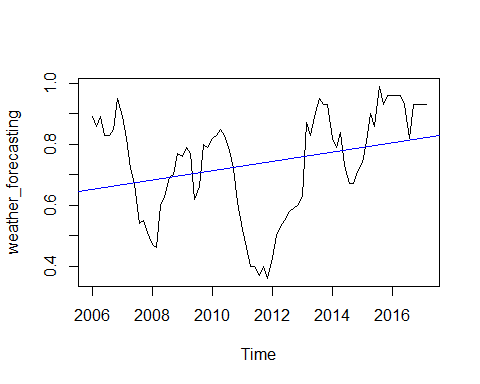
Analysis PART: plotting the series

plot(weather\_forecasting)



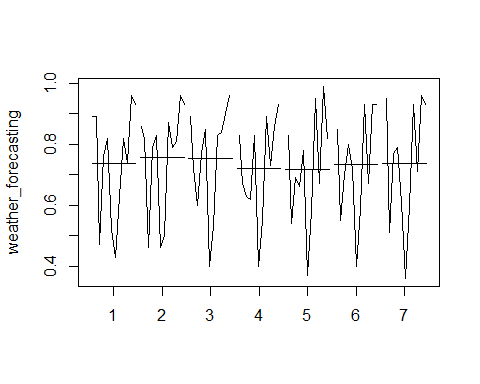
regression

plot(weather\_forecasting)  
abline(reg=lm(weather\_forecasting~time(weather\_forecasting)), col="blue")



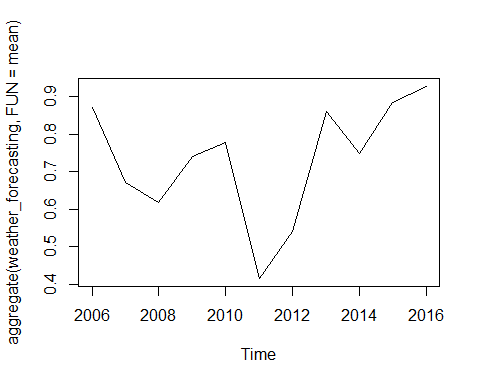
month-wise (range and mean) plot

monthplot(weather\_forecasting)



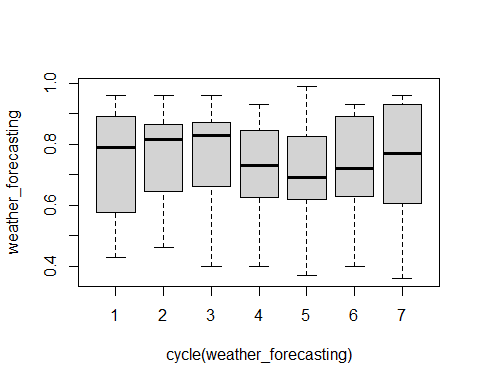
year-wise mean plot

plot(aggregate(weather\_forecasting,FUN=mean))



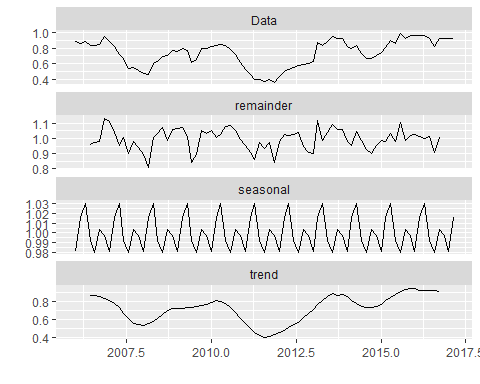
month-wise box plot

boxplot(weather\_forecasting~cycle(weather\_forecasting))



visualisation

decompose\_fc = decompose(weather\_forecasting,"multiplicative")  
autoplot(decompose\_fc)

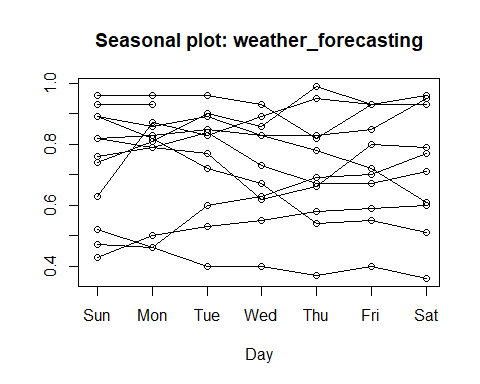


FORECASTING / PREDICTING

library(forecast)

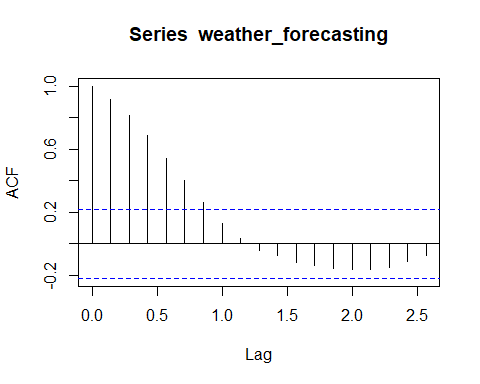
Monthwise predicting

seasonplot(weather\_forecasting)

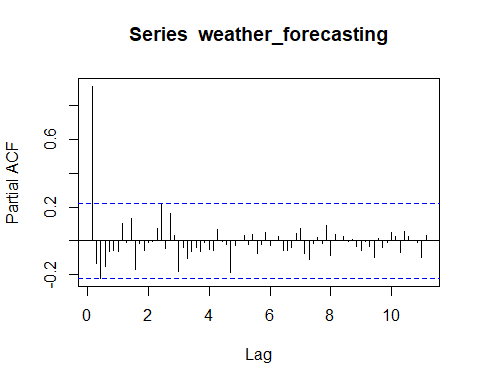


Autocorrelation and Partial Autocorrelation plots

acf(weather\_forecasting) #for MA - Q



pacf(weather\_forecasting, lag=length(weather\_forecasting),pl=TRUE) # for AR - P



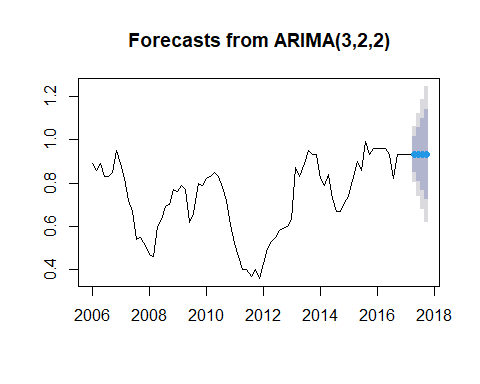
ARIMA

AR (P - autoregressive lags) + I (d - order of differentiation) + MA (Q - moving average)  
correlation between previous time period with current  
noise | averaging / smoothing

fit\_arima = arima(weather\_forecasting, order=c(3, 2, 2))  
accuracy(fit\_arima)

## ME RMSE MAE MPE MAPE MASE  
## Training set 0.00494041 0.06445767 0.04667016 0.7026935 6.57561 0.9430757  
## ACF1  
## Training set -0.0154765

newdata\_arima = forecast(fit\_arima, 4)  
plot(newdata\_arima)



**MFE: Mean Forecast Error or ME: Mean Error (for forecast)**

From accuracy(fit\_arima), MFE or ME = 0.00494041

**MAD: Mean Absolute Deviation or MAE: Mean Absolute Error**

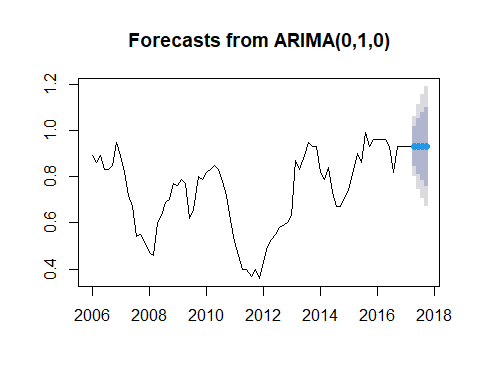
From accuracy(fit\_arima), MAD or MAE = 0.04667016

Auto ARIMA

fit\_autoarima = auto.arima(weather\_forecasting)  
accuracy(fit\_autoarima)

## ME RMSE MAE MPE MAPE MASE  
## Training set 0.0005175949 0.06606487 0.04887203 -0.3998971 7.069131 0.2600729  
## ACF1  
## Training set 0.1100771

newdata\_autoarima = forecast(fit\_autoarima, 4)  
plot(newdata\_autoarima)



**MFE: Mean Forecast Error or ME: Mean Error (for forecast)**

From accuracy(fit\_autoarima), MFE or ME = 0.0005175949

**MAD: Mean Absolute Deviation or MAE: Mean Absolute Error**

From accuracy(fit\_autoarima), MAD or MAE = 0.04887203

**Conclusion:**  
Forecasting, including MFE & MAD, has been successfully performed on sample data as well as on weatherHistory dataset.