PRACTICE

Pranav Gopalkrishna, 1940223

2022-03-22

# QUESTION 1: Simulating & studying ARMA(1, 2) process

## Aim

Generate an ARMA(1,2) process of size 1000. Is the process stationary? Comment on its ACF and PACF plots.

## Introduction to ARMA(1, 2) process

ARMA(1, 2) is the combination of an AR(1) (autoregressive process of order 1) and an MA(2) (moving average of order 2) process, and is given by

The part in yellow is the AR(1) part, and the part in green is the MA(2) part.

## Simulating an ARMA(1, 2) process

z = arima.sim(model = list(ar = c(0.3), ma = c(0.6, 0.7)), 1000)

By design, an ARMA model is a stationary time series model i.e. it is a stationary time series process used as a model for actual data. Hence, data based on this model will likely be stationary as well. To confirm, we perform the augmented Dickey-Fuller test. We can also judge the stationarity based on the ACF and PACF plots.

## Augmented Dickey-Fuller test

This test is used to estimate the truth or hypothe of the hypothesis

library(tseries)

adf.test(z)

*Warning in adf.test(z): p-value smaller than printed p-value*

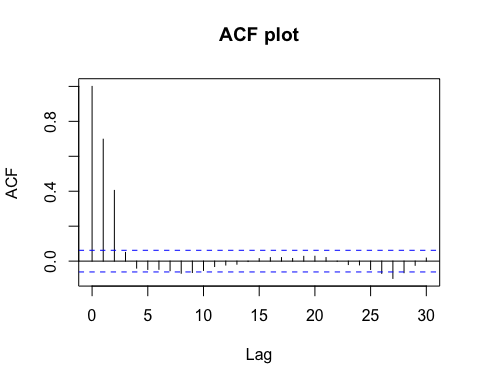
**Augmented Dickey-Fuller Test**

**Data**: z  
***Dickey-Fuller*** = -10.296, ***Lag order*** = 9, ***p-value*** = 0.01  
**Alternative hypothesis**: stationary

As we can see, p-value ≤ 0.01, which indicates that, given a 0.05 level of significance, we may reject the null hypothesis that the data is not stationary. Hence, we may accept the alternative hypothesis that the data is stationary, confirming our original statements.

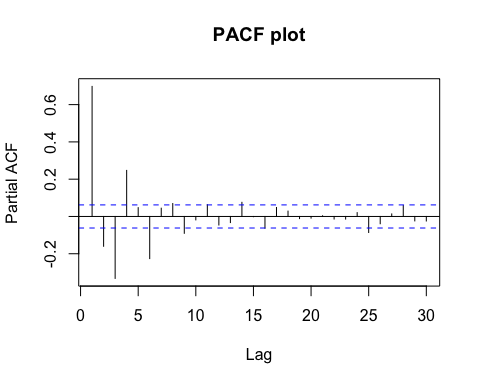
## ACF and PACF plots

acf(z,  
 main = "ACF plot")



From the above plot, we find that past observations are positively significantly correlated to the current observation upto lag 10 (excluding lags 5 and 6). However, the ACF seems to be some damped oscillatory function of the lag, suggesting weak stationarity.

pacf(z,  
 main = "PACF plot")



From the above plot, we find that past observations are significantly correlated to the current observation upto lag 7, when we keep the effect of the in-between lags constant. The signs of the autocorrelation coefficients are not the same for every lag, showing some dampened oscillatory movement.

# QUESTION 2: Applying moving average smoothing & differencing to monthly prices of agricultural products

## Aim

Choose two-time series data sets with non-stationary components and demonstrate the method of moving average, method of differencing, and the method of seasonal differencing to extract the stationary components.

## Dataset

The following data contains

data = read.csv("~/Documents/Study/computerScience/programming/r/data/agriculturalRawMaterial.csv")  
data = data[c(1, 2, 4)]  
head(data)

**Month | Coarse.wool.Price | Copra.Price**  
Apr-90 | 482.34 | 236  
May-90 | 447.26 | 234  
Jun-90 | 440.99 | 216  
Jul-90 | 418.44 | 205  
Aug-90 | 418.44 | 198  
Sep-90 | 412.18 | 196

z1 = data$Coarse.wool.Price  
z2 = data$Copra.Price  
months = data$Month

## Defining and formatting the ‘Month’ variable

***(for better date summary)***

t = c()  
for(x in months)  
{  
 x = paste("01", x, sep = "-")  
 t = c(t, x)  
}  
t = as.Date(t, format = "%d-%b-%y")  
# %b => abbreviated month  
# %y => 2 digit year  
# It can recognize century on its own.  
# So for example, '93' will be interpreted as '1993'.

### Summarising dates

summary(t)

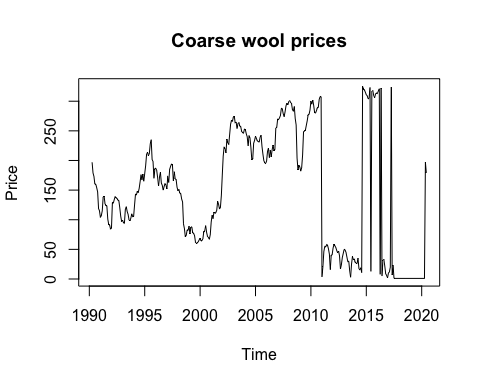
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Minimum** | **1st Quartile** | **Median** | **Mean** | **3rd Quartile** | **Maximum** |
| 1990-04-01 | 1997-10-01 | 2005-04-01 | 2005-04-01 | 2012-10-01 | 2020-04-01 |

## Converting to time series

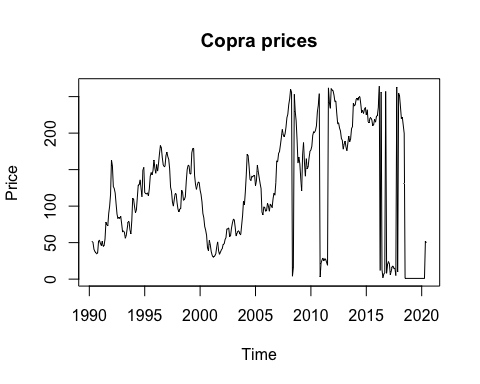
Z1 = ts(z1, start = c(1990, 4, 1), end = c(2020, 6, 1), frequency = 12)  
Z2 = ts(z2, start = c(1990, 4, 1), end = c(2020, 6, 1), frequency = 12)  
# frequency = 12 -> monthly frequency

## Time plots before smoothing or differencing

ts.plot(Z1,  
 main = "Coarse wool prices",  
 ylab = "Price")



ts.plot(Z2,  
 main = "Copra prices",  
 ylab = "Price")



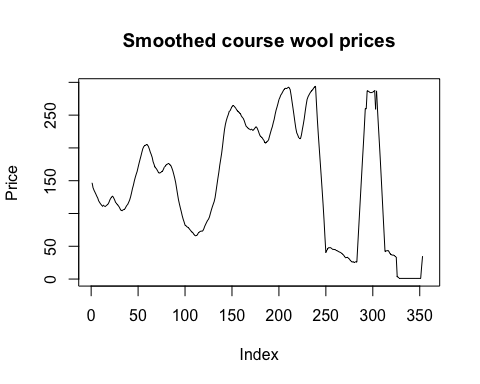
## Moving average smoothing

This is a method of estimating trend at particular time points using the moving average of a fixed number of observations around the particular time points.

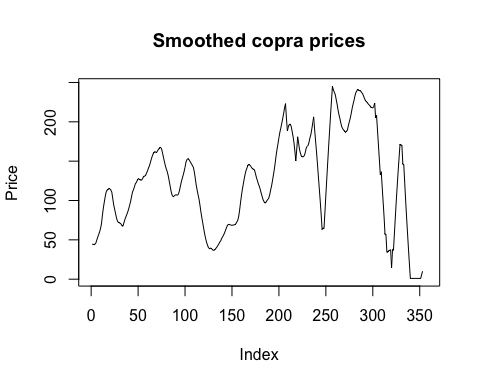
mas = function(ts, order) # mas => moving average smoothing  
{  
 ma = c()  
 # If order is n, that means n = 2d + 1, where  
 # [t-d, t+d] is the interval of time points for which  
 # average is taken.  
 d = (order - 1)/2  
 for(i in c(d+1:length(ts)-d-1))  
 {  
 sum = 0  
 for(j in c(-d:d))  
 {  
 sum = sum + ts[i+j]  
 }  
 ma = c(ma, sum/order)  
 }  
 return(ma)  
}

The order of smoothing must be chosen based on requirement, which often depends on the level of periodic and irregular fluctuations in the data. In the given data, there is a high degree of irregular fluctuations, and hence, I will choose a higher order.

plot(mas(Z1, 11),  
 type = 'l',  
 main = 'Smoothed course wool prices',  
 ylab = 'Price')



plot(mas(Z2, 11),  
 type = 'l',  
 main = 'Smoothed copra prices',  
 ylab = 'Price')

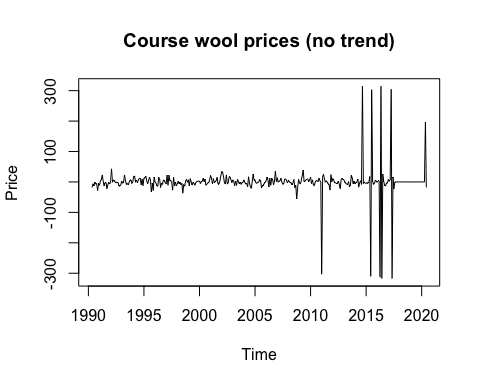


From the above, we can see that both datasets display no clear trend.

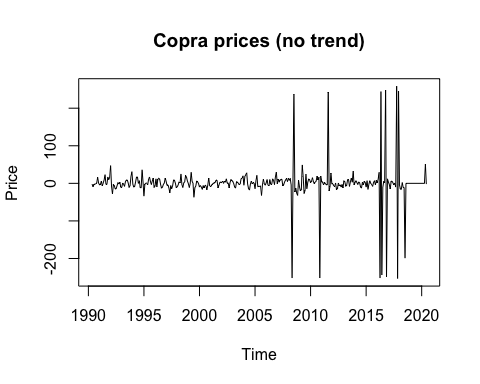
## Method of differencing

### *1st order differencing: (to remove trend)*

Z1\_notrend = diff(Z1)  
ts.plot(Z1\_notrend,  
 main = "Course wool prices (no trend)",  
 ylab = "Price")



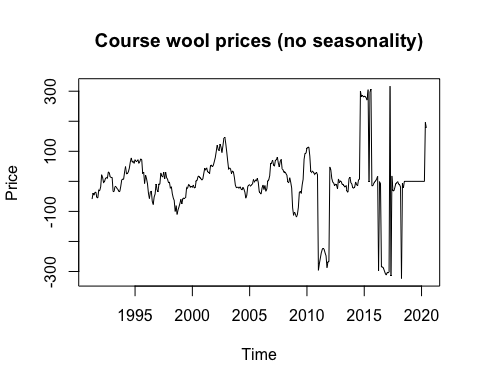
Z2\_notrend = diff(Z2)  
ts.plot(Z2\_notrend,  
 main = "Copra prices (no trend)",  
 ylab = "Price")



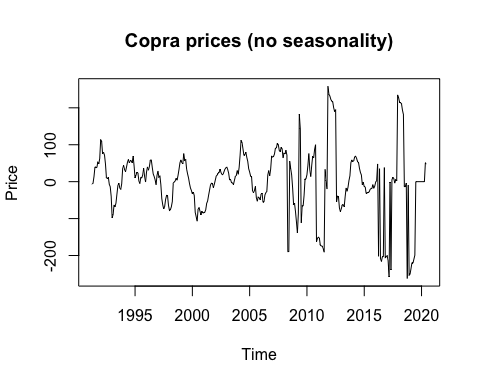
### *Seasonal differencing*

Our data is monthly, so one seasonal period is 12. Hence, we do differencing at a lag of 12, i.e.

Z1\_noseasonality = diff(Z1, 12)  
ts.plot(Z1\_noseasonality,  
 main = "Course wool prices (no seasonality)",  
 ylab = "Price")

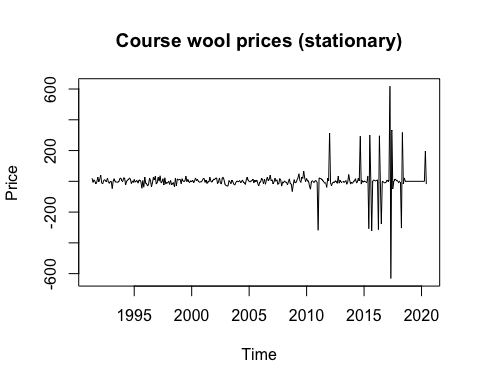


Z2\_noseasonality = diff(Z2, 12)  
ts.plot(Z2\_noseasonality,  
 main = "Copra prices (no seasonality)",  
 ylab = "Price")

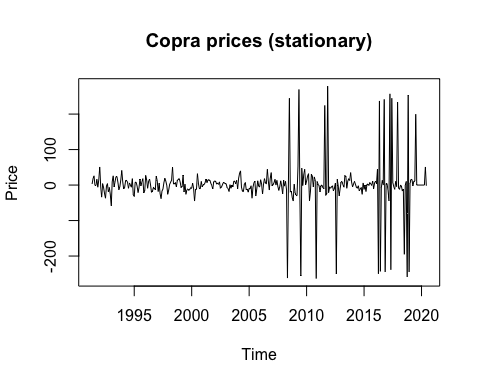


### *Combining both the above differencing methods*

Z1\_stationary = diff(diff(Z1), 12)  
ts.plot(Z1\_stationary,  
 main = "Course wool prices (stationary)",  
 ylab = "Price")



Z2\_stationary = diff(diff(Z2), 12)  
ts.plot(Z2\_stationary,  
 main = "Copra prices (stationary)",  
 ylab = "Price")



From the above time plots, we observe that

* Mean is constant at 0
* Variance is bounded, with occasional random spikes

Furthermore, there is no visible seasonality or cyclic fluctuation in the data, suggesting stationarity. This data is easier to study and forecast upon, compared to the original data.