HW4

Oluwafemi Shobowale

4/27/2022

1,

AR(2)

$$V_t = \phi_o + \phi_{1_{vt-1}} + \phi_{2_{vt-2}} + a_t$$

$$\mu_{vt} = \frac{\phi_o}{1 - \phi_1 - \phi_2}$$

1a, The mean of the time series is 0.167.

```
#r = 0 + 0.9 lag_r +at
phi_0 = 0.10
phi_1=0
phi_2=0.4
mean_r = phi_0/ (1-phi_1 - phi_2)
mean_r

## [1] 0.1666667
```

$$V_t = 0.1 + 0.42_{vt-2} + a_t$$

$$1 - 0B - 0.4B^2 = 0$$

1b, Both absolute roots > 1 , hence the series is STATIONARY

```
# Compute the roots of the characteristic polynomial for v_t= 0.10 + 0.4_vt-2
+at
#
# Its characteristic polynomial is 1 - 0B - .4B^2
polyroot(c(1, 0, -.4))   # Note, two complex roots and they are complex
conjugates!

## [1]  1.581139+0i -1.581139+0i
```

1c,Computing a 1-step and 2-step ahead forcast of AR(2).

1-step = 0.108

2-step = 0.208

$$\hat{V}_t(1) = \phi_o + \phi_{1_{vt}} + \phi_{2_{vt-1}}$$

$$\widehat{V}_t(1) = 0.1 + +0_{vt} + 0.42_{vt-1}$$

$$\widehat{V}_t(2) = 0.1 + +\widehat{V}_t(1)$$

```
###1step ahead
phi_0 = 0.10
phi_1=0
phi_2=0.4
vt= -0.01
vt_1= 0.02
one_step = phi_0 + phi_1*vt + phi_2*vt_1
one_step

## [1] 0.108

###1step ahead
two_step = phi_0 + one_step
two_step

## [1] 0.208
```

1d, lag-1 and lag-2 autocorrelations for an AR(2) process.

lag-1 auto correlation = 0

lag2- autp correlation = 0.4

$$\rho_1 = \frac{\phi_1}{1 - \phi_2}$$

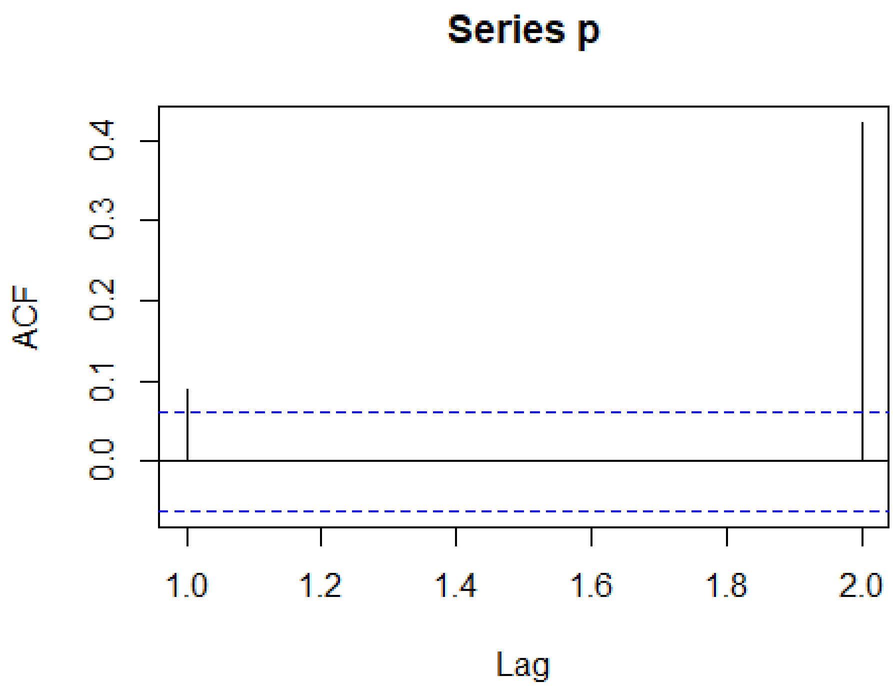
$$\rho_2 = \frac{\phi_1^2 + \phi_2(1 - \phi_2)}{1 - \phi_2}$$

```
rho_1= phi_1/(1-phi_2)
rho_1

## [1] 0

rho_2 = phi_1^2+phi_2*(1-phi_2)/(1-phi_2)
rho_2

## [1] 0.4

############################################################
# Creating an AR(2) process
############################################################

# Create a series of 1000 white-noise with a sd = .1
set.seed(335)
a = rnorm(1000, 0, .14)
#plot(a, pch=16, cex=.2)
```

```r
# Now, we start the array with a starting value, it is the
# initial "seed" value of the series.  Play with this value
# to see how it affects the series.  We also choose a constant
# "theta" for the model
p = rep(0, 1000)
p[1] = 0
p[2] = 0
theta_0= 0.10
theta_1 = 0
theta_2 = 0.4

# Now, we loop through the elements of p one by one and
# apply the formula for the AR(2) process.
#
#            p_i = p_i-1 + a_i
#
# Our random shocks are in the array "a"

for (i in 3:1000)
{
  p[i] = theta_0 + theta_1 * p[i-1] + theta_2 * p[i-2] + a[i]
}
p = ts(p)
```

Both the mean and lag autocorrelation are equal to the theroretical calculation

```
## [1] 0.1688576
```




```
##
## Autocorrelations of series 'p', by lag
##
##     1     2
## 0.090 0.422
```

2 MA(2) PROCESSES

$$V_t = \theta_0 + a_t + \theta_{1_{at-1}} + \theta_{2_{at-2}}$$

$$\mu_{vt} = \theta_0$$

$$\sigma_v^2 = \sigma_a^2(1 + \theta_1^2 + \theta_2^2)$$

2a,

$$V_t = 5 + a_t - 0.5a_{t-1} + 0.25a_{t-2}$$

The mean of series 9 is 5

The variance of the series is 0.0328

```
b=0.5
var_x= 0.025*(1+b^2+0.25^2)
var_x

## [1] 0.0328125
```

2b, MA(2) is stationary beacuse MA models are always stationary.

2c, Forcast 1,2,3 steps of MA(2) model.

$$\widehat{V}_t(1) = 5 - 0.5a_t - 0.25a_{t-1}$$

$$\widehat{V}_t(K) = \mu - - - - - - - - - - - - - - - for - K > 1$$

1-step forcast = 5

For multiple step ahead for k>1 = mean

Hence 2-step and 3-step forcast = mean = 5

```
one_step_ma = 5 - 0.5*-0.01 - 0.25*0.02
one_step_ma

## [1] 5
```

2d, Autocorrelation for MA(2)

for MA(2) a non-zero autocorrlation on occurs at lag_1

phi_1 = -0.5952381

$$\rho_1 = \frac{\phi_1 + \phi_1 * \phi_2}{1 + \phi_1^2 + \phi_2^2}$$

$$\rho_k = 0 - - - - - - - - for - k > 2$$

```
ma_rho_1 = -0.5+(-.5*0.25)/(1+(-0.5^2)+0.25^2)
ma_rho_1

## [1] -0.6538462
```

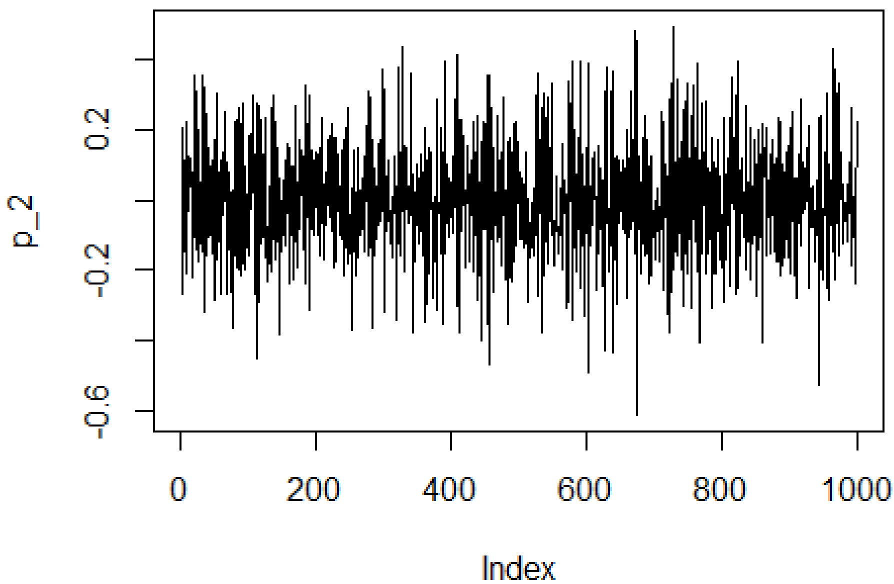
2e,

Since for the simulated result the initail value is zero , this makes the mean different from the theoretical calculation where the inital value is 5 and mean is 5.

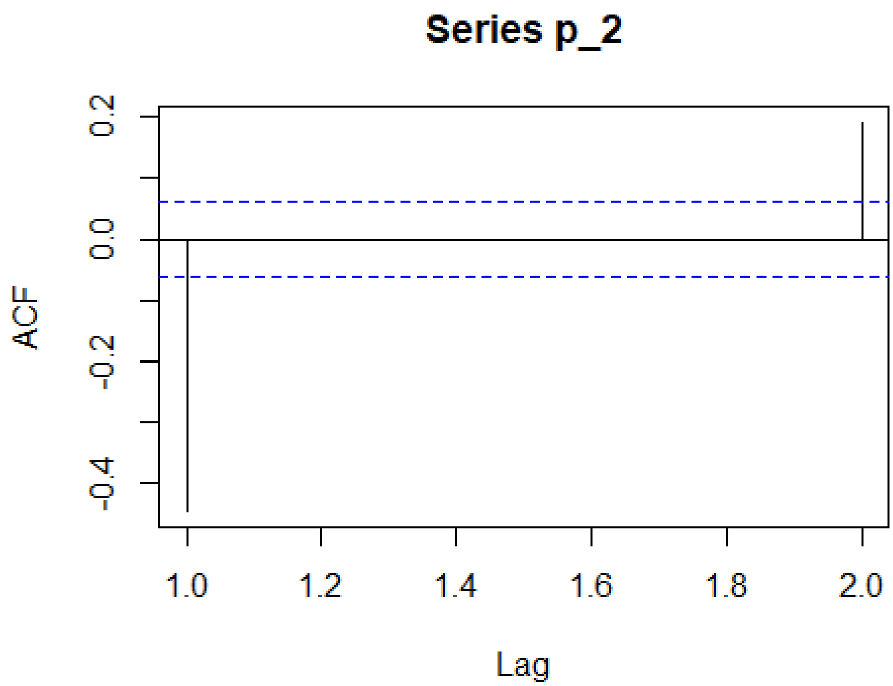
The autocorrelation at lag 1 is slightly different , while the correlation at lag 2 is zero.

```
# Create a series of 1000 white-noise with a sd = .15
ma_a = rnorm(1001, 0, .15)

rw = ts(cumsum(ma_a))
```

```r
# Now, create the MA(2) process
p_2 = ma_a[1:999] - .5 * ma_a[2:1000] + 0.25 * ma_a[3:1000]
plot(p_2, type="l")
```




```r
ac_f=acf(p_2,lag.max = 2)
```




```
ac_f

##
## Autocorrelations of series 'p_2', by lag
##
##      1      2
## -0.445  0.191

mean(p_2)

## [1] -0.0001731993
```

3, EXTRA CREDIT

MA(3) AUTOCORRELATION AT LAG 1 & 2

$$\rho_1 = \frac{\phi_1 + \phi_1 * \phi_2 + \phi_2 * \phi_3}{1 + \phi_1^2 + \phi_2^2 + \phi_3^2}$$

$$\rho_2 = \frac{\phi_2 + \phi_1 * \phi_3}{1 + \phi_1^2 + \phi_2^2 + \phi_3^2}$$

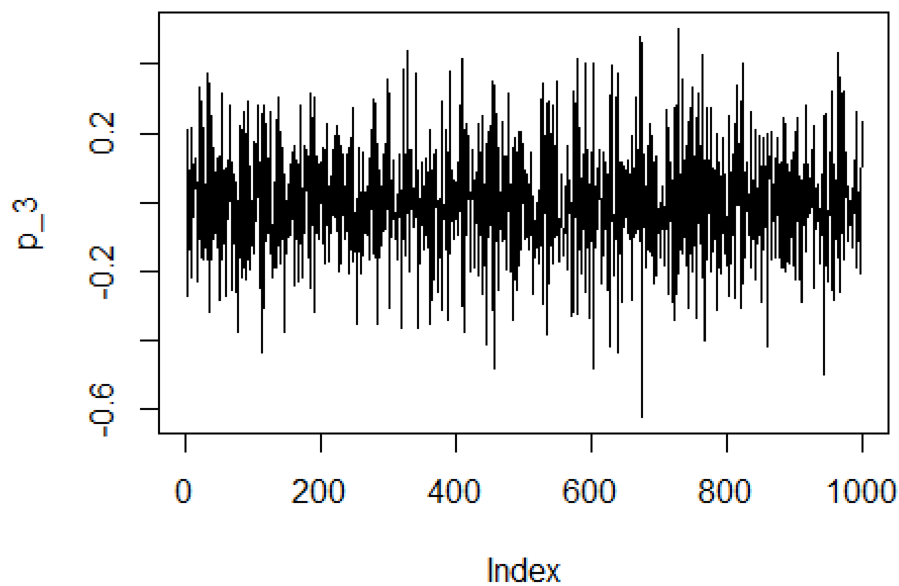
3b , Autocorrelation at lag 1 = -0.594518

```
ma_3_phi = (-.5) + (-.5*0.25) / (1+ (-.5)^2+ (0.25^2) +(-0.1)^2  )
ma_3_phi
```

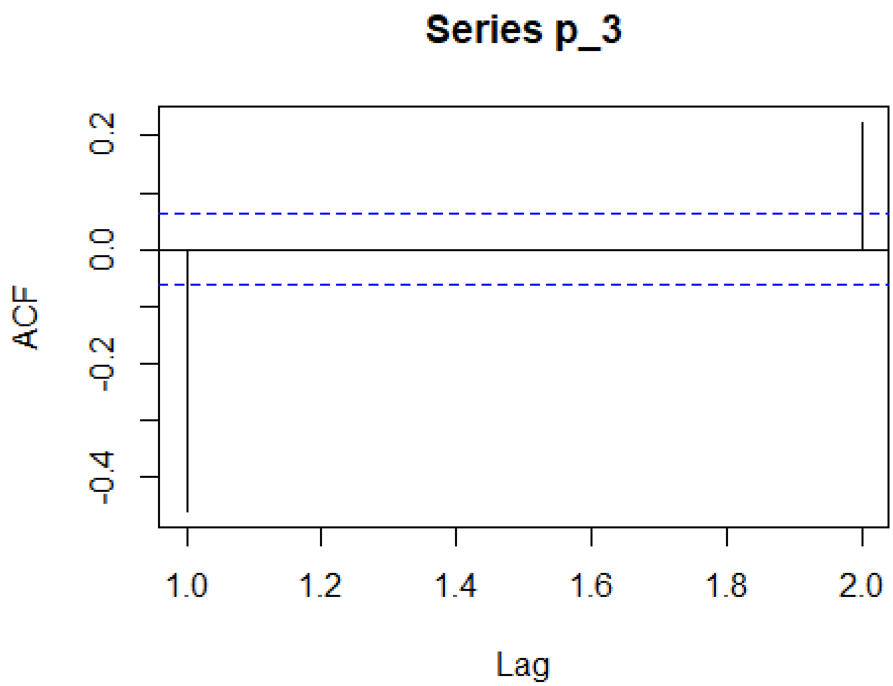
```
## [1] -0.594518
```

3c, The autocorrelation at lag 1 are similar to the auto correlation as calculated.

```
# Create a series of 1000 white-noise with a sd = .15
ma_3_a = rnorm(1001, 0, .15)

rw = ts(cumsum(ma_a))


# Now, create the MA(2) process
p_3 = ma_a[1:999] - .5 * ma_a[2:1000] + 0.25 * ma_a[3:1000]- 0.1
*ma_a[4:1000]
plot(p_3, type="l")
```




```
ac_f3=acf(p_3,lag.max = 2)
```




```
ac_f3

##
## Autocorrelations of series 'p_3', by lag
##
##      1      2
## -0.460  0.223

mean(p_3)

## [1] -0.0001413151
```

4

```
library(readxl)

## Warning: package 'readxl' was built under R version 4.0.5

df1<- read.csv("NAPM.csv")
df1$date = as.Date(df1$date,format = "%m/%d/%y")
df1$date =as.character(df1$date)
df1_ts = ts(df1$index,c(2019,01),frequency=365)
```

4a, Auto arima suggest a Series: df1_ts ARIMA(3,0,2) all coeffienct are highly significant.

```
# Auto.arima
fit2b = auto.arima(df1_ts, seasonal = "F")
fit2b

## Series: df1_ts
## ARIMA(3,0,2) with non-zero mean
##
## Coefficients:
##          ar1      ar2     ar3      ma1     ma2     mean
##       2.1341  -2.0191  0.8263  -1.1207  0.9138  51.2594
## s.e.  0.0520   0.0774  0.0417   0.0334  0.0397   1.3040
##
## sigma^2 = 4.272:  log likelihood = -925.1
## AIC=1864.2   AICc=1864.47   BIC=1892.68

coeftest(fit2b)

##
## z test of coefficients:
##
##            Estimate Std. Error z value  Pr(>|z|)
## ar1        2.134092   0.051954  41.076 < 2.2e-16 ***
## ar2       -2.019108   0.077396 -26.088 < 2.2e-16 ***
## ar3        0.826309   0.041684  19.823 < 2.2e-16 ***
## ma1       -1.120745   0.033411 -33.544 < 2.2e-16 ***
## ma2        0.913831   0.039679  23.031 < 2.2e-16 ***
## intercept 51.259385   1.303996  39.309 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4b Auto arima with fit criteriion as BIC, using the BIC as a fit criterion gives ARIMA(1,0,2) likewise all coeficient are significant.

Comparing the models we see that MA(2) has the highest σ^2 =23.7. AR(2) σ^2 =4.439. Using the autoarima with bic and without bic had both σ^2 of 4.284 and 4.272 respectfully. Comparing the models with simplicity we can go with the ARIMA(1,0,2)

```
# Auto.arima
fit3b = auto.arima(df1_ts, seasonal = "F", ic= "bic")
fit3b

## Series: df1_ts
## ARIMA(1,0,2) with non-zero mean
##
## Coefficients:
##          ar1     ma1     ma2     mean
##       0.8786  0.1705  0.2442  51.3660
```

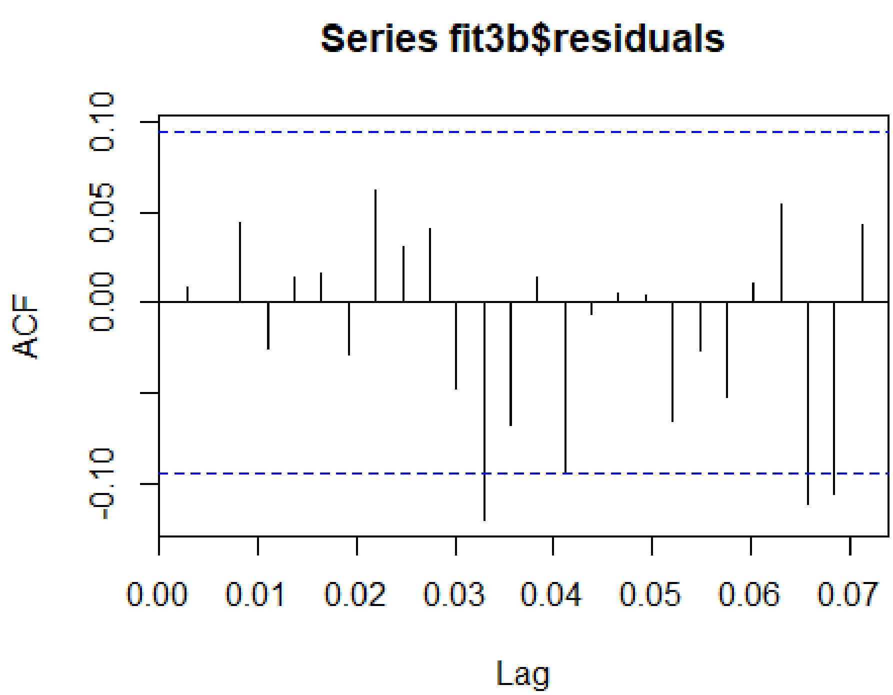
```
## s.e.   0.0259  0.0503  0.0525   1.1358
##
## sigma^2 = 4.284:  log likelihood = -926.33
## AIC=1862.66   AICc=1862.8   BIC=1883
```

```
coeftest(fit3b)
```

```
##
## z test of coefficients:
##
##            Estimate Std. Error z value  Pr(>|z|)
## ar1        0.878615   0.025922 33.8951 < 2.2e-16 ***
## ma1        0.170533   0.050307  3.3898 0.0006993 ***
## ma2        0.244205   0.052464  4.6547 3.244e-06 ***
## intercept 51.366036   1.135791 45.2249 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4C, The model I am picking is the ARIMA(1,0,2) this has the highest p-value while analysising the residual. It also has the lowest σ^2 .

```
acf(fit3b$residuals)
```




```
Box.test(fit3b$residuals, lag=10, type="Ljung")
```

```
##
##  Box-Ljung test
##
```

```
## data:  fit3b$residuals
## X-squared = 4.563, df = 10, p-value = 0.9184
```

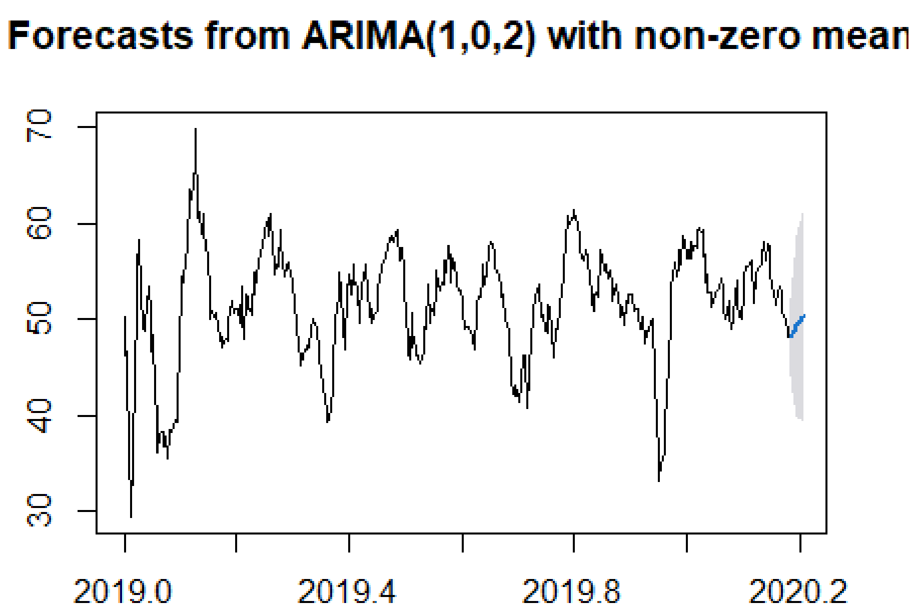
4d, 5-step ahead forecast. From the forcast plot and upward movement seem to excist.

```
fore_arima = forecast::forecast(fit3b, h=5, level = c(95))
df_arima = as.data.frame(fore_arima)
df_arima
```

```
##           Point Forecast    Lo 95    Hi 95
## 2020.1836       48.11247 44.05576 52.16918
## 2020.1863       48.38277 42.50304 54.26249
## 2020.1890       48.74489 41.19868 56.29110
## 2020.1918       49.06306 40.44810 57.67801
## 2020.1945       49.34260 39.98574 58.69946
```

4e,From the forcast plot and upward movement seem to excist towards the mean.

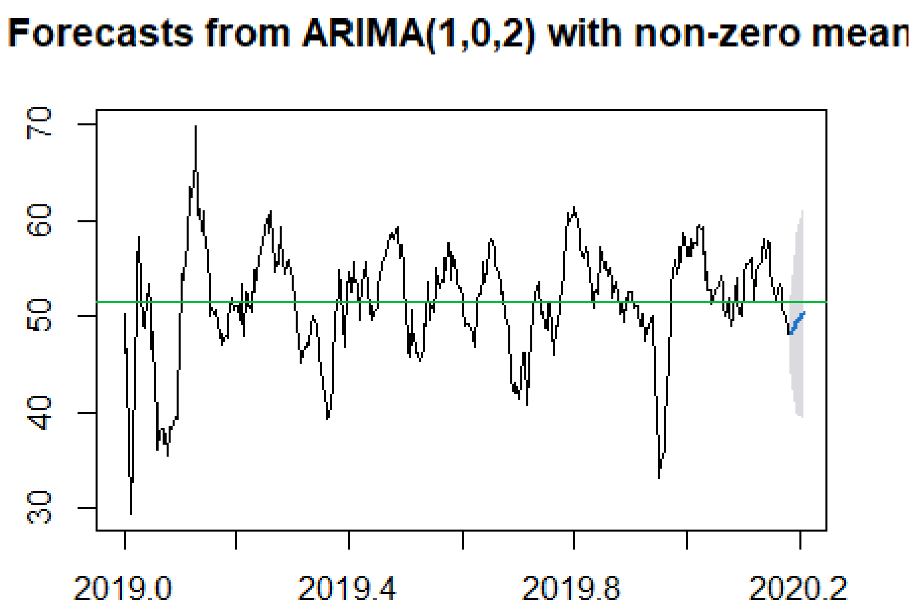
```
# Now, compute forecasts
plot(forecast(fit3b, h=10, level = c(95)), xlim=c(2019, 2020.2))
```



4f, The model predicts suggest a gradual increase as PMI increase toward the mean as seen above. Hence the manufacuting economy is expanding.

4g, Forcast for ARMA model will decay to the current level of the series and remain there. Converges to the mean.

```
b=mean(df1_ts)
# Now, compute forecasts
plot(forecast(fit3b, h=10, level = c(95)), xlim=c(2019, 2020.2))
abline(h=b, col=3)
```



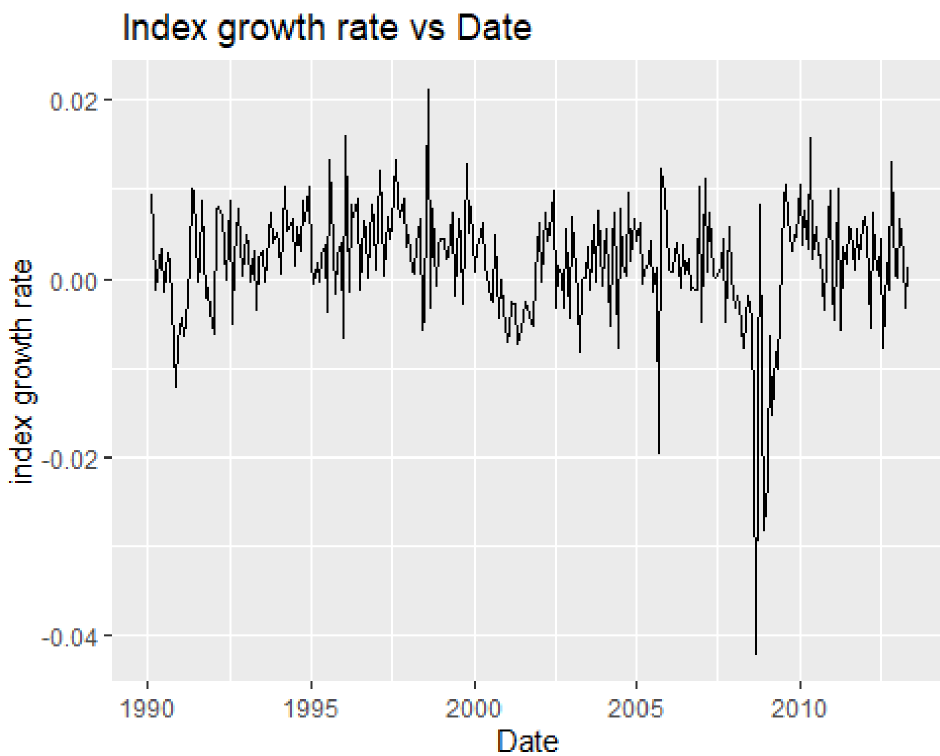
5

5a Import data

```
library(readxl)
df2<- read.csv("indpro.csv")
df2$date = as.Date(df2$date,format = "%m/%d/%y")
df2$date =as.character(df2$date)
df2_ts = ts(df2$rate,c(1990,02),frequency=12)
```

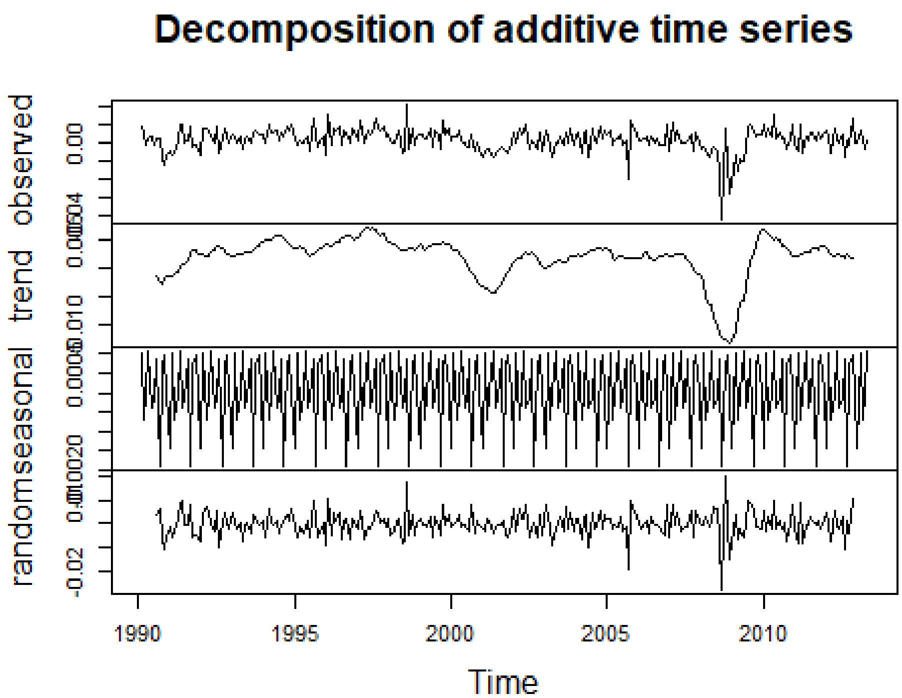
5b, Time plot and analyze series for stationarity, trends and strong seasonality

```
ggplot(df2_ts,aes(x= time(df2_ts) ,y= df2_ts
,group=1))+geom_line()+labs(title = " Index growth rate vs
Date")+xlab("Date")+ylab("index growth rate")
```



The series is stationary, there seem to be no distinct trend in the sereies also there is no strong seaoninality in the series beacause the random scale is larger than the seasonal range.

```
plot(decompose(df2_ts))
```



5c Using the Jarque Bera test we reject the null hypotesis , hence index rate is not normally distributed.

```
library(tseries)
jarque.bera.test(df2_ts)

##
##  Jarque Bera Test
##
## data:  df2_ts
## X-squared = 896.15, df = 2, p-value < 2.2e-16
```

5d Both Dickey-Fuller test rejects non-stationarity and KPSS fail to reject stationarity hence series is stationary and no further transformation is needed.

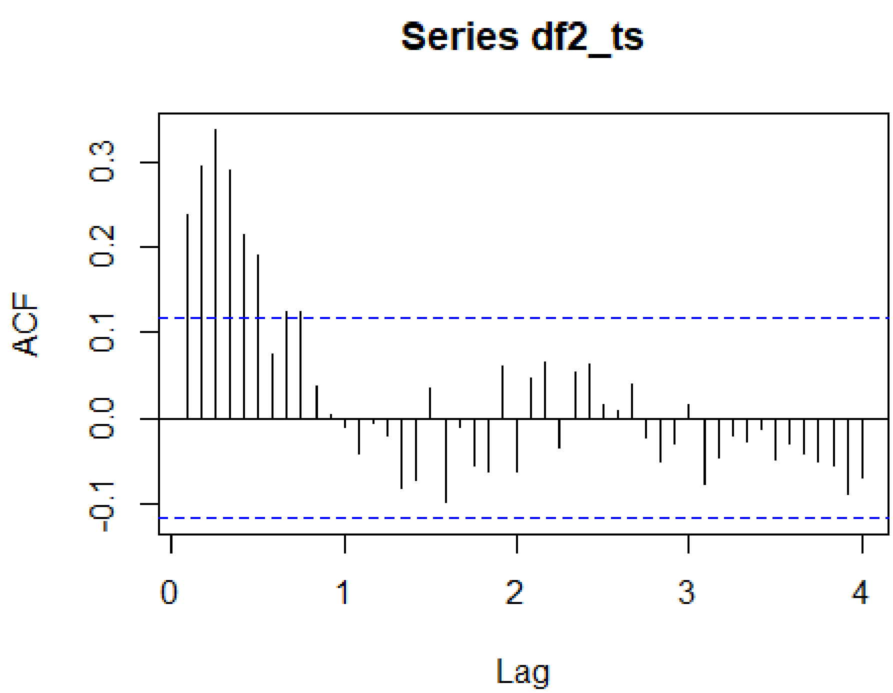
```
#
#   Dickey-Fuller:  The series has a unit root ---> non-stationary
#   KPSS          :  The series is stationary
library(TSA)
adf.test(df2_ts)    # What can we conclude?

##
##  Augmented Dickey-Fuller Test
##
## data:  df2_ts
## Dickey-Fuller = -4.5041, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```r
kpss.test(df2_ts, null="Level")   # What can we conclude?

##
##  KPSS Test for Level Stationarity
##
## data:  df2_ts
## KPSS Level = 0.33824, Truncation lag parameter = 5, p-value = 0.1
```

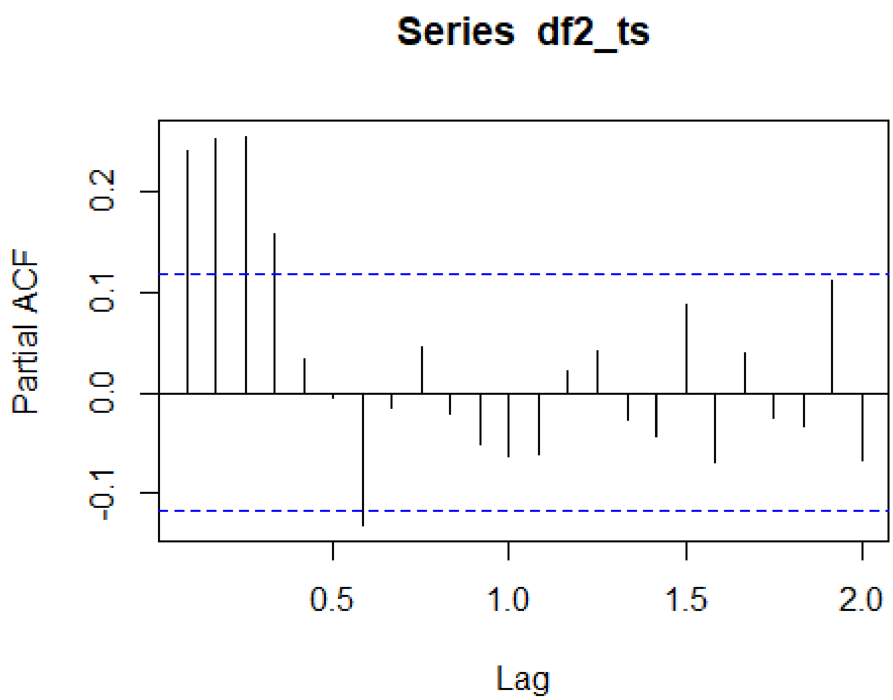
5e The ACF and PACF do not show any distinct sign of non-stationarity , but the EACF shows sign of stationarity since the top row of the EACF has no strong line of X's.

Model shows a little bit of both AR and MA, the EACF suggest a ARIMA(1,0,1) process, also the ACF and PACF show some MA(1) and AR(1) attribute respectfully.

```r
# ACF Analysis
acf(df2_ts, lag.max = 48)
```




```r
pacf(df2_ts)
```




```
eacf(df2_ts)

## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x o x x o o  o  o  o
## 1 x o o o o o o o o o o  o  o  o
## 2 x x o o o o x o o o o  o  o  o
## 3 x x x o o o o o o o o  o  o  o
## 4 x o o x o o o o o o o  o  o  o
## 5 x o x x x o o o o o o  o  o  o
## 6 o x x x x x o o o o o  o  o  o
## 7 o x x x x o x o o o o  o  o  o
```

5f, The coefficient of the model are significant but the intercept (mean) is not significant the resedual fail to reject the the null hypotesis hence residual has no correlation.

ARMA (1,1)

$$V_t = 0.89_{v_{t-1}} - 0.83_{at-1}$$

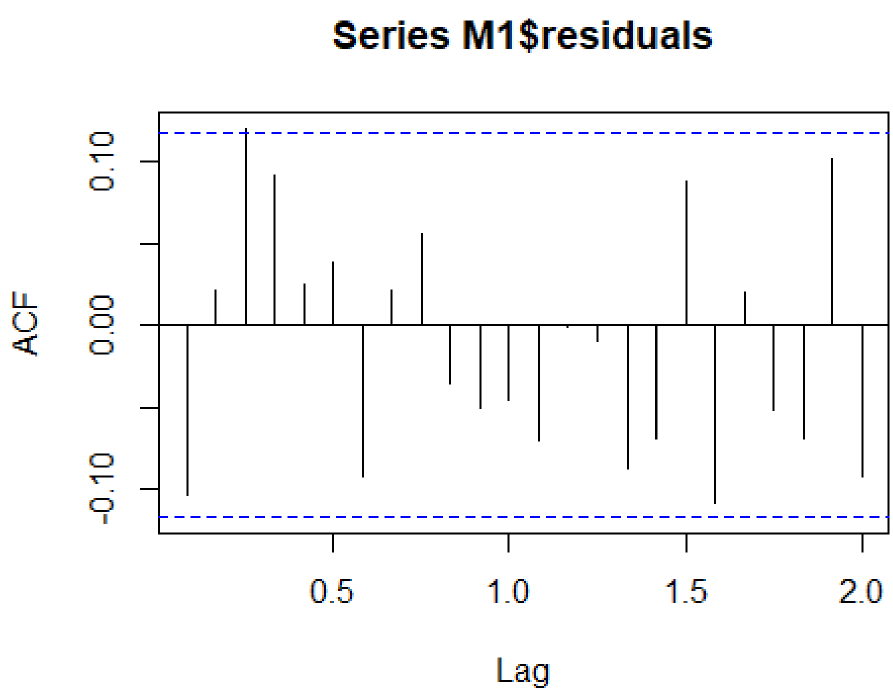
```
M1 = Arima(df2_ts, order=c(1, 0, 1))
M1

## Series: df2_ts
## ARIMA(1,0,1) with non-zero mean
```

```
##
## Coefficients:
##          ar1      ma1    mean
##       0.8919  -0.6967  0.0017
## s.e.  0.0407   0.0583  0.0010
##
## sigma^2 = 3.692e-05:  log likelihood = 1032.99
## AIC=-2057.97   AICc=-2057.83   BIC=-2043.43
```

```
coeftest(M1)
```

```
##
## z test of coefficients:
##
##              Estimate  Std. Error  z value Pr(>|z|)
## ar1        0.89188598  0.04069736  21.9151  < 2e-16 ***
## ma1       -0.69671463  0.05825331 -11.9601  < 2e-16 ***
## intercept  0.00174898  0.00099419   1.7592  0.07854 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
acf(M1$residuals)
```




```
Box.test(M1$residuals, lag=48, type="Ljung")
```

```
##
##  Box-Ljung test
##
```

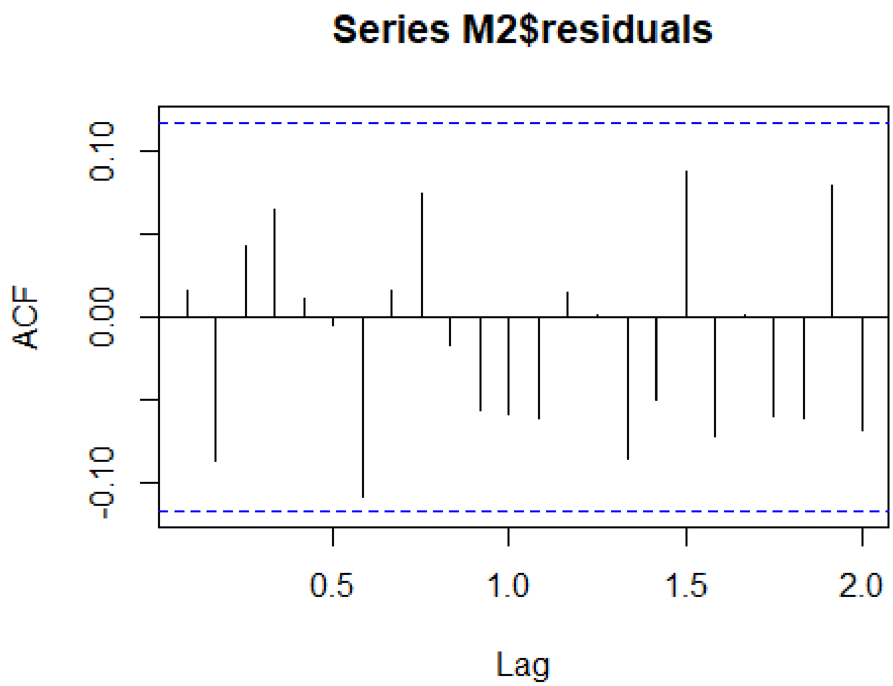
```
## data:  M1$residuals
## X-squared = 50.153, df = 48, p-value = 0.388
```

5g, Auto ARIMA with BIC suggest a ARIMA(1,0,2) all coefficent are significant in the model but AR(1) and MA(1) are highly significant. The residual fail to reject white noise hence has no correlation

ARMA (1,2)

$$V_t = 0.89_{v_{t-1}} - 0.83_{at-1} - 0.21_{at-2}$$

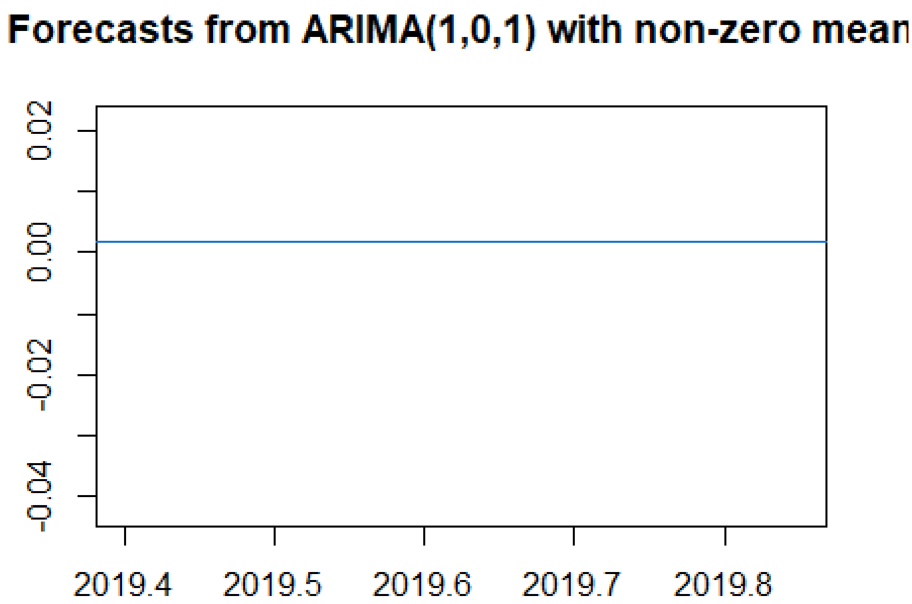
```
# Auto.arima
M2 = auto.arima(df2_ts,ic= "bic")
M2

## Series: df2_ts
## ARIMA(1,0,2) with zero mean
##
## Coefficients:
##          ar1      ma1     ma2
##       0.8813  -0.8302  0.2172
## s.e.  0.0421   0.0716  0.0693
##
## sigma^2 = 3.604e-05:  log likelihood = 1036.24
## AIC=-2064.49   AICc=-2064.34   BIC=-2049.95

coeftest(M2)

##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1  0.881250   0.042066  20.9491 < 2.2e-16 ***
## ma1 -0.830236   0.071645 -11.5882 < 2.2e-16 ***
## ma2  0.217189   0.069277   3.1351  0.001718 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

acf(M2$residuals)
```



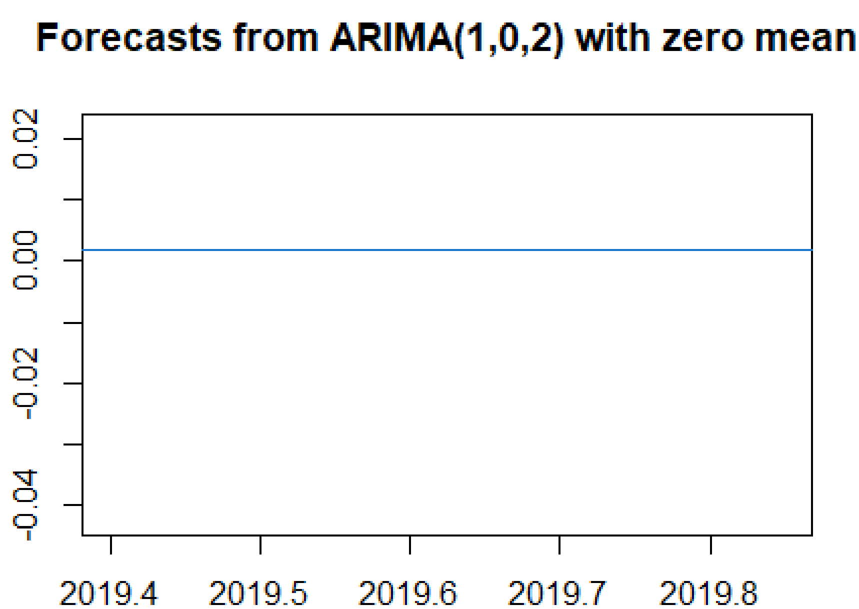

```
Box.test(M2$residuals, lag=10, type="Ljung")

##
##  Box-Ljung test
##
## data:  M2$residuals
## X-squared = 9.1922, df = 10, p-value = 0.514
```

5h Forcast behavior for each series are different, for M1 the forcast somewhat converge to the mean of the series. While the M2 model the forcast somewhat deviates from the mean.

```
# Now, compute forecasts
plot(forecast(M1, h=10, level = c(95)), xlim=c(2019.4, 2019.85))
abline(h=mean(df2_ts), col=4)
```




```
plot(forecast(M2, h=10, level = c(95)), xlim=c(2019.4, 2019.85))
abline(h=mean(df2_ts), col=4)
```



5i, From the M1 and M2 , I tend to choose M1. The M2 model ARIMA(1,0,2) seem to have a non significant coeffiecent for M2, the M1 had all coefficent to be significant it also passed the residual test and has a better forcast behaviour compared with M2. I pick M1.

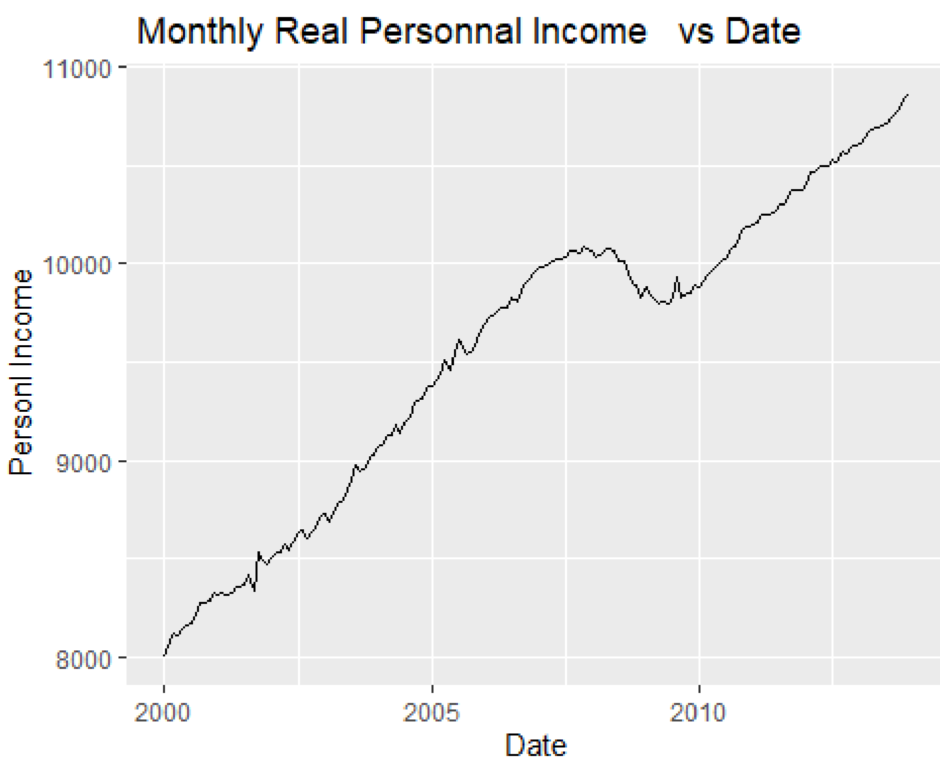
6

6a Import data

```
library(readxl)
df3<- read.csv("consump.csv")
df3$date = as.Date(df3$date,format = "%m/%d/%Y")
df3$date =as.character(df3$date)
df3_ts = ts(df3$pers_inc,c(2000,01),frequency=12)
```

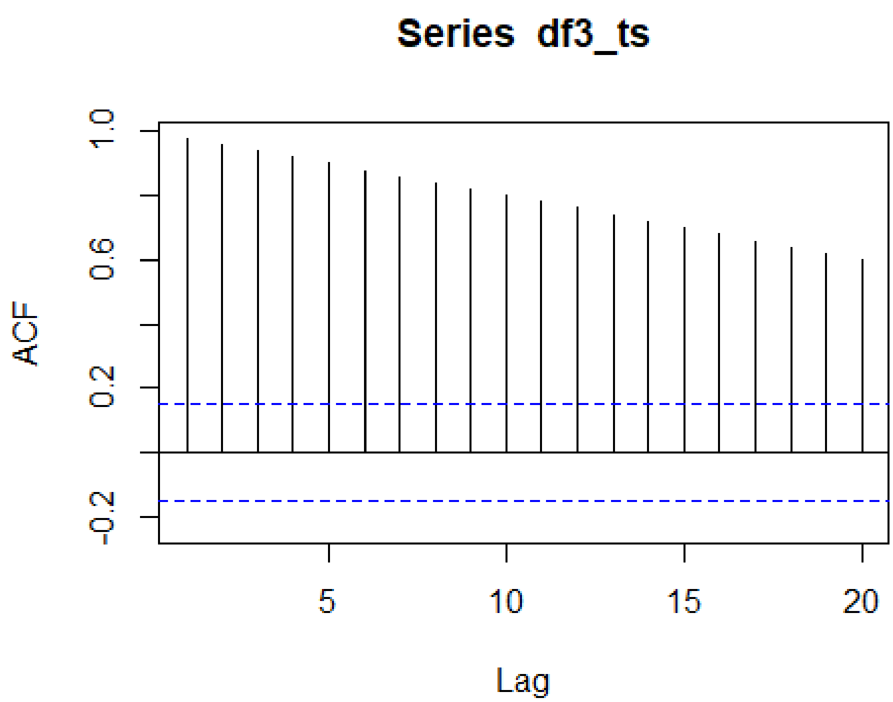
6b, Personal income show an exponential growth over time, with an up-ward trend , non stationary.

```
ggplot(df3_ts,aes(x= time(df3_ts) ,y= df3_ts
,group=1))+geom_line()+labs(title = " Monthly Real Personnal Income   vs
Date")+xlab("Date")+ylab(" Personl Income")
```

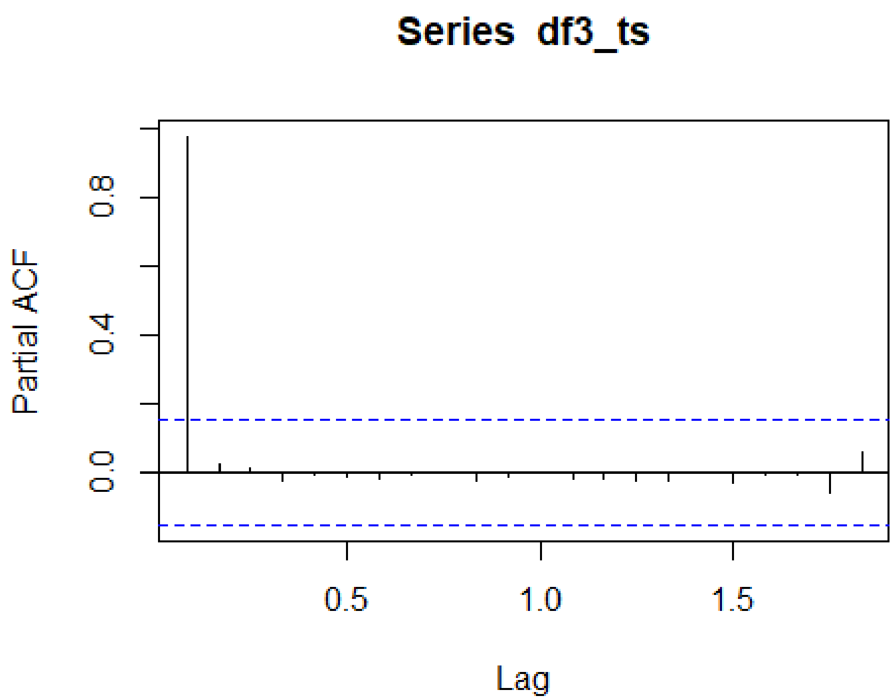


1b, PACF indicates AR(1) ACF shows a sharp fall off but not exponential which indicates non-stationary series. The top row of the EACF has constant line's of X hence series is NON-STATIONARY

```
# ACF Analysis
Acf(df3_ts, lag.max = 20)
```



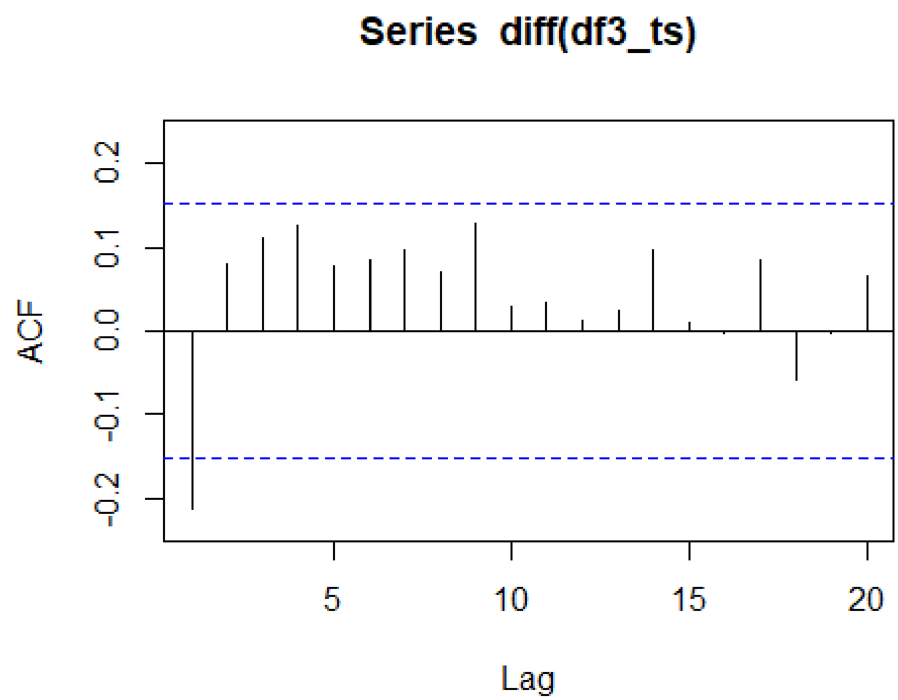

```
pacf(df3_ts)
```



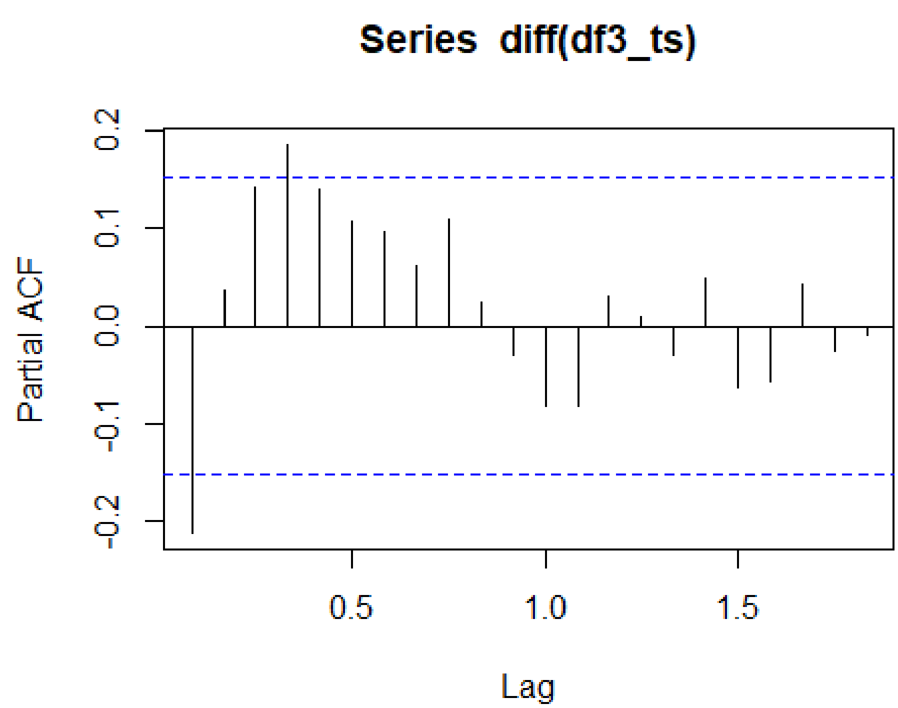

```
eacf(df3_ts)

## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x x x  x  x  x
## 1 x o o o o o o o o o o  o  o  o
## 2 x o o o o o o o o o o  o  o  o
## 3 x o o o o o o o o o o  o  o  o
## 4 x x x x o o o o o o o  o  o  o
## 5 x o o o o o o o o o o  o  o  o
## 6 x o o o o o o o o o o  o  o  o
## 7 x x o o o x o o o o o  o  o  o
```

6c, The difference of the series shows no sign of non-stationary behaviour the acf show some sign of over-differencing which is indicated in the flipping signs in the ACF plot.

```
# ACF Analysis
Acf(diff(df3_ts), lag.max = 20)
```




```r
pacf(diff(df3_ts))
```




```r
eacf(diff(df3_ts))
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x o o o o o o o o o o  o  o  o
## 1 x o o o o o o o o o o  o  o  o
## 2 x o o o o o o o o o o  o  o  o
## 3 x x x x o o o o o o o  o  o  o
## 4 x o o o o o o o o o o  o  o  o
## 5 x x o o o o o o o o o  o  o  o
## 6 x x o o o x o o o o o  o  o  o
## 7 x o o o o x o o o o o  o  o  o
```

6d, Dickey Fuller can not reject null , hence series is non-stationary. An ARIMA(p,d,q) is apporate.

```
#
#   Dickey-Fuller:  The series has a unit root ---> non-stationary
#   KPSS          :  The series is stationary
library(TSA)
adf.test(df3_ts)    # What can we conclude?

##
##  Augmented Dickey-Fuller Test
##
## data:  df3_ts
## Dickey-Fuller = -1.6612, Lag order = 5, p-value = 0.7177
## alternative hypothesis: stationary

kpss.test(df3_ts, null="Level")   # What can we conclude?

##
##  KPSS Test for Level Stationarity
##
## data:  df3_ts
## KPSS Level = 3.2373, Truncation lag parameter = 4, p-value = 0.01
```

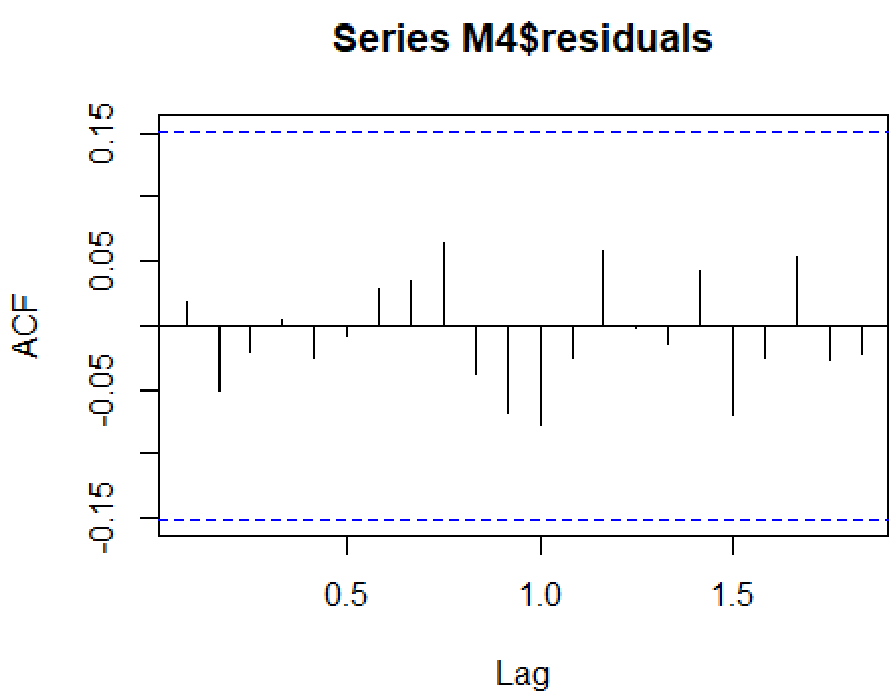
6e, From the result a differencing is apporate. From the information gathered from the initail ACF and PACF my initial model was ARIMA(1,1,1) after tuning and obsering the residual ACF the final model is ARIMA (1,1,2).

```
M4 = Arima(df3_ts, order=c(1, 1, 2))
M4

## Series: df3_ts
## ARIMA(1,1,2)
##
## Coefficients:
##          ar1      ma1     ma2
##       0.9804  -1.2851  0.4237
## s.e.  0.0176   0.0765  0.0768
##
## sigma^2 = 1100:  log likelihood = -820.93
## AIC=1649.87   AICc=1650.11   BIC=1662.34
```

```
coeftest(M4)

##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1  0.980367   0.017559  55.8338 < 2.2e-16 ***
## ma1 -1.285079   0.076545 -16.7887 < 2.2e-16 ***
## ma2  0.423688   0.076823   5.5151 3.485e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

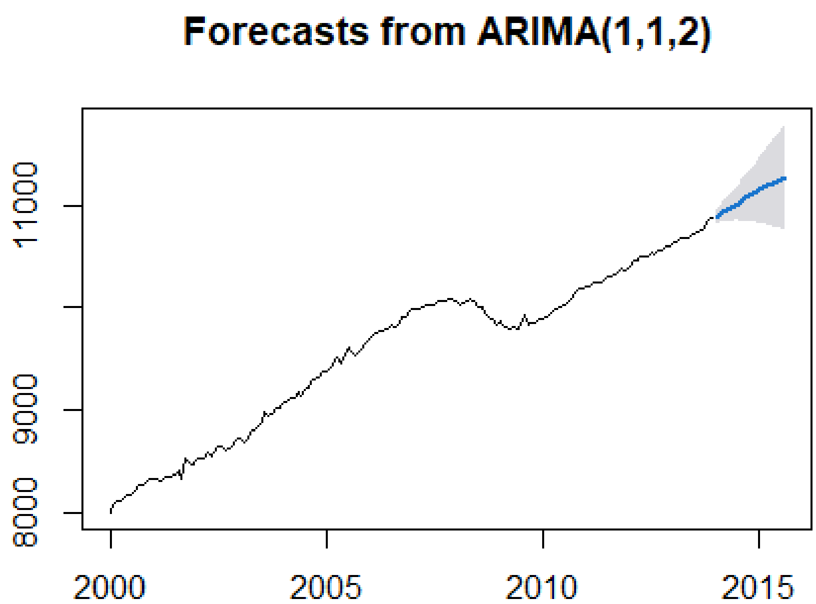
6f, The model captures all dynamic monthly behaviours.

```
acf(M4$residuals)
```




```
Box.test(M4$residuals, lag=20, type="Ljung")

##
##  Box-Ljung test
##
## data:  M4$residuals
## X-squared = 6.6331, df = 20, p-value = 0.9977
```

6g, The 20 step ahead forecast seem to be flow the path of the trend.

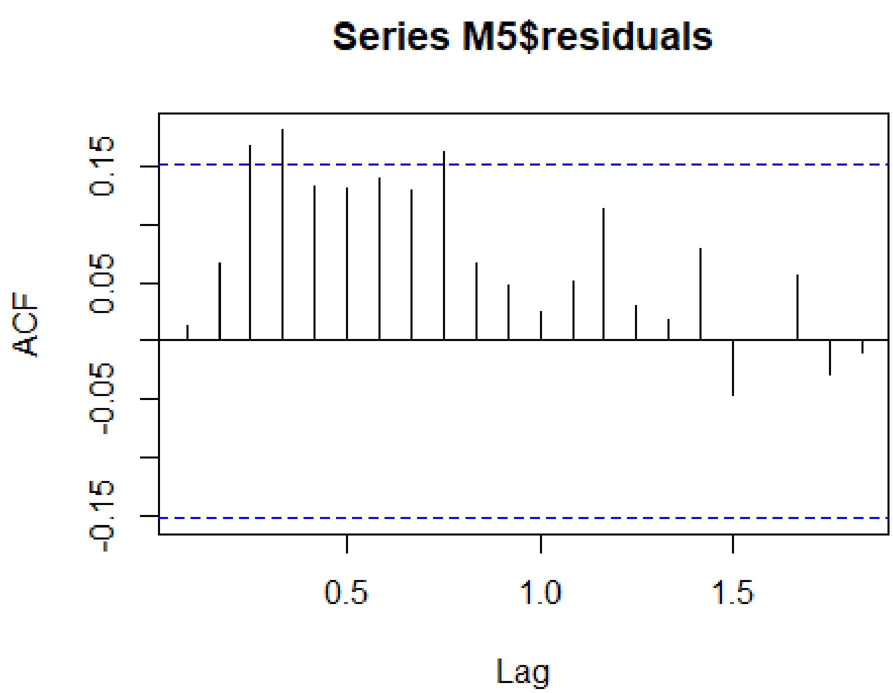


6h, Using Auto ARIMA with BIC the model suggest a ARIMA(1,1,0) and includes a drift in the model. The residual ACF seem not to capture all the required monthly information. The ARIMA model fails the Ljung-Box test by rejecting white noise. Hence the hand tuned model performs better than AUTO ARIMA's model.

```
# Auto.arima
M5 = auto.arima(df3_ts,ic= "bic")
M5

## Series: df3_ts
## ARIMA(1,1,0) with drift
##
## Coefficients:
##           ar1    drift
##       -0.2147  17.1087
## s.e.   0.0760   2.1843
##
## sigma^2 = 1188:  log likelihood = -827.11
## AIC=1660.22   AICc=1660.36   BIC=1669.57

coeftest(M5)
```

```
##
## z test of coefficients:
##
##         Estimate Std. Error z value  Pr(>|z|)
## ar1   -0.214674   0.076047 -2.8229  0.004759 **
## drift 17.108730   2.184287  7.8326 4.777e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
acf(M5$residuals)
```




```r
Box.test(M5$residuals, lag=20, type="Ljung")
```

```
##
##  Box-Ljung test
##
## data:  M5$residuals
## X-squared = 35.056, df = 20, p-value = 0.01981
```