PSTAT 174 Final Project

Jay Shreedhar

2023-03-26

Results

House Price Index data: https://fred.stlouisfed.org/series/ATNHPIUS41940Q

Unemployment Rate data: https://fred.stlouisfed.org/series/SANJ906URN

Before reading in the data, I manually removed all rows prior to 1990 and after Jan 1, 2022 so that all the time series matched up. This also means we do not have data after Q1 2022 in the `hpi1990` and `unemployment` time series — fortunately, the original dataset contains data for Q2-Q4 2022, so we can compare the model forecast to actual measurements and find out how accurate it is.

```r
# open necessary libraries, go into necessary directory
library(astsa)
library(forecast)
library(dplyr)
setwd('/Users/shobhanashreedhar/R/PSTAT 174')
# read CSV file
hpicsv <- read.csv('RealEstateTimeSeries.csv')
hpi1990csv <- read.csv('RealEstateTimeSeries1990.csv')
unempcsv <- read.csv('UnemploymentRate.csv')
```

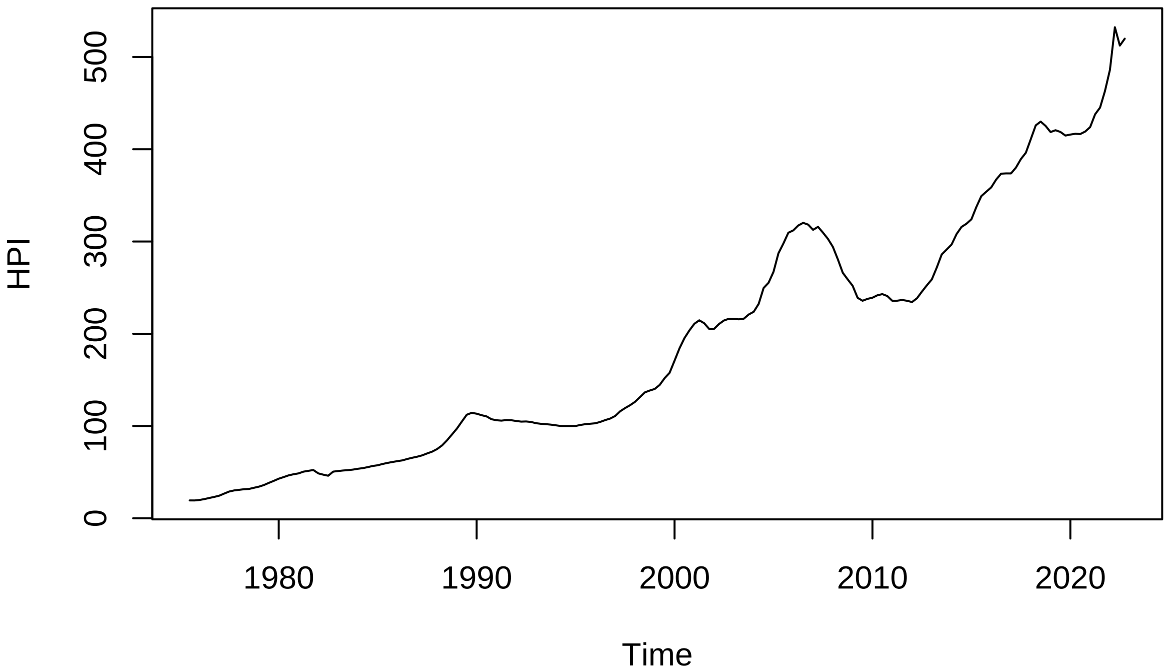
```r
# convert to time series
hpi <- ts(subset(hpicsv, select=-DATE), start = c(1975, 3), frequency=4)
hpi1990 <- ts(subset(hpi1990csv, select=-DATE), start=c(1990, 1), frequency=4)
unempl <- ts(subset(unempcsv, select=-DATE), start=c(1990, 1), frequency=12)
```

```r
# convert non-quarterly time series to quarterly
unemployment <- aggregate.ts(unempl, nfrequency=4, FUN='mean')
```

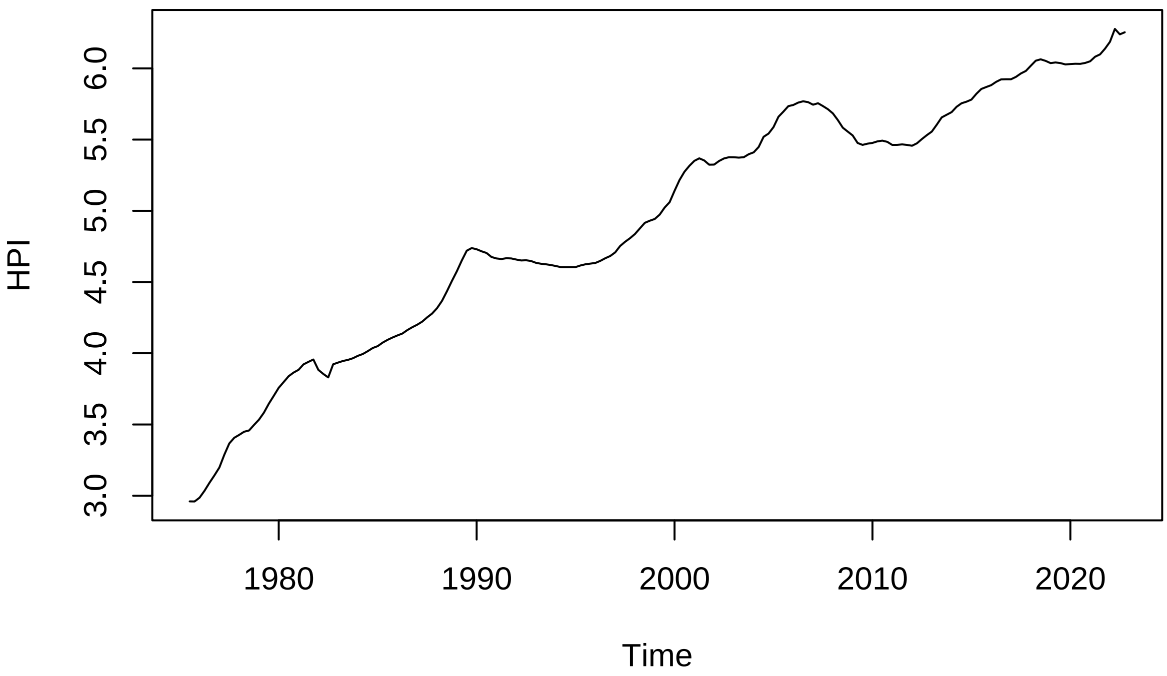
Fitting a univariate SARIMA model

Initial investigation

```r
# plotting the time series and log of the time series to see which is better modeled
plot.ts(hpi)
```




```
plot.ts(log(hpi))
```



Determining stationarity

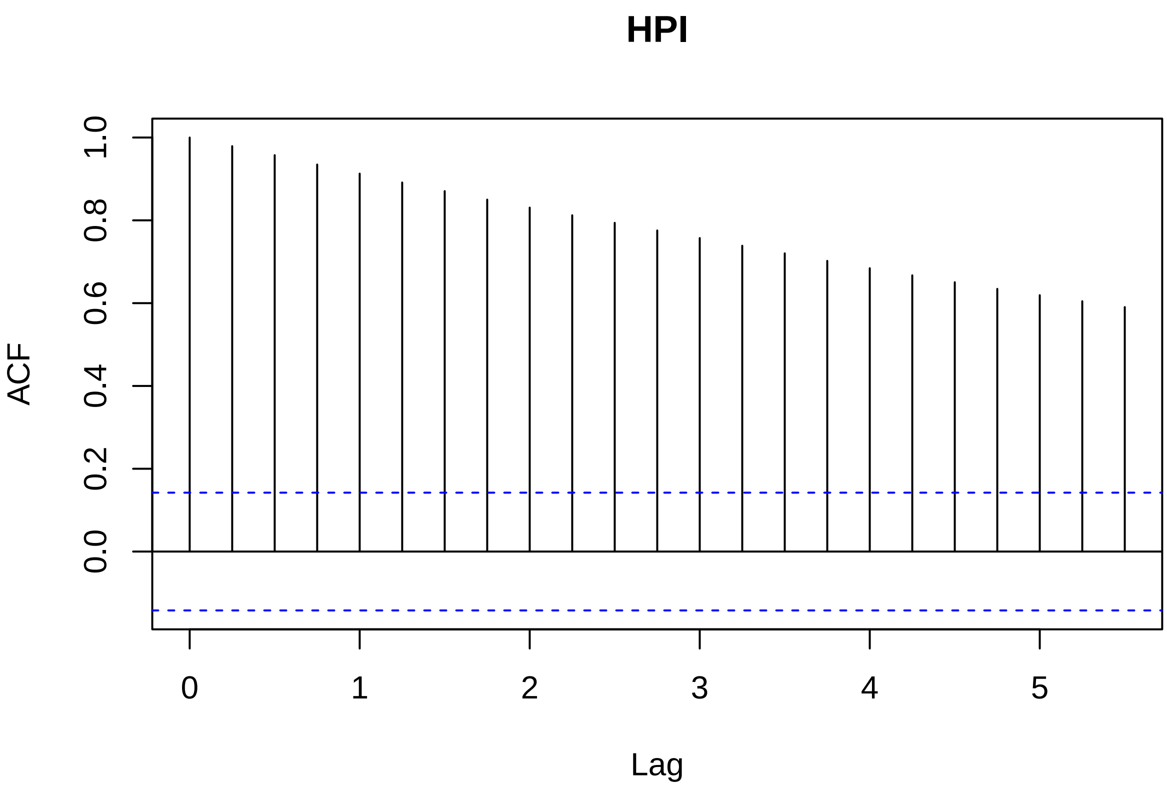
```
# running auto arima to find the best models
# and AIC values for both the original time
# series and the log
auto.arima(hpi, seasonal=TRUE)
```

```
## Series: hpi
## ARIMA(2,1,2)(1,0,0)[4] with drift
##
## Coefficients:
##          ar1      ar2      ma1     ma2    sar1  drift
##       1.2937  -0.7194  -0.9395  0.6340  0.5099  2.623
## s.e.  0.1330   0.1338   0.1526  0.1351  0.1056  1.237
##
## sigma^2 = 28.08:  log likelihood = -581.11
## AIC=1176.22   AICc=1176.84   BIC=1198.91
```

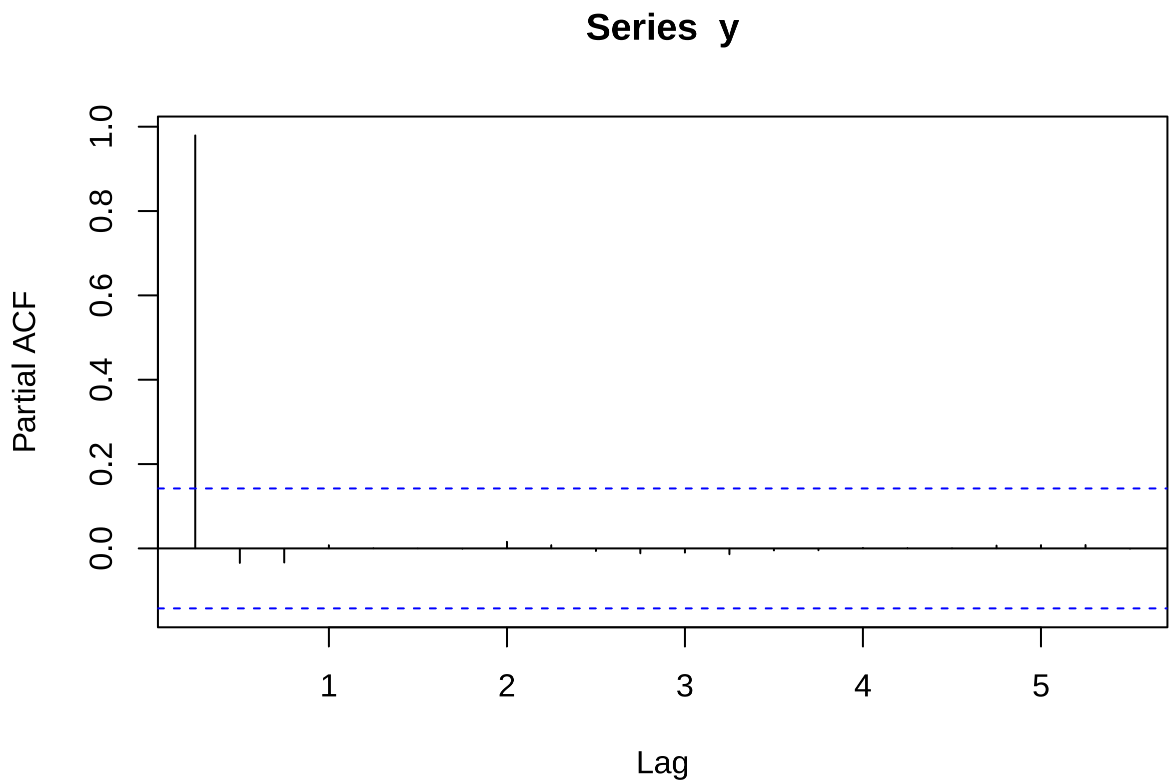
```
auto.arima(log(hpi), seasonal=TRUE)
```

```
## Series: log(hpi)
## ARIMA(1,1,0)(2,0,1)[4] with drift
##
## Coefficients:
##          ar1     sar1    sar2    sma1   drift
##       0.6439  -0.6037  0.2249  0.8422  0.0172
## s.e.  0.0557   0.2471  0.0946  0.2370  0.0054
##
## sigma^2 = 0.0004149:  log likelihood = 469.83
## AIC=-927.65   AICc=-927.19   BIC=-908.2
```

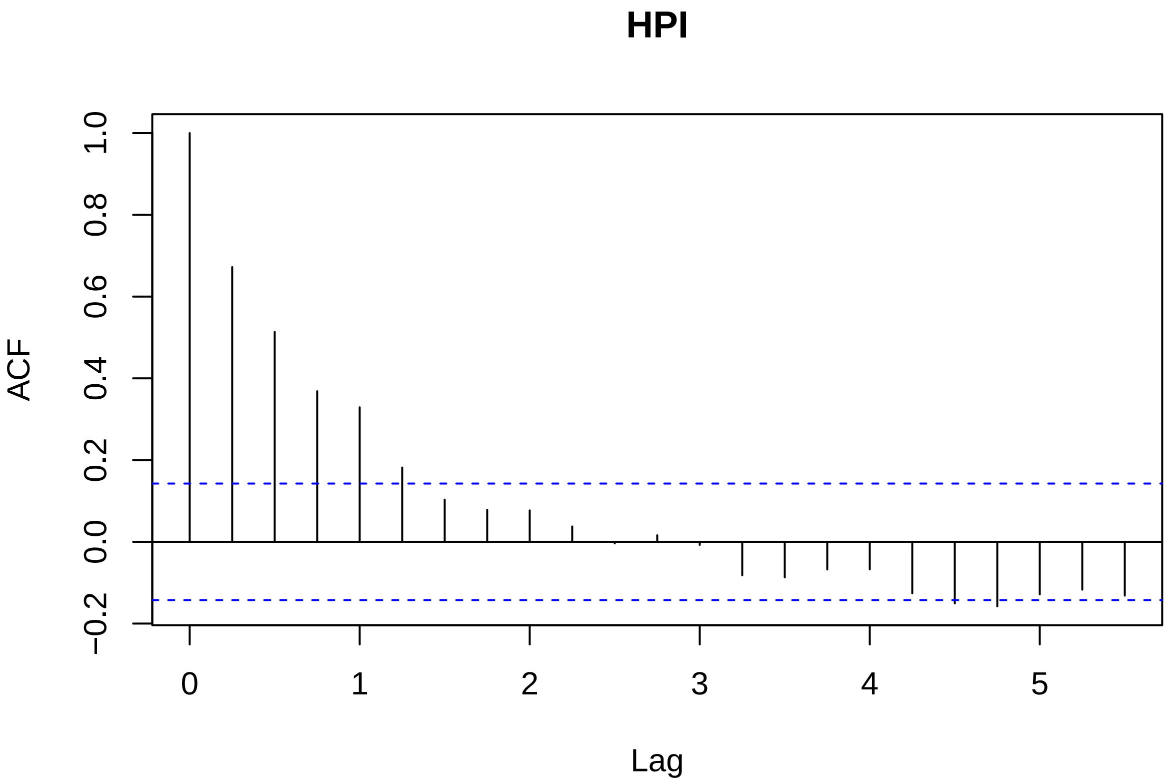
```
y <- log(hpi)

acf(y)
```



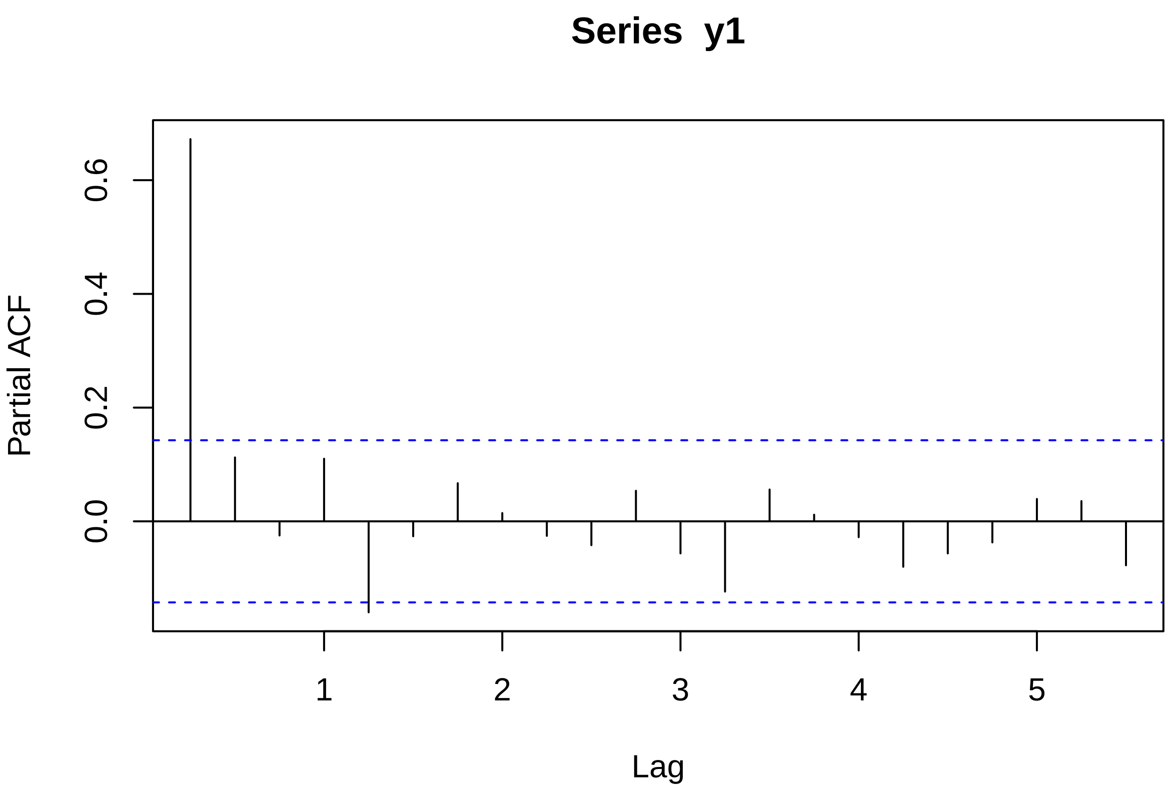
```
pacf(y)
```



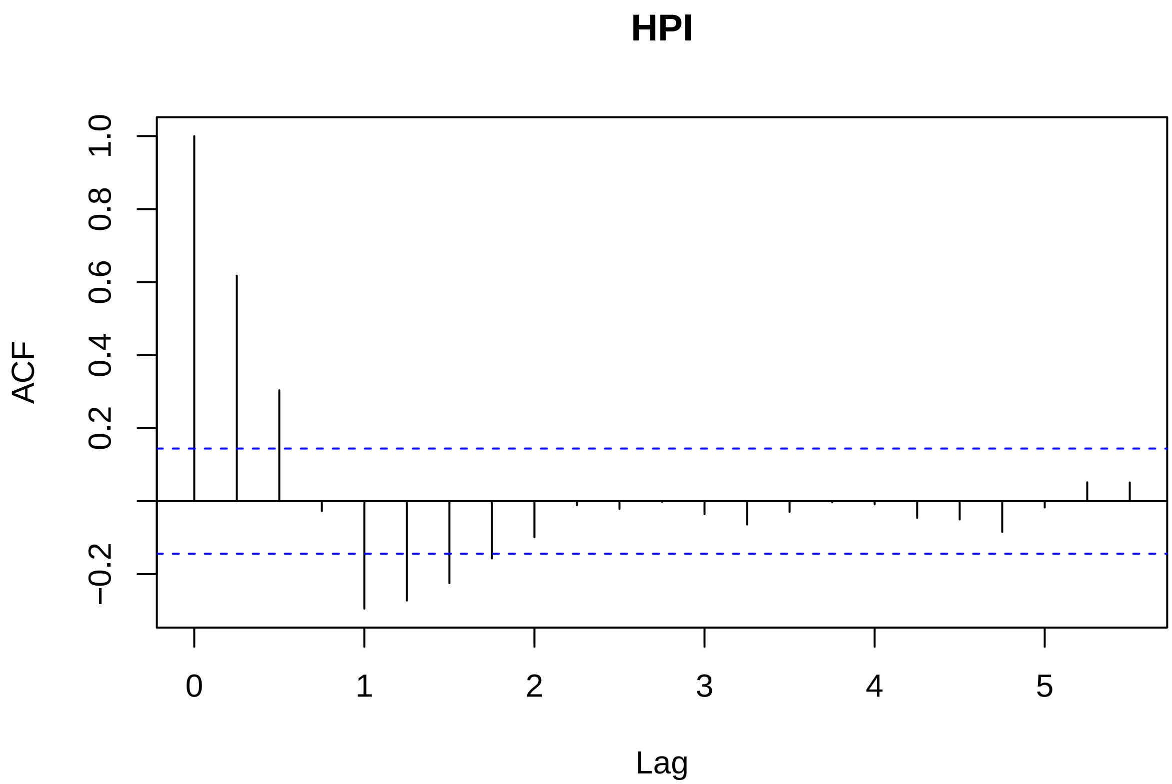

```
y1 <- diff(y)
acf(y1)
```


HPI
ACF
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
1
2
3
4
5
Lag

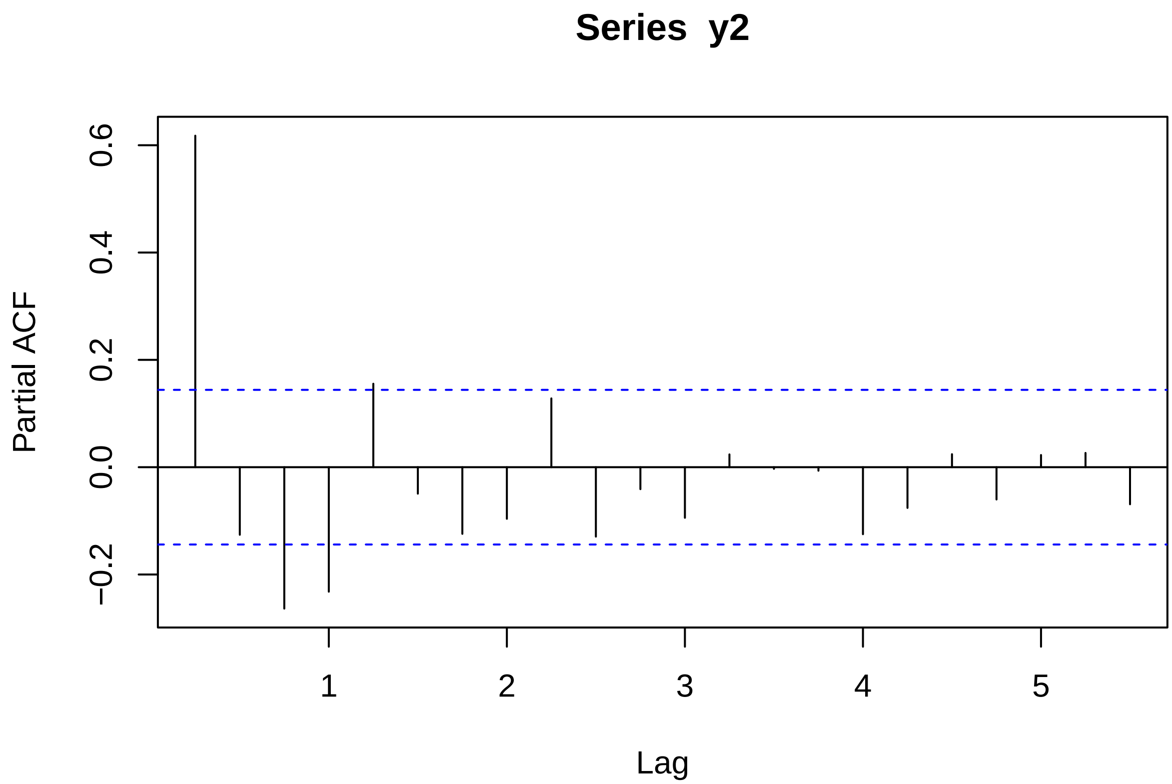
```
pacf(y1)
```


Series y1
Partial ACF
0.6
0.4
0.2
0.0
1
2
3
4
5
Lag

```
y2 <- diff(y1, 4)
acf(y2)
```




```
pacf(y2)
```



Model fitting and diagnostics

```
auto.arima(y2)
```

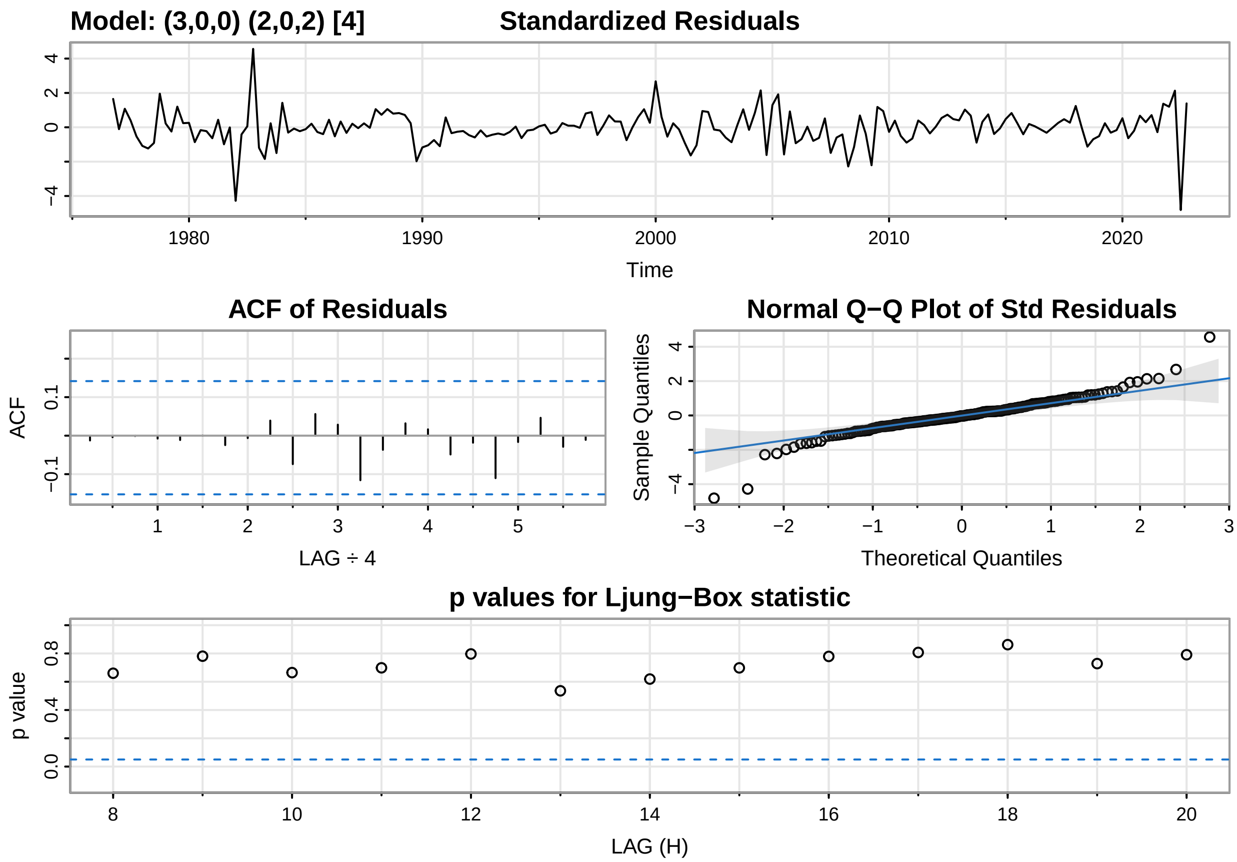
```
## Series: y2
## ARIMA(3,0,0)(2,0,2)[4] with zero mean
##
## Coefficients:
##          ar1     ar2      ar3      sar1    sar2     sma1     sma2
##       0.6283  0.1692  -0.1780  -0.6039  0.2207  -0.1097  -0.7963
## s.e.  0.0736  0.0953   0.0861   0.2664  0.1115   0.2527   0.2372
##
## sigma^2 = 0.0004134:  log likelihood = 457.76
## AIC=-899.53   AICc=-898.71   BIC=-873.77
```

```
sarima(y2, 3, 0, 0, 2, 0, 2, 4)
```

```
## initial  value -3.474409
## iter   2 value -3.795586
## iter   3 value -3.847739
## iter   4 value -3.879596
## iter   5 value -3.900654
## iter   6 value -3.904244
## iter   7 value -3.906610
## iter   8 value -3.909272
## iter   9 value -3.915059
```

```
## iter  10 value -3.917297
## iter  11 value -3.926107
## iter  12 value -3.927552
## iter  13 value -3.932583
## iter  14 value -3.935091
## iter  15 value -3.935882
## iter  16 value -3.940039
## iter  17 value -3.940247
## iter  18 value -3.940375
## iter  19 value -3.940388
## iter  20 value -3.940463
## iter  21 value -3.940474
## iter  22 value -3.940564
## iter  23 value -3.940617
## iter  24 value -3.940696
## iter  25 value -3.940701
## iter  26 value -3.940772
## iter  27 value -3.940938
## iter  28 value -3.942755
## iter  29 value -3.943203
## iter  30 value -3.944455
## iter  31 value -3.946614
## iter  32 value -3.947867
## iter  33 value -3.949419
## iter  34 value -3.953589
## iter  35 value -3.955186
## iter  36 value -3.957084
## iter  37 value -3.957350
## iter  38 value -3.957422
## iter  39 value -3.957488
## iter  40 value -3.957503
## iter  41 value -3.957514
## iter  41 value -3.957514
## final  value -3.957514
## converged
## initial  value -3.868435
## iter   2 value -3.872455
## iter   3 value -3.884702
## iter   4 value -3.894161
## iter   5 value -3.895412
## iter   6 value -3.895655
## iter   7 value -3.895765
## iter   8 value -3.895784
## iter   9 value -3.895788
## iter  10 value -3.895790
## iter  11 value -3.895795
## iter  12 value -3.895807
## iter  13 value -3.895828
## iter  14 value -3.895855
## iter  15 value -3.895880
## iter  16 value -3.895889
## iter  17 value -3.895889
## iter  17 value -3.895889
## iter  17 value -3.895889
```

```
## final  value -3.895889
## converged
```



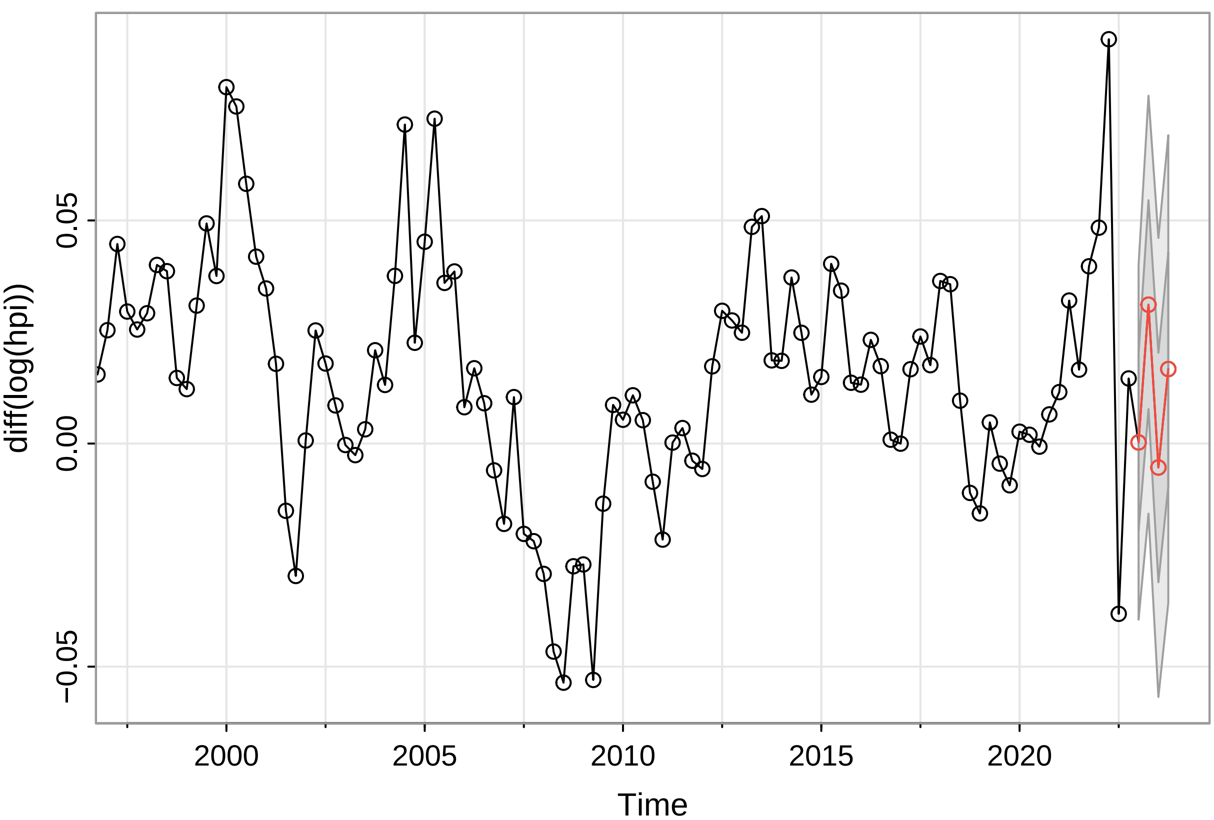

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1     ar2      ar3     sar1    sar2     sma1     sma2   xmean
##       0.6216  0.1679  -0.1828  -0.6053  0.2289  -0.1171  -0.8176  -4e-04
## s.e.  0.0738  0.0952   0.0862   0.2485  0.1084   0.2370   0.2251   4e-04
##
## sigma^2 estimated as 0.000393:  log likelihood = 458.24,  aic = -898.47
##
## $degrees_of_freedom
## [1] 177
##
## $ttable
##       Estimate     SE t.value p.value
## ar1     0.6216 0.0738  8.4212  0.0000
## ar2     0.1679 0.0952  1.7625  0.0797
## ar3    -0.1828 0.0862 -2.1203  0.0354
## sar1   -0.6053 0.2485 -2.4355  0.0159
```

```
## sar2     0.2289 0.1084  2.1118  0.0361
## sma1    -0.1171 0.2370 -0.4939  0.6220
## sma2    -0.8176 0.2251 -3.6328  0.0004
## xmean   -0.0004 0.0004 -1.0281  0.3053
##
## $AIC
## [1] -4.856603
##
## $AICc
## [1] -4.852181
##
## $BIC
## [1] -4.699937
```

Forecasting next 12 months

Next, we will forecast the next 4 quarters of house price index fluctuations.

```
sarima.for(diff(log(hpi)), 4, 3, 0, 0, 2, 0, 2, 4)
```




```
## $pred
##                Qtr1          Qtr2          Qtr3          Qtr4
## 2023  0.0002527363  0.0311223609 -0.0053680461  0.0166946824
##
## $se
##             Qtr1       Qtr2       Qtr3       Qtr4
## 2023 0.01986268 0.02341661 0.02571850 0.02620474
```

Multivariate SARIMAX model

Finding the best model parameters

```r
y3 <- diff(diff(log(hpi1990)), 4)
y4 <- diff(diff(log(unemployment)), 4)
auto.arima(y3, xreg = y4, ic = "aic", trace = TRUE)
```

```
##
##  Regression with ARIMA(2,0,2)(1,0,1)[4] errors : Inf
##  Regression with ARIMA(0,0,0)           errors : -556.9725
##  Regression with ARIMA(1,0,0)(1,0,0)[4] errors : -650.0031
##  Regression with ARIMA(0,0,1)(0,0,1)[4] errors : -633.9687
##  Regression with ARIMA(0,0,0)           errors : -558.6216
##  Regression with ARIMA(1,0,0)           errors : -620.1698
##  Regression with ARIMA(1,0,0)(2,0,0)[4] errors : -655.2818
##  Regression with ARIMA(1,0,0)(2,0,1)[4] errors : Inf
##  Regression with ARIMA(1,0,0)(1,0,1)[4] errors : Inf
##  Regression with ARIMA(0,0,0)(2,0,0)[4] errors : -564.9108
##  Regression with ARIMA(2,0,0)(2,0,0)[4] errors : -653.3301
##  Regression with ARIMA(1,0,1)(2,0,0)[4] errors : -653.336
##  Regression with ARIMA(0,0,1)(2,0,0)[4] errors : -627.5157
##  Regression with ARIMA(2,0,1)(2,0,0)[4] errors : -657.4378
##  Regression with ARIMA(2,0,1)(1,0,0)[4] errors : -654.3674
##  Regression with ARIMA(2,0,1)(2,0,1)[4] errors : Inf
##  Regression with ARIMA(2,0,1)(1,0,1)[4] errors : Inf
##  Regression with ARIMA(3,0,1)(2,0,0)[4] errors : -655.8921
##  Regression with ARIMA(2,0,2)(2,0,0)[4] errors : -655.7946
##  Regression with ARIMA(1,0,2)(2,0,0)[4] errors : -651.581
##  Regression with ARIMA(3,0,0)(2,0,0)[4] errors : -651.4907
##  Regression with ARIMA(3,0,2)(2,0,0)[4] errors : -651.4772
##  Regression with ARIMA(2,0,1)(2,0,0)[4] errors : -659.1988
##  Regression with ARIMA(2,0,1)(1,0,0)[4] errors : -656.1503
##  Regression with ARIMA(2,0,1)(2,0,1)[4] errors : Inf
##  Regression with ARIMA(2,0,1)(1,0,1)[4] errors : Inf
##  Regression with ARIMA(1,0,1)(2,0,0)[4] errors : -655.0996
##  Regression with ARIMA(2,0,0)(2,0,0)[4] errors : -655.0937
##  Regression with ARIMA(3,0,1)(2,0,0)[4] errors : -657.6506
##  Regression with ARIMA(2,0,2)(2,0,0)[4] errors : -657.5536
##  Regression with ARIMA(1,0,0)(2,0,0)[4] errors : -657.0446
##  Regression with ARIMA(1,0,2)(2,0,0)[4] errors : -653.3442
##  Regression with ARIMA(3,0,0)(2,0,0)[4] errors : -653.2531
##  Regression with ARIMA(3,0,2)(2,0,0)[4] errors : -653.2361
##
##  Best model: Regression with ARIMA(2,0,1)(2,0,0)[4] errors
```

```
## Series: y3
## Regression with ARIMA(2,0,1)(2,0,0)[4] errors
##
## Coefficients:
##           ar1     ar2     ma1     sar1     sar2  UnempRate
##       -0.0794  0.6276  0.9826  -0.5744  -0.2113    -0.0045
```

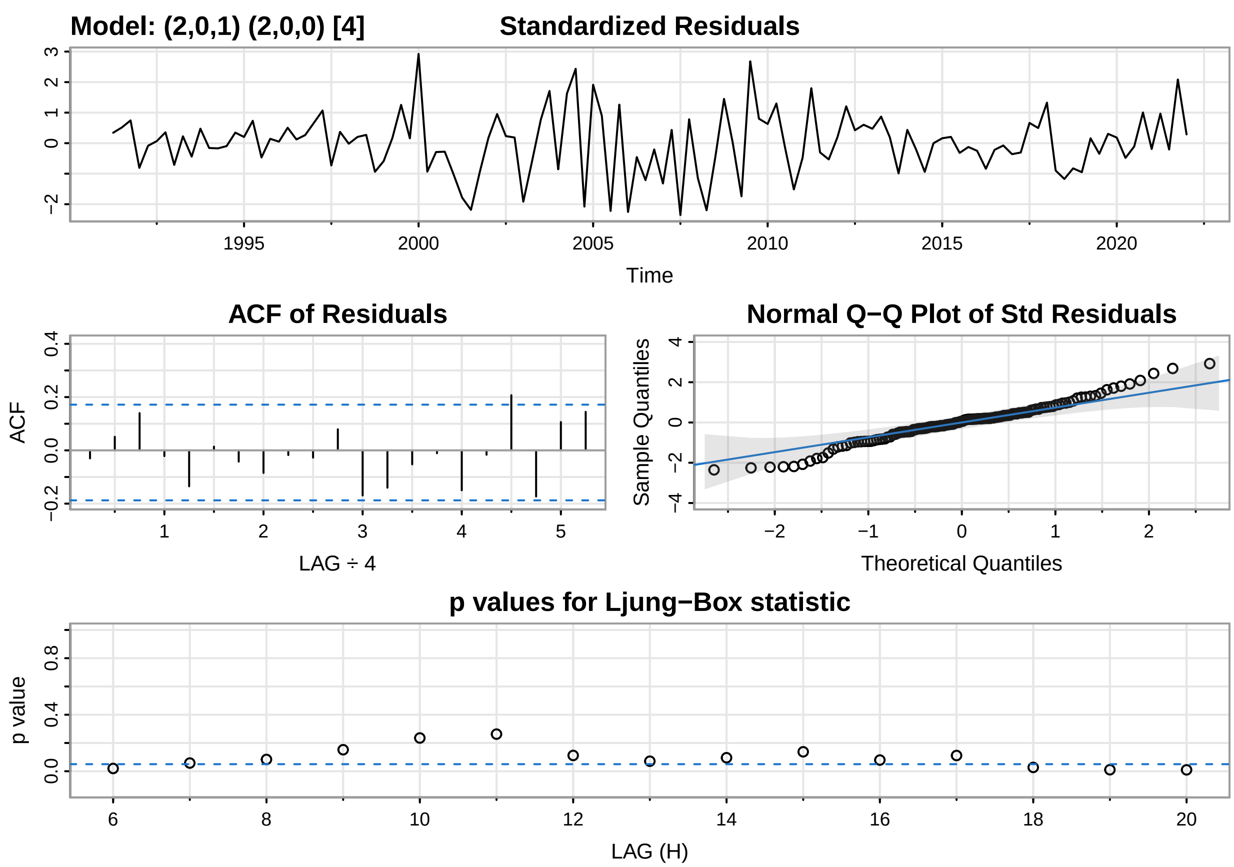
```
## s.e.   0.0801  0.0858  0.0325   0.0987   0.0920     0.0072
##
## sigma^2 = 0.0002622:  log likelihood = 336.6
## AIC=-659.2   AICc=-658.23   BIC=-639.46
```

Diagnostics

```r
sarima(y3, xreg=y4, 2, 0, 1, 2, 0, 0, 4)
```

```
## initial  value -3.653337
## iter   2 value -3.884881
## iter   3 value -4.032138
## iter   4 value -4.053068
## iter   5 value -4.088736
## iter   6 value -4.089446
## iter   7 value -4.090388
## iter   8 value -4.090537
## iter   9 value -4.091879
## iter  10 value -4.095461
## iter  11 value -4.098142
## iter  12 value -4.103219
## iter  13 value -4.110901
## iter  14 value -4.122679
## iter  15 value -4.124615
## iter  16 value -4.126899
## iter  17 value -4.128802
## iter  18 value -4.132120
## iter  19 value -4.134825
## iter  20 value -4.136508
## iter  21 value -4.145111
## iter  22 value -4.146051
## iter  23 value -4.149162
## iter  24 value -4.150909
## iter  25 value -4.152045
## iter  26 value -4.152095
## iter  27 value -4.154626
## iter  28 value -4.155330
## iter  29 value -4.155879
## iter  30 value -4.156122
## iter  31 value -4.156637
## iter  32 value -4.156904
## iter  33 value -4.158915
## iter  34 value -4.160277
## iter  35 value -4.161345
## iter  36 value -4.161990
## iter  36 value -4.161990
## iter  37 value -4.163578
## iter  38 value -4.164724
## iter  39 value -4.166503
## iter  40 value -4.168597
## iter  41 value -4.168688
## iter  41 value -4.168688
```

```
## iter  42 value -4.168700
## iter  42 value -4.168700
## iter  43 value -4.168702
## iter  43 value -4.168702
## iter  43 value -4.168702
## final  value -4.168702
## converged
## initial  value -4.121494
## iter   2 value -4.130762
## iter   3 value -4.131413
## iter   4 value -4.131638
## iter   5 value -4.133477
## iter   6 value -4.134027
## iter   7 value -4.134306
## iter   8 value -4.134369
## iter   9 value -4.134406
## iter  10 value -4.134413
## iter  11 value -4.134413
## iter  11 value -4.134413
## iter  11 value -4.134413
## final  value -4.134413
## converged
```



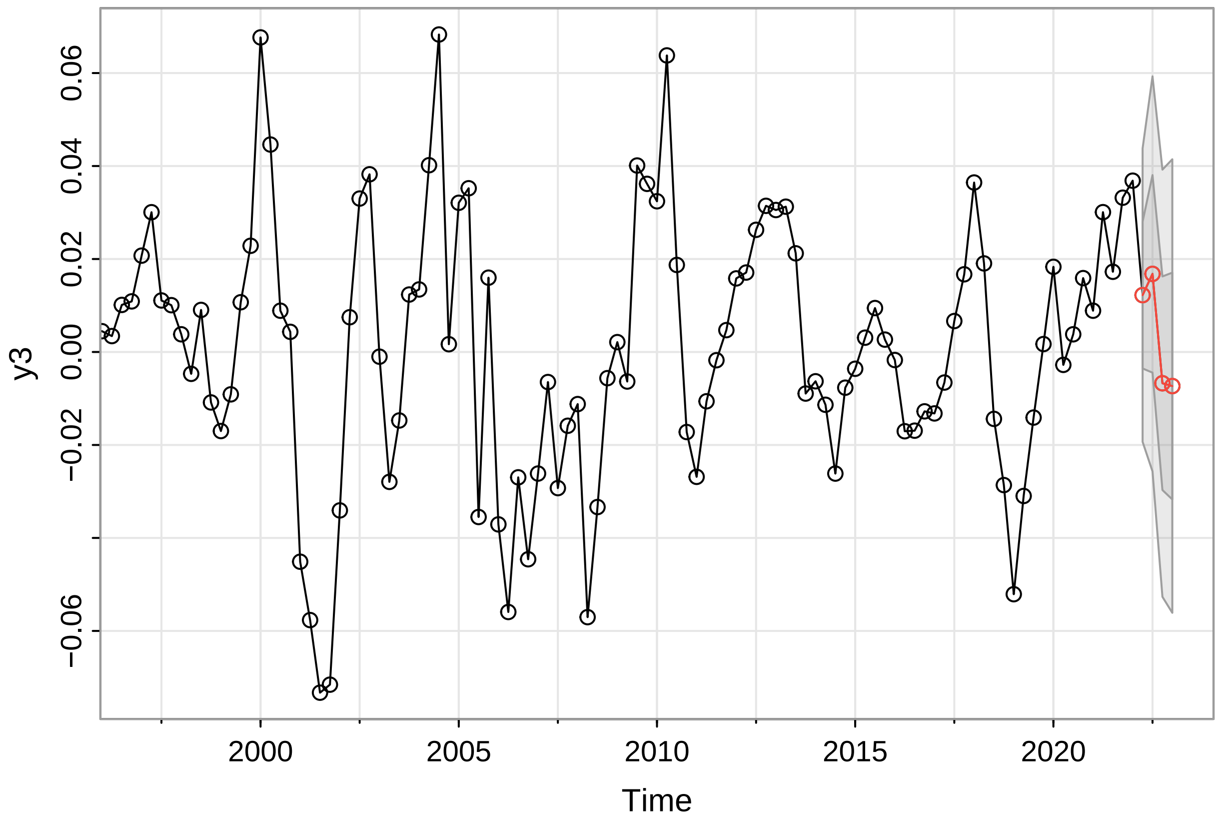

```
## $fit
##
## Call:
```

```
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), xreg = xreg, transform.pars = trans, fixed = fixed, optim.control = list(trace =
##     REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1     ar2     ma1      sar1      sar2  intercept  UnempRate
##       -0.0810  0.6263  0.9826  -0.5744  -0.2116     0.0017    -0.0044
## s.e.   0.0801  0.0858  0.0325   0.0986   0.0919     0.0034     0.0072
##
## sigma^2 estimated as 0.000249:  log likelihood = 336.72,  aic = -657.44
##
## $degrees_of_freedom
## [1] 117
##
## $ttable
##           Estimate     SE t.value p.value
## ar1        -0.0810 0.0801 -1.0109  0.3141
## ar2         0.6263 0.0858  7.2997  0.0000
## ma1         0.9826 0.0325 30.2310  0.0000
## sar1       -0.5744 0.0986 -5.8245  0.0000
## sar2       -0.2116 0.0919 -2.3028  0.0231
## intercept   0.0017 0.0034  0.4886  0.6260
## UnempRate  -0.0044 0.0072 -0.6108  0.5425
##
## $AIC
## [1] -5.301918
##
## $AICc
## [1] -5.294131
##
## $BIC
## [1] -5.119964
```

Forecasting next 12 months

```r
model <- sarima.for(y3, n.ahead=4, p=2, d=0, q=1, P=2, D=0, Q=0, S=4,
                    xreg = y4, newxreg=diff(diff(log(c(2.966666666667, 2.1333333,
                                                       2.266666666666667, 2.233333333333333, 3.15)))))
```

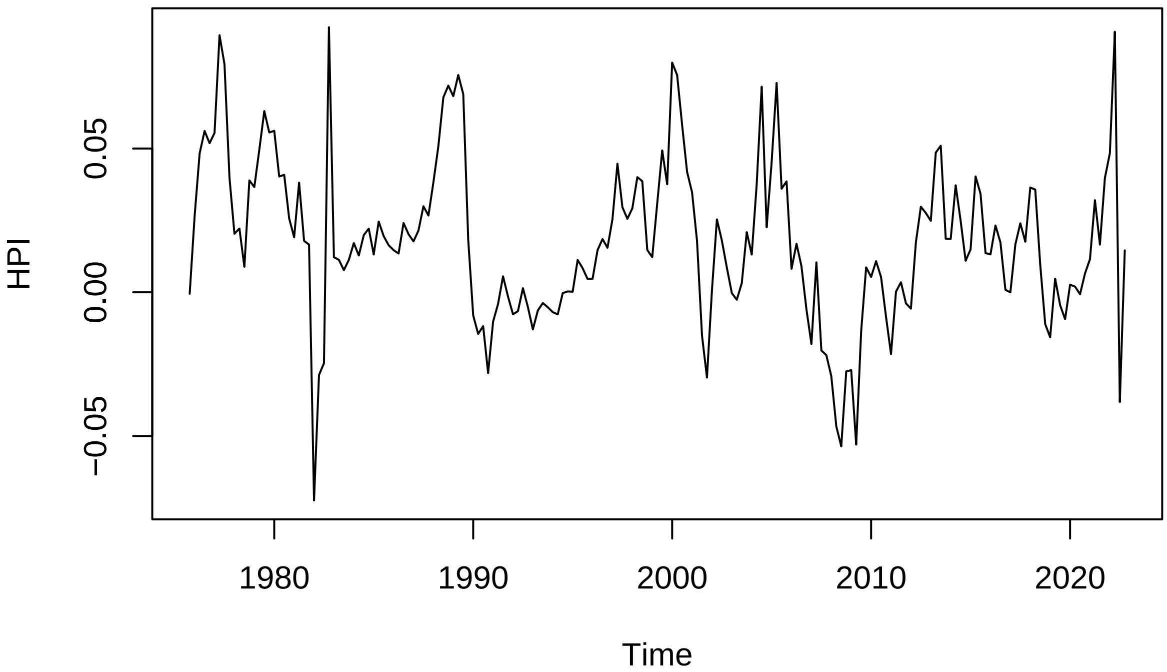
```
## Warning in cbind(intercept = rep(1, n.ahead), newxreg): number of rows of
## result is not a multiple of vector length (arg 2)
```




```
model
```

```
## $pred
##              Qtr1         Qtr2         Qtr3         Qtr4
## 2022                0.012234841  0.016803456 -0.006722654
## 2023 -0.007325699
##
## $se
##            Qtr1       Qtr2       Qtr3       Qtr4
## 2022            0.01578242 0.02124662 0.02297112
## 2023 0.02439127
```

```
plot.ts(diff(log(hpi)))
```


HPI
0.05
0.00
−0.05
1980
1990
2000
2010
2020
Time