Project 2 Applied Econometrics

David Contento, Adam Jacobson, and Ross Lewis

February 11, 2019

Part I) Introduction

Our data for this project consists of 2 time series variables: the measure of price level CPI and the measure of money supply M2. CPI measures price levels by examining the change in prices for some representative basket of goods while M2 measures money supply consisting of currency in the hands of the public, demand deposits, some overnight repurchase agreements, savings deposits, money market mutual funds, and other miscellaneous items. The standard view on the relation between money supply and price level is that of long run money neutrality. That is any change in money supply will produce a corresponding change in the price level over the long run. Consequently changes in money supply can only effect nominal variables in the long run not real variables. We will use a VAR time series model to examine the statistical relationship between CPI and M2 over time. Our data is sourced from FRED (Federal Reserve Econominc Databse).

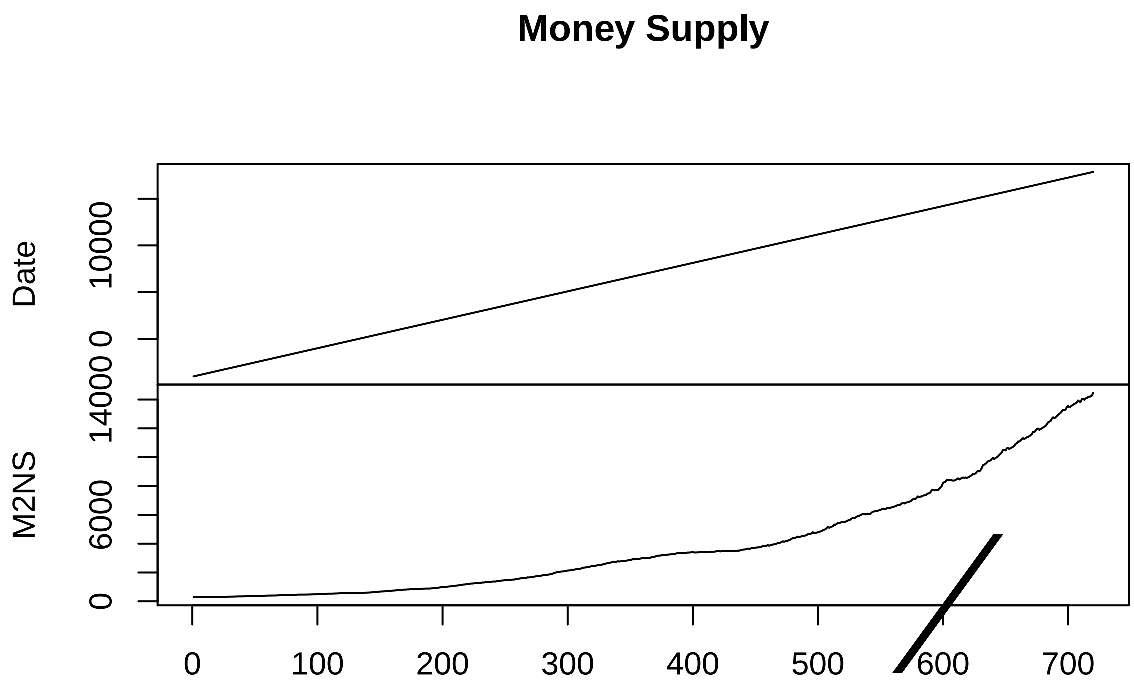
Part II) Results

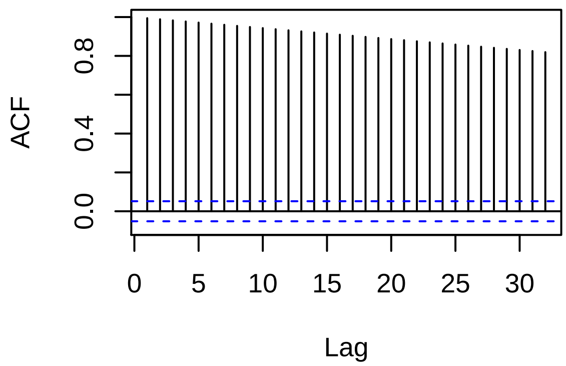
```
library(forecast)
library(ggplot2)
library(tseries)
setwd('C:/Users/rossw/Documents/MAE Program/Q2/Applied Econometrics 403B/Project 2')
moneySupply = read.csv("M2NS.csv", header = TRUE)
cpi = read.csv("CPIAUCNS.csv", header = TRUE)

moneySupply$Date <- as.Date(moneySupply$DATE, format= "%Y-%m-%d")
moneySupply = moneySupply[,c('Date','M2NS')]
cpi$Date <- as.Date(cpi$DATE, format= "%Y-%m-%d")
cpi = cpi[,c('Date','CPIAUCNS')]
```

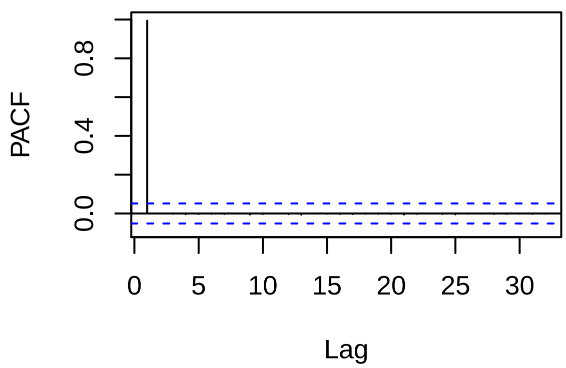
A

This is our ACF and PACF for M2. Our ACF slowly declines over time and our PACF has a strong spike at lag 1 and no spikes at all outside of that which suggests an AR model of order 1.

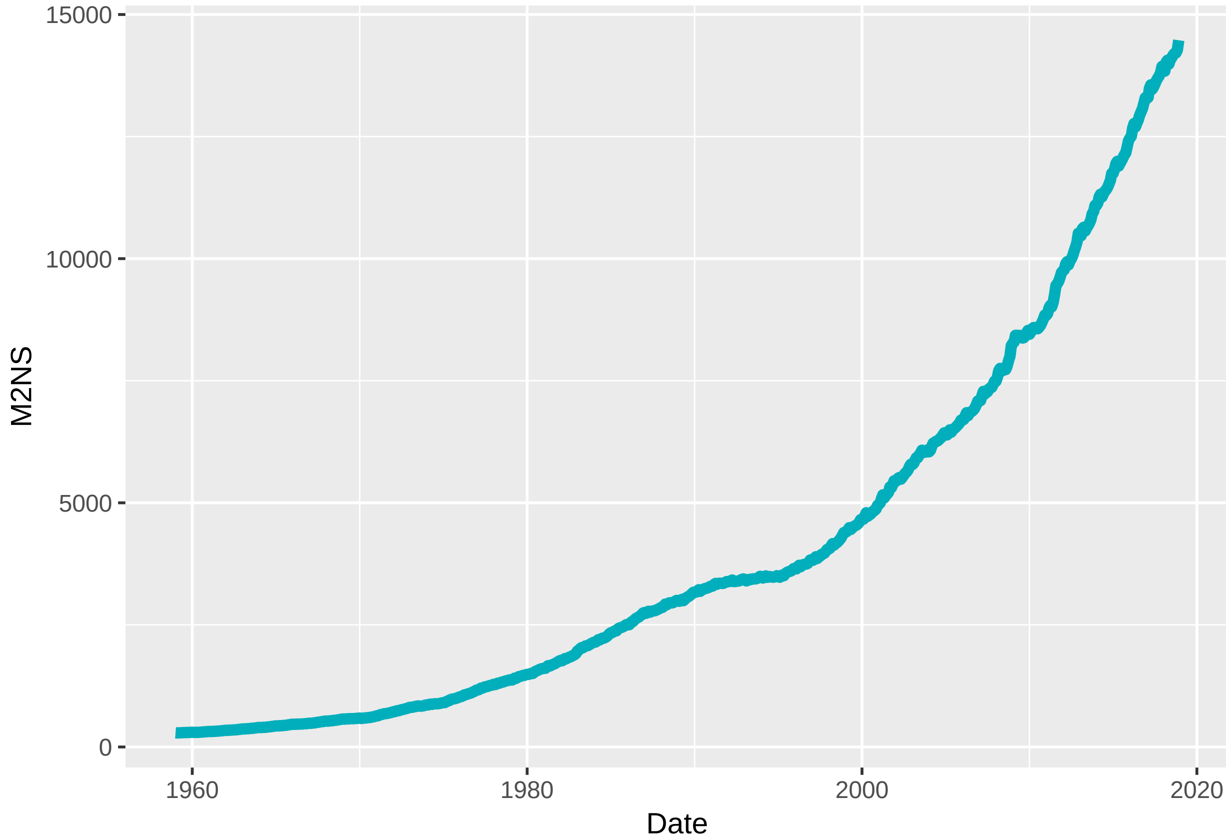
```
tsdisplay(moneySupply,main="Money Supply")
```


Money Supply
Date
10000
0
M2NS
14000
6000
0
0
100
200
300
400
500
600
700



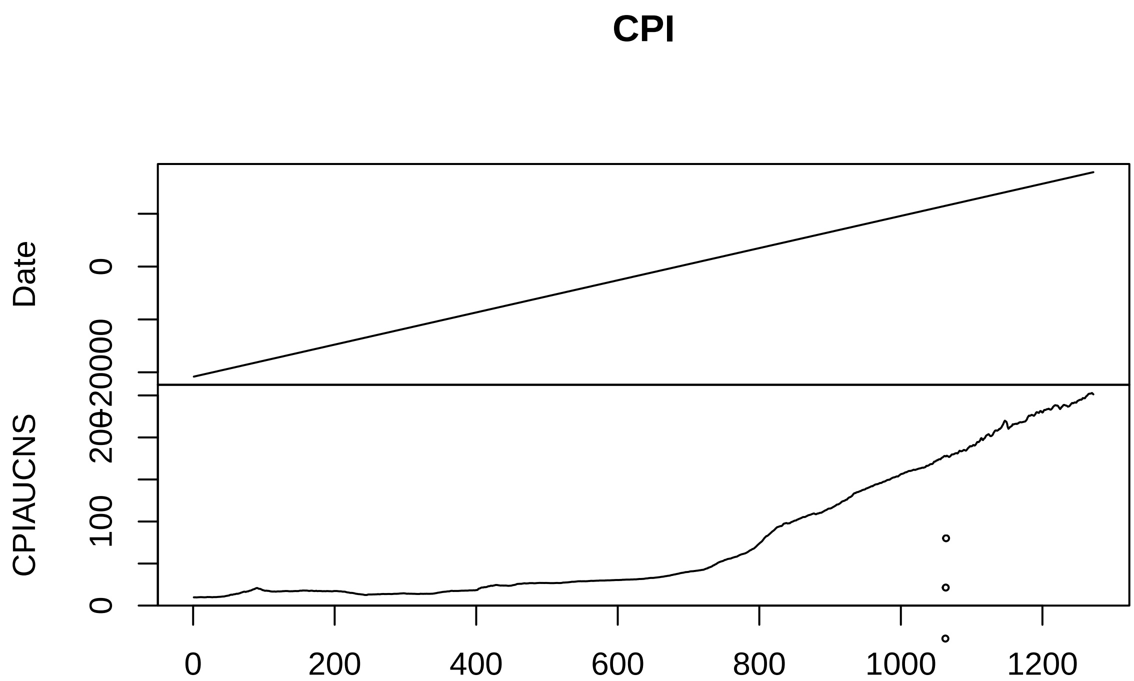


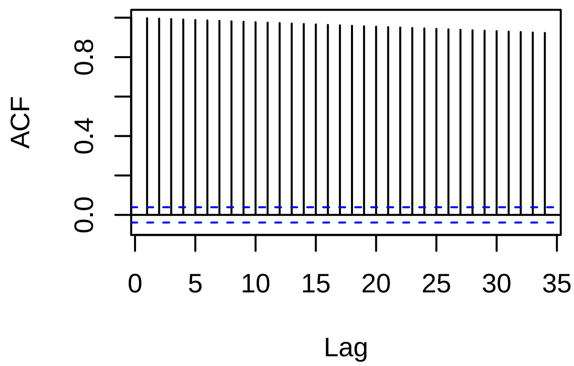

```r
#A graph of M2 over time. It shows its gradual increase since 1960
ggplot(data = moneySupply, aes(x = Date, y = M2NS)) +
  geom_line(color = "#00AFBB", size = 2)
```

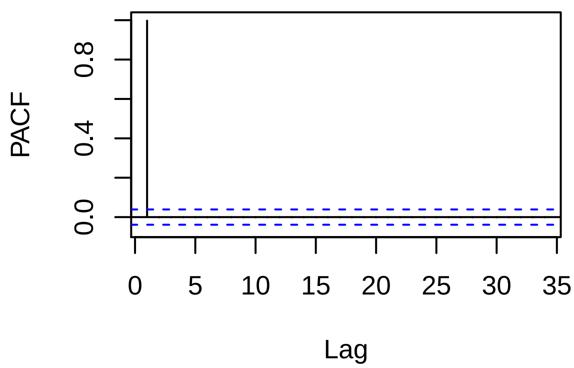


Our ACF and PACF for CPI. Like the ACF and PACF for M2 the CPI ACF and PACF shows the ACF declining slowly over the lags and the PACF spiking at lag 1 and not significant after that.

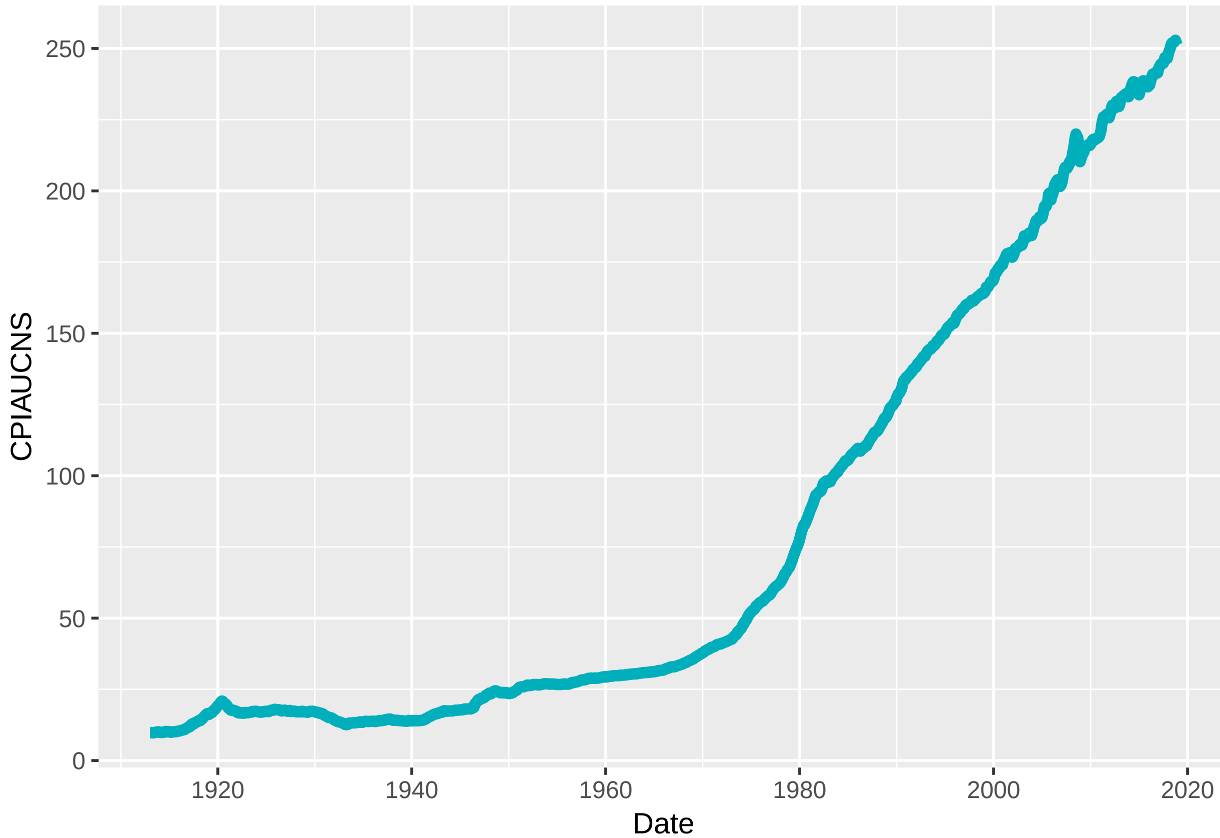
```
tsdisplay(cpi,main="CPI")
```


CPI
Date
CPIAUCNS
0
-20000
200
100
0
200
400
600
800
1000
1200





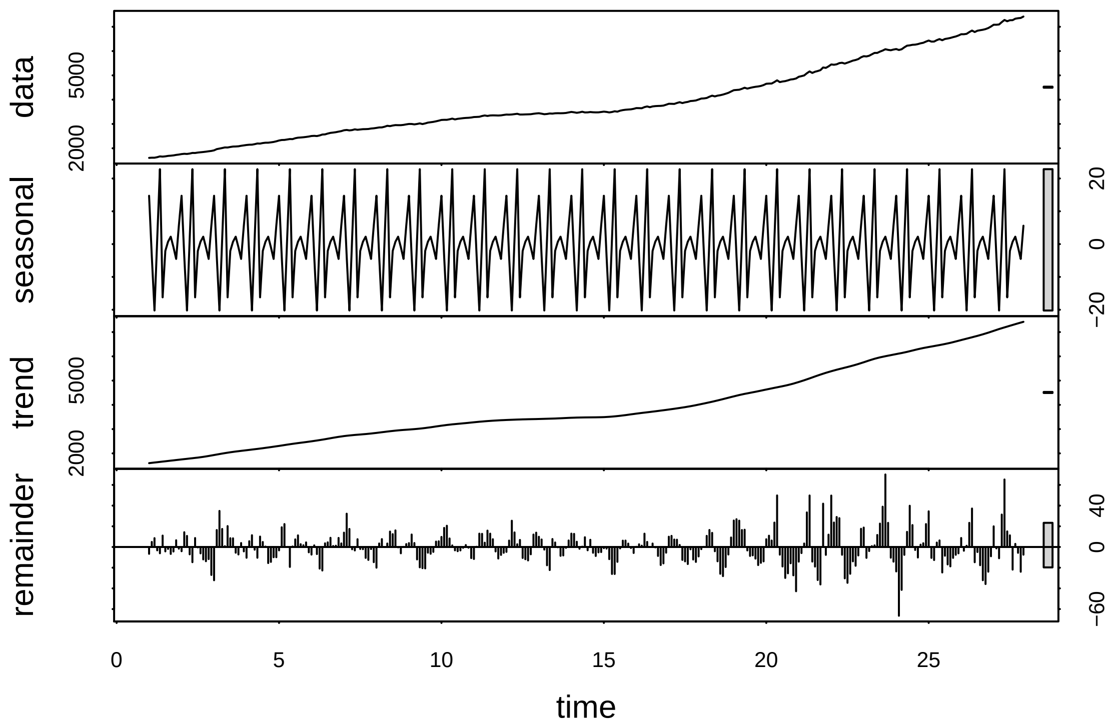

```
ggplot(data = cpi, aes(x = Date, y = CPIAUCNS)) +
  geom_line(color = "#00AFBB", size = 2)
```




```
cpiTs = ts(subset(cpi, Date > "1980-11-01" & Date <  "2007-12-01"), frequency=12)
moneySupplyTs = ts(subset(moneySupply, Date > "1980-11-01" & Date <  "2007-12-01"), frequency=12)
data = merge(cpiTs,moneySupplyTs)
```

b

```
msModel=tslm(moneySupplyTs~trend+season)
plot(stl(moneySupplyTs[,'M2NS'], s.window="periodic"))
```



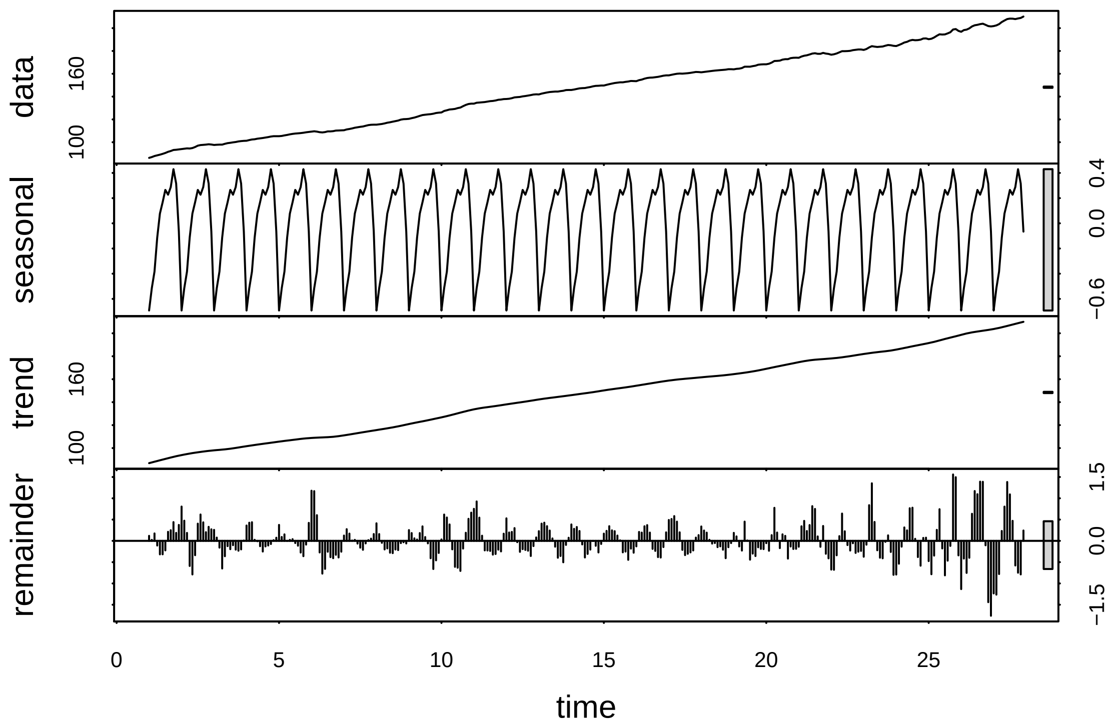

```r
summary(msModel)
```

```
## Response Date :
##
## Call:
## tslm(formula = Date ~ trend + season)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41026 -0.20513 -0.08547  0.16239  0.41026
##
## Coefficients:
##                Estimate Std. Error    t value Pr(>|t|)
## (Intercept)  3.956e+03  5.992e-02  66023.916  < 2e-16 ***
## trend        3.044e+01  1.665e-04 182780.255  < 2e-16 ***
## season2      5.627e-01  7.625e-02      7.379 1.46e-12 ***
## season3      1.125e+00  7.625e-02     14.759  < 2e-16 ***
## season4     -1.090e+00  7.625e-02    -14.291  < 2e-16 ***
## season5     -5.271e-01  7.625e-02     -6.912 2.71e-11 ***
## season6     -9.644e-01  7.625e-02    -12.647  < 2e-16 ***
## season7     -4.017e-01  7.626e-02     -5.268 2.58e-07 ***
## season8     -8.390e-01  7.626e-02    -11.002  < 2e-16 ***
## season9     -2.764e-01  7.626e-02     -3.624 0.000339 ***
## season10     2.863e-01  7.626e-02      3.754 0.000207 ***
## season11    -1.510e-01  7.627e-02     -1.980 0.048605 *
## season12     4.117e-01  7.627e-02      5.398 1.34e-07 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2802 on 311 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.788e+09 on 12 and 311 DF,  p-value: < 2.2e-16
##
##
## Response M2NS :
##
## Call:
## tslm(formula = M2NS ~ trend + season)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -640.85 -351.52   88.26  233.52  853.71
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1312.4260    81.3603  16.131   <2e-16 ***
## trend         16.2103     0.2261  71.694   <2e-16 ***
## season2      -14.9584   103.5315  -0.144    0.885
## season3      -27.9761   103.5322  -0.270    0.787
## season4       -5.6900   103.5335  -0.055    0.956
## season5       16.1441   103.5352   0.156    0.876
## season6      -24.7698   103.5374  -0.239    0.811
## season7      -12.4542   103.5401  -0.120    0.904
## season8      -10.4755   103.5433  -0.101    0.919
## season9       -9.7710   103.5470  -0.094    0.925
## season10     -10.0775   103.5512  -0.097    0.923
## season11     -10.7396   103.5559  -0.104    0.917
## season12       3.8390   103.5611   0.037    0.970
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 380.4 on 311 degrees of freedom
## Multiple R-squared:  0.943,  Adjusted R-squared:  0.9408
## F-statistic:   429 on 12 and 311 DF,  p-value: < 2.2e-16
```

None of our seasonal components are even close to statistical significance. The trend and intercept are highly statistically significant however. This suggests the data is not seasonal but shows a strong trend which is consistent with a prima facie look at the data

```r
cpiModel=tslm(cpiTs~trend+season)
plot(stl(cpiTs[,'CPIAUCNS'], s.window="periodic"))
```




```r
summary(cpiModel)
```

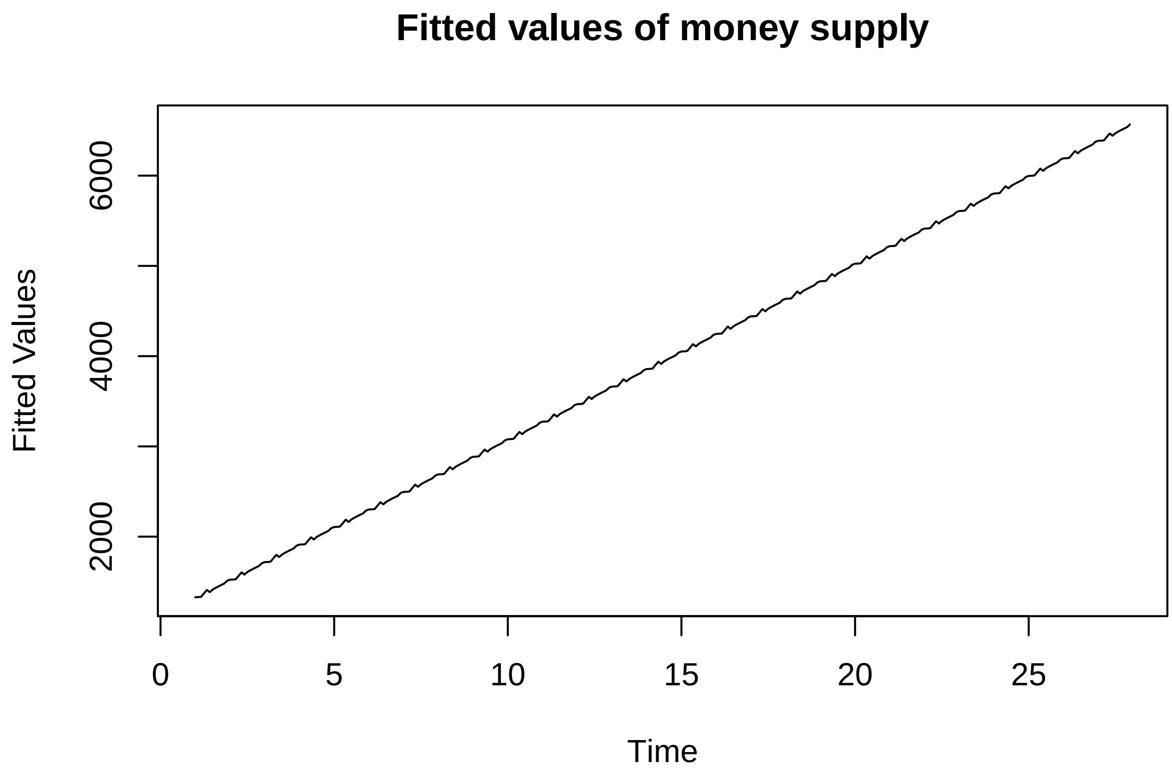
```
## Response Date :
##
## Call:
## tslm(formula = Date ~ trend + season)
##
## Residuals:
##       Min       1Q   Median       3Q      Max
## -0.41026 -0.20513 -0.08547  0.16239  0.41026
##
## Coefficients:
##                Estimate Std. Error    t value Pr(>|t|)
## (Intercept)  3.956e+03  5.992e-02  66023.916  < 2e-16 ***
## trend        3.044e+01  1.665e-04 182780.255  < 2e-16 ***
## season2      5.627e-01  7.625e-02      7.379 1.46e-12 ***
## season3      1.125e+00  7.625e-02     14.759  < 2e-16 ***
## season4     -1.090e+00  7.625e-02    -14.291  < 2e-16 ***
## season5     -5.271e-01  7.625e-02     -6.912 2.71e-11 ***
## season6     -9.644e-01  7.625e-02    -12.647  < 2e-16 ***
## season7     -4.017e-01  7.626e-02     -5.268 2.58e-07 ***
## season8     -8.390e-01  7.626e-02    -11.002  < 2e-16 ***
## season9     -2.764e-01  7.626e-02     -3.624 0.000339 ***
## season10     2.863e-01  7.626e-02      3.754 0.000207 ***
## season11    -1.510e-01  7.627e-02     -1.980 0.048605 *
## season12     4.117e-01  7.627e-02      5.398 1.34e-07 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2802 on 311 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.788e+09 on 12 and 311 DF,  p-value: < 2.2e-16
##
##
## Response CPIAUCNS :
##
## Call:
## tslm(formula = CPIAUCNS ~ trend + season)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3547 -1.6739  0.2201  1.4181  4.8704
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 86.836387   0.405077 214.370   <2e-16 ***
## trend        0.363580   0.001126 322.973   <2e-16 ***
## season2      0.237012   0.515463   0.460   0.6460
## season3      0.428358   0.515467   0.831   0.4066
## season4      0.692666   0.515473   1.344   0.1800
## season5      0.878493   0.515482   1.704   0.0893 .
## season6      0.906135   0.515493   1.758   0.0798 .
## season7      0.942665   0.515506   1.829   0.0684 .
## season8      0.862307   0.515522   1.673   0.0954 .
## season9      0.880875   0.515541   1.709   0.0885 .
## season10     1.060739   0.515561   2.057   0.0405 *
## season11     0.984047   0.515585   1.909   0.0572 .
## season12     0.670134   0.515611   1.300   0.1947
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.894 on 311 degrees of freedom
## Multiple R-squared:  0.997,  Adjusted R-squared:  0.9969
## F-statistic:  8710 on 12 and 311 DF,  p-value: < 2.2e-16
```

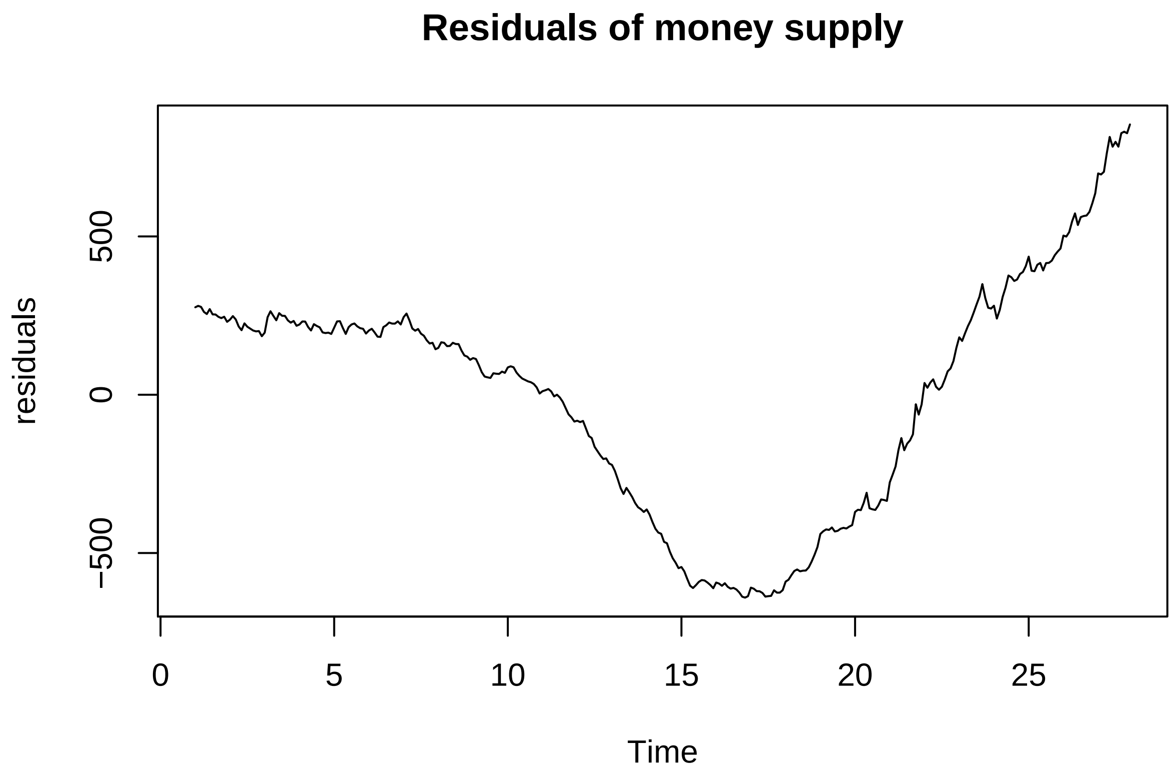
Similar to the above only the trend and intercept are statistically significant. This is also consistent with a prima facie visual examination of the data.

c

```
plot(msModel$fitted.values[,'M2NS'], ylab="Fitted Values", main="Fitted values of money supply")
```

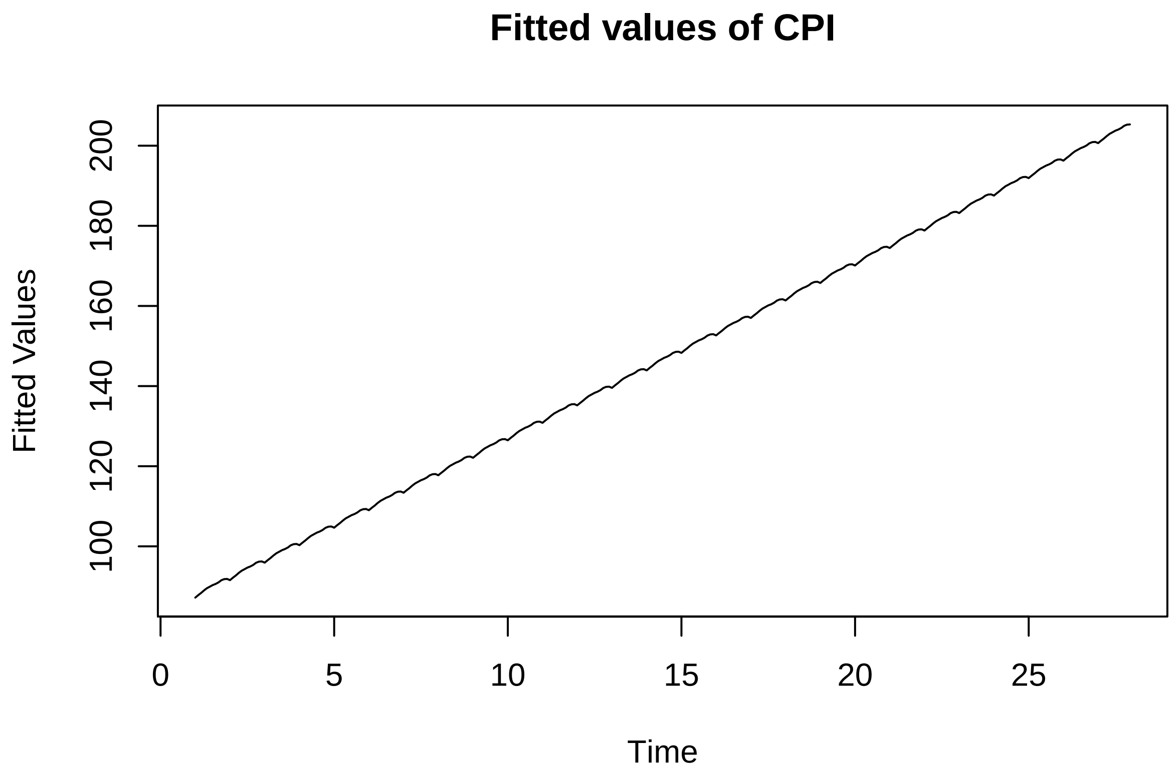



```r
plot(msModel$residuals[,'M2NS'], ylab="residuals", main="Residuals of money supply")
```

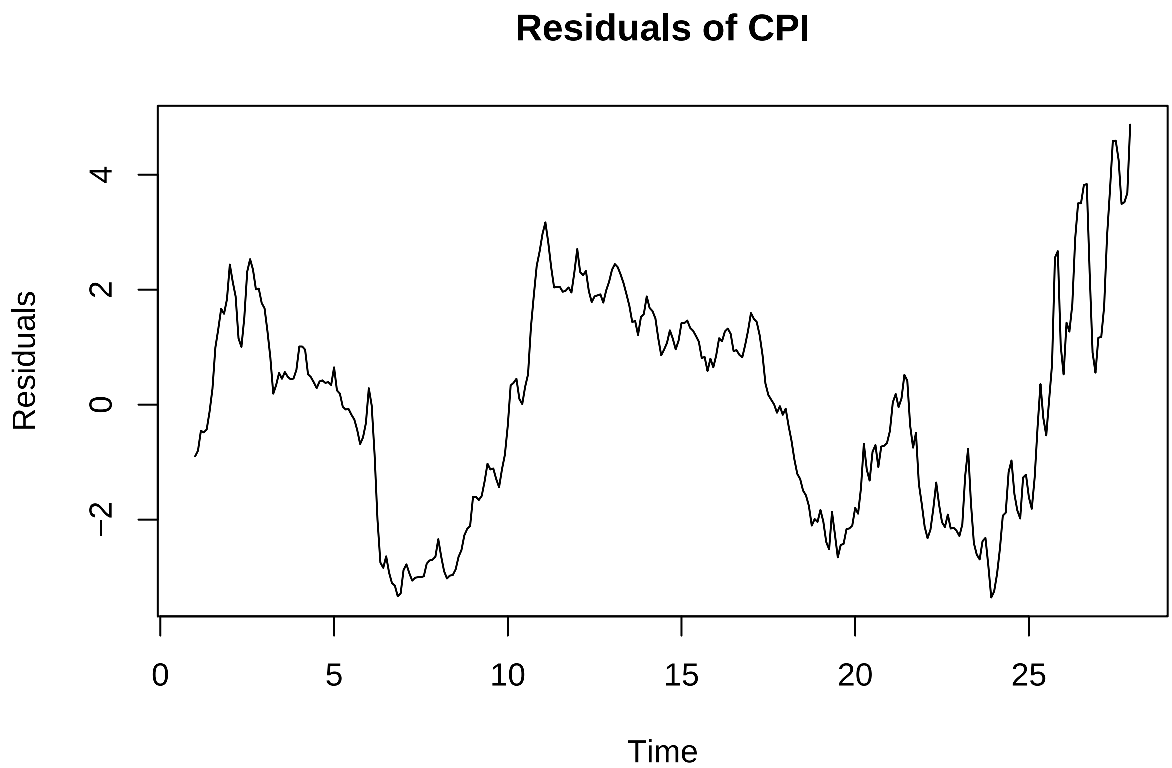


The fitted values for M2 show a strong linear trend. The residuals are relatively flat until they dip down rapidly and then rapidly go up.

```
plot(cpiModel$fitted.values[,'CPIAUCNS'], ylab="Fitted Values", main="Fitted values of CPI")
```




```
plot(cpiModel$residuals[,'CPIAUCNS'], ylab="Residuals", main="Residuals of CPI")
```

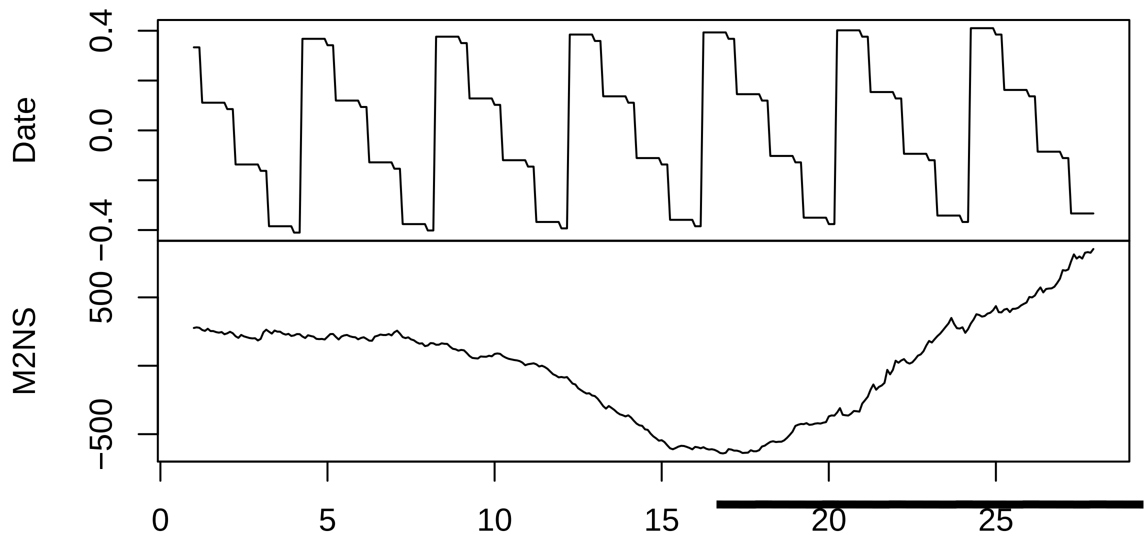


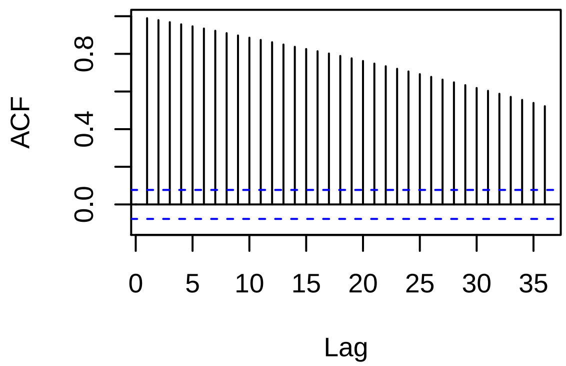
The fitted values here also show a strong and consistent linear trend while the residuals are much more unstable compared with M2.

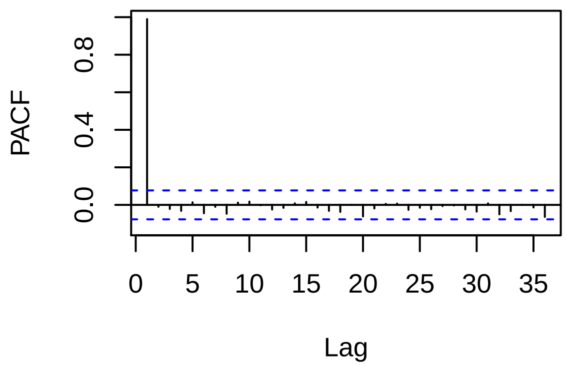
e

```
#acf and pacf of money supply
tsdisplay(msModel$residuals,main="Money Supply")
```

Money Supply

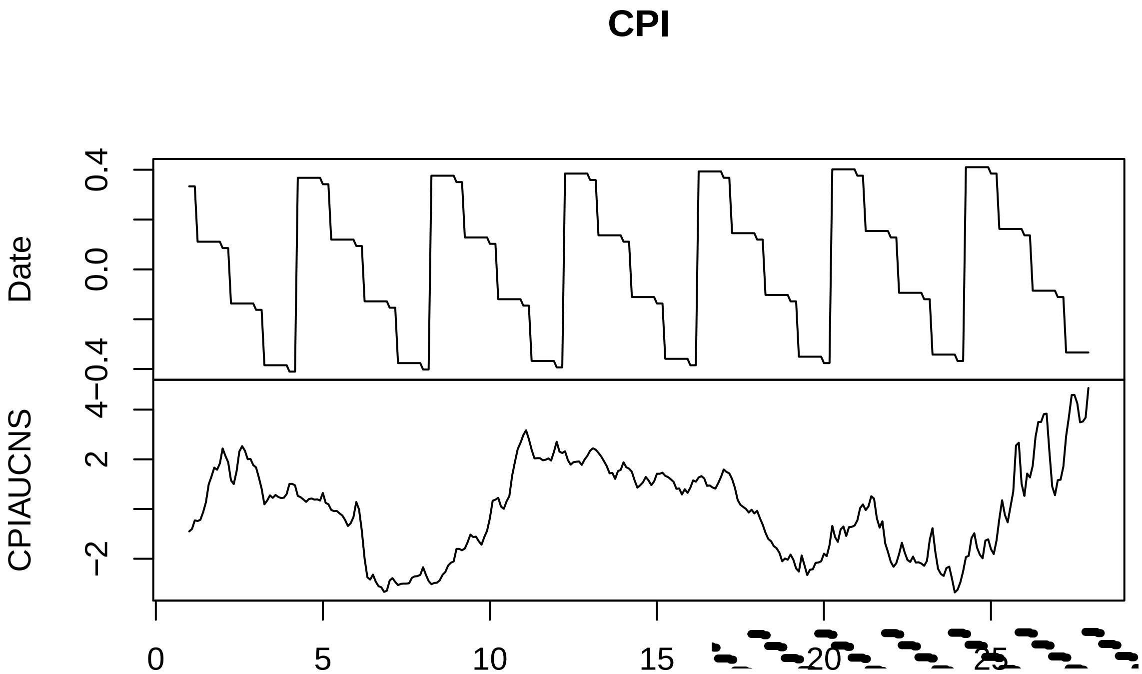


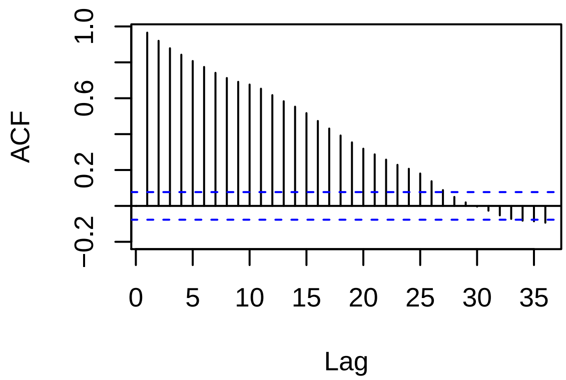


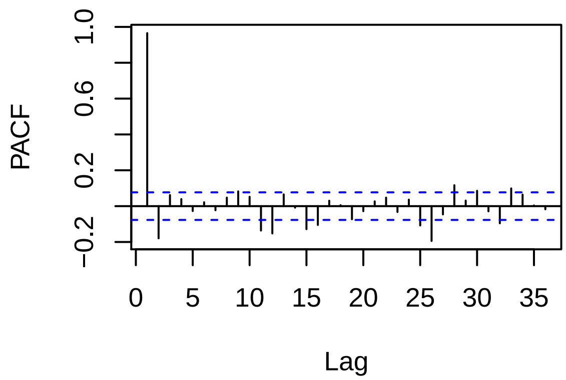


The M2 residuals ACF and PACF look just like the ACF and PACF for M2 itself. ACF declines over time while PACF spikes at lag 1.

```
#acf and pacf of CPI
tsdisplay(cpiModel$residuals,main="CPI")
```


CPI
Date
CPIAUCNS
0.4
0.0
−0.4
4
2
−2
0
5
10
15
20
25





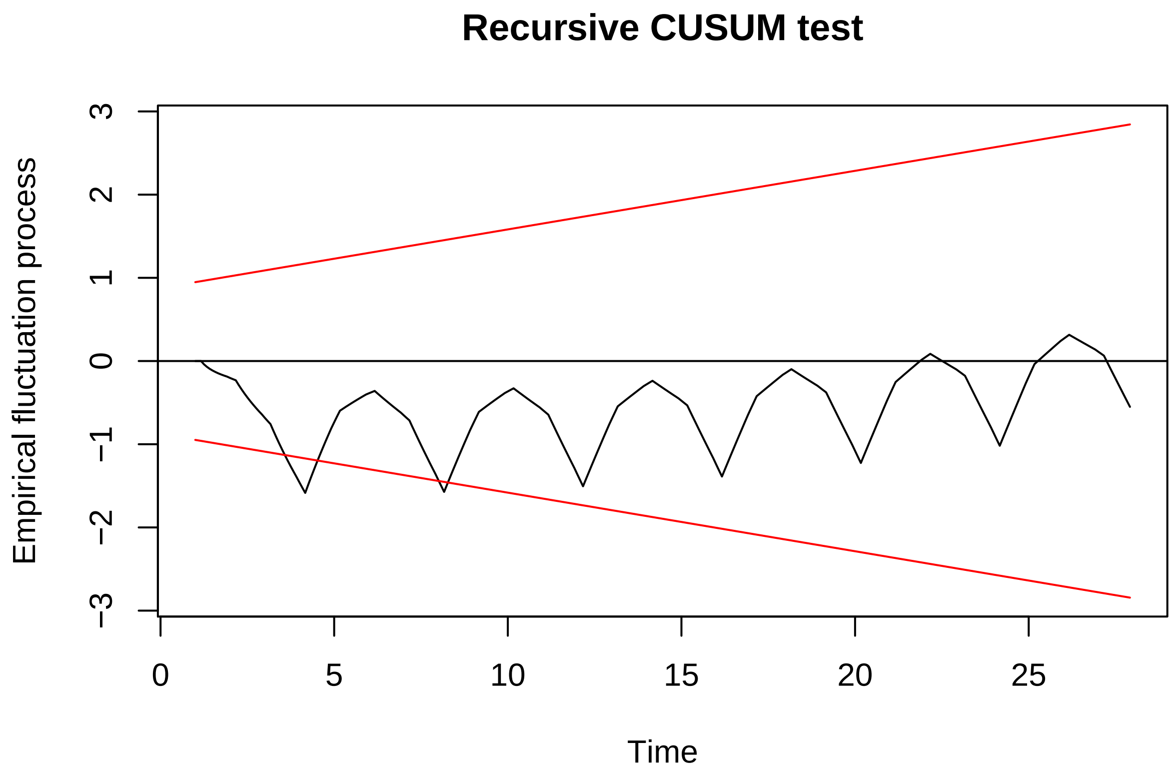
The residuals for CPI also show a decline in the ACF over time. The PACF spikes strongly at lag 1 with weak spikes at lags 2, 11, 12, 15, 16, 25, 26

f

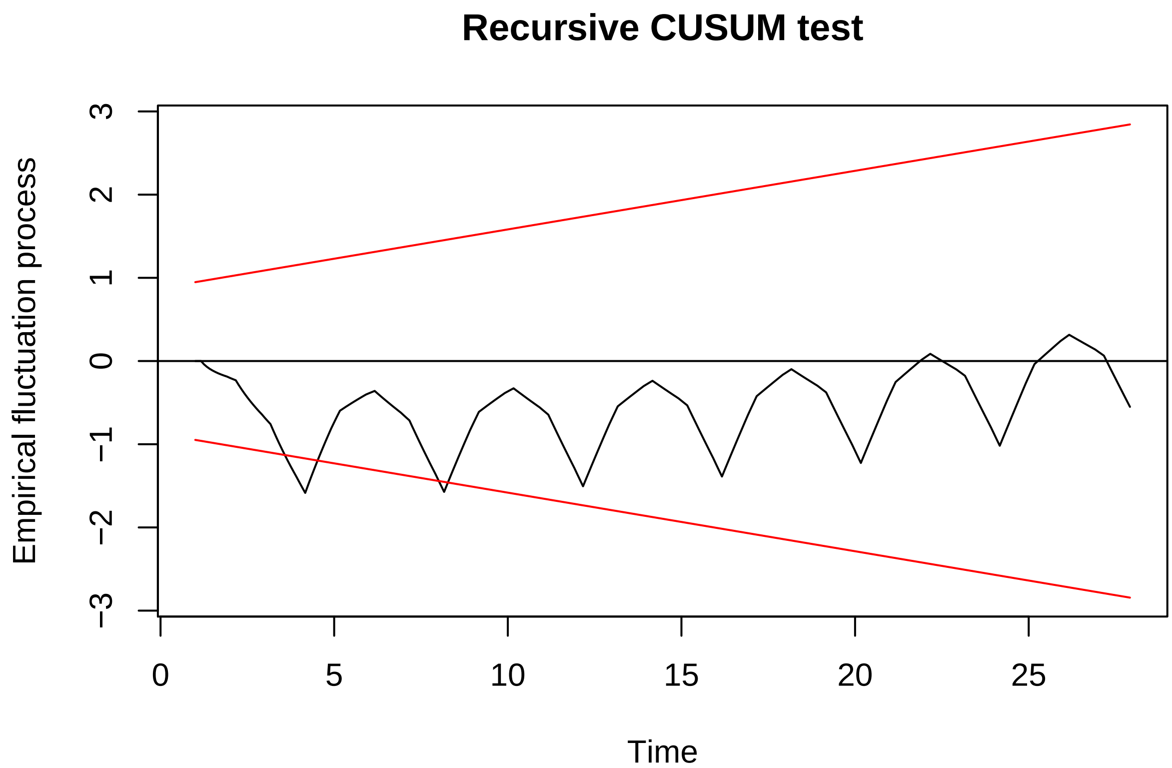
```r
library(strucchange)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

## Warning: package 'sandwich' was built under R version 3.5.2
```

```r
#CUSUM for Money supply
plot(efp(msModel$residuals~1, type = "Rec-CUSUM"))
```



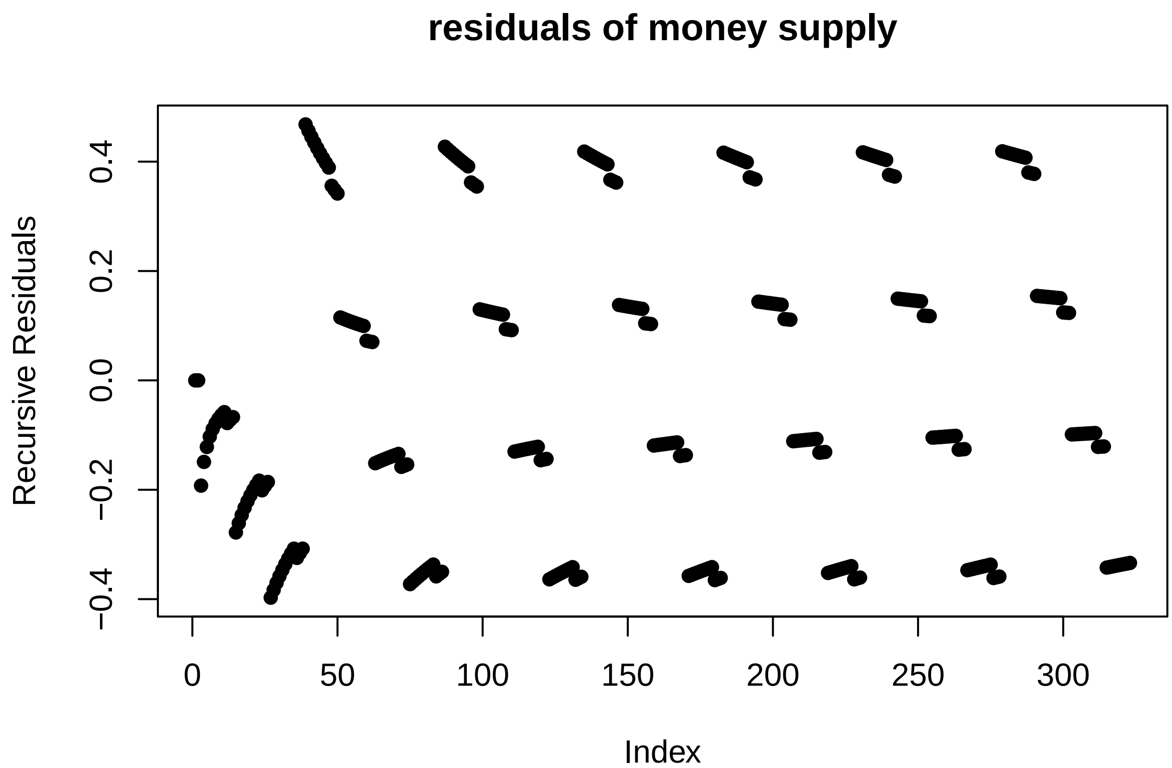

```r
#CUSUM for CPI
plot(efp(cpiModel$residuals~1, type = "Rec-CUSUM"))
```



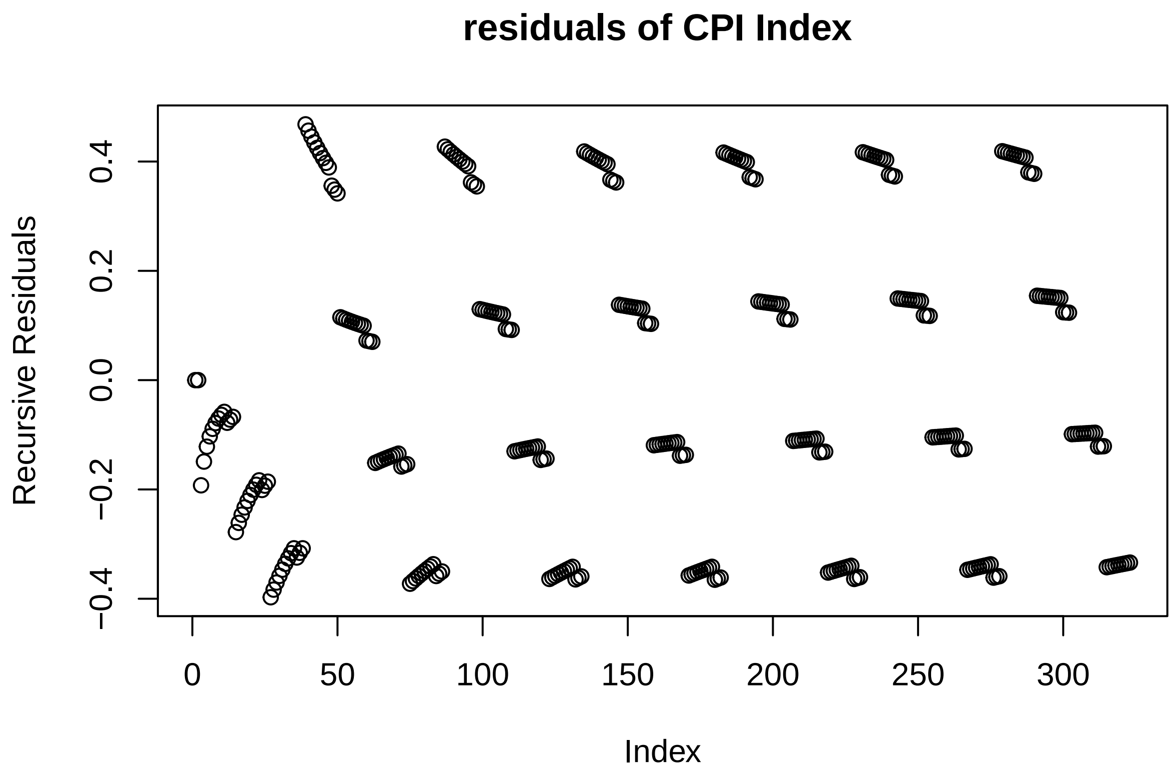
The CUSUM charts for both M2 and CPI look similar with small deviations outside the error bands in the begining but no indications of structural breaks.

g

```
y=recresid(msModel$residuals~1)
plot(y, pch=16,ylab="Recursive Residuals", main="residuals of money supply")
```




```r
y=recresid(cpiModel$residuals~1)
plot(y, ylab="Recursive Residuals", main="residuals of CPI Index ")
```



h

```r
#Diagnostic statistics for CPI
summary(cpiModel)
```

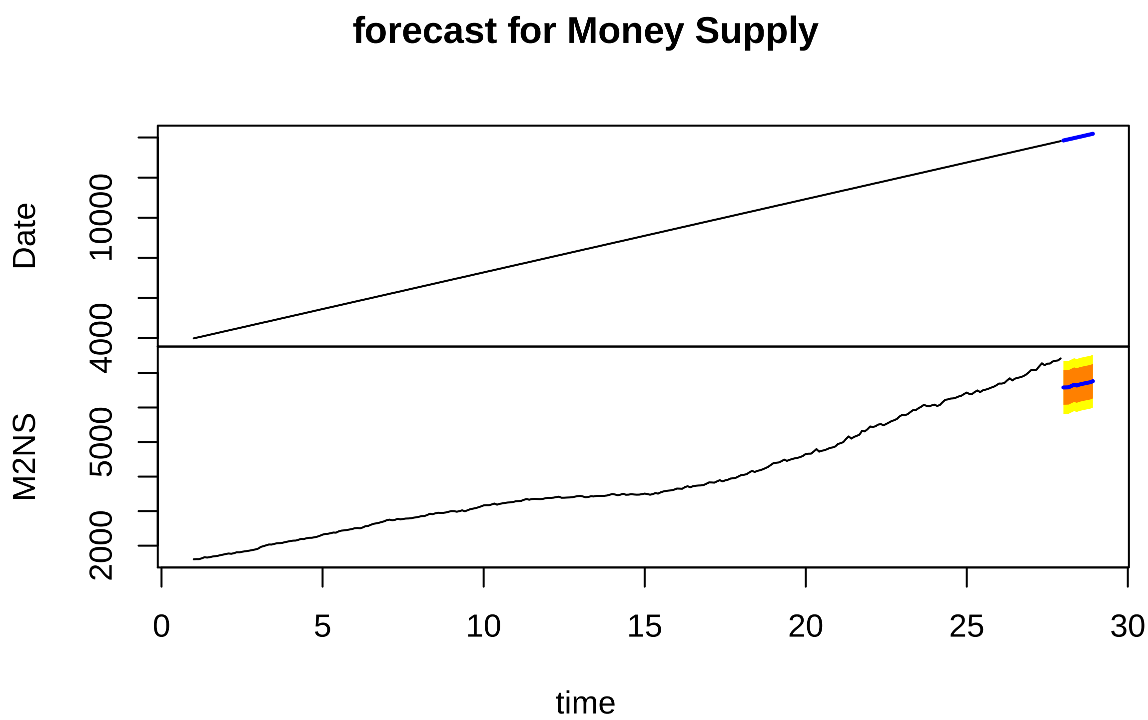
```
## Response Date :
##
## Call:
## tslm(formula = Date ~ trend + season)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41026 -0.20513 -0.08547  0.16239  0.41026
##
## Coefficients:
##               Estimate Std. Error    t value Pr(>|t|)
## (Intercept)  3.956e+03  5.992e-02  66023.916  < 2e-16 ***
## trend        3.044e+01  1.665e-04 182780.255  < 2e-16 ***
## season2      5.627e-01  7.625e-02      7.379 1.46e-12 ***
## season3      1.125e+00  7.625e-02     14.759  < 2e-16 ***
## season4     -1.090e+00  7.625e-02    -14.291  < 2e-16 ***
## season5     -5.271e-01  7.625e-02     -6.912 2.71e-11 ***
## season6     -9.644e-01  7.625e-02    -12.647  < 2e-16 ***
## season7     -4.017e-01  7.626e-02     -5.268 2.58e-07 ***
## season8     -8.390e-01  7.626e-02    -11.002  < 2e-16 ***
```

```
## season9     -2.764e-01  7.626e-02     -3.624 0.000339 ***
## season10     2.863e-01  7.626e-02      3.754 0.000207 ***
## season11    -1.510e-01  7.627e-02     -1.980 0.048605 *
## season12     4.117e-01  7.627e-02      5.398 1.34e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2802 on 311 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.788e+09 on 12 and 311 DF,  p-value: < 2.2e-16
##
##
## Response CPIAUCNS :
##
## Call:
## tslm(formula = CPIAUCNS ~ trend + season)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3547 -1.6739  0.2201  1.4181  4.8704
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 86.836387   0.405077 214.370   <2e-16 ***
## trend        0.363580   0.001126 322.973   <2e-16 ***
## season2      0.237012   0.515463   0.460   0.6460
## season3      0.428358   0.515467   0.831   0.4066
## season4      0.692666   0.515473   1.344   0.1800
## season5      0.878493   0.515482   1.704   0.0893 .
## season6      0.906135   0.515493   1.758   0.0798 .
## season7      0.942665   0.515506   1.829   0.0684 .
## season8      0.862307   0.515522   1.673   0.0954 .
## season9      0.880875   0.515541   1.709   0.0885 .
## season10     1.060739   0.515561   2.057   0.0405 *
## season11     0.984047   0.515585   1.909   0.0572 .
## season12     0.670134   0.515611   1.300   0.1947
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.894 on 311 degrees of freedom
## Multiple R-squared:  0.997,  Adjusted R-squared:  0.9969
## F-statistic:  8710 on 12 and 311 DF,  p-value: < 2.2e-16
```

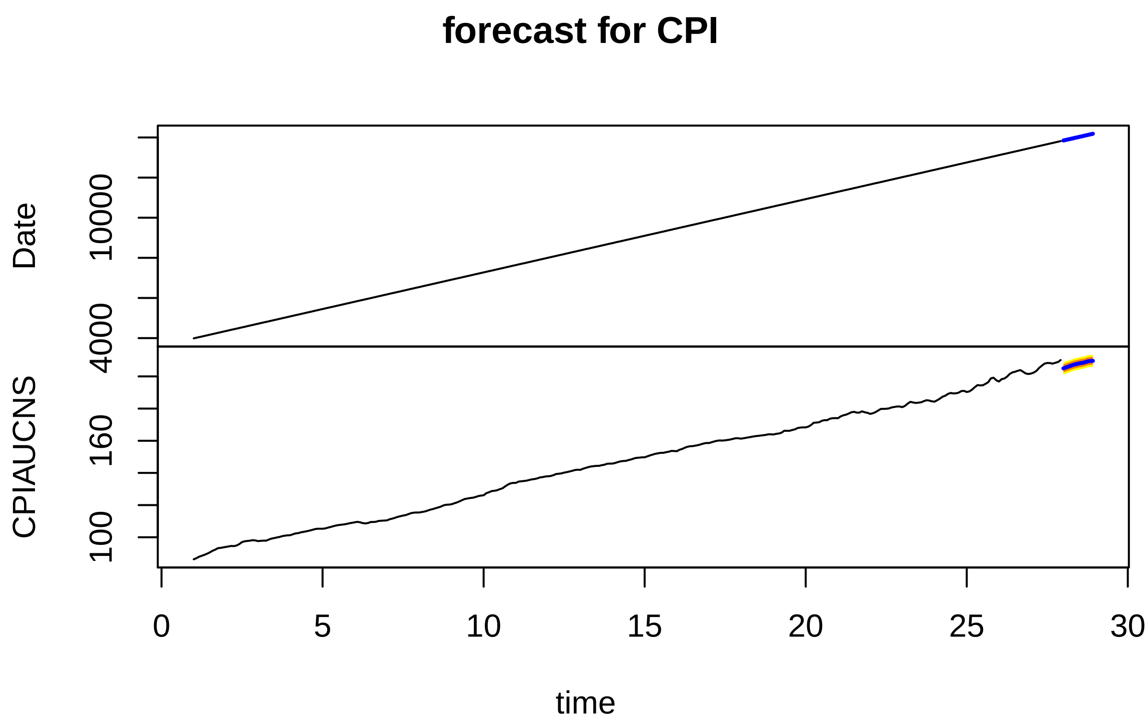
The adjusted R-squared is extremely high at 0.9969 and the F-Statistic has a p-value well under 0.05 level indicating the model is statistically significant.

i

```
plot(forecast(msModel,h=12),shadecols="oldstyle", main="forecast for Money Supply")
```




```r
plot(forecast(cpiModel,h=12),shadecols="oldstyle", main="forecast for CPI ")
```



The forecast for M2 shows relatively wide error bands reflecting its relatively greatly instability. The CPI has far smaller error bands which is unsurpring given its relatively greater stability.

j

```
library(vars)
var_model=VAR(data[,c('CPIAUCNS','M2NS')],p=4)
summary(var_model)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: CPIAUCNS, M2NS
## Deterministic variables: const
## Sample size: 320
## Log Likelihood: -1608.832
## Roots of the characteristic polynomial:
## 1.005 0.9955 0.6465 0.6465 0.6033 0.6033 0.377 0.1628
## Call:
## VAR(y = data[, c("CPIAUCNS", "M2NS")], p = 4)
##
##
## Estimation results for equation CPIAUCNS:
## =========================================
## CPIAUCNS = CPIAUCNS.l1 + M2NS.l1 + CPIAUCNS.l2 + M2NS.l2 + CPIAUCNS.l3 + M2NS.l3 + CPIAUCNS.l4 + M2N
```

```
##
##              Estimate Std. Error t value Pr(>|t|)
## CPIAUCNS.l1  1.4741452  0.0569879  25.868  < 2e-16 ***
## M2NS.l1     -0.0006665  0.0009298  -0.717   0.4740
## CPIAUCNS.l2 -0.7496579  0.1009019  -7.430 1.06e-12 ***
## M2NS.l2      0.0007451  0.0012271   0.607   0.5442
## CPIAUCNS.l3  0.2178749  0.1008902   2.160   0.0316 *
## M2NS.l3     -0.0002221  0.0012258  -0.181   0.8563
## CPIAUCNS.l4  0.0521993  0.0566784   0.921   0.3578
## M2NS.l4      0.0002820  0.0009393   0.300   0.7642
## const        0.5943315  0.1866623   3.184   0.0016 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.3911 on 311 degrees of freedom
## Multiple R-Squared: 0.9999,  Adjusted R-squared: 0.9999
## F-statistic: 2.956e+05 on 8 and 311 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation M2NS:
## =====================================
## M2NS = CPIAUCNS.l1 + M2NS.l1 + CPIAUCNS.l2 + M2NS.l2 + CPIAUCNS.l3 + M2NS.l3 + CPIAUCNS.l4 + M2NS.l4
##
##             Estimate Std. Error t value Pr(>|t|)
## CPIAUCNS.l1 -6.89192    3.42506  -2.012  0.04506 *
## M2NS.l1      0.94741    0.05588  16.953  < 2e-16 ***
## CPIAUCNS.l2 11.40195    6.06436   1.880  0.06102 .
## M2NS.l2     -0.20792    0.07375  -2.819  0.00512 **
## CPIAUCNS.l3 -9.69846    6.06365  -1.599  0.11074
## M2NS.l3      0.42175    0.07368   5.725 2.44e-08 ***
## CPIAUCNS.l4  4.99230    3.40645   1.466  0.14378
## M2NS.l4     -0.15201    0.05645  -2.693  0.00747 **
## const       16.33775   11.21868   1.456  0.14632
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 23.51 on 311 degrees of freedom
## Multiple R-Squared: 0.9998,  Adjusted R-squared: 0.9998
## F-statistic: 1.737e+05 on 8 and 311 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##          CPIAUCNS     M2NS
## CPIAUCNS   0.1530  -0.2105
## M2NS      -0.2105 552.5528
##
## Correlation matrix of residuals:
##          CPIAUCNS     M2NS
## CPIAUCNS  1.00000 -0.02289
## M2NS     -0.02289  1.00000
```

Our Var model shows the CPI lags 1, 2, and 3 are significant for CPI but none of the M2 lags are significant. This suggests previous levels of M2 do not predict CPI. M2 however is affected by its own lags at 1, 2, 3, and 4. It also has a statistically siginificant coefficient at lag 1 for CPI. CPI not being influnced by M2 lags is surprising but M2 being affected by the CPI lag likely reflects the Federal Reserve using price levels to determine the fed funds rate.

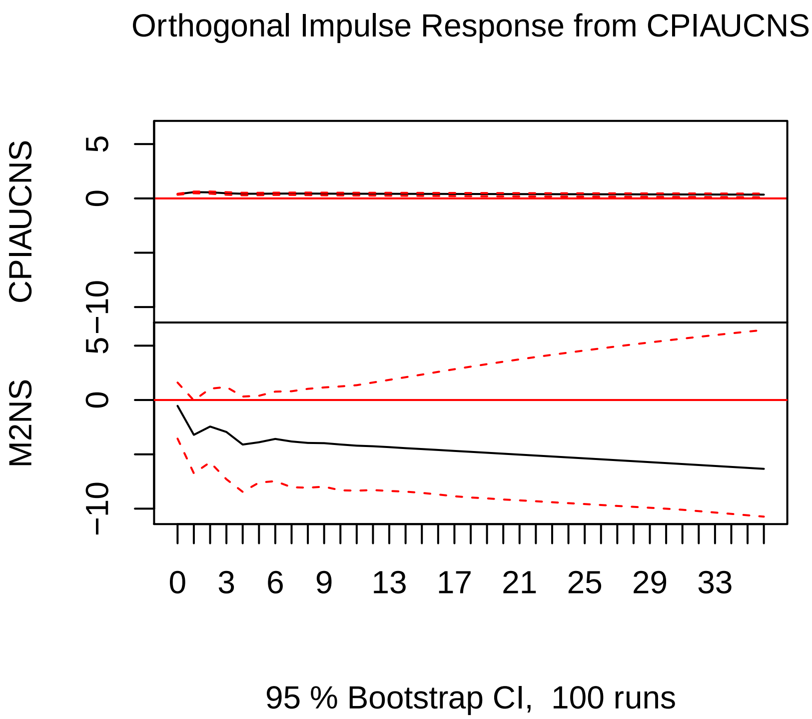
k

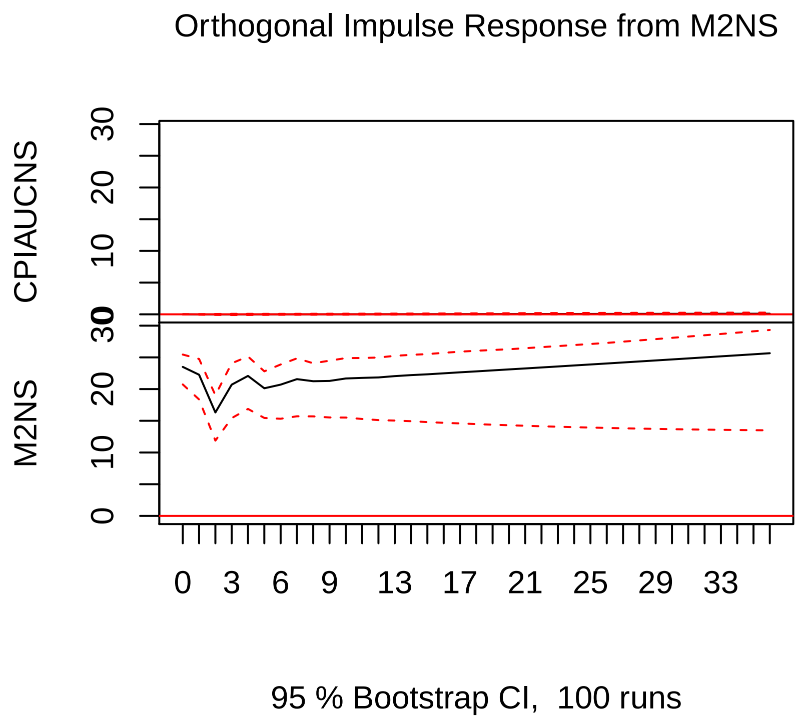
```r
#impluse response function
irf(var_model)
```

```
##
## Impulse response coefficients
## $CPIAUCNS
##        CPIAUCNS       M2NS
##  [1,] 0.3911123 -0.5381651
##  [2,] 0.5769149 -3.2053790
##  [3,] 0.5589913 -2.4415237
##  [4,] 0.4761175 -2.9413899
##  [5,] 0.4296266 -4.0995468
##  [6,] 0.4284905 -3.8883564
##  [7,] 0.4419990 -3.5823914
##  [8,] 0.4483788 -3.8177959
##  [9,] 0.4449938 -3.9503379
## [10,] 0.4380095 -3.9706370
## [11,] 0.4323997 -4.0929994
##
## $M2NS
##            CPIAUCNS     M2NS
##  [1,]  0.000000e+00 23.50028
##  [2,] -1.566276e-02 22.26438
##  [3,] -2.041827e-02 16.31523
##  [4,] -1.786280e-02 20.70146
##  [5,] -1.439696e-02 22.08051
##  [6,] -9.735257e-03 20.12701
##  [7,] -5.475530e-03 20.70263
##  [8,] -2.710708e-03 21.57195
##  [9,]  4.063884e-05 21.24395
## [10,]  3.382810e-03 21.28669
## [11,]  6.768186e-03 21.67731
##
##
## Lower Band, CI= 0.95
## $CPIAUCNS
##        CPIAUCNS      M2NS
##  [1,] 0.3386305 -3.107873
##  [2,] 0.4827498 -6.422459
##  [3,] 0.4385348 -5.961840
##  [4,] 0.3325560 -7.003928
##  [5,] 0.2991862 -8.171558
##  [6,] 0.3040693 -7.512458
##  [7,] 0.3317449 -7.067142
##  [8,] 0.3433930 -7.386988
```

```
##  [9,] 0.3326895 -7.490202
## [10,] 0.3242824 -7.616137
## [11,] 0.3138209 -7.778248
##
## $M2NS
##          CPIAUCNS     M2NS
##  [1,]  0.00000000 20.78389
##  [2,] -0.05564498 18.66574
##  [3,] -0.10962009 11.76353
##  [4,] -0.11442985 15.79589
##  [5,] -0.10856887 17.19911
##  [6,] -0.09787410 15.29392
##  [7,] -0.09403556 15.37369
##  [8,] -0.09143411 16.08453
##  [9,] -0.08580692 15.88241
## [10,] -0.07962878 15.69936
## [11,] -0.07384848 15.72927
##
##
## Upper Band, CI= 0.95
## $CPIAUCNS
##        CPIAUCNS        M2NS
##  [1,] 0.4339869  1.90980229
##  [2,] 0.6414715  0.12974357
##  [3,] 0.6243773  1.01435442
##  [4,] 0.5538894  1.48992181
##  [5,] 0.5216524  0.17189729
##  [6,] 0.5171033 -0.01105163
##  [7,] 0.5169592  0.81301251
##  [8,] 0.5083322  0.89062359
##  [9,] 0.4993931  0.82710064
## [10,] 0.4934498  1.08925447
## [11,] 0.4906499  1.36396928
##
## $M2NS
##         CPIAUCNS     M2NS
##  [1,] 0.00000000 25.22701
##  [2,] 0.02854451 24.36487
##  [3,] 0.06380613 18.50929
##  [4,] 0.09042948 22.53722
##  [5,] 0.08552589 23.58016
##  [6,] 0.07990818 21.74056
##  [7,] 0.08067944 22.23572
##  [8,] 0.08497031 22.76371
##  [9,] 0.09296146 22.55798
## [10,] 0.10265126 22.64268
## [11,] 0.10769309 22.85656
```

```
plot(irf(var_model, n.ahead=36))
```


Orthogonal Impulse Response from CPIAUCNS
CPIAUCNS
M2NS
5
0
−10
5
0
−10
0
3
6
9
13
17
21
25
29
33
95 % Bootstrap CI, 100 runs



The impulse reponse function for CPI shows barely any response at all at any point. The impulse response function for M2 however shows a response that slowly grows in magnitude over time.

l

```r
#Granger Causality
gmon<-ts(moneySupplyTs[,2],start=1908,freq=12)
gcpi<-ts(cpiTs[,2],start=1908,freq=12)
#granger test CPI
grangertest(gcpi ~ gmon, order = 8)
```

```
## Granger causality test
##
## Model 1: gcpi ~ Lags(gcpi, 1:8) + Lags(gmon, 1:8)
## Model 2: gcpi ~ Lags(gcpi, 1:8)
##   Res.Df Df      F Pr(>F)
## 1    299
## 2    307 -8 1.8205 0.0728 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#granger test Money Supply
grangertest(gmon ~ gcpi, order = 8)
```

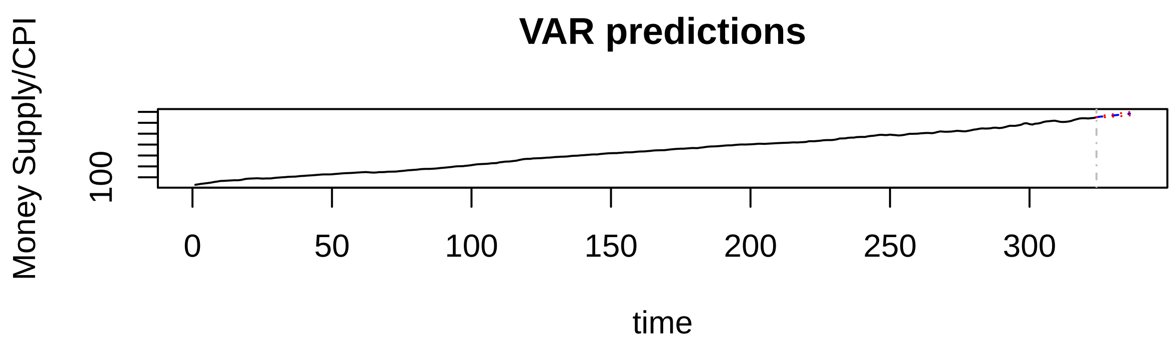
```
## Granger causality test
##
```

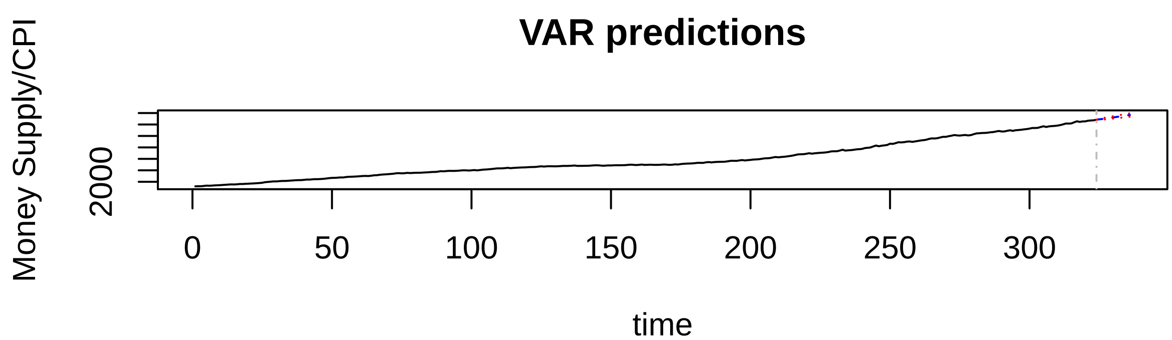
```
## Model 1: gmon ~ Lags(gmon, 1:8) + Lags(gcpi, 1:8)
## Model 2: gmon ~ Lags(gmon, 1:8)
##   Res.Df Df    F    Pr(>F)
## 1    299
## 2    307 -8 7.54 3.548e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The granger causality test for M2 on CPI shows that it is not statistically significant suggesting the M2 lags are not leading to changes in CPI. The granger causality test for CPI on M2 however is statistically significant. This suggests lags in CPI do predict M2. Like in the comment above the most plausible explanation for this is that the Federal Reserve is observing CPI and using that variable (among others) to determine interest rate setting policies. Through the interet rate mechanisms the fed uses it is ultimately changing M2 in response to it's observations of CPI.

m

```r
#VAR prediction
varPredict = predict(object=var_model, n.ahead=12)
plot(varPredict, main="VAR predictions", xlab="time", ylab="Money Supply/CPI")
```





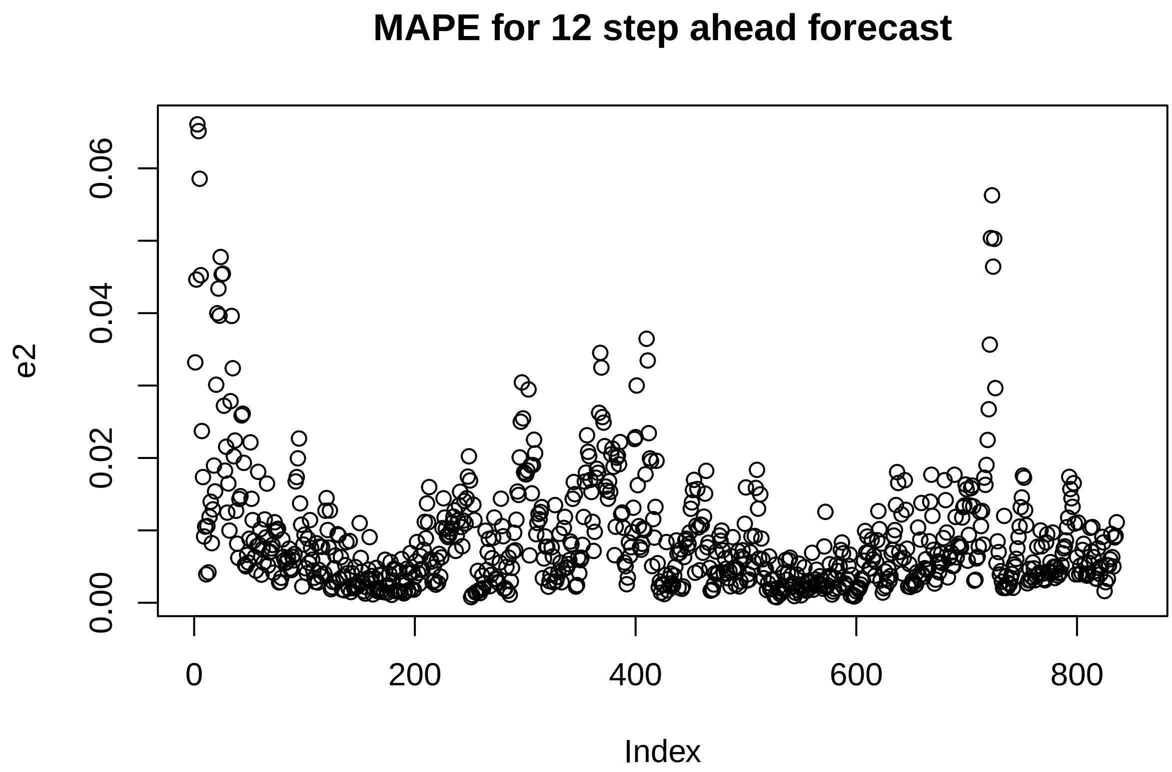
The VAR model seems to give far better forecasts for both CPI and M2 based on how much smaller the error bands are.

n-a

```r
#Recursive window with 12 steps ahead forecast

cpi2=cpi[2]
library(foreach)
library(MLmetrics)
```

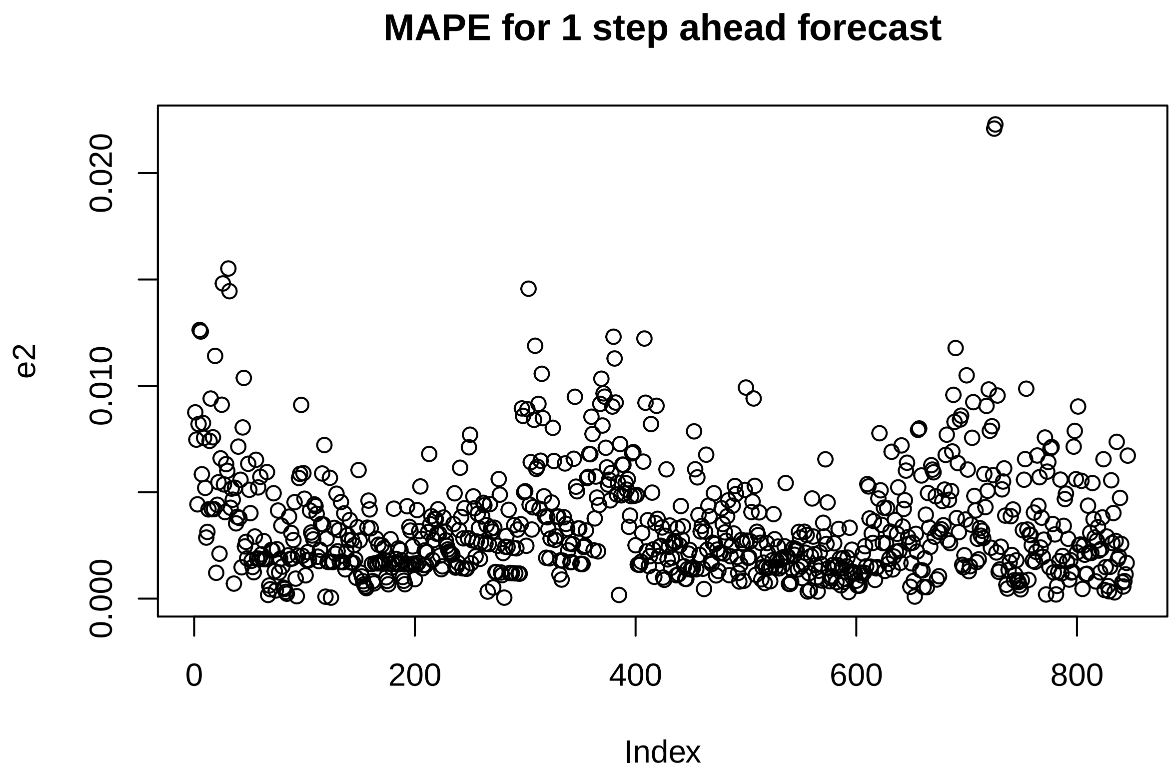
```
##
## Attaching package: 'MLmetrics'

## The following object is masked from 'package:base':
##
##     Recall
```

```r
library(forecast)
e2=numeric()
w_size=424
n_windows=848
forecasts=foreach(i=1:n_windows, .combine = rbind) %do% {
  y_in=cpi2[1:(w_size+i),]
  fit=auto.arima(y_in)
  f1=forecast(fit, h=12)
  f1=as.numeric(f1$mean)
  f2=cpi2[(w_size+1+i):(w_size+12+i),]
  e2[i]=MAPE(f1,f2)
}

plot(e2, main="MAPE for 12 step ahead forecast")
```



n-b

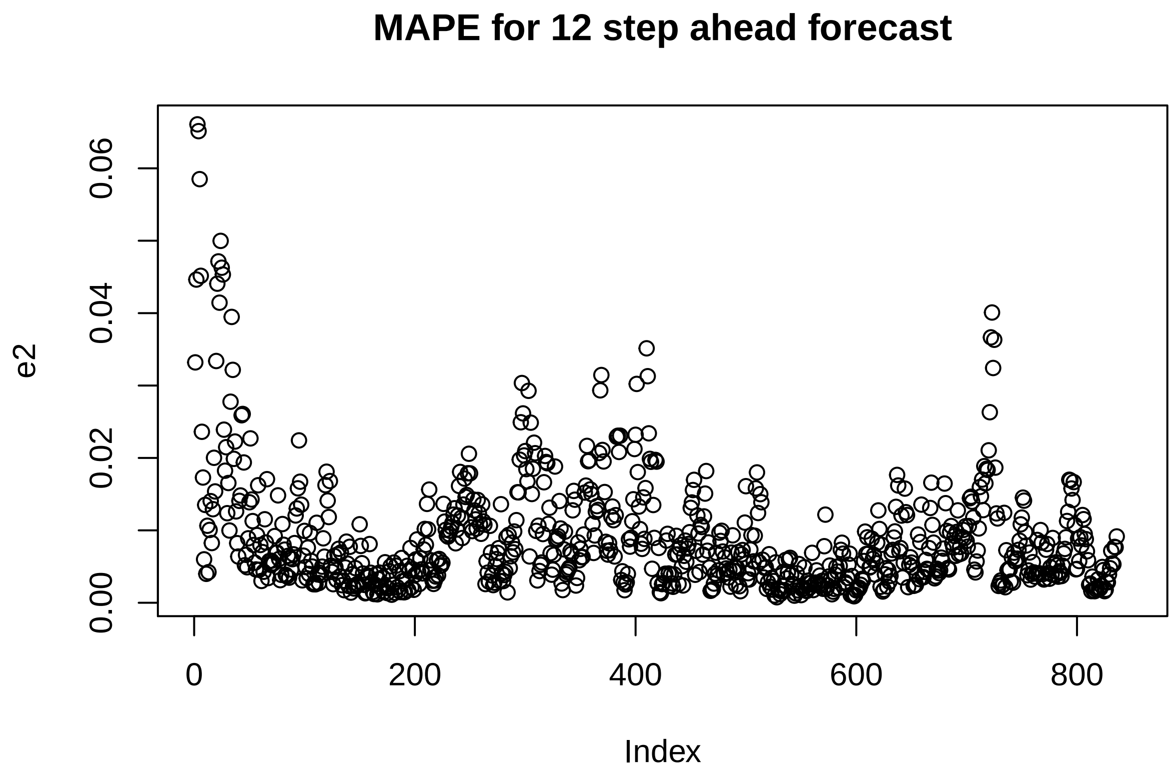
```
#recursive window with 1 step ahead forecast
forecasts=foreach(i=1:n_windows, .combine = rbind) %do% {
  y_in=cpi2[1:(w_size+i),]
  fit=auto.arima(y_in)
  f1=forecast(fit, h=1)
  f1=as.numeric(f1$mean)
  f2=cpi2[(w_size+1+i):(w_size+2+i),]
  e2[i]=MAPE(f1,f2)
}

plot(e2, main="MAPE for 1 step ahead forecast")
```



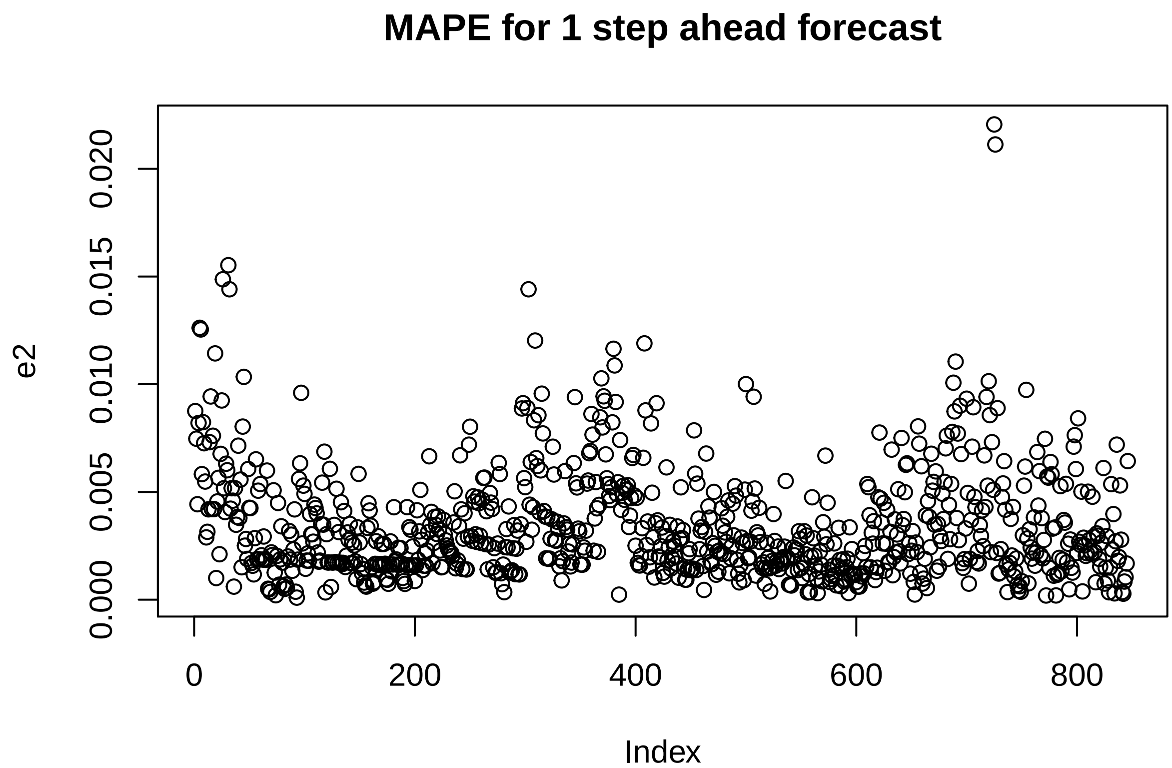
n-c

The model performs better for short term forecasts. This is likely because it is an AR(1) process which tends to have forecast error grow over time.

```
#rolling window with 12 step ahead forecast
forecasts=foreach(i=1:n_windows, .combine = rbind) %do% {
  y_in=cpi2[i:(w_size+i),]
  fit=auto.arima(y_in)
  f1=forecast(fit, h=12)
  f1=as.numeric(f1$mean)
  f2=cpi2[(w_size+1+i):(w_size+12+i),]
  e2[i]=MAPE(f1,f2)
}

plot(e2, main="MAPE for 12 step ahead forecast")
```




```r
#Rolling window with one step ahead forecast
forecasts=foreach(i=1:n_windows, .combine = rbind) %do% {
  y_in=cpi2[i:(w_size+i),]
  fit=auto.arima(y_in)
  f1=forecast(fit, h=1)
  f1=as.numeric(f1$mean)
  f2=cpi2[(w_size+1+i):(w_size+2+i),]
  e2[i]=MAPE(f1,f2)
}

plot(e2, main="MAPE for 1 step ahead forecast")
```



n-e

Recursive backtesting and moving average showed similar levels of error. This is likely because it is an AR(1) process so only one lag has an effect. Consequently recursive and moving average estimations will produce similar results since they both look at the one lag that matters.

Part IV) References

We got all of our data from the FRED database website. the data ranges from 1980 to 2007 with a monthly time frame. The links below lead to the where we got the data

Money supply: https://l.messenger.com/l.php?u=https%3A%2F%2Ffred.stlouisfed.org%2Fseries%2FM2NS&h=AT0ZEuzIBO_mXQyn6dXWonfEGrMW6N7seOx29LuZzRMhOFzU0w3d9PI7sMPF127fQ4f2FZnG5-qRDWSSp7PK78M_X1g8ZNckMHmp8vPYzRvUuoMbTNUnNIm3FFb1-Khu2e8

CPI: https://fred.stlouisfed.org/series/CPIAUCNS

Part V) Results

We conclude that both movements in M2 and CPI are very 'clean' AR(1) processes. Our most interesting finding wasthat lags in M2 were not strong predictors of movements in CPI like a simple univariate understanding would suggest. What was even more interesting was that CPI lags were significant in predicting movements in M2. This reflects the endogenous relationship between the two variables since you would expect an inflation targeting central bank like the US Federal Reserve to react to changes in prices levels in a way that would impact money supply measures like M2. Further investigations should look at other model specifications and other money supply measures. We could also use all of the variables from the full model from the quantity theory of money rather than just M2.