Price Forecast

Import dataset

#IMPORT DATASET  
library(readr)  
food\_prices\_large\_Nig <- read\_csv("C:/Users/Elite/Downloads/food\_prices/food\_prices/food\_prices\_large Nig.csv")

## Rows: 14 Columns: 12  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (8): Country, State/Province, City\_Market, Product, Currency\_name, Buyer...  
## dbl (4): Id, Month, Year, Price  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

View(food\_prices\_large\_Nig)  
attach(food\_prices\_large\_Nig)

## 

Load Library

#Load library  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.1 ✔ purrr 1.0.1  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(forecast)

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

library(TSA)

## Registered S3 methods overwritten by 'TSA':  
## method from   
## fitted.Arima forecast  
## plot.Arima forecast  
##   
## Attaching package: 'TSA'  
##   
## The following object is masked from 'package:readr':  
##   
## spec  
##   
## The following objects are masked from 'package:stats':  
##   
## acf, arima  
##   
## The following object is masked from 'package:utils':  
##   
## tar

library(caret)

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

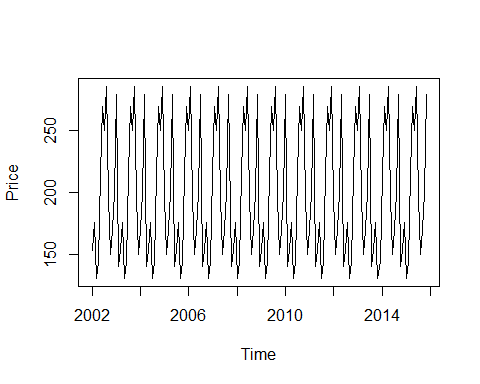
library(ggplot2)

data preparation

#data preparation  
Price <- ts (food\_prices\_large\_Nig$Price,start = c(2002,1),  
 end = c(2015,11),frequency = 12)

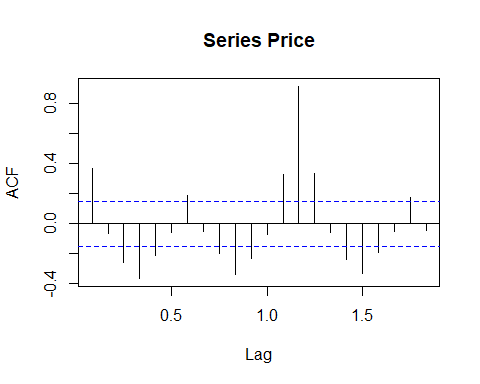
Data exploration

#data exploration  
plot(Price)



correlation

#annual correlation  
acf(Price)



Best way to forecast

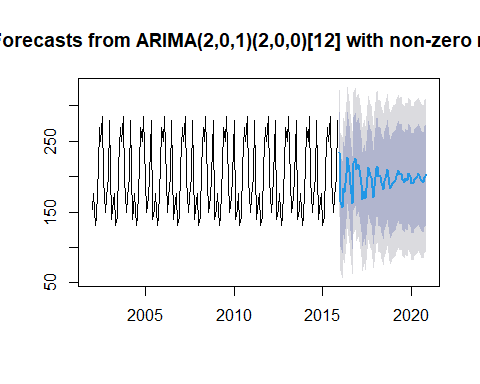
RMSE,MAE OR MAP I.e the model should be as low as possible

#model evaluation for better forecast  
model <- auto.arima(Price)  
accuracy(model)

## ME RMSE MAE MPE MAPE MASE  
## Training set 0.02709125 45.23378 34.78098 -5.107706 18.85601 0.530308  
## ACF1  
## Training set -0.006319977

For the next 60 month

#forecating 60 MONTHS(5YEARS) 12\*5 =60  
forecast\_price <- forecast(model, h = 60)  
plot(forecast\_price)



Mean for the months for the next five years

#VIEW FORCAST FOR MONTH  
forecast\_price$mean

## Jan Feb Mar Apr May Jun Jul Aug  
## 2015   
## 2016 164.9538 155.8821 183.9551 177.4818 204.7646 227.1264 206.9617 189.0618  
## 2017 216.6684 210.3203 188.7624 167.4870 178.7878 169.0469 199.1176 213.1456  
## 2018 211.8199 214.1937 200.6294 199.7959 191.4811 181.7979 193.7424 202.4339  
## 2019 192.0945 194.7843 200.6548 208.3430 202.9376 205.0537 195.8122 191.9717  
## 2020 190.8982 190.4323 196.3002 197.7667 200.0047 203.8805 198.0987 194.3269  
## Sep Oct Nov Dec  
## 2015 233.5129  
## 2016 162.7510 218.4196 225.5380 210.3486  
## 2017 201.8290 199.2125 171.1690 185.6858  
## 2018 210.3893 189.5485 182.7016 190.4713  
## 2019 197.3288 195.1443 204.4107 200.2517  
## 2020 192.2136 199.5383 203.4516

Mean forcast

plot(forecast\_price)  
lines(forecast\_price$mean, col = "PINK")

