**Step 1**: Load all the required libraries

library(tidyverse)  
library(lubridate)  
library(forecast)  
library(tseries)

**Step 2:** Take input of the data and use glimpse() to have a look on it

pricedata <- read\_csv("foodprice.csv")

glipmse(pricedata)

**Output:**

Rows: 5,105  
Columns: 17  
$ date <chr> "15-12-2006", "15-01-2007", "15-02-2007", "15-03-2007", "...  
$ cmname <chr> "Wheat flour - Retail", "Wheat flour - Retail", "Wheat fl...  
$ unit <chr> "KG", "KG", "KG", "KG", "KG", "KG", "KG", "KG", "KG", "KG...  
$ category <chr> "cereals and tubers", "cereals and tubers", "cereals and ...  
$ price <dbl> 23.0, 25.5, 25.5, 26.0, 26.0, 26.5, 26.5, 27.5, 29.5, 30....  
$ currency <chr> "BDT", "BDT", "BDT", "BDT", "BDT", "BDT", "BDT", "BDT", "...  
$ country <chr> "Bangladesh", "Bangladesh", "Bangladesh", "Bangladesh", "...  
$ station <chr> "Barisal", "Barisal", "Barisal", "Barisal", "Barisal", "B...  
$ stationID <dbl> 575, 575, 575, 575, 575, 575, 575, 575, 575, 575, 575, 57...  
$ mktname <chr> "Barisal Division", "Barisal Division", "Barisal Division...  
$ mktid <dbl> 112, 112, 112, 112, 112, 112, 112, 112, 112, 112, 112, 11...  
$ cmid <dbl> 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 5...  
$ ptid <dbl> 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 1...  
$ umid <dbl> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, ...  
$ catid <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...  
$ sn <chr> "112\_58\_15\_5", "112\_58\_15\_5", "112\_58\_15\_5", "112\_58\_15\_5...  
$ default <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...

**Step 3:** Take a look on the “date” variable. We need to convert this into date and time. And then use class() to check the data type of date.

pricedata <- pricedata %>%  
 mutate(Date = dmy(date)) %>%  
 select(-date)  
class(pricedata$Date)

**Output:**

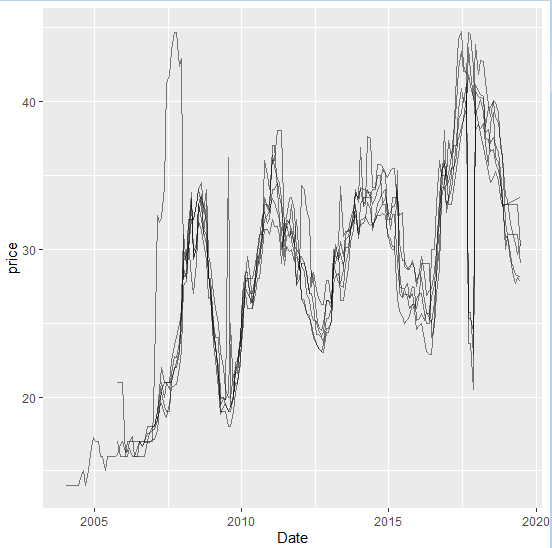
> class(pricedata$Date)  
[1] "Date"

**Step 4:** For this code, We will forecast on price of rice(coarse) only. Filter on “cmname” variable to get the information about rice only and store it in rice\_ts. ggplot() is used to plot the result.

rice\_price <- filter(pricedata, cmname == "Rice (coarse) - Retail")

ggplot(rice\_price, aes(x = Date, y = price, group = station)) + geom\_line(alpha = 0.5)

**Output:**

Price of rice in many different stations

**Step 5:** Because the price differ from station to station, we took the median price of each months and summarize the data, then converted the summarized data into time series using ts().

summary\_rice\_price <- rice\_price %>%  
 group\_by(Date)%>%  
 summarize(median\_price = median(price))

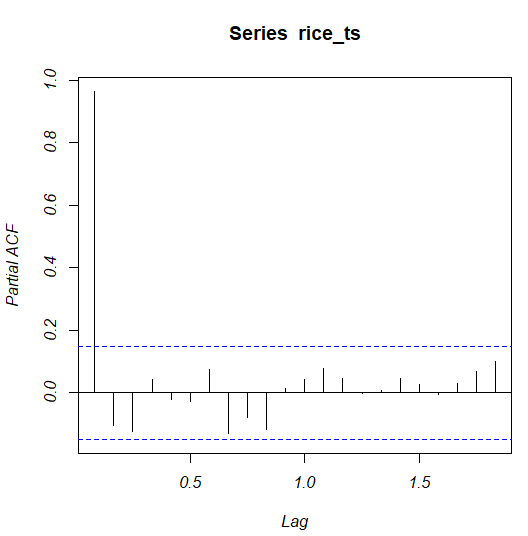
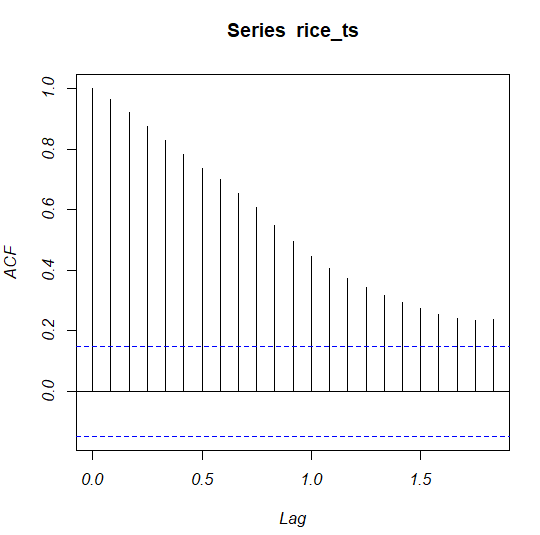
max\_Date = max(summary\_rice\_price$Date)  
min\_Date = min(summary\_rice\_price$Date)

rice\_ts <- ts(summary\_rice\_price$median\_price, end=c(year(max\_Date), month(max\_Date)),start=c(year(min\_Date), month(min\_Date)),frequency=12)

**Step 6:** Check the ACF and PACF plots to decide whether or not to go with ARIMA model.

acf <- acf(rice\_ts)  
pacf <- pacf(rice\_ts)

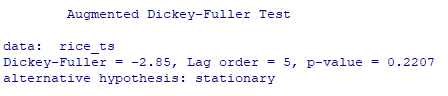
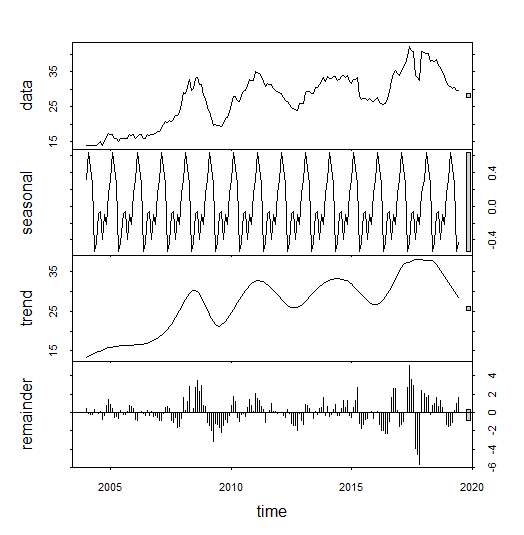
**Output:** As both of the figures shows a geometric pattern, ARIMA is a suitable model to predict price.

ACF and PACF graphs for rice\_ts

**Step 8:** Check if the model starionary using stl(). And to be more sure, perform Augmented Dickey-Fuller test on rice\_ts using adf.test().

plot(stl(rice\_ts,s.window = "periodic"))  
adf.test(rice\_ts)

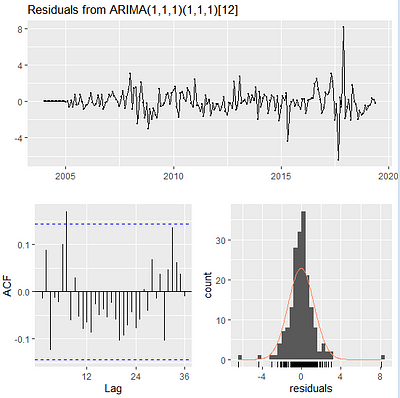
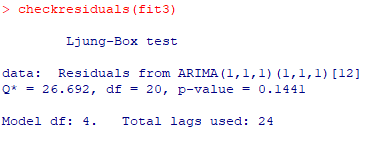
**Output:** The bar on the right side of the component “seasonal” confirms the seasonality of the time series. Also the p-value of ADF test rejects the possibility of the time series to be stationary.



**Step 9:** Fit the model using auto.arima() and set seasonal = TRUE . Or manually by tring out Arima() with different order and season. Note that, for tring out manully, we’ll need to choose the model with lowest AIC AICc and BIC value. Use checkresiduals() to confirm the fitted model does not contain lack of fit.

fit3 <- Arima(rice\_ts, order = c(1,1,3), seasonal = c(1, 1, 1))  
checkresiduals(fit3)

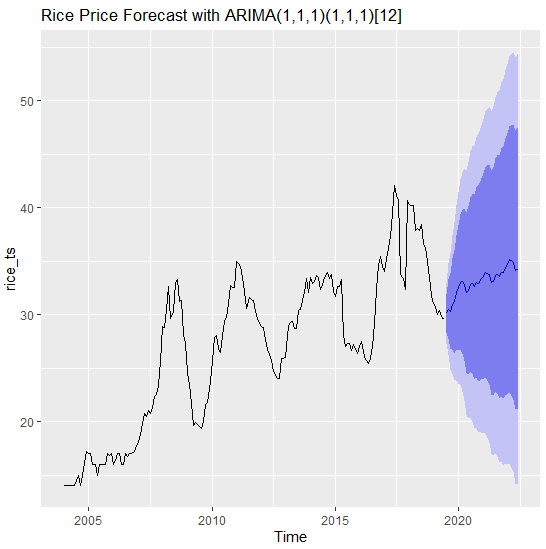
**Output:** The fitted model rejects lac of fit that if confirmed by Ljung-Box test having a p-value well above 0.05 and also confirms a binomial distribution on residuals.

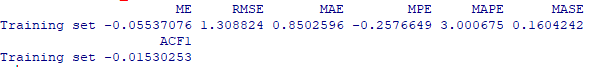
Check if the model is good enough for forecasting

**Step 10:** Use forecast() on the fitted model and use autoplot() to viewthe result. Also use accuracy() to check the accuracy of the forecast.

rice\_forecast <- forecast(fit3, h = 36)  
autoplot(rice\_forecast, main = "Rice Price Forecast with ARIMA(1,1,1)(1,1,1)[12]")  
accuracy(rice\_forecast)

**Output:**





This is a simple way to forecast a time series data in R.