4550 Final Project

Michael Cao, Winston Park

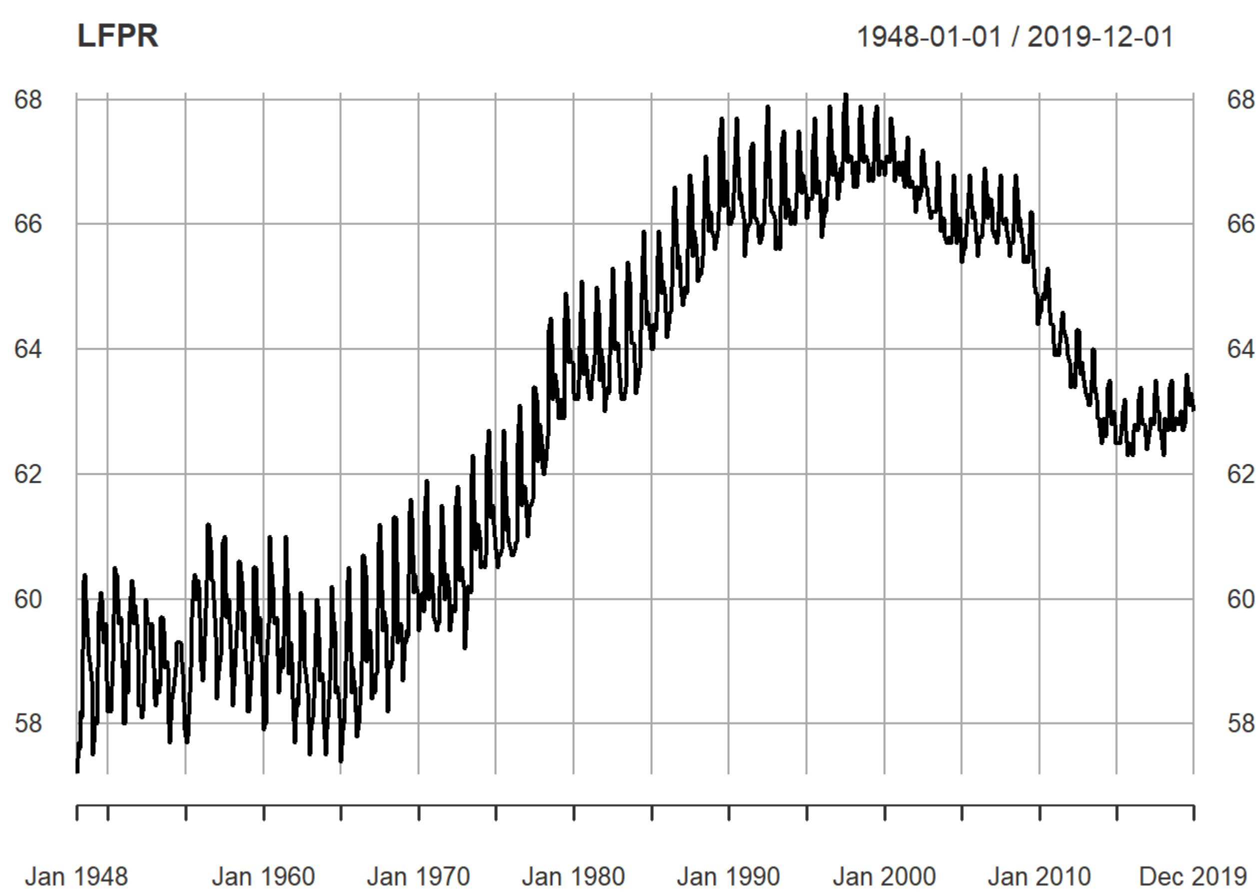
Pre model-fitting

i. Importing Raw Data

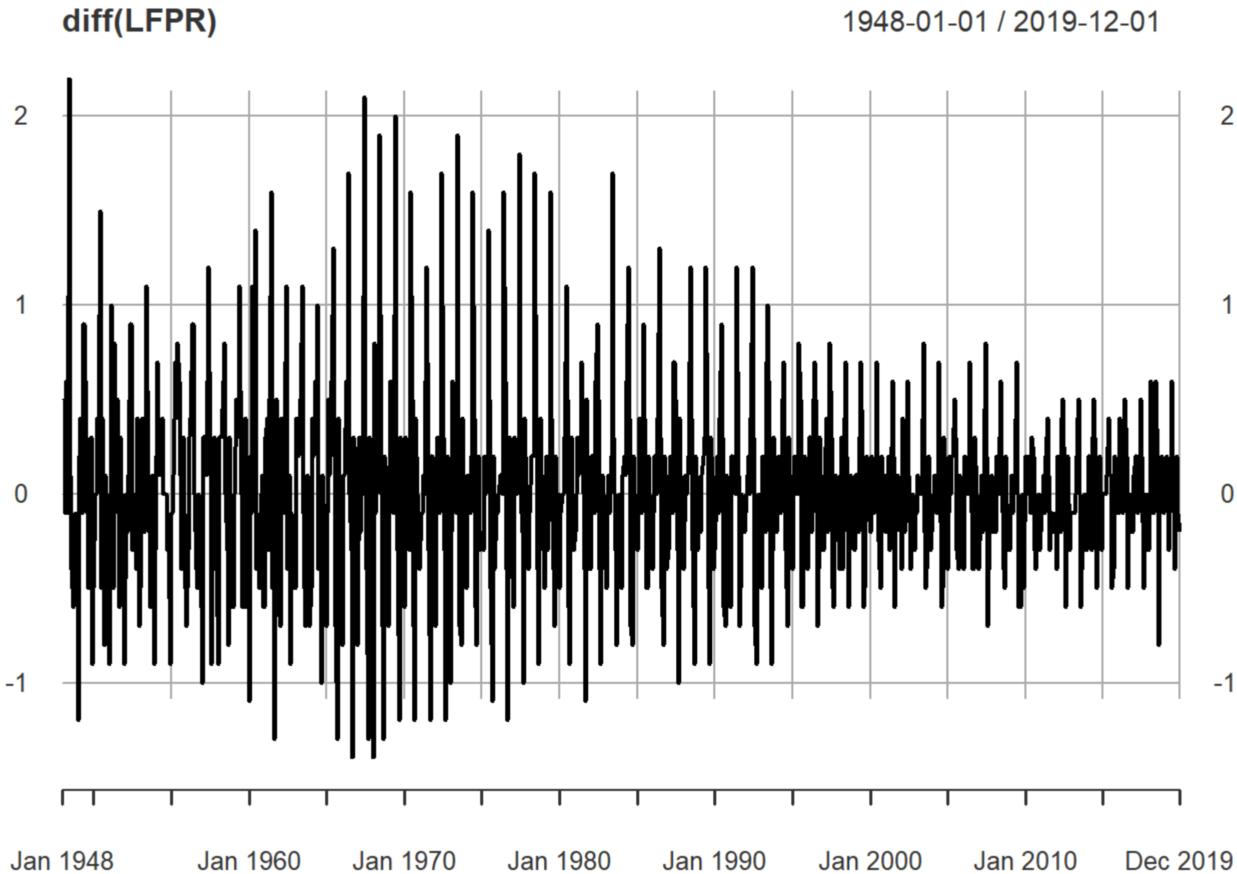
```
LFPR <- read.csv('LFPR.csv')[1:864,] # cut off before COVID
UNRATE <- read.csv('UNRATE.csv')[1:864,]
INFLATION <- read.csv('INFLATION.csv')[1:864,]
POPLEVEL <- read.csv('POPLEVEL.csv')[1:864,]

LFPR <- xts(LFPR[,2], order.by=as.Date(LFPR$DATE))
UNRATE <- xts(UNRATE[,2], order.by=as.Date(UNRATE$DATE))
INFLATION <- xts(INFLATION[,2], order.by=as.Date(INFLATION$DATE))
POPLEVEL <- xts(POPLEVEL[,2], order.by=as.Date(POPLEVEL$DATE))
```

ii. Visualize Attempted Transformations

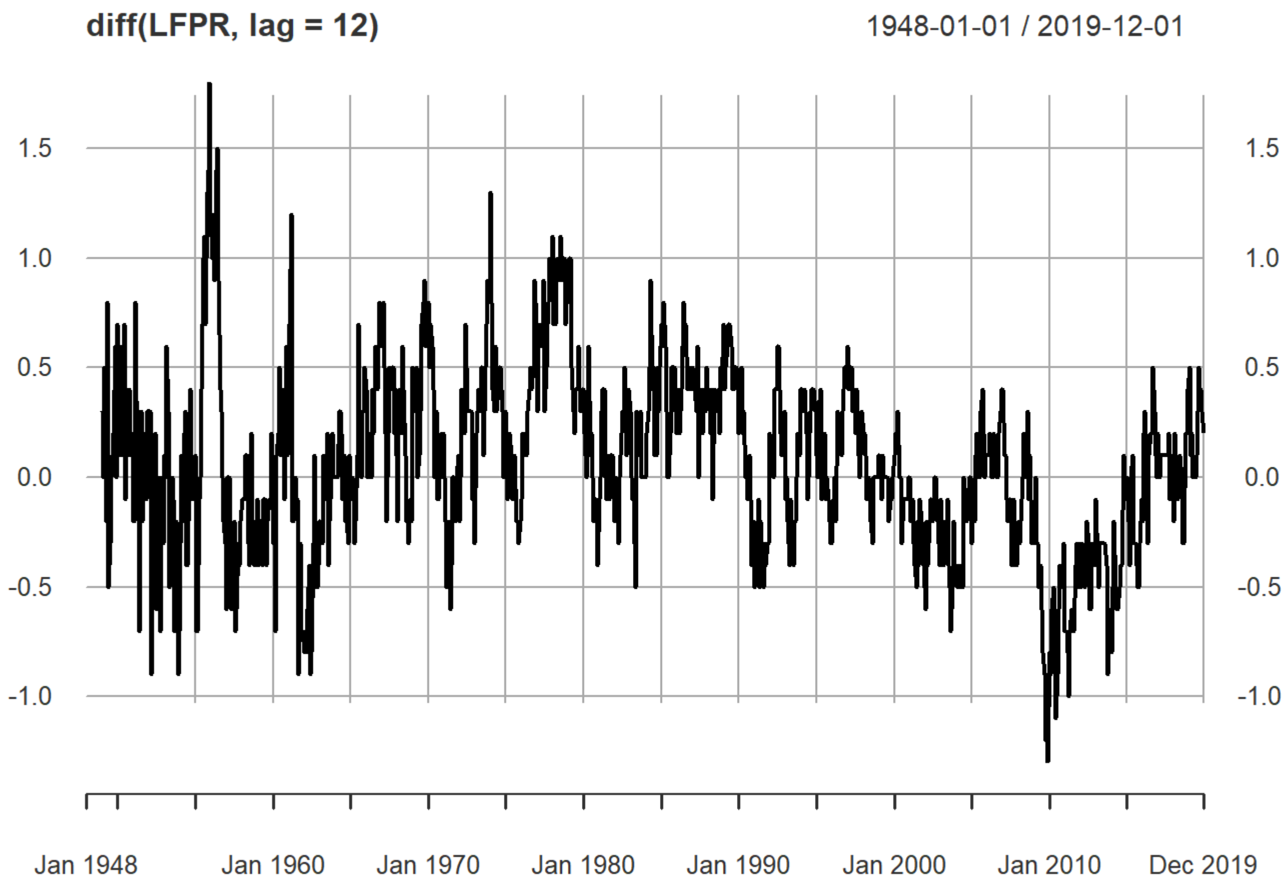
```
plot(LFPR)
```



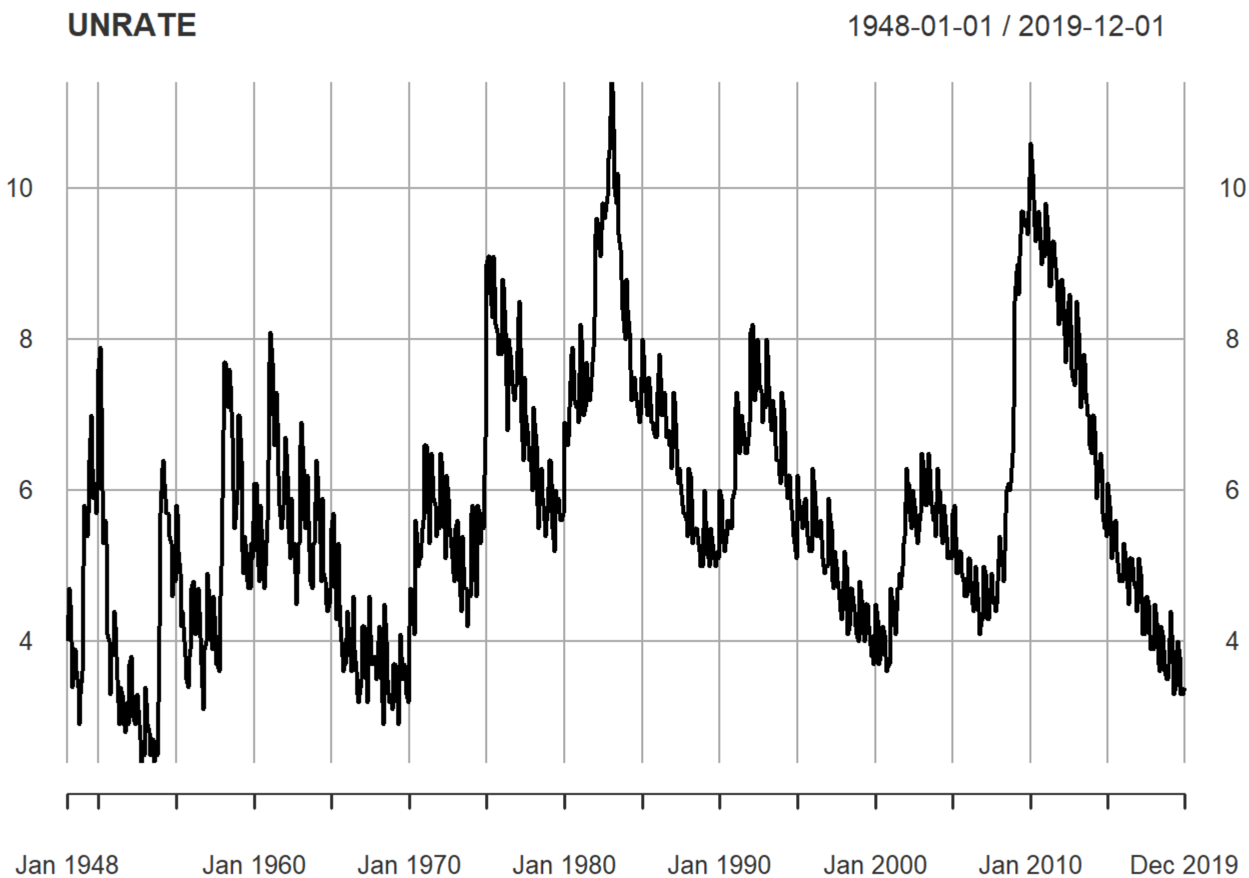

```
plot(diff(LFPR))
```



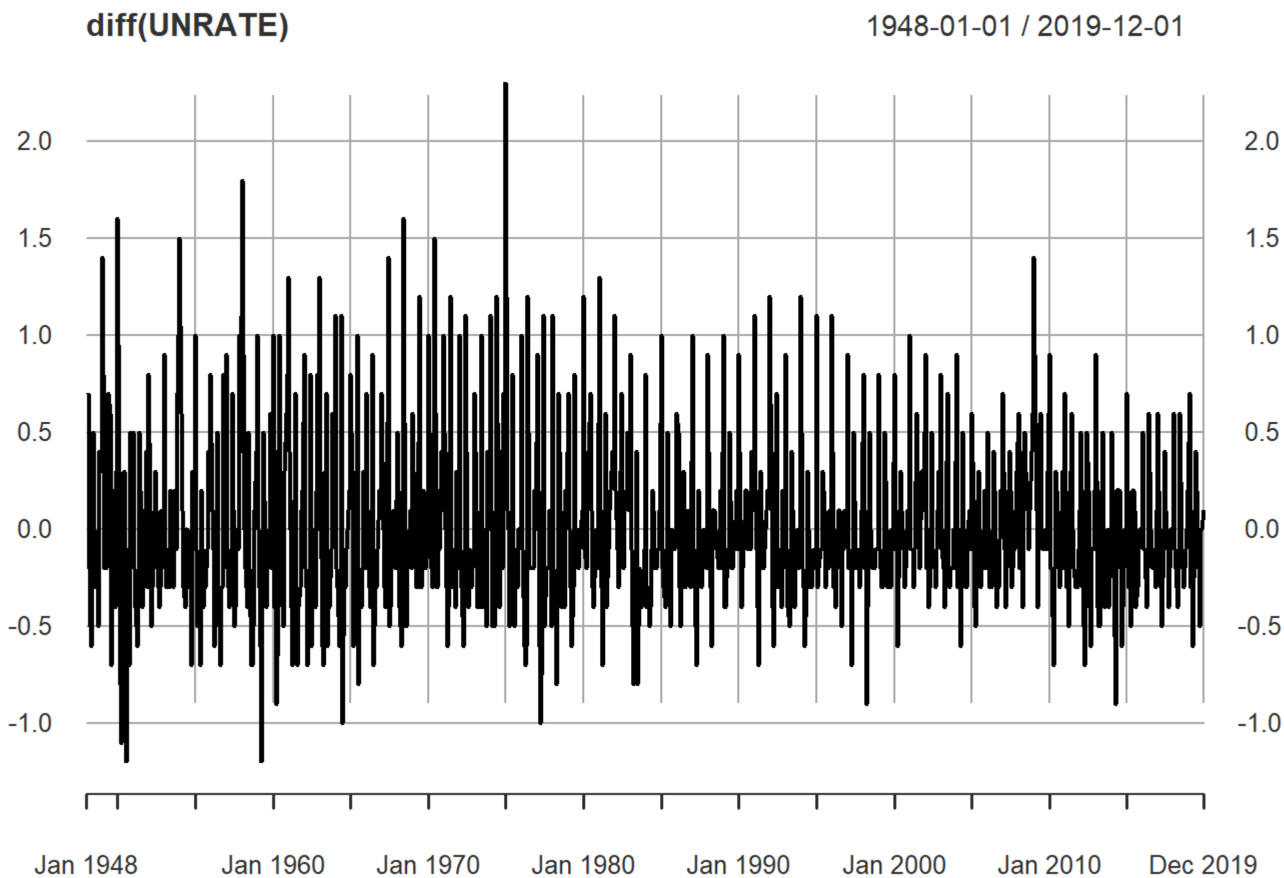

```
plot(diff(LFPR, lag = 12)) # t-12
```


diff(LFPR, lag = 12)
1948-01-01 / 2019-12-01
1.5
1.0
0.5
0.0
-0.5
-1.0
Jan 1948
Jan 1960
Jan 1970
Jan 1980
Jan 1990
Jan 2000
Jan 2010
Dec 2019

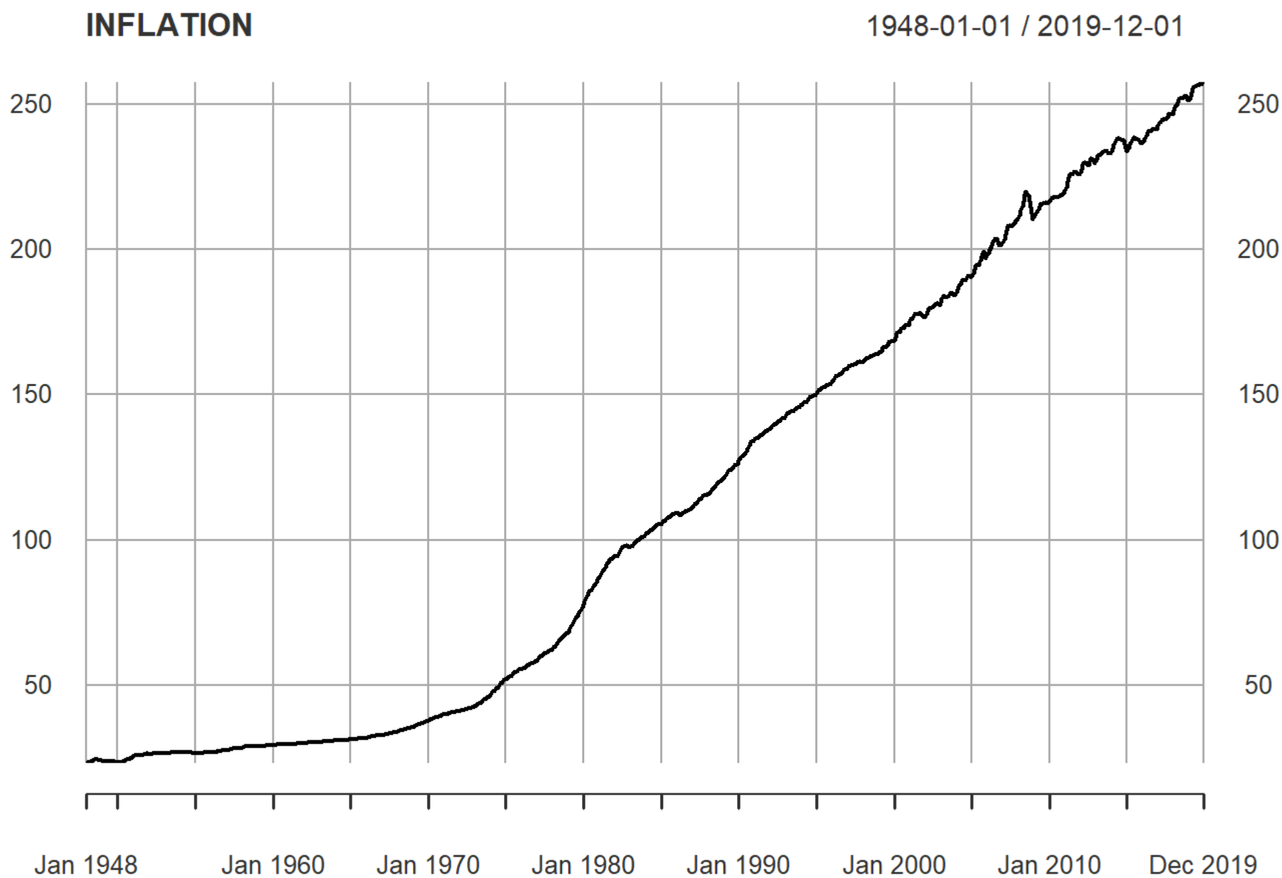
```
plot(UNRATE)
```



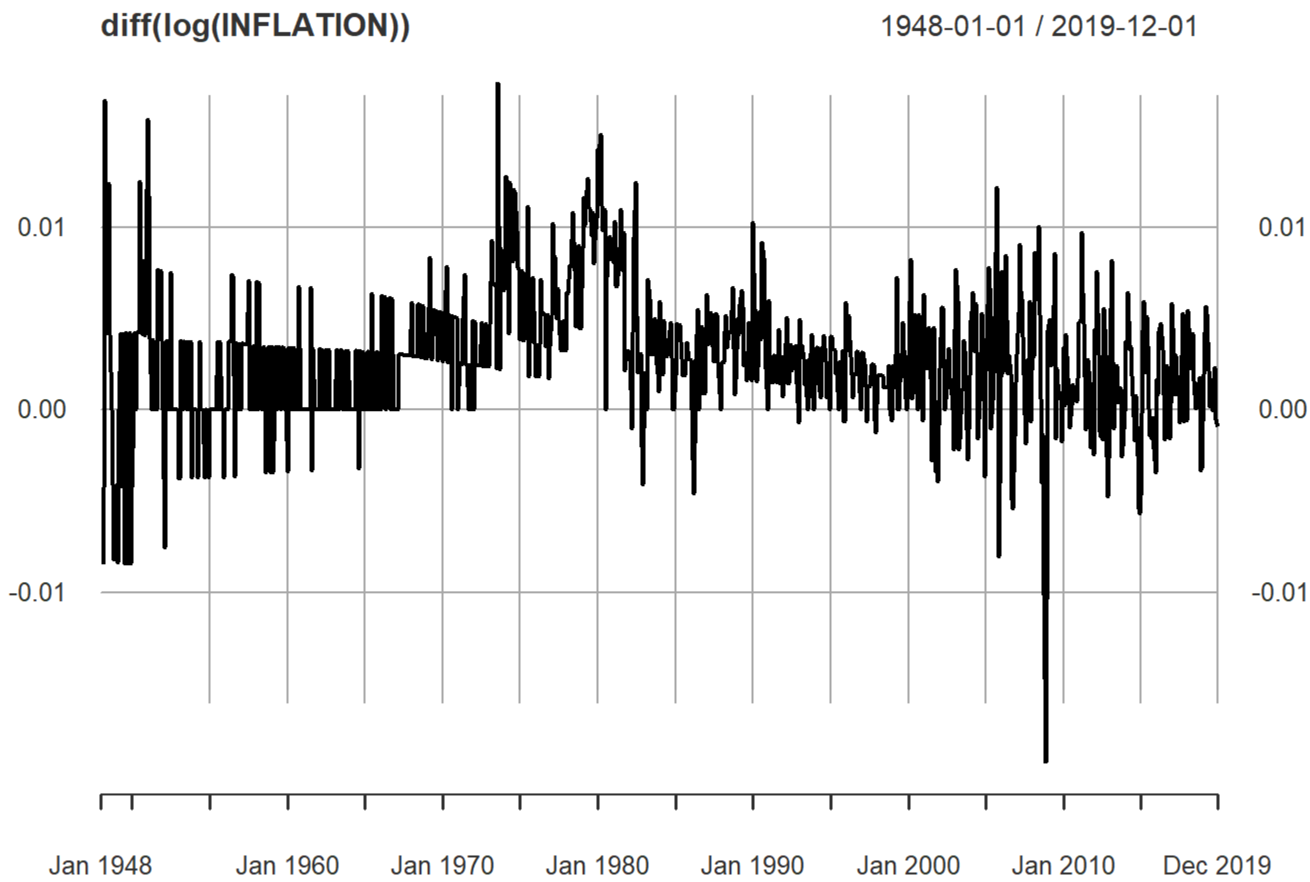

```
plot(diff(UNRATE))
```



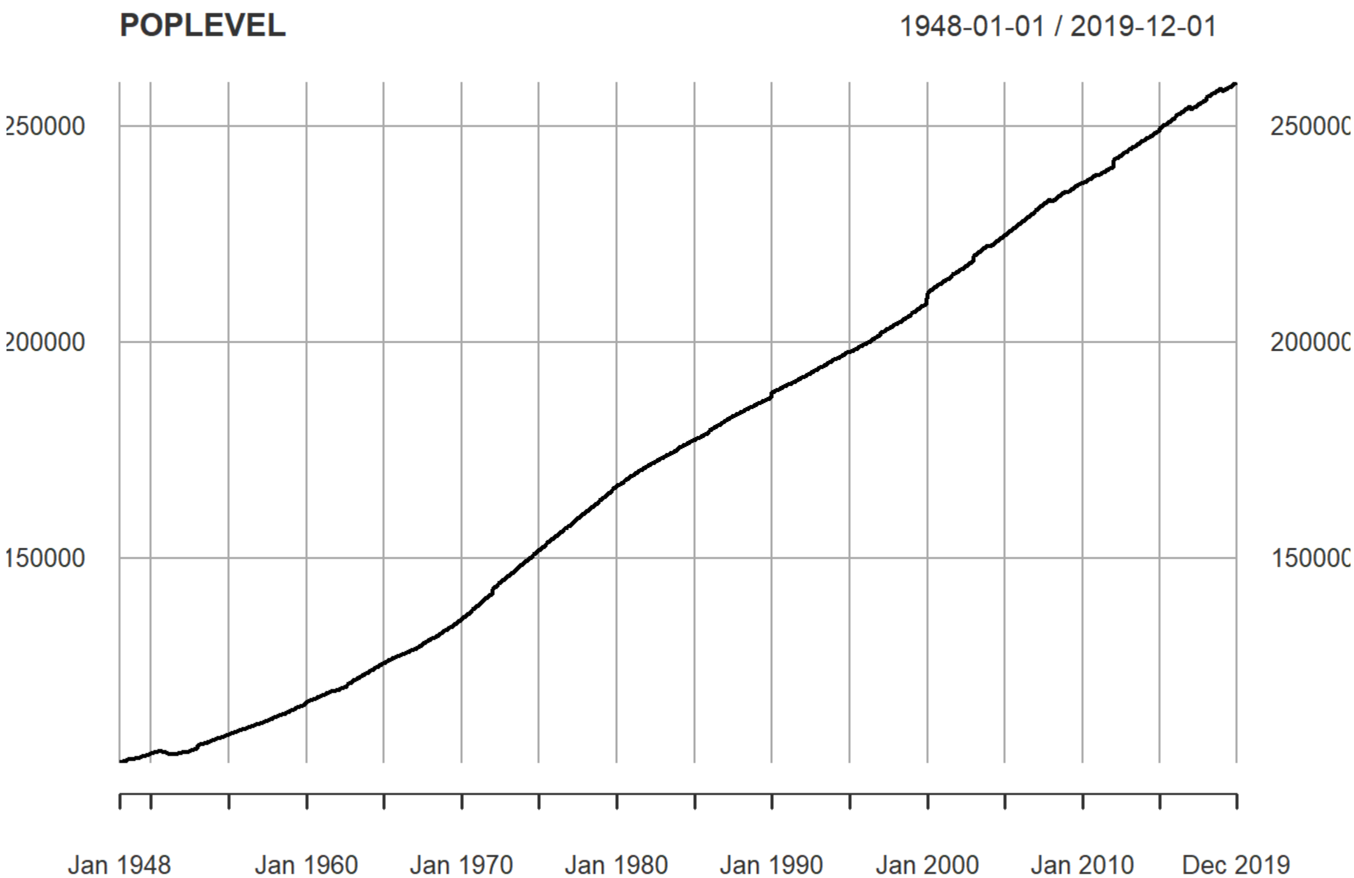

```
plot(INFLATION)
```



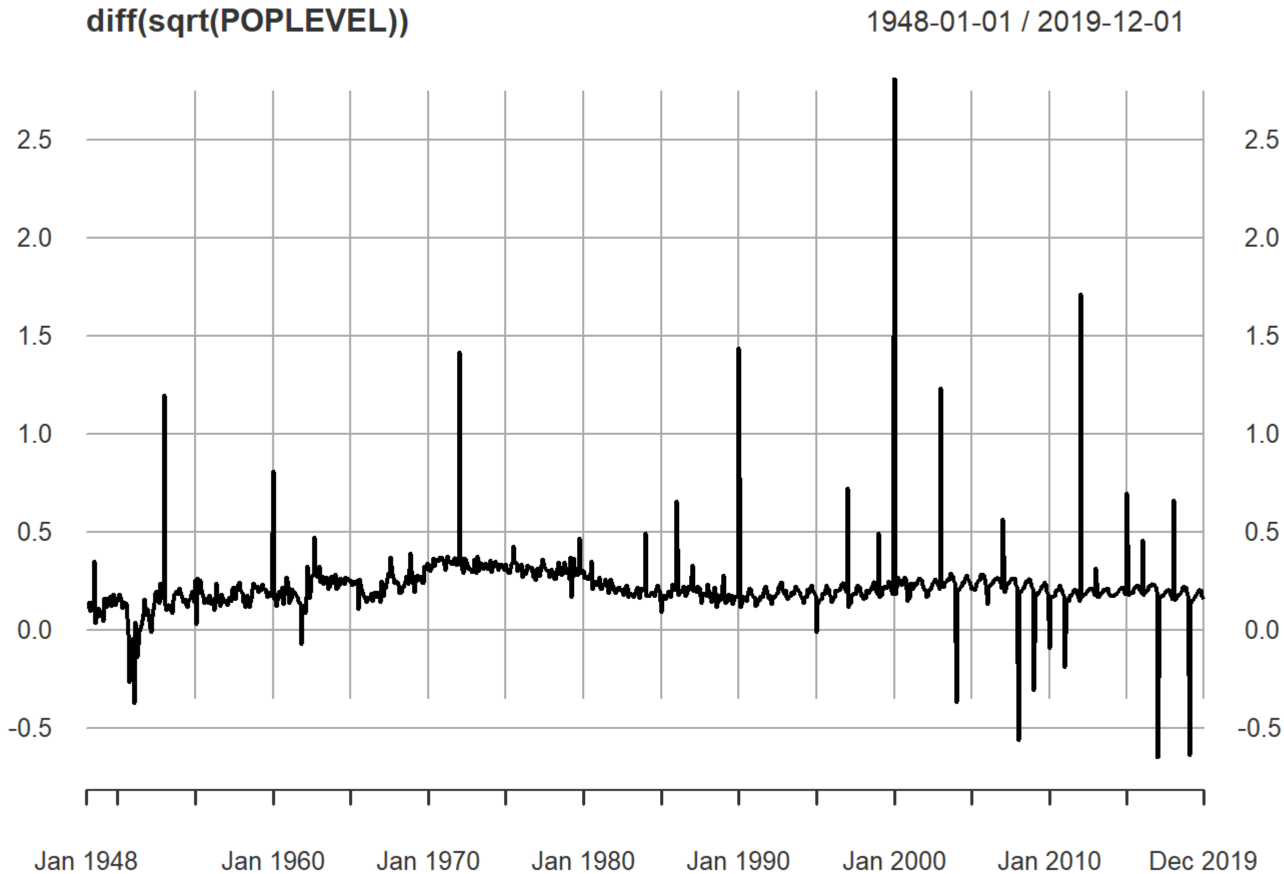

```
plot(diff(log(INFLATION)))
```



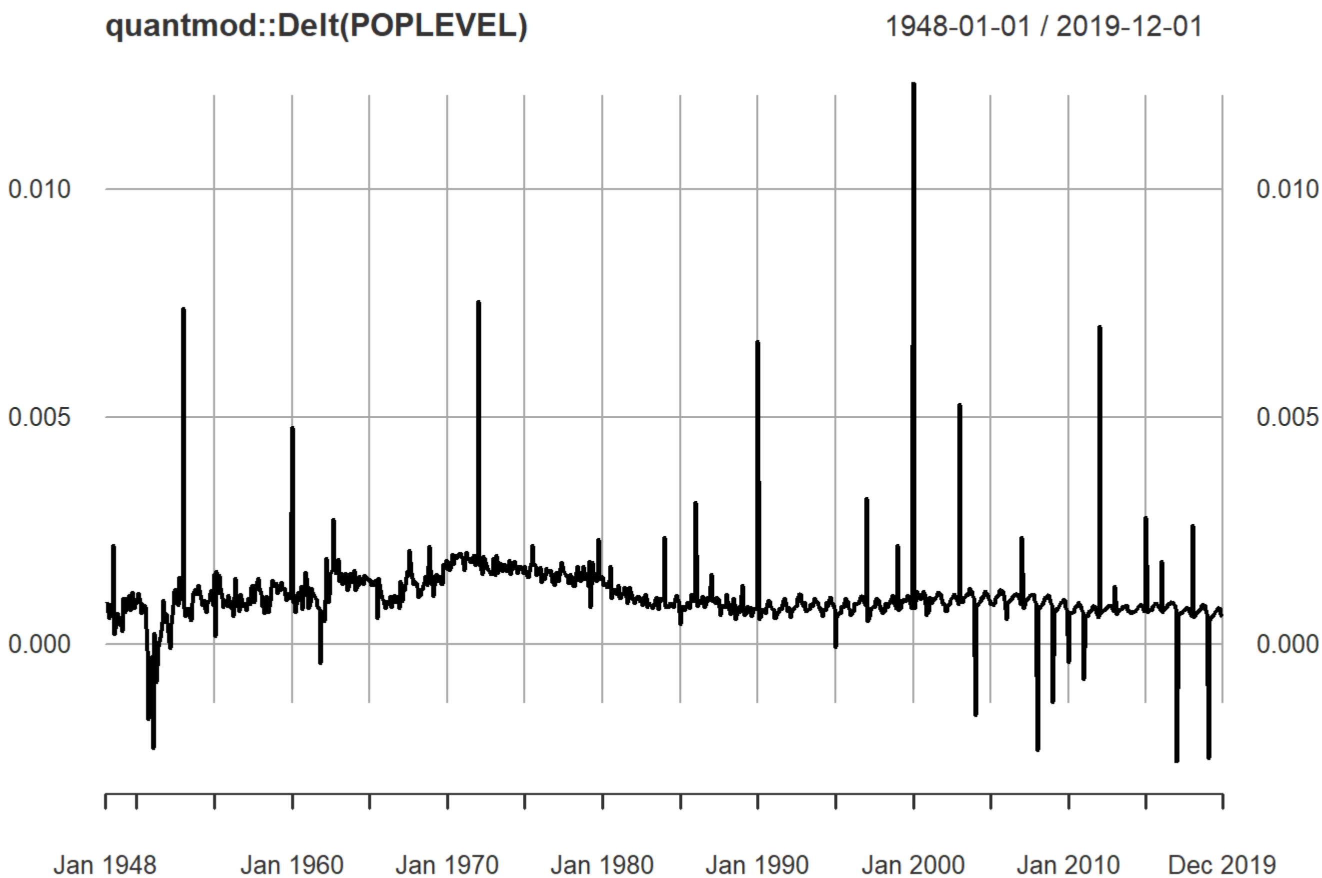

```
plot(POPLEVEL)
```




```
plot(diff(sqrt(POPLEVEL)))
```




```
plot(quantmod::Delt(POPLEVEL))
```



iii. Stationary Assumptions

```
tseries::adf.test(LFPR) # not stationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  LFPR
## Dickey-Fuller = 0.46557, Lag order = 9, p-value = 0.99
## alternative hypothesis: stationary
```

```
forecast::ndiffs(LFPR) # d=1 yields stationarity
```

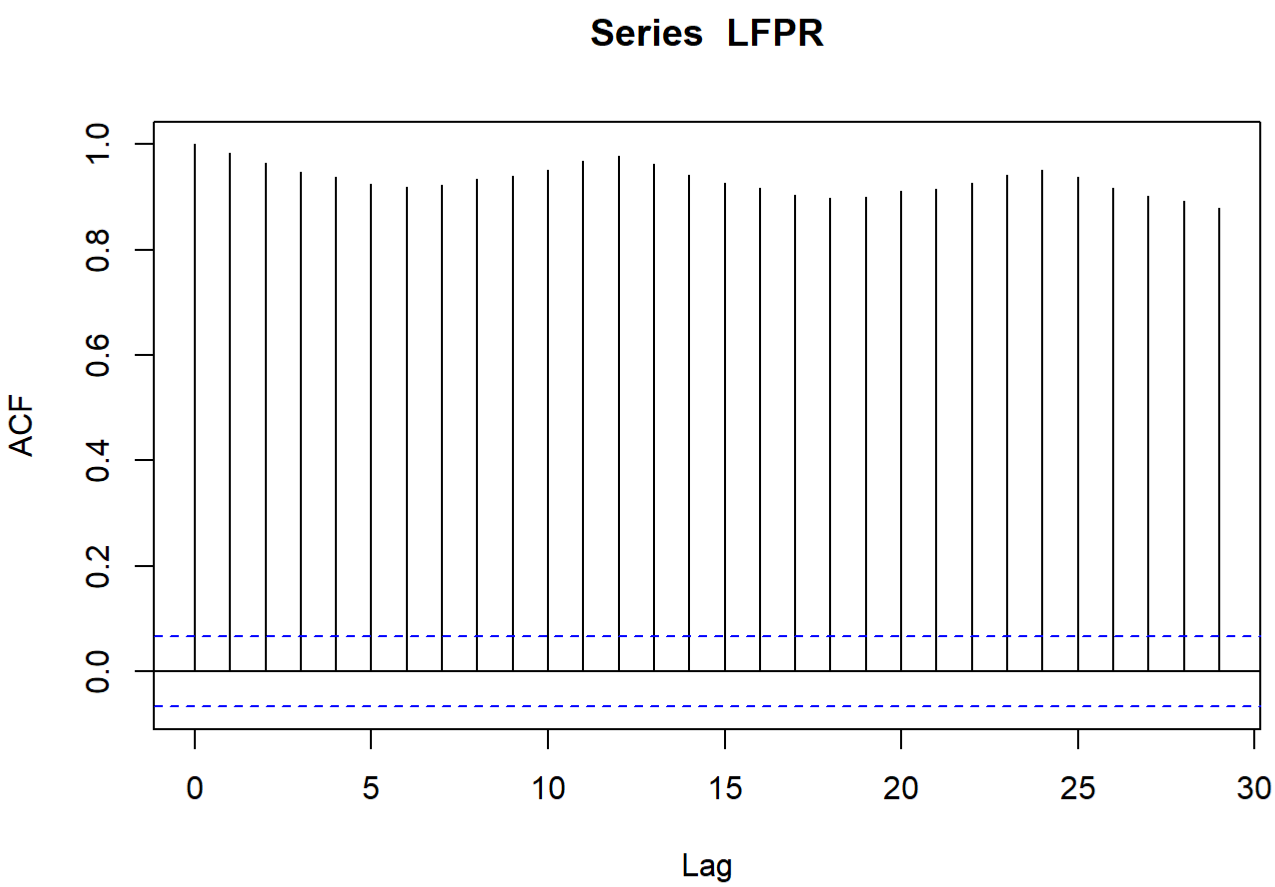
```
## [1] 1
```

```
tseries::adf.test(diff(LFPR)[-1])
```

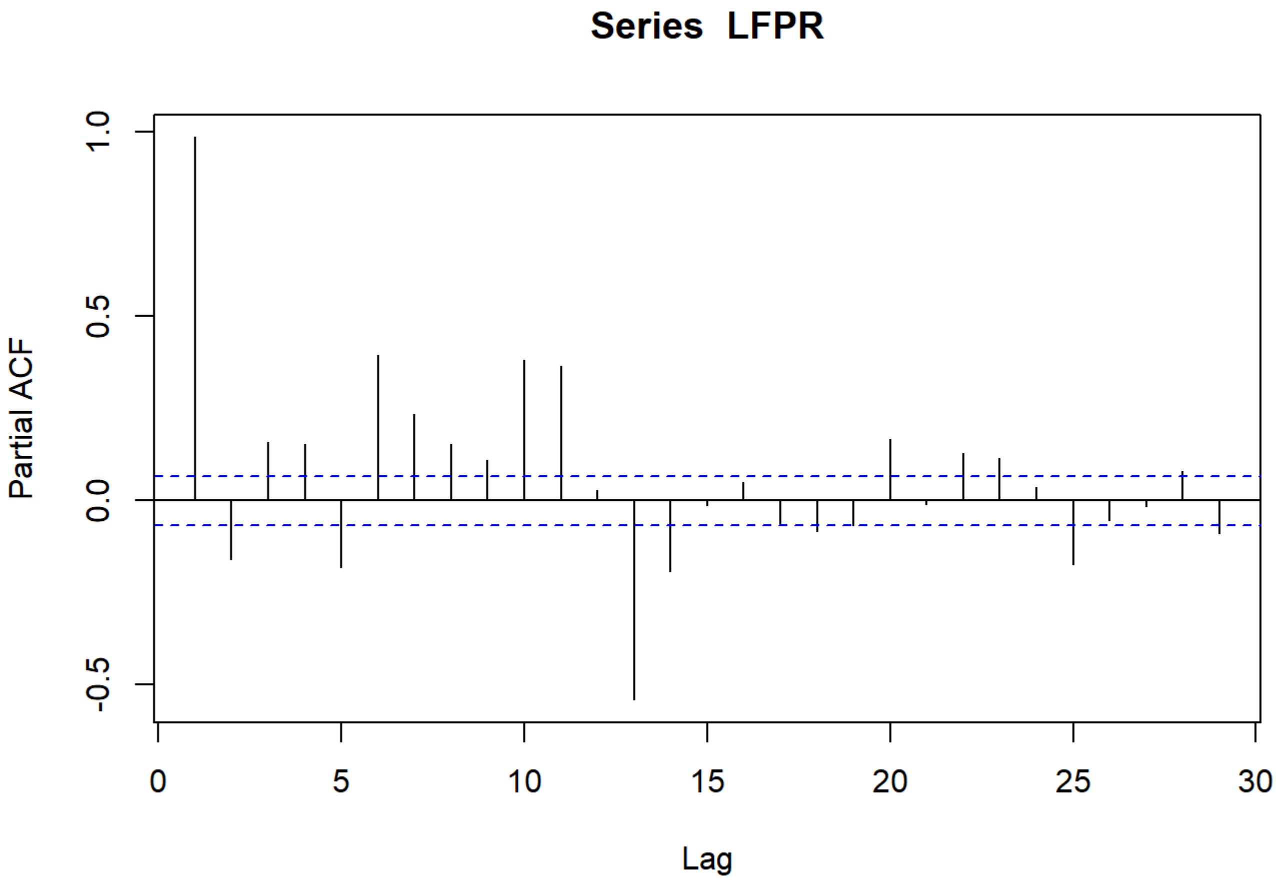
```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(LFPR)[-1]
## Dickey-Fuller = -34.627, Lag order = 9, p-value = 0.01
## alternative hypothesis: stationary
```

iv. ACF and PACF

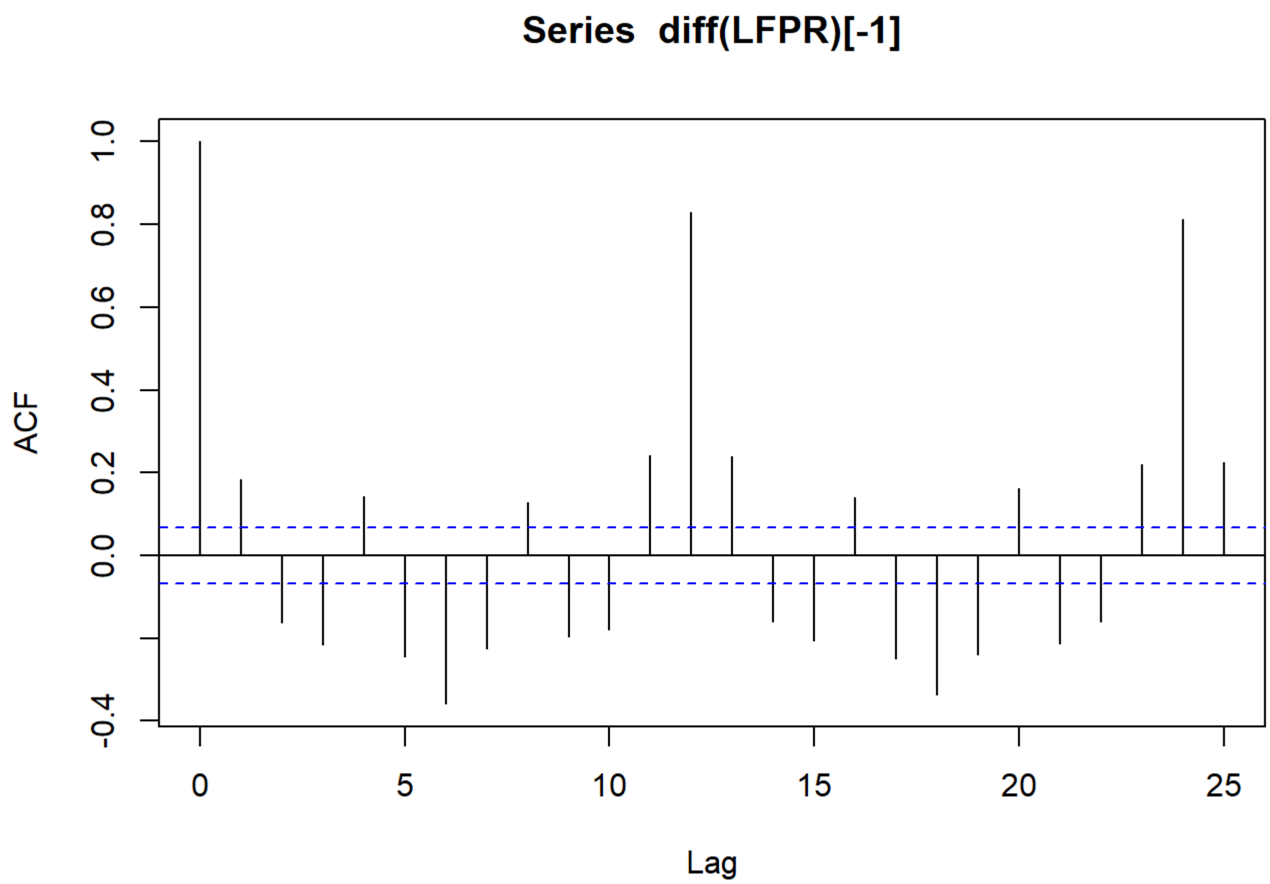
```
acf(LFPR)
```



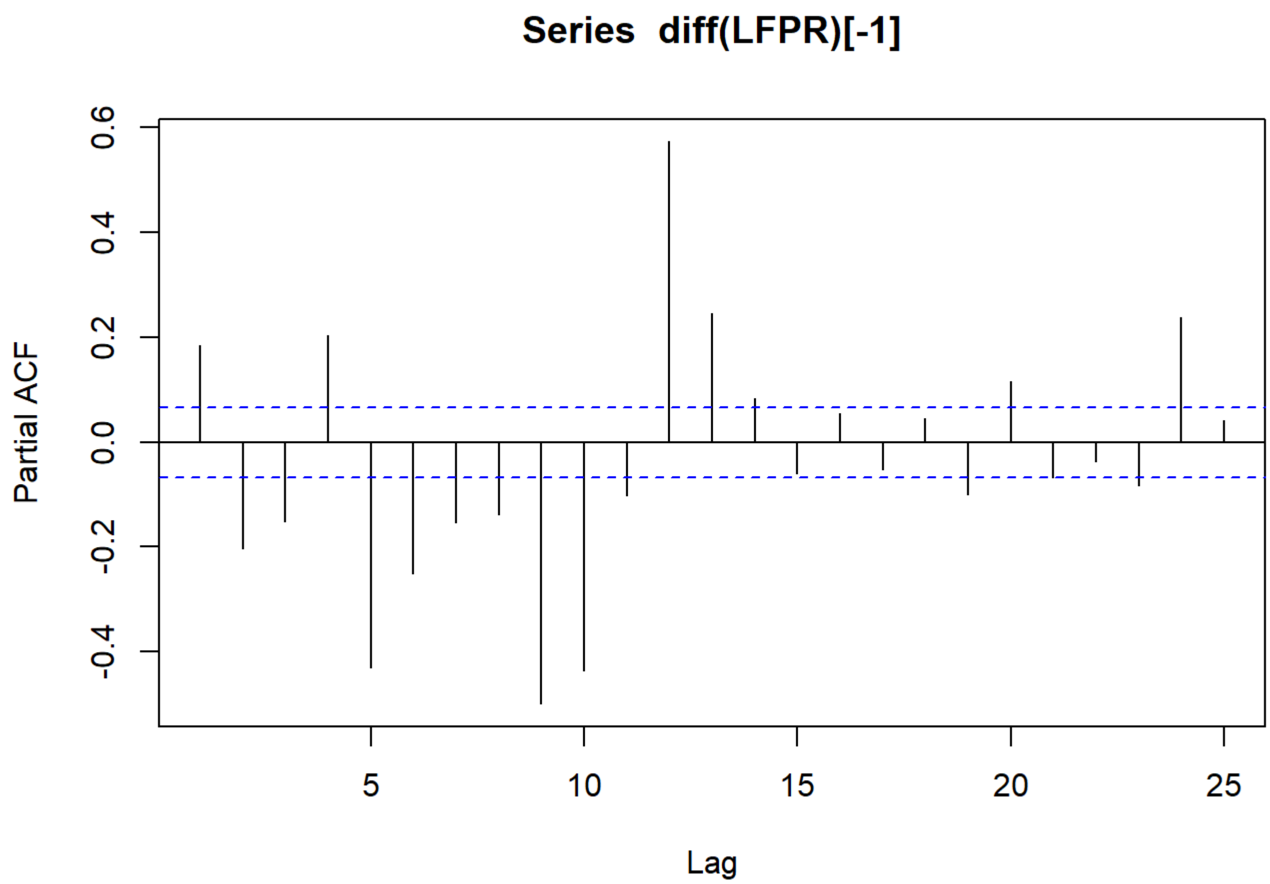

```
pacf(LFPR)
```




```
acf(diff(LFPR)[-1], lag.max=25) # suggests periodicity
```



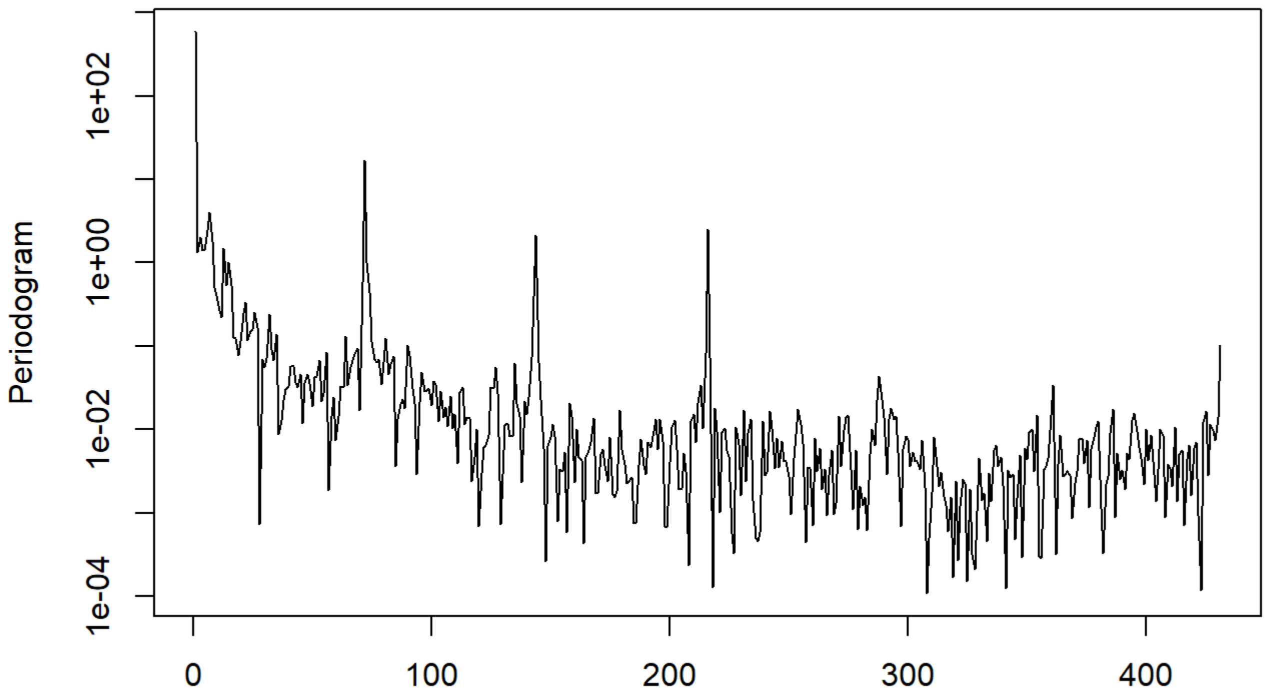

```
pacf(diff(LFPR)[-1], lag.max=25)
```



Model 1: SARIMA Model

i. Periodogram

```
tmp1 <- abs(fft(LFPR))^2/(2*pi*length(LFPR))
plot(2:(length(LFPR)/2)-1, tmp1[2:(length(LFPR)/2)],
     ylab="Periodogram", log="y", type='l',xlab='')
```



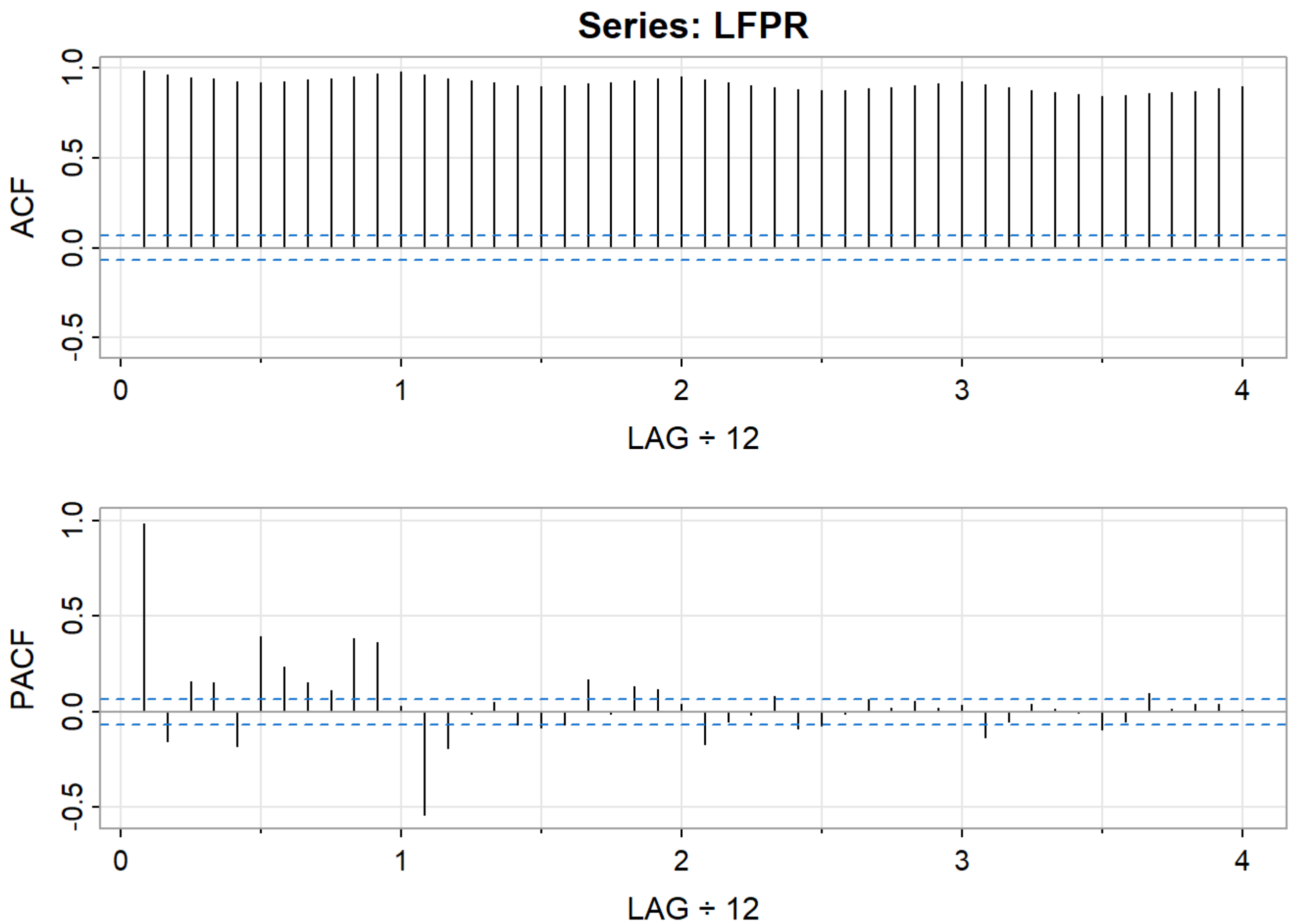

```
ordered <- order(tmp1[1:(length(LFPR)/2)], decreasing = T)
head(ordered[ordered > 12], 3) - 1 # peaks - use as frequency
```

```
## [1]  72 216 144
```

Since the peaks in the periodogram are all to multiples of 12, it might be due to underlying economic cycles of 1 year, 1/2 year, or 1/4 year for LFPR, which suggests the presence of harmonics in the data. Therefore, we consider seasonally differencing the data, while still model using a SARIMA framework with S=12 (since 1/12 is the main frequency).

ii. Construct SARIMA Model

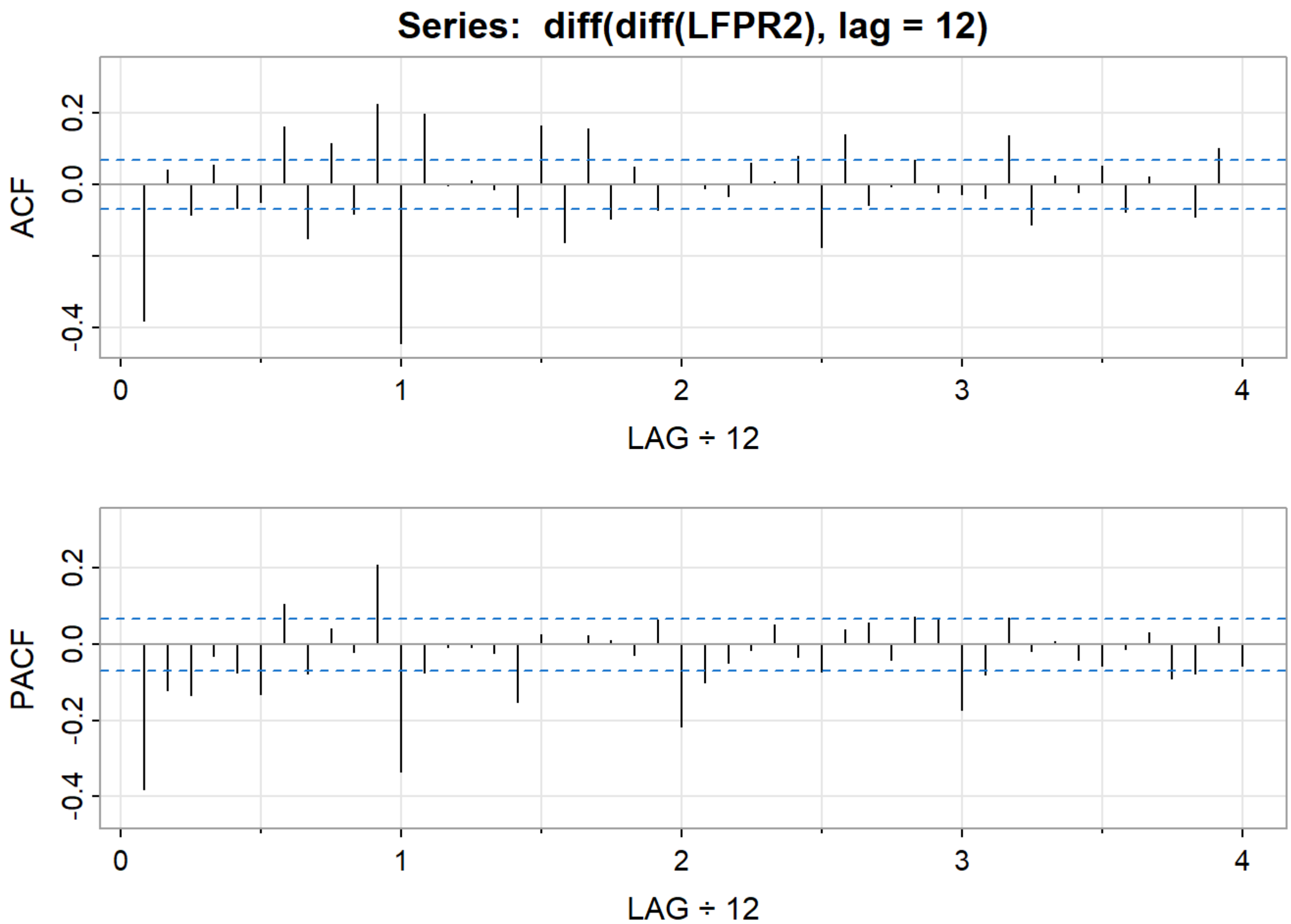
```
LFPR2 <- ts(LFPR, frequency = 12)
acf2(LFPR2, main = 'Series: LFPR')
```




```
##      [,1]  [,2] [,3] [,4]  [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## ACF  0.98  0.96 0.95 0.94  0.92 0.92 0.92 0.93 0.94  0.95  0.97  0.98  0.96
## PACF 0.98 -0.16 0.16 0.15 -0.18 0.40 0.23 0.15 0.11  0.38  0.36  0.03 -0.54
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF   0.94  0.93  0.92  0.90  0.90  0.90  0.91  0.92  0.93  0.94  0.95  0.94
## PACF -0.19 -0.01  0.05 -0.07 -0.08 -0.07  0.17 -0.01  0.13  0.12  0.04 -0.17
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
## ACF   0.92  0.90  0.89  0.88  0.87  0.88  0.89  0.89  0.90  0.92  0.92  0.91
## PACF -0.05 -0.02  0.08 -0.09 -0.07 -0.01  0.07  0.02  0.05  0.02  0.03 -0.14
##      [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF   0.89  0.87  0.86  0.85  0.84  0.85  0.86  0.86  0.87  0.89  0.89
## PACF -0.05  0.04  0.01 -0.01 -0.10 -0.05  0.10  0.01  0.04  0.04  0.01
```

Based on the PACF plot, we see a significant spike of the PACF plot a 1 year. We consider modeling LFPR with S=12, while checking with the seasonally differenced data to confirm this.

```
acf2(diff(diff(LFPR2), lag=12))
```




```
##       [,1]  [,2]  [,3]  [,4]  [,5]  [,6] [,7]  [,8] [,9] [,10] [,11] [,12]
## ACF  -0.38  0.04 -0.08  0.06 -0.06 -0.05 0.16 -0.15 0.11 -0.08  0.22 -0.44
## PACF -0.38 -0.12 -0.13 -0.03 -0.07 -0.13 0.10 -0.08 0.04 -0.02  0.21 -0.33
##      [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## ACF   0.20  0.00  0.01 -0.01 -0.09  0.16 -0.16  0.15 -0.10  0.05 -0.07  0.00
## PACF -0.08 -0.01 -0.01 -0.02 -0.15  0.03  0.00  0.02  0.01 -0.03  0.06 -0.22
##      [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36]
## ACF  -0.01 -0.03  0.06  0.01  0.08 -0.17  0.14 -0.06 -0.01  0.07 -0.02 -0.03
## PACF -0.10 -0.05 -0.02  0.05 -0.03 -0.07  0.04  0.06 -0.04  0.07  0.06 -0.17
##      [,37] [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF  -0.04  0.14 -0.11  0.03 -0.02  0.05 -0.08  0.02  0.00 -0.09  0.10  0.00
## PACF -0.08  0.07 -0.02  0.01 -0.04 -0.06 -0.01  0.03 -0.09 -0.08  0.05 -0.06
```

Based on the ACF and PACF of the seasonally differenced series, we conclude that AR order of 1 and MA order of 2 should be used on the seasonal component, as the periodic trend does not persist.

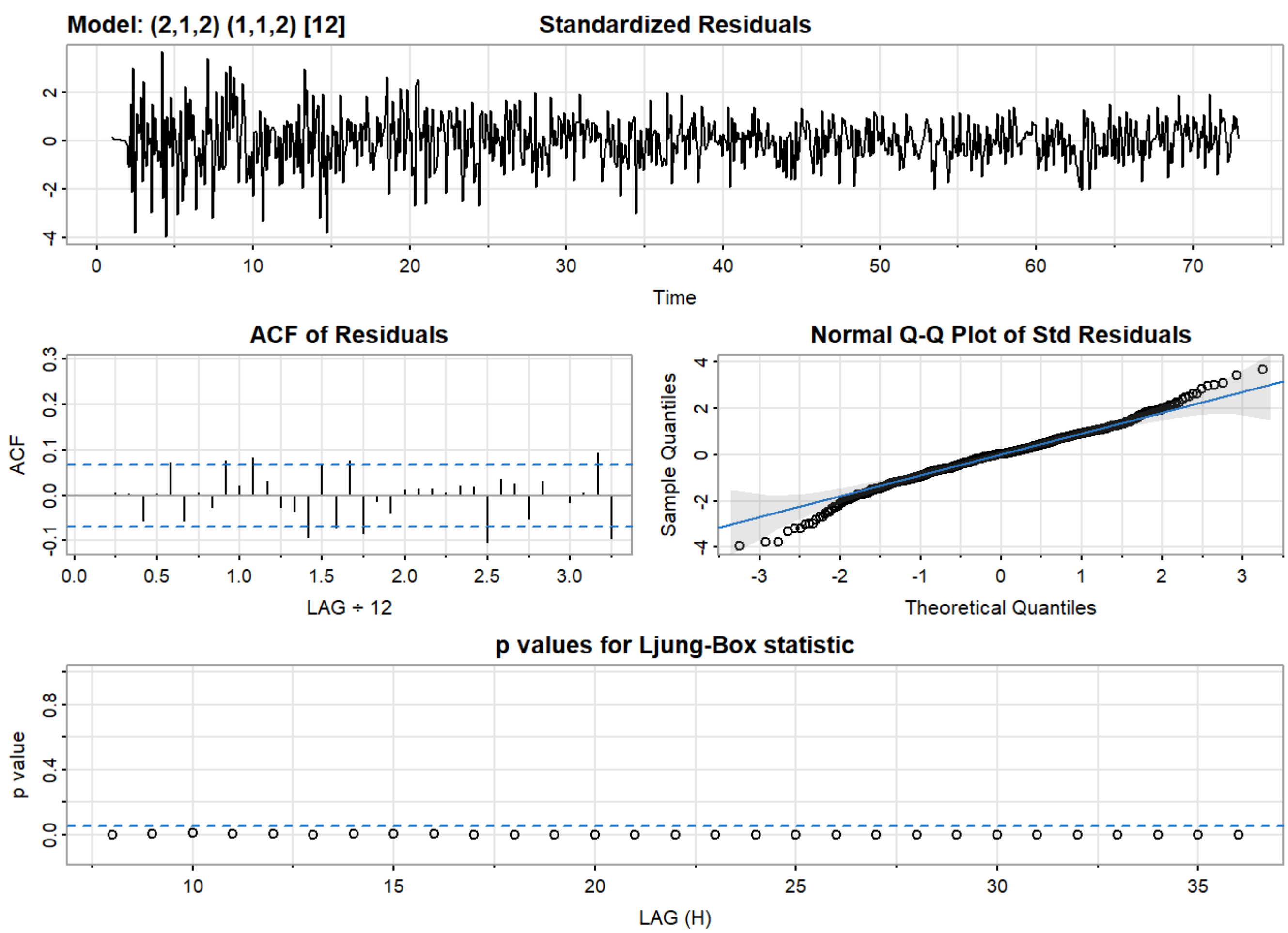
```
best.sarima.model <- auto.arima(LFPR2, seasonal=T)
                               # ARIMA(2,1,2)(2,0,0)[12]
                               # AIC: -0.2088
print(best.sarima.model)
```

```
## Series: LFPR2
## ARIMA(2,1,2)(1,1,2)[12]
##
## Coefficients:
##           ar1     ar2      ma1      ma2    sar1     sma1     sma2
##       -0.0857  0.2879  -0.2965  -0.3449  0.0169  -0.6660  -0.0517
## s.e.   0.2233  0.0797   0.2248   0.1459  0.3453   0.3436   0.2373
##
## sigma^2 = 0.04662:  log likelihood = 96.85
## AIC=-177.7   AICc=-177.53   BIC=-139.73
```

```
sarima(LFPR2,p=2,d=1,q=2,P=1,D=1,Q=2,S=12)
```

```
## initial  value -1.299172
## iter   2 value -1.407603
## iter   3 value -1.549282
## iter   4 value -1.553284
## iter   5 value -1.555372
## iter   6 value -1.557306
## iter   7 value -1.561361
## iter   8 value -1.562188
## iter   9 value -1.563271
## iter  10 value -1.563775
## iter  11 value -1.563926
## iter  12 value -1.563936
## iter  13 value -1.563943
## iter  14 value -1.563989
## iter  15 value -1.563998
## iter  16 value -1.564001
## iter  17 value -1.564001
## iter  18 value -1.564014
## iter  19 value -1.564019
## iter  20 value -1.564020
## iter  21 value -1.564021
## iter  22 value -1.564022
## iter  23 value -1.564025
## iter  24 value -1.564029
## iter  25 value -1.564030
## iter  26 value -1.564031
## iter  27 value -1.564031
## iter  28 value -1.564031
## iter  29 value -1.564031
## iter  29 value -1.564031
## iter  29 value -1.564031
## final  value -1.564031
## converged
## initial  value -1.530796
## iter   2 value -1.531166
## iter   3 value -1.531906
## iter   4 value -1.531943
## iter   5 value -1.531974
## iter   6 value -1.531984
## iter   7 value -1.532014
## iter   8 value -1.532043
## iter   9 value -1.532063
## iter  10 value -1.532098
## iter  11 value -1.532165
## iter  12 value -1.532291
## iter  13 value -1.532468
## iter  14 value -1.532667
## iter  15 value -1.532692
## iter  16 value -1.532694
## iter  17 value -1.532695
## iter  18 value -1.532696
## iter  19 value -1.532698
```

```
## iter  20 value -1.532703
## iter  21 value -1.532711
## iter  22 value -1.532719
## iter  23 value -1.532728
## iter  24 value -1.532736
## iter  25 value -1.532740
## iter  26 value -1.532746
## iter  27 value -1.532746
## iter  28 value -1.532746
## iter  28 value -1.532746
## iter  28 value -1.532746
## final  value -1.532746
## converged
```



```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control = list
(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1     ar2      ma1      ma2    sar1     sma1     sma2
##       -0.0857  0.2879  -0.2965  -0.3449  0.0169  -0.6660  -0.0517
## s.e.   0.2233  0.0797   0.2248   0.1459  0.3453   0.3436   0.2373
##
## sigma^2 estimated as 0.04618:  log likelihood = 96.85,  aic = -177.7
##
## $degrees_of_freedom
## [1] 844
##
## $ttable
##      Estimate     SE t.value p.value
## ar1   -0.0857 0.2233 -0.3839  0.7012
## ar2    0.2879 0.0797  3.6133  0.0003
## ma1   -0.2965 0.2248 -1.3192  0.1874
## ma2   -0.3449 0.1459 -2.3635  0.0183
## sar1   0.0169 0.3453  0.0489  0.9610
## sma1  -0.6660 0.3436 -1.9380  0.0530
## sma2  -0.0517 0.2373 -0.2180  0.8275
##
## $AIC
## [1] -0.2088144
##
## $AICc
## [1] -0.2086583
##
## $BIC
## [1] -0.1641948
```

Although the ACF plot and QQ-plot of the residuals both look good, confirming a part of the assumptions, the standardized residual plot shows significant sign of heteroskedasticity - as the variability of the residuals decreases when lag value increases, and the Ljung-Box statistic has p-values very close to 0, indicating non-independent errors. We decide to fix this by fitting a GARCH model on the residuals of our SARIMA model.

```
sarima_resid <- residuals(best.sarima.model)
Box.test(sarima_resid, lag = 24, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  sarima_resid
## X-squared = 54.031, df = 24, p-value = 0.0004222
```

```
Box.test(sarima_resid, lag = 60, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  sarima_resid
## X-squared = 130.23, df = 60, p-value = 4.15e-07
```

A Ljung-Box test on the residuals confirms the presence of serial correlation.

iii. Analyze Residuals with GARCH

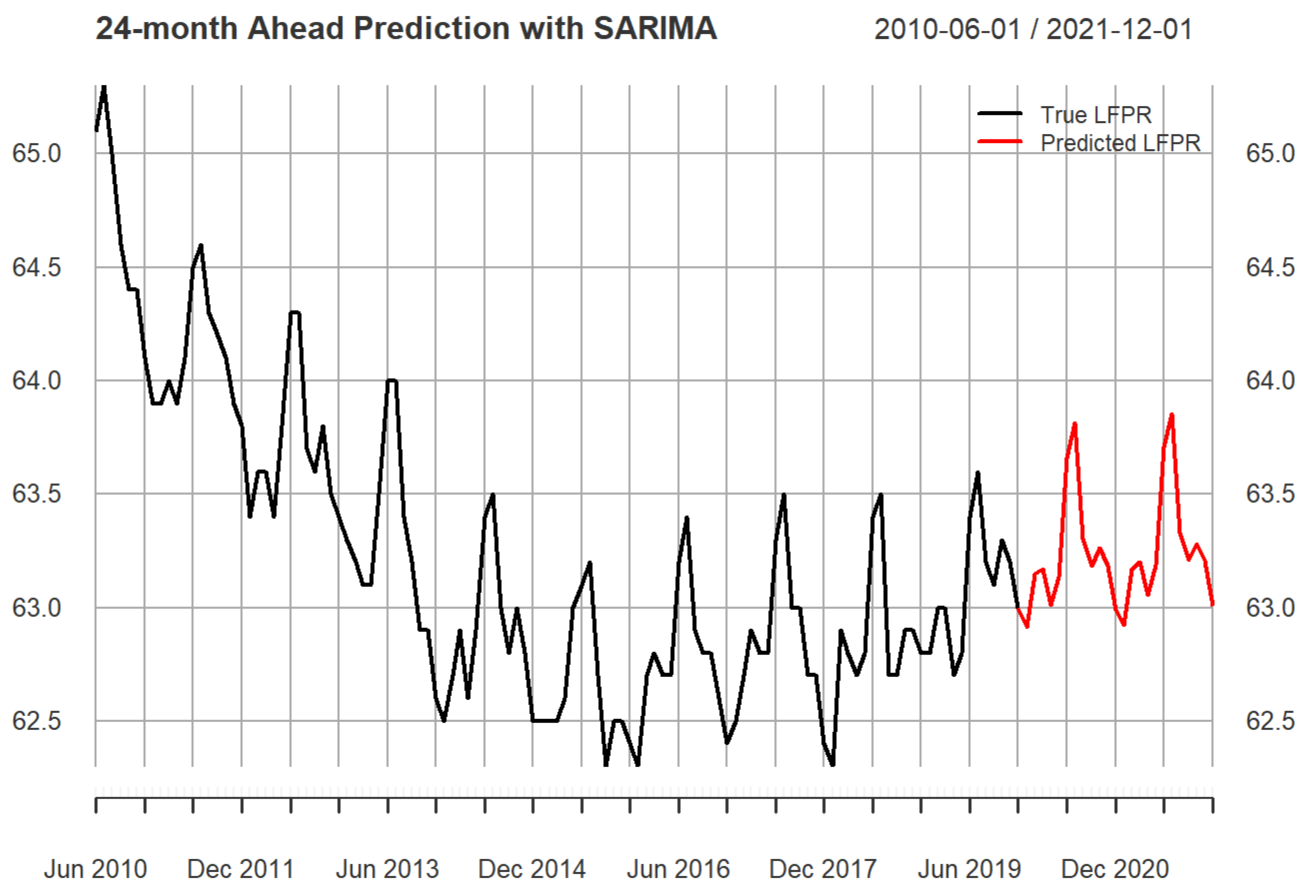
```
spec <- ugarchspec(mean.model = list(armaOrder= c(0,0)),
                   variance.model = list(garchOrder= c(1,2)))
best.garch.model <- ugarchfit(spec, data = sarima_resid)
print(best.garch.model)
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(1,2)
## Mean Model   : ARFIMA(0,0,0)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     -0.003519    0.006443 -0.54612 0.584986
## omega   0.000338    0.000290  1.16877 0.242497
## alpha1  0.051266    0.017508  2.92811 0.003410
## beta1   0.092516    0.021481  4.30678 0.000017
## beta2   0.846713    0.021396 39.57402 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     -0.003519    0.006862 -0.51280 0.608091
## omega   0.000338    0.000343  0.98576 0.324250
## alpha1  0.051266    0.021544  2.37954 0.017334
## beta1   0.092516    0.013721  6.74273 0.000000
## beta2   0.846713    0.013760 61.53220 0.000000
##
## LogLikelihood : 163.4386
##
## Information Criteria
## ------------------------------------
##
## Akaike       -0.36676
## Bayes        -0.33920
## Shibata      -0.36682
## Hannan-Quinn -0.35621
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.4336  0.5102
## Lag[2*(p+q)+(p+q)-1][2]    0.4895  0.6985
## Lag[4*(p+q)+(p+q)-1][5]    1.4419  0.7542
## d.o.f=0
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      2.395  0.1217
## Lag[2*(p+q)+(p+q)-1][8]     5.634  0.2739
```

```
## Lag[4*(p+q)+(p+q)-1][14]     6.498  0.5719
## d.o.f=3
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[4]    0.6232 0.500 2.000  0.4299
## ARCH Lag[6]    0.8754 1.461 1.711  0.7838
## ARCH Lag[8]    1.0034 2.368 1.583  0.9238
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  1.122
## Individual Statistics:
## mu     0.10370
## omega  0.08255
## alpha1 0.27676
## beta1  0.13354
## beta2  0.13945
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.28 1.47 1.88
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value   prob sig
## Sign Bias           0.5393 0.5898
## Negative Sign Bias  0.2395 0.8108
## Positive Sign Bias  1.0712 0.2844
## Joint Effect        1.2340 0.7449
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     31.00      0.04037
## 2    30     42.88      0.04675
## 3    40     48.22      0.14781
## 4    50     69.56      0.02823
##
##
## Elapsed time : 0.108125
```

In general, a higher GARCH order captures longer-term persistence in volatility and allows for more complex volatility dynamics. It considers a larger number of past squared residuals in the conditional variance equation. On the other hand, a higher ARCH order captures shorter-term volatility clustering or heteroscedasticity in the data. Here, a GARCH order of 1 and ARCH order of 2 returns p-value much higher than 0.05 for different lags on both the weighted Ljung-Box test of on standardized residuals and standardized squared residuals. Although the Person goodness-of-fit test does not return p-values greater than 0.05 for some lags, it solves the issue of heteroscedasticity in our original SARIMA residuals.

```
addLegend("topright", legend.names = c('True LFPR', 'Predicted LFPR'),
          col = c('black', 'red'), lty = 1, lwd = 2, cex = 0.8)
```

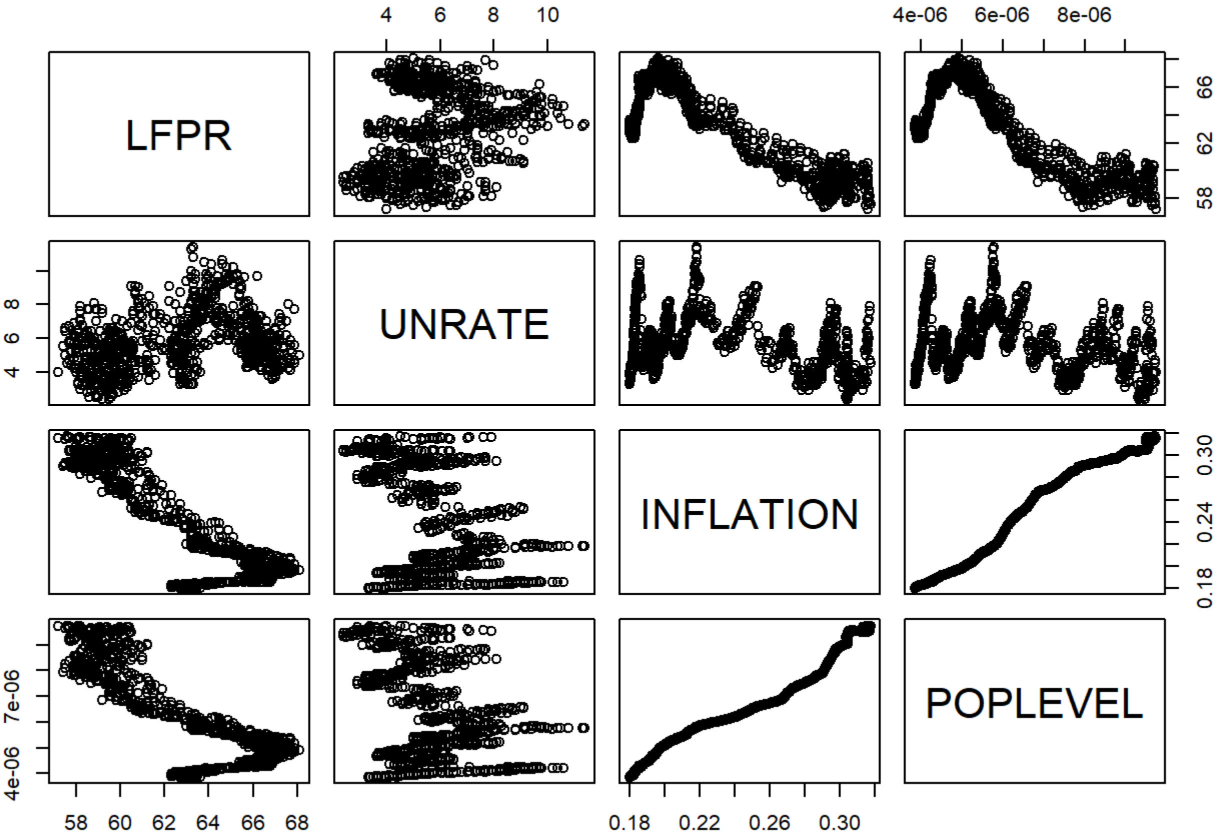


Model 2: Vector Autoregressive and Granger Causality

i. Variable Selection

```
dat <- xts::cbind.xts(LFPR,
                      UNRATE,
                      1/log(INFLATION),
                      1/POPLEVEL)
names(dat) <- c('LFPR', 'UNRATE', 'INFLATION', 'POPLEVEL')
```

```
pairs(data.frame(dat))
```




```
cor(dat)
```

```
##                 LFPR     UNRATE  INFLATION   POPLEVEL
## LFPR       1.0000000  0.2547739 -0.8712033 -0.8247537
## UNRATE     0.2547739  1.0000000 -0.3161169 -0.3087008
## INFLATION -0.8712033 -0.3161169  1.0000000  0.9893472
## POPLEVEL  -0.8247537 -0.3087008  0.9893472  1.0000000
```

```
summary(lm(LFPR~UNRATE, data=dat)) # very poor R^2
```

```
##
## Call:
## lm(formula = LFPR ~ UNRATE, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.7823 -2.4758 -0.1224  2.8652  5.5837
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 60.24621    0.35552 169.459  < 2e-16 ***
## UNRATE       0.46002    0.05947   7.735 2.88e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.957 on 862 degrees of freedom
## Multiple R-squared:  0.06491,    Adjusted R-squared:  0.06382
## F-statistic: 59.84 on 1 and 862 DF,  p-value: 2.876e-14
```

```
summary(lm(LFPR~INFLATION, data=dat))
```

```
##
## Call:
## lm(formula = LFPR ~ INFLATION, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6694 -0.9499  0.2099  1.1661  3.1992
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  76.2446     0.2615   291.6   <2e-16 ***
## INFLATION   -56.6464     1.0872   -52.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.501 on 862 degrees of freedom
## Multiple R-squared:  0.759,  Adjusted R-squared:  0.7587
## F-statistic:  2715 on 1 and 862 DF,  p-value: < 2.2e-16
```

```
summary(lm(LFPR~POPLEVEL, data=dat))
```

```
##
## Call:
## lm(formula = LFPR ~ POPLEVEL, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7838 -1.4120  0.1888  1.4355  3.6229
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.141e+01  2.076e-01  344.03   <2e-16 ***
## POPLEVEL    -1.367e+06  3.192e+04  -42.82   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.729 on 862 degrees of freedom
## Multiple R-squared:  0.6802, Adjusted R-squared:  0.6798
## F-statistic:  1834 on 1 and 862 DF,  p-value: < 2.2e-16
```

Based on this, we drop UNRATE from our dataframe, and only keep LFPR with 1/log(INFLATION) and 1/POPLEVEL.

```
dat <- dat[ , !names(dat)=='UNRATE']
```

ii. Construct VAR Model

```
freq = 1/12
df = data.frame(trend = 1:nrow(dat),
                cos1 = cos(2*pi*freq*1:nrow(dat)),
                sin1 = sin(2*pi*freq*1:nrow(dat)))
VARselect(dat, type = "const", exogen = df, lag.max=30)$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     25     14     14     25
```

```
                                                   # AIC: -58.919
var.model <- VAR(dat, p = 25, type = "const", exogen = df)
```

```
freq = 1/12
df = data.frame(cos1 = cos(2*pi*freq*1:nrow(dat)),
                sin1 = sin(2*pi*freq*1:nrow(dat)))
VARselect(dat, type = "const", exogen = df, lag.max=30)$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     25     14     14     25
```

```
                                                    # AIC: -58.900
var.model <- VAR(dat, p = 25, type = "const", exogen = df)
```

```
df = data.frame(trend = 1:nrow(dat))
VARselect(dat, type = "const", exogen = df, lag.max=30)$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     25     14     14     25
```

```
                                                    # AIC: -58.901
var.model <- VAR(dat, p = 25, type = "const", exogen = df)
```

```
VARselect(dat, type = "const", lag.max=30)$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     25     14     14     25
```

```
                                     # AIC: -58.881
var.model <- VAR(dat, p = 25, type = "const")
```

```
freq = 1/12
freq1 = 1/6

df = data.frame(cos1 = cos(2*pi*freq*1:nrow(dat)),
                sin1 = sin(2*pi*freq*1:nrow(dat)),
                cos2 = cos(2*pi*freq1*1:nrow(dat)),
                sin2 = sin(2*pi*freq1*1:nrow(dat)))
VARselect(dat, type = "const", exogen = df, lag.max=30)$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     25     14     14     25
```

```
                                                    # AIC: -58.914
var.model <- VAR(dat, p = 25, type = "const", exogen = df)
```

```
freq = 1/12
freq1 = 1/6

df = data.frame(trend = 1:nrow(dat),
                cos1 = cos(2*pi*freq*1:nrow(dat)),
                sin1 = sin(2*pi*freq*1:nrow(dat)),
                cos2 = cos(2*pi*freq1*1:nrow(dat)),
                sin2 = sin(2*pi*freq1*1:nrow(dat)))
VARselect(dat, type = "const", exogen = df, lag.max=30)$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     25     14     14     25
```

```
                                                    # AIC: -58.933
var.model <- VAR(dat, p = 25, type = "const", exogen = df)
```

```
freq = 1/12
freq1 = 1/6
freq2 = 1/4

df = data.frame(cos1 = cos(2*pi*freq*1:nrow(dat)),
                sin1 = sin(2*pi*freq*1:nrow(dat)),
                cos2 = cos(2*pi*freq1*1:nrow(dat)),
                sin2 = sin(2*pi*freq1*1:nrow(dat)),
                cos3 = cos(2*pi*freq2*1:nrow(dat)),
                sin3 = sin(2*pi*freq2*1:nrow(dat)))
VARselect(dat, type = "const", exogen = df, lag.max=30)$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     25     15     14     25
```

```
                                                    # AIC: -58.928
var.model <- VAR(dat, p = 25, type = "const", exogen = df)
```

```
freq = 1/12
freq1 = 1/6
freq2 = 1/4

df = data.frame(trend = 1:nrow(dat),
                cos1 = cos(2*pi*freq*1:nrow(dat)),
                sin1 = sin(2*pi*freq*1:nrow(dat)),
                cos2 = cos(2*pi*freq1*1:nrow(dat)),
                sin2 = sin(2*pi*freq1*1:nrow(dat)),
                cos3 = cos(2*pi*freq2*1:nrow(dat)),
                sin3 = sin(2*pi*freq2*1:nrow(dat)))
VARselect(dat, type = "const", exogen = df, lag.max=30)$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     25     15     14     25
```

```
                                                    # BEST MODEL!
                                                    # AIC: -58.947
best.var.model <- VAR(dat, p = 25, type = "const", exogen = df)
```

```
summary(best.var.model)$varresult$LFPR
```

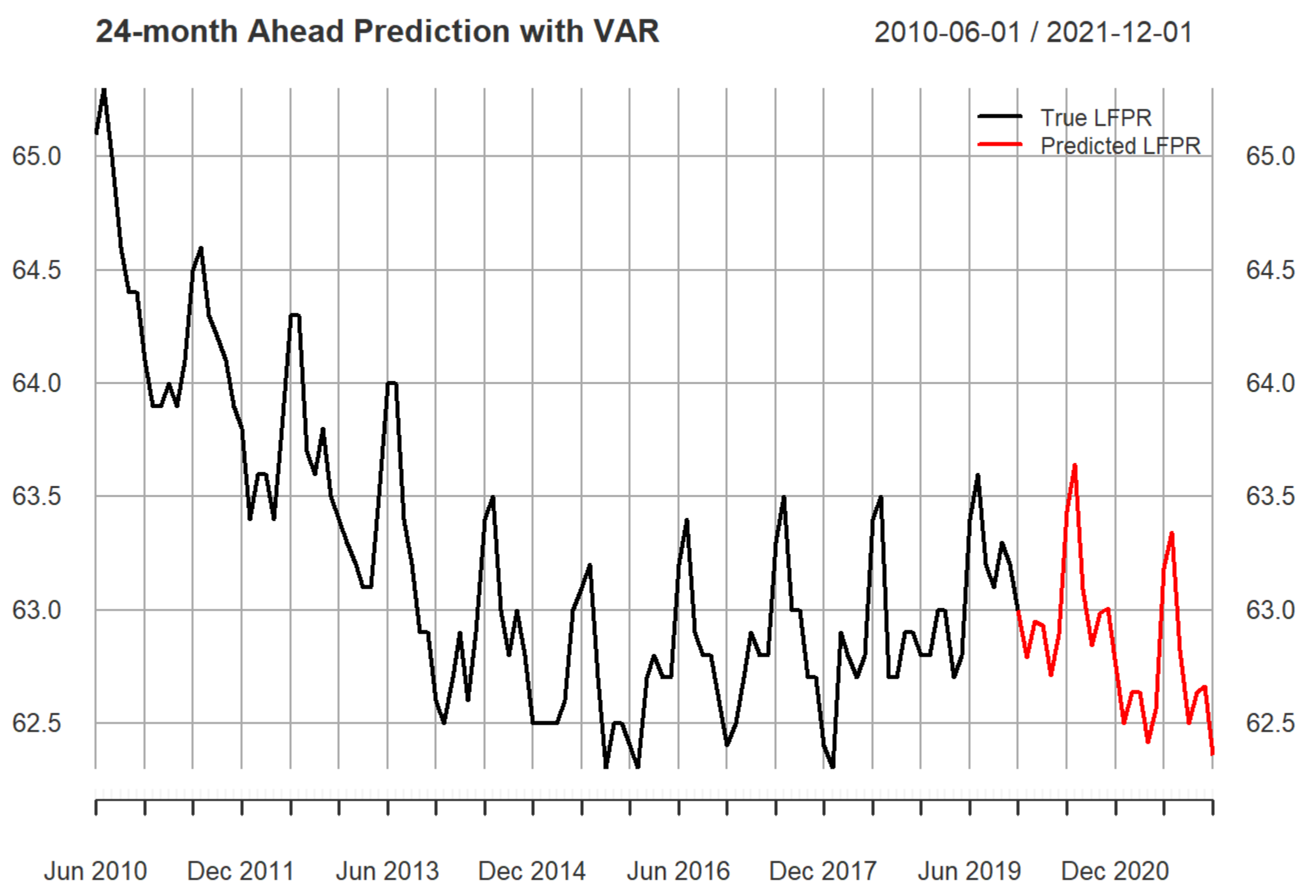
```
##
## Call:
## lm(formula = y ~ -1 + ., data = datamat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.65445 -0.11719  0.00663  0.12516  0.60602
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## LFPR.l1        6.615e-01  3.570e-02  18.529  < 2e-16 ***
## INFLATION.l1  -9.924e+01  4.487e+01  -2.212  0.02728 *
## POPLEVEL.l1   -2.559e+06  1.738e+06  -1.472  0.14146
## LFPR.l2        1.219e-01  4.199e-02   2.903  0.00380 **
## INFLATION.l2   6.054e+01  7.152e+01   0.846  0.39756
## POPLEVEL.l2    4.393e+05  2.566e+06   0.171  0.86411
## LFPR.l3       -3.001e-03  4.177e-02  -0.072  0.94275
## INFLATION.l3   9.534e+01  7.021e+01   1.358  0.17490
## POPLEVEL.l3    8.669e+05  2.560e+06   0.339  0.73499
## LFPR.l4        7.337e-02  4.179e-02   1.756  0.07950 .
## INFLATION.l4  -6.323e+01  6.925e+01  -0.913  0.36154
## POPLEVEL.l4    1.092e+06  2.546e+06   0.429  0.66804
## LFPR.l5       -4.827e-02  4.181e-02  -1.155  0.24862
## INFLATION.l5  -1.788e+01  6.835e+01  -0.262  0.79369
## POPLEVEL.l5   -3.232e+06  2.552e+06  -1.267  0.20571
## LFPR.l6        2.824e-02  4.181e-02   0.675  0.49964
## INFLATION.l6   1.162e+02  6.831e+01   1.701  0.08943 .
## POPLEVEL.l6    7.174e+06  2.555e+06   2.807  0.00513 **
## LFPR.l7        1.176e-01  4.167e-02   2.824  0.00487 **
## INFLATION.l7  -9.588e+01  6.671e+01  -1.437  0.15109
## POPLEVEL.l7   -2.613e+06  2.568e+06  -1.018  0.30909
## LFPR.l8       -8.127e-02  4.208e-02  -1.931  0.05383 .
## INFLATION.l8  -1.040e+02  6.588e+01  -1.579  0.11472
## POPLEVEL.l8   -3.777e+04  2.568e+06  -0.015  0.98827
## LFPR.l9        4.043e-02  4.202e-02   0.962  0.33631
## INFLATION.l9   6.992e+01  6.544e+01   1.068  0.28565
## POPLEVEL.l9   -7.547e+05  2.543e+06  -0.297  0.76671
## LFPR.l10      -8.585e-03  4.174e-02  -0.206  0.83709
## INFLATION.l10  1.500e+02  6.528e+01   2.298  0.02184 *
## POPLEVEL.l10   2.791e+06  2.534e+06   1.101  0.27113
## LFPR.l11       9.068e-02  4.152e-02   2.184  0.02927 *
## INFLATION.l11 -1.095e+02  6.514e+01  -1.680  0.09329 .
## POPLEVEL.l11  -1.205e+06  2.526e+06  -0.477  0.63354
## LFPR.l12       3.629e-01  4.170e-02   8.703  < 2e-16 ***
## INFLATION.l12 -8.841e+01  6.543e+01  -1.351  0.17702
## POPLEVEL.l12  -2.951e+06  2.520e+06  -1.171  0.24186
## LFPR.l13      -1.858e-01  4.299e-02  -4.321 1.76e-05 ***
## INFLATION.l13  1.057e+02  6.570e+01   1.609  0.10801
## POPLEVEL.l13  -1.392e+06  2.534e+06  -0.550  0.58279
## LFPR.l14      -9.289e-02  4.145e-02  -2.241  0.02531 *
## INFLATION.l14  4.177e+01  6.577e+01   0.635  0.52560
## POPLEVEL.l14  -2.743e+04  2.524e+06  -0.011  0.99133
```

```
## LFPR.l15      -1.027e-01  4.134e-02  -2.485  0.01319 *
## INFLATION.l15  1.924e+01  6.581e+01   0.292  0.77007
## POPLEVEL.l15   2.857e+06  2.515e+06   1.136  0.25624
## LFPR.l16       2.704e-02  4.154e-02   0.651  0.51521
## INFLATION.l16 -1.143e+02  6.566e+01  -1.741  0.08204 .
## POPLEVEL.l16   3.140e+06  2.514e+06   1.249  0.21215
## LFPR.l17      -8.126e-02  4.141e-02  -1.962  0.05008 .
## INFLATION.l17  2.089e+01  6.578e+01   0.318  0.75093
## POPLEVEL.l17  -4.652e+06  2.520e+06  -1.846  0.06527 .
## LFPR.l18       7.169e-02  4.146e-02   1.729  0.08421 .
## INFLATION.l18  1.001e+02  6.572e+01   1.524  0.12798
## POPLEVEL.l18   2.740e+06  2.534e+06   1.081  0.27998
## LFPR.l19      -1.235e-01  4.141e-02  -2.981  0.00296 **
## INFLATION.l19 -3.893e+01  6.527e+01  -0.596  0.55109
## POPLEVEL.l19  -2.609e+06  2.525e+06  -1.033  0.30185
## LFPR.l20       1.303e-01  4.123e-02   3.161  0.00163 **
## INFLATION.l20 -5.547e+01  6.454e+01  -0.859  0.39034
## POPLEVEL.l20  -1.679e+06  2.519e+06  -0.667  0.50514
## LFPR.l21      -1.338e-01  4.097e-02  -3.266  0.00114 **
## INFLATION.l21 -5.140e+00  6.414e+01  -0.080  0.93615
## POPLEVEL.l21   2.479e+06  2.521e+06   0.983  0.32580
## LFPR.l22       5.192e-02  4.106e-02   1.264  0.20648
## INFLATION.l22  3.565e+00  6.205e+01   0.057  0.95419
## POPLEVEL.l22   5.172e+05  2.521e+06   0.205  0.83751
## LFPR.l23      -8.750e-03  4.101e-02  -0.213  0.83108
## INFLATION.l23 -6.429e+01  5.949e+01  -1.081  0.28018
## POPLEVEL.l23   1.666e+06  2.518e+06   0.662  0.50842
## LFPR.l24       2.246e-01  4.052e-02   5.545 4.08e-08 ***
## INFLATION.l24  1.417e+02  5.923e+01   2.393  0.01697 *
## POPLEVEL.l24  -2.991e+06  2.522e+06  -1.186  0.23604
## LFPR.l25      -1.633e-01  3.481e-02  -4.692 3.22e-06 ***
## INFLATION.l25 -7.197e+01  3.818e+01  -1.885  0.05980 .
## POPLEVEL.l25   8.235e+05  1.698e+06   0.485  0.62781
## const          4.022e+00  1.051e+00   3.828  0.00014 ***
## trend         -1.245e-03  3.088e-04  -4.030 6.14e-05 ***
## cos1          -7.064e-02  2.706e-02  -2.610  0.00923 **
## sin1          -2.038e-02  2.743e-02  -0.743  0.45774
## cos2           5.629e-02  2.113e-02   2.664  0.00788 **
## sin2          -4.581e-02  2.119e-02  -2.162  0.03094 *
## cos3          -4.213e-02  3.022e-02  -1.394  0.16368
## sin3          -9.494e-02  2.990e-02  -3.175  0.00156 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2024 on 756 degrees of freedom
## Multiple R-squared:  0.9959, Adjusted R-squared:  0.9955
## F-statistic:  2258 on 82 and 756 DF,  p-value: < 2.2e-16
```

```
summary(best.var.model)$corres
```

```
##                    LFPR   INFLATION    POPLEVEL
## LFPR       1.00000000  0.05154618 -0.01352365
## INFLATION  0.05154618  1.00000000 -0.08796712
## POPLEVEL  -0.01352365 -0.08796712  1.00000000
```

```
addLegend("topright", legend.names = c('True LFPR', 'Predicted LFPR'),
          col = c('black', 'red'), lty = 1, lwd = 2, cex = 0.8)
```



iii. Granger Causality

```
df_INFLATION = data.frame(trend = 1:nrow(dat),
                          cos1 = cos(2*pi*freq*1:nrow(dat)),
                          sin1 = sin(2*pi*freq*1:nrow(dat)),
                          cos2 = cos(2*pi*freq1*1:nrow(dat)),
                          sin2 = sin(2*pi*freq1*1:nrow(dat)),
                          cos3 = cos(2*pi*freq2*1:nrow(dat)),
                          sin3 = sin(2*pi*freq2*1:nrow(dat)),
                          dat$POPLEVEL)
VARselect(dat[, c('LFPR', 'INFLATION')],
          type = "const",
          exogen = df_INFLATION,
          lag.max = 50)$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     49     15     14     49
```

```
var.model <- VAR(dat[, c('LFPR', 'INFLATION')],
                 p=49, type = "const", exogen = df_INFLATION)
causality(var.model, cause = "INFLATION")$Granger
```

```
##
##  Granger causality H0: INFLATION do not Granger-cause LFPR
##
## data:  VAR object var.model
## F-Test = 1.6565, df1 = 49, df2 = 1416, p-value = 0.003235
```

```
df_POPLEVEL = data.frame(trend = 1:nrow(dat),
                         cos1 = cos(2*pi*freq*1:nrow(dat)),
                         sin1 = sin(2*pi*freq*1:nrow(dat)),
                         cos2 = cos(2*pi*freq1*1:nrow(dat)),
                         sin2 = sin(2*pi*freq1*1:nrow(dat)),
                         cos3 = cos(2*pi*freq2*1:nrow(dat)),
                         sin3 = sin(2*pi*freq2*1:nrow(dat)),
                         dat$INFLATION)
VARselect(dat[, c('LFPR', 'POPLEVEL')],
          type = "const",
          exogen = df_POPLEVEL,
          lag.max = 50)$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     25     15     14     25
```

```
var.model <- VAR(dat[, c('LFPR', 'POPLEVEL')],
                 p=49, type = "const", exogen = df_POPLEVEL)
causality(var.model, cause = "POPLEVEL")$Granger
```

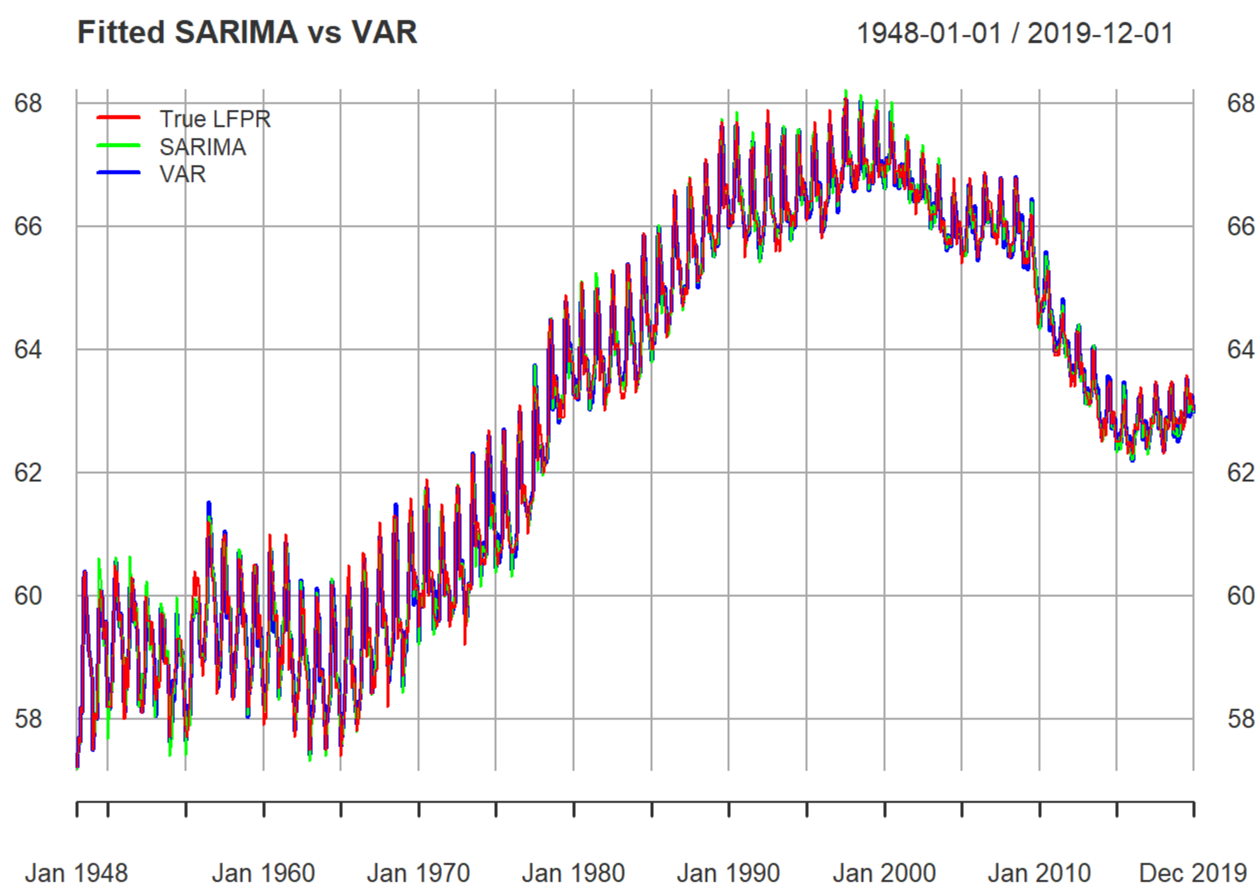
```
##
##  Granger causality H0: POPLEVEL do not Granger-cause LFPR
##
## data:  VAR object var.model
## F-Test = 1.3161, df1 = 49, df2 = 1416, p-value = 0.07242
```

Model Comparison

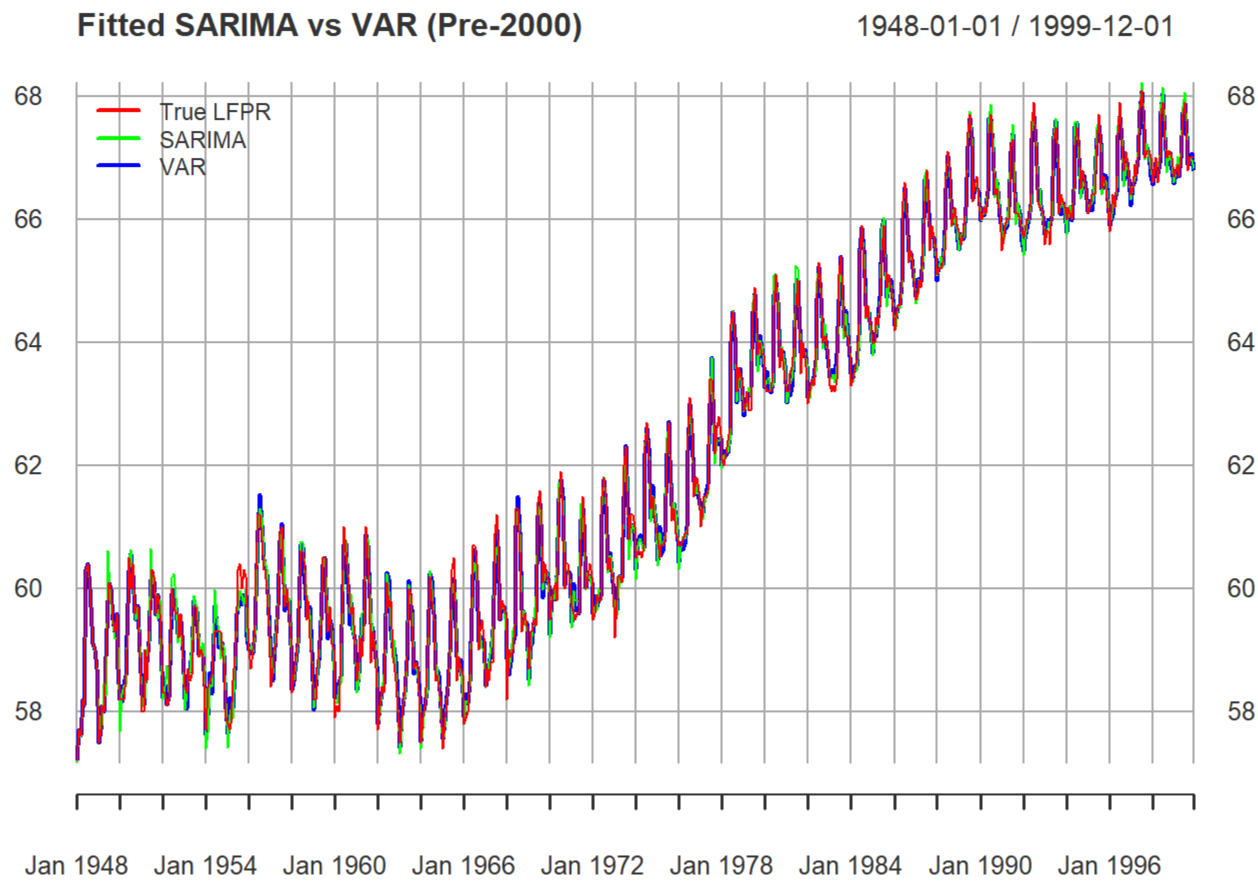
i. Predictions Plots

```
date_range <- seq(as.Date("1948-01-01"),
                  as.Date("2019-12-01"), by = "1 month")

sarima_fitted <- fitted(best.sarima.model)
sarima_fitted <- xts(sarima_fitted, order.by = date_range)

var_fitted <- c(as.numeric(LFPR[1:best.var.model$p]),
                fitted(best.var.model)[,'LFPR'])
var_fitted <- xts(var_fitted, order.by = date_range)

combined_xts <- merge(LFPR, sarima_fitted, var_fitted)
```

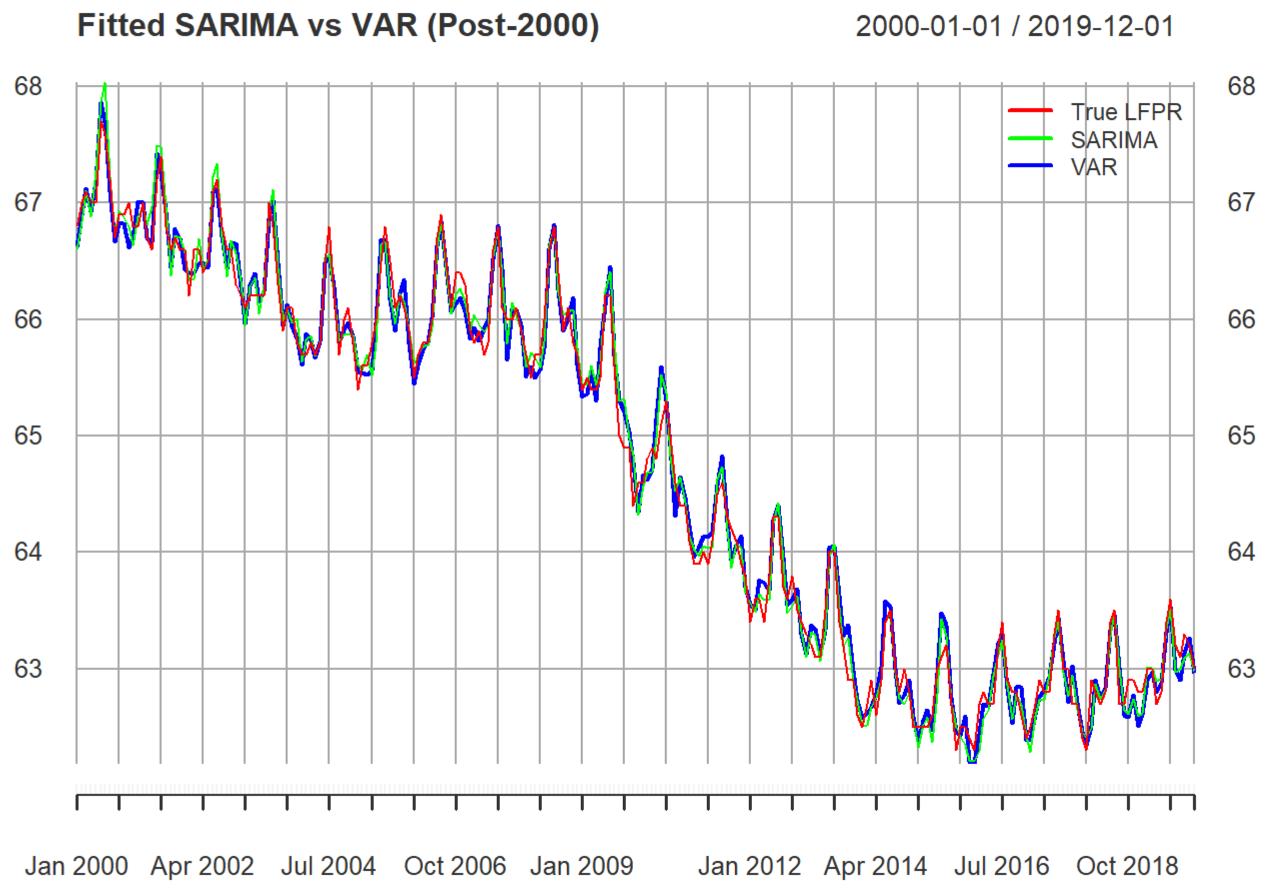
```
addLegend("topleft", legend.names = c('True LFPR', 'SARIMA', 'VAR'),
          col = colors, lty = 1, lwd = 2, cex = 0.8)
```




```
addLegend("topleft", legend.names = c('True LFPR', 'SARIMA', 'VAR'),
          col = colors, lty = 1, lwd = 2, cex = 0.8)
```



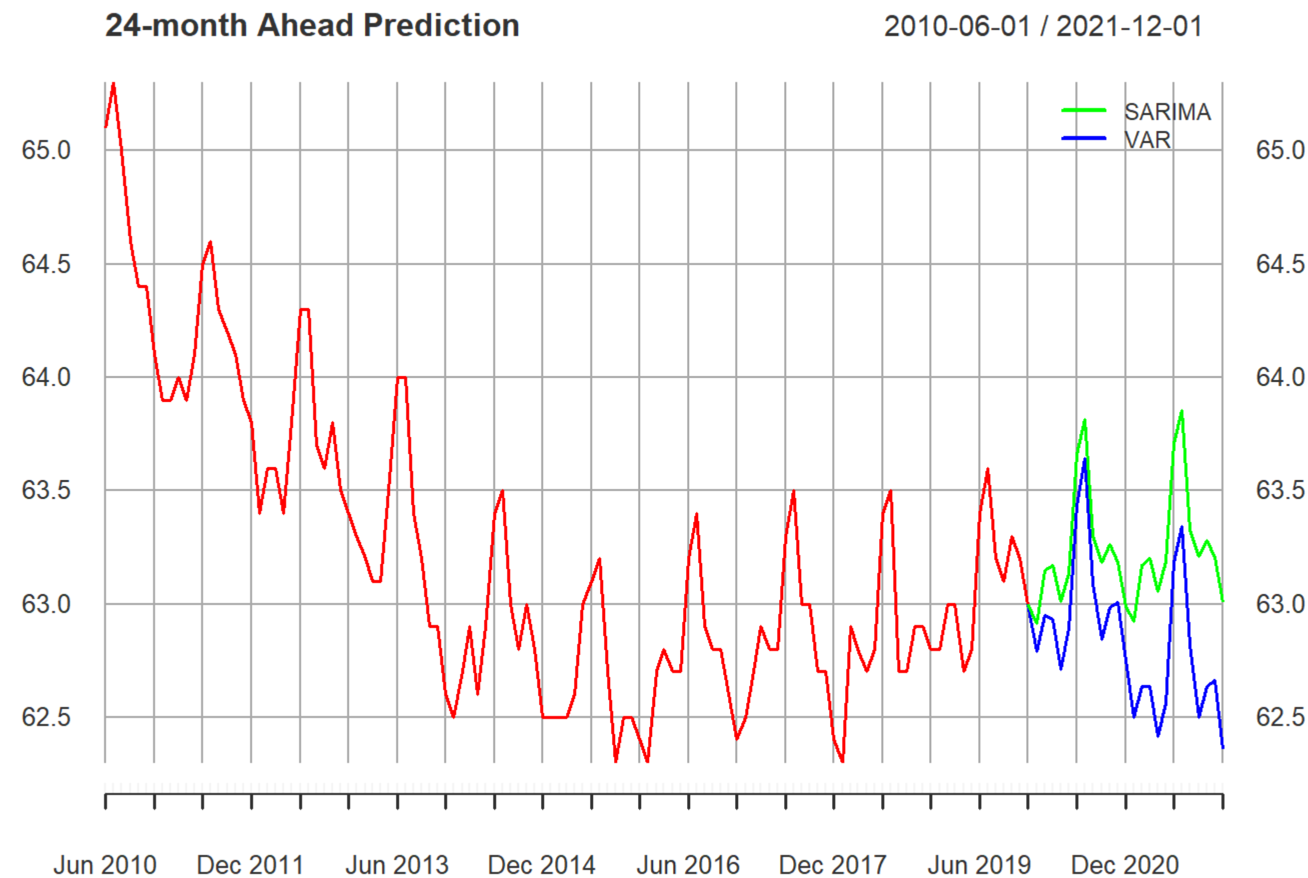

```
addLegend("topright", legend.names = c('True LFPR', 'SARIMA', 'VAR'),
          col = colors, lty = 1, lwd = 2, cex = 0.8)
```



The plots above shows the fitted lines of both model-fitting methods without new predictions. The SARIMA model is much more volatile compared to the VAR model, especially in the more recent data, indicating that the VAR model is a much-better fitted model.

```r
ahead = 24
dates_ahead <- seq(as.Date("2020-01-01"),
                   as.Date("2020-01-01")+ months(ahead-1),
                   by = "1 month")

# VAR Predictions
df2 = data.frame(trend = 1:ahead+nrow(dat),
                 cos1 = cos(2*pi*freq*(1:ahead+nrow(dat))),
                 sin1 = sin(2*pi*freq*(1:ahead+nrow(dat))),
                 cos2 = cos(2*pi*freq1*(1:ahead+nrow(dat))),
                 sin2 = sin(2*pi*freq1*(1:ahead+nrow(dat))),
                 cos3 = cos(2*pi*freq2*(1:ahead+nrow(dat))),
                 sin3 = sin(2*pi*freq2*(1:ahead+nrow(dat))))
pred.var <- predict(best.var.model, dumvar = df2,
                    n.ahead = ahead)$fcst$LFPR[,1]
pred.var <- rbind(LFPR[864], xts(pred.var, order.by = dates_ahead))

# SARIMA Predictions
pred.sarima <- sarima.for(LFPR, n.ahead = ahead, plot = F,
                          p=2, d=1, q=2, P=1, D=1, Q=2, S=12)$pred
pred.sarima <- rbind(LFPR[864], xts(pred.sarima, order.by = dates_ahead))
```

```r
addLegend("topright", col = colors[2:3], lty = 1, lwd = 2, cex = 0.8,
          legend.names = c('SARIMA', 'VAR'))
```



This plot here shows the 24-step ahead prediction using each model.

ii. Out-of-sample Error (2010-2019)

```
ahead = 120
dates_ahead <- seq(as.Date("2020-01-01") - months(ahead),
                   as.Date("2019-12-01"), by = "1 month")
LFPR_train <- LFPR[1:(864-ahead)]
LFPR_test <- LFPR[(864-ahead):864]
```

```
# SARIMA Model MSE
pred.sarima <- sarima.for(LFPR_train, n.ahead = ahead, plot = F,
                          p=2, d=1, q=2, P=1, D=1, Q=2, S=12)$pred
pred.sarima <- xts(pred.sarima, order.by = dates_ahead)
mse.sarima <- (1/ahead)*sum((LFPR_test - pred.sarima)^2)
print(mse.sarima)
```

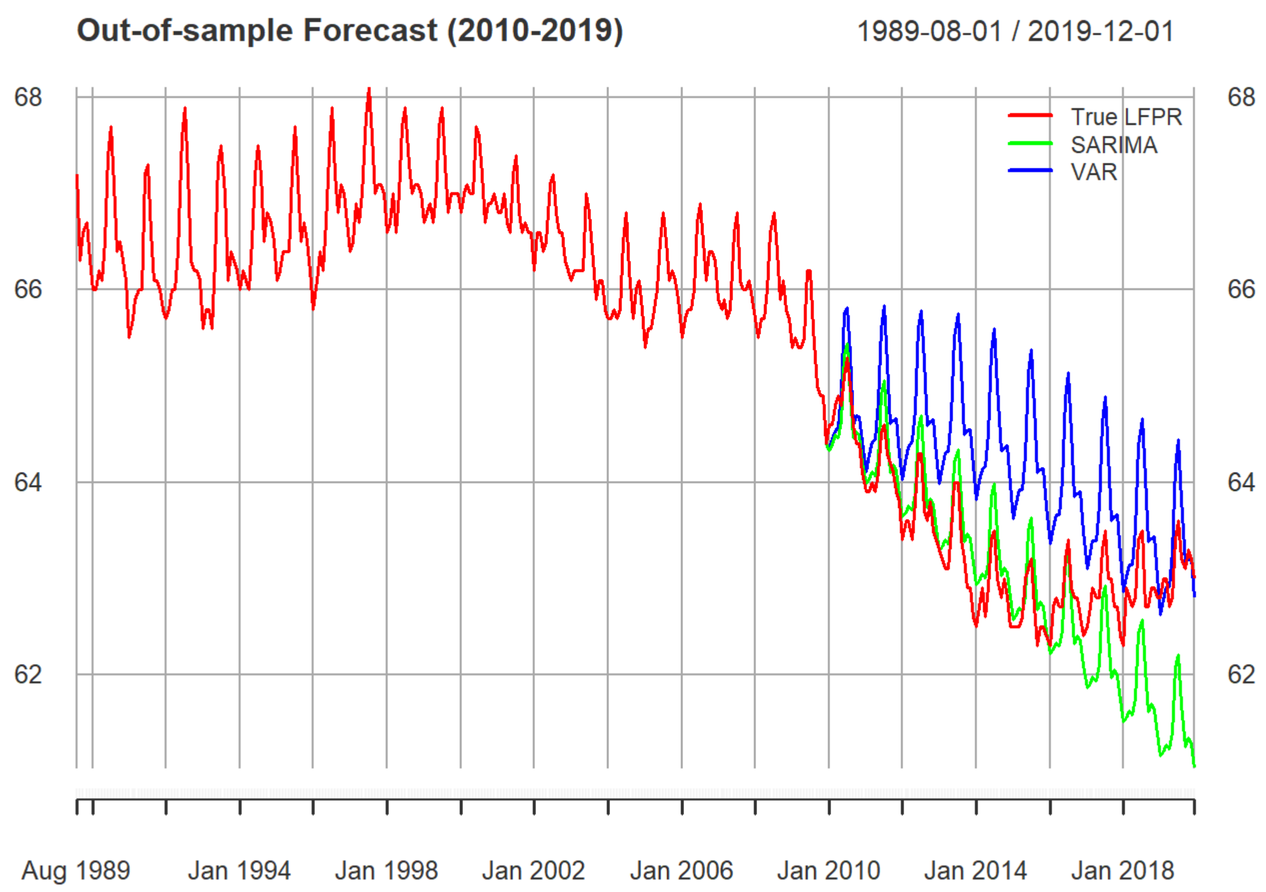
```
## [1] 0.5161911
```

```
# VAR Model MSE
dat_train <- dat[1:(864-ahead),]
df = data.frame(trend = 1:nrow(dat_train),
                cos1 = cos(2*pi*freq*1:nrow(dat_train)),
                sin1 = sin(2*pi*freq*1:nrow(dat_train)),
                cos2 = cos(2*pi*freq1*1:nrow(dat_train)),
                sin2 = sin(2*pi*freq1*1:nrow(dat_train)),
                cos3 = cos(2*pi*freq2*1:nrow(dat_train)),
                sin3 = sin(2*pi*freq2*1:nrow(dat_train)))
test.var.model <- VAR(dat_train, p = 25, type = "const", exogen = df)
df2 = data.frame(trend = 1:ahead+nrow(dat_train),
                 cos1 = cos(2*pi*freq*(1:ahead+nrow(dat_train))),
                 sin1 = sin(2*pi*freq*(1:ahead+nrow(dat_train))),
                 cos2 = cos(2*pi*freq1*(1:ahead+nrow(dat_train))),
                 sin2 = sin(2*pi*freq1*(1:ahead+nrow(dat_train))),
                 cos3 = cos(2*pi*freq2*(1:ahead+nrow(dat_train))),
                 sin3 = sin(2*pi*freq2*(1:ahead+nrow(dat_train))))
pred.var <- predict(test.var.model, dumvar = df2,
                    n.ahead = ahead)$fcst$LFPR[,1]
pred.var <- xts(pred.var, order.by = dates_ahead)
mse.var <- (1/ahead)*sum((LFPR_test - pred.var)^2)
print(mse.var)
```

```
## [1] 1.208616
```

It should be noted that although in this case the SARIMA model has a lower out-of-sample error than the VAR model, in certain cases VAR performs better (i.e. ahead = a different number). The trend is very hard to predict since if the cutoff of a training set is a turning point, the predictions would fail miserably. However, the TBATS model (see the last part) is still able to capture this surprisingly well, although we are not including it into our report, since it is not a model type that can clearly answer our research questions despite a superior fit.

```
addLegend("topright", col = colors, lty = 1, lwd = 2, cex = 0.8,
          legend.names = c('True LFPR', 'SARIMA', 'VAR'))
```



Additional Ideas (not included in our report)

TBATS State-Space Model (specifically designed to handle multiple seasonal patterns)

```
tbats.model <- tbats(LFPR2, seasonal.periods = 12)
print(tbats.model) # AIC: 3531.981
```

```
## TBATS(1, {5,0}, 0.803, {<12,5>})
##
## Call: tbats(y = LFPR2, seasonal.periods = 12)
##
## Parameters
##   Alpha: -0.0547951
##   Beta: 0.1215533
##   Damping Parameter: 0.802768
##   Gamma-1 Values: 0.001166057
##   Gamma-2 Values: -0.0002729506
##   AR coefficients: 0.642191 0.07988 -0.275283 0.155326 -0.334712
##
## Seed States:
##                 [,1]
##  [1,] 58.6594239490
##  [2,]  0.0844542233
##  [3,] -0.6661912538
##  [4,]  0.2110512057
##  [5,] -0.2543174101
##  [6,]  0.0007449031
##  [7,]  0.0219903493
##  [8,] -0.1735589402
##  [9,] -0.1331289634
## [10,]  0.0810083729
## [11,] -0.0323481556
## [12,] -0.0107419920
## [13,]  0.0000000000
## [14,]  0.0000000000
## [15,]  0.0000000000
## [16,]  0.0000000000
## [17,]  0.0000000000
##
## Sigma: 0.2545989
## AIC: 3531.981
```

```
tbats.model <- tbats(LFPR2, seasonal.periods = c(12,6))
print(tbats.model) # AIC: 3563.784
```

```
## TBATS(1, {5,2}, -, {<6,3>, <12,1>})
##
## Call: tbats(y = LFPR2, seasonal.periods = c(12, 6))
##
## Parameters
##   Alpha: 0.2437036
##   Gamma-1 Values: 7.911621e-05 0.0002711854
##   Gamma-2 Values: 5.678655e-05 -0.0001815337
##   AR coefficients: 0.656409 -0.880917 0.33581 0.099742 -0.298949
##   MA coefficients: -0.095559 0.890334
##
## Seed States:
##                [,1]
##  [1,]  5.874795e+01
##  [2,]  2.108277e-01
##  [3,]  6.425073e-04
##  [4,] -1.060403e-01
##  [5,] -1.334927e-01
##  [6,] -3.266369e-02
##  [7,] -1.030372e+12
##  [8,] -6.669111e-01
##  [9,] -1.742185e-01
## [10,]  0.000000e+00
## [11,]  0.000000e+00
## [12,]  0.000000e+00
## [13,]  0.000000e+00
## [14,]  0.000000e+00
## [15,]  0.000000e+00
## [16,]  0.000000e+00
##
## Sigma: 0.259028
## AIC: 3563.784
```

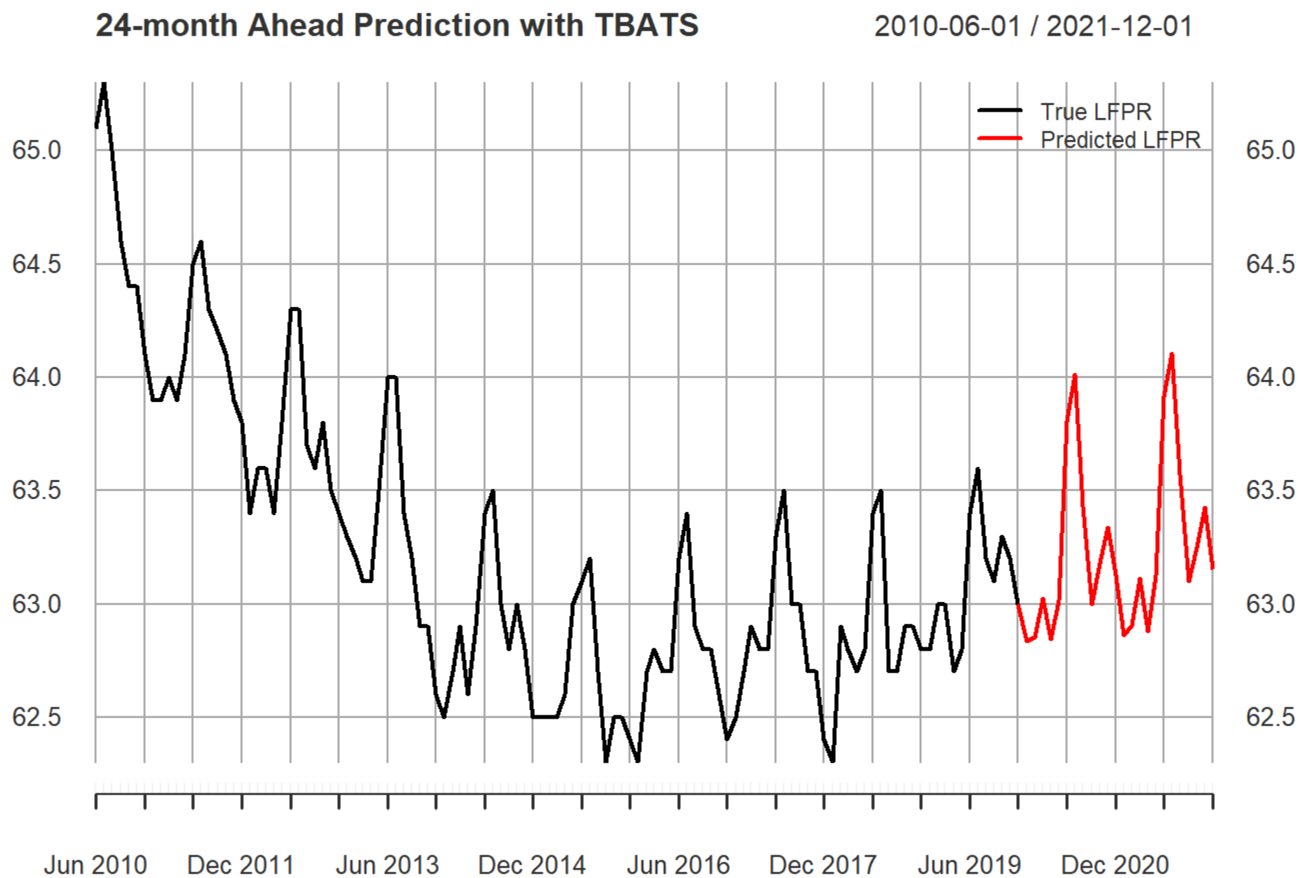
```
best.tbats.model <- tbats(LFPR2, seasonal.periods = c(216,144,72,12))
print(best.tbats.model) # AIC: 3463.498 - BEST MODEL!
```

```
## TBATS(1, {5,4}, -, {<12,6>, <72,1>, <144,1>, <216,1>})
##
## Call: tbats(y = LFPR2, seasonal.periods = c(216, 144, 72, 12))
##
## Parameters
##   Alpha: 0.3068854
##   Gamma-1 Values: 2.463633e-05 1.338189e-05 8.932051e-06 -4.384574e-05
##   Gamma-2 Values: -5.184307e-05 -1.956346e-05 1.414685e-05 -2.559407e-05
##   AR coefficients: 0.71264 0.127657 0.250369 -0.920967 0.185717
##   MA coefficients: -0.342932 -0.272198 -0.405352 0.858937
##
## Seed States:
##                [,1]
##  [1,]  5.872659e+01
##  [2,] -6.667927e-01
##  [3,]  2.109461e-01
##  [4,] -2.543316e-01
##  [5,]  7.609436e-04
##  [6,]  2.201827e-02
##  [7,] -1.050125e-01
##  [8,] -1.737676e-01
##  [9,] -1.332824e-01
## [10,]  8.083714e-02
## [11,] -3.259014e-02
## [12,] -1.121721e-02
## [13,] -1.012062e+12
## [14,] -3.171551e-02
## [15,]  4.385161e-02
## [16,]  8.437852e-02
## [17,] -2.322144e-02
## [18,] -8.728584e-02
## [19,]  1.654087e-01
## [20,]  0.000000e+00
## [21,]  0.000000e+00
## [22,]  0.000000e+00
## [23,]  0.000000e+00
## [24,]  0.000000e+00
## [25,]  0.000000e+00
## [26,]  0.000000e+00
## [27,]  0.000000e+00
## [28,]  0.000000e+00
##
## Sigma: 0.2393836
## AIC: 3463.498
```

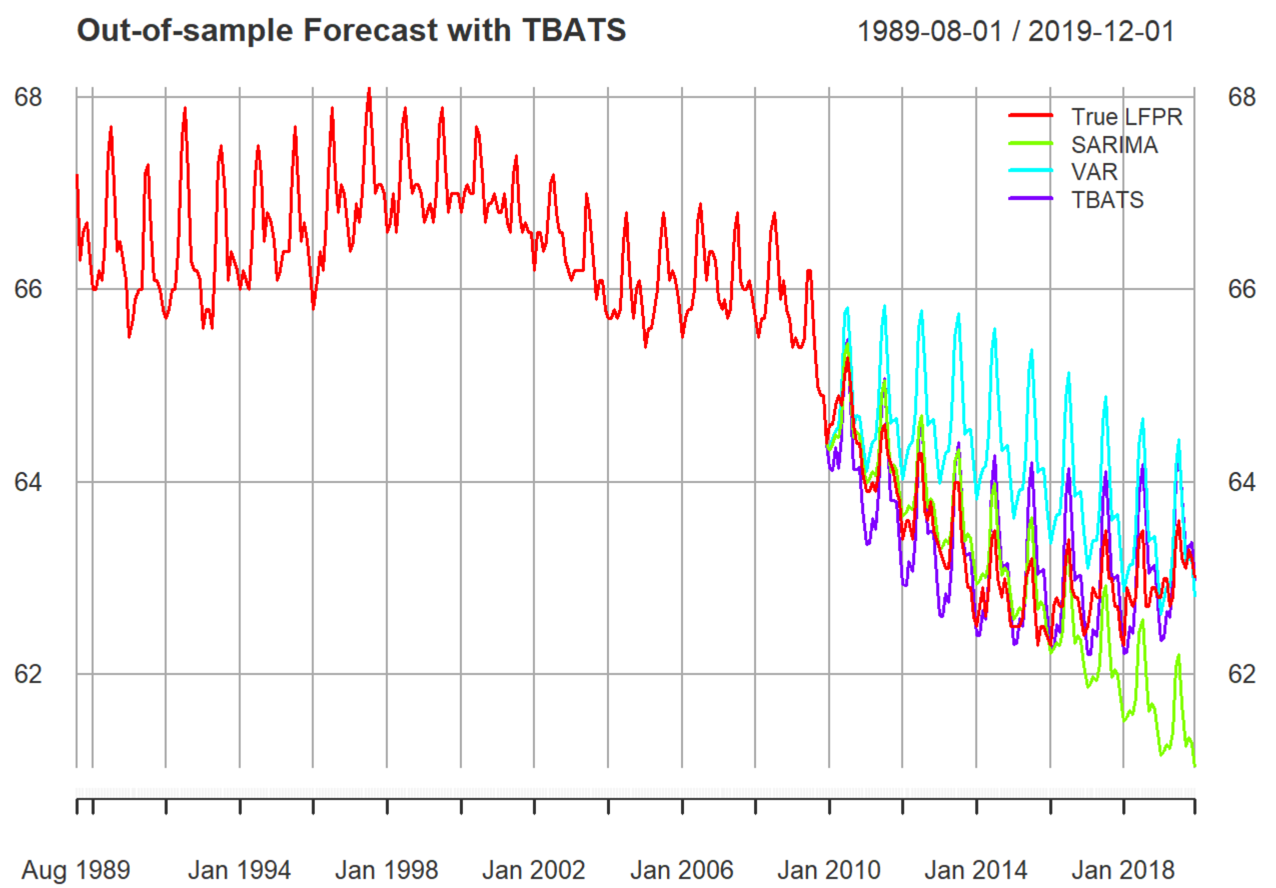
```
tbats.model <- tbats(LFPR2, seasonal.periods = c(216,144,72,12,6))
print(tbats.model) # AIC: 3580.569
```

```
## TBATS(1, {5,2}, 1, {<6,3>, <12,1>, <72,1>, <144,1>, <216,1>})
##
## Call: tbats(y = LFPR2, seasonal.periods = c(216, 144, 72, 12, 6))
##
## Parameters
##   Alpha: 0.1476255
##   Beta: 0.01030555
##   Damping Parameter: 1
##   Gamma-1 Values: 3.249832e-05 -2.255698e-05 6.026648e-05 8.103686e-05 -2.161838e-06
##   Gamma-2 Values: 5.350942e-08 -6.561046e-05 -6.008953e-05 -7.296117e-06 3.599386e-05
##   AR coefficients: 0.722855 -0.896288 0.416529 0.084244 -0.28431
##   MA coefficients: -0.096109 0.894072
##
## Seed States:
##                [,1]
##  [1,]  5.836782e+01
##  [2,]  7.778865e-02
##  [3,]  2.111885e-01
##  [4,]  8.837070e-04
##  [5,] -1.060968e-01
##  [6,] -1.331794e-01
##  [7,] -3.255988e-02
##  [8,] -1.033577e+12
##  [9,] -6.660599e-01
## [10,] -1.735432e-01
## [11,] -8.069036e-03
## [12,]  4.523260e-02
## [13,]  1.787307e-01
## [14,] -2.045797e-02
## [15,]  1.249093e-01
## [16,]  1.695543e-01
## [17,]  0.000000e+00
## [18,]  0.000000e+00
## [19,]  0.000000e+00
## [20,]  0.000000e+00
## [21,]  0.000000e+00
## [22,]  0.000000e+00
## [23,]  0.000000e+00
##
## Sigma: 0.2573525
## AIC: 3580.569
```

```
addLegend("topright", legend.names = c('True LFPR', 'Predicted LFPR'),
          col = c('black', 'red'), lty = 1, lwd = 2, cex = 0.8)
```




```
addLegend("topright", col = colors, lty = 1, lwd = 2, cex = 0.8,
          legend.names = c('True LFPR', 'SARIMA', 'VAR', 'TBATS'))
```



TBATS model, when incorporating long term cycles, outperforms both SARIMA and VAR models.