STAT 626 Project

Ken Marciel

6/12/2021

US Meat Production

Packages

```
library(readxl) # for read_excel function
library(astsa) # for tsplot function
library(dplyr) # data cleaning
```

Data

```
# read data from Excel file
setwd("C:/Users/keoka/OneDrive - Texas A&M University/Courses/STAT_626/Project/Data/Ken/")
meat <- read_excel("MeatStatsFull.xlsx", sheet = 2)
head(meat)
```

```
## # A tibble: 6 x 17
##   `Red meat and po~ ...2   ...3  ...4  ...5  ...6  ...7  ...8  ...9  ...10 ...11
##   <chr>             <chr>  <chr> <chr> <chr> <lgl> <chr> <chr> <chr> <chr> <chr>
## 1 Type 1/           Comme~ <NA>  <NA>  <NA>  NA    <NA>  Fede~ <NA>  <NA>  <NA>
## 2 <NA>              Beef ~ Veal~ Pork~ Lamb~ NA    Tota~ Beef~ Veal~ Pork~ Lamb~
## 3 Jan-Apr 2021      9241.5 18.5  9635  48.7  NA    1894~ 9088~ 17.9  9590~ 43.8
## 4 Jan-Apr 2020      8748.2 22.1  9457~ 46    NA    1827~ 8622~ 21.4  9414~ 41.8
## 5 Apr-2021          2346.3 4.3   2343~ 13.6  NA    4708  2307~ 4.1   2333~ 12
## 6 Mar-2021          2480.6 4.9   2548~ 13.9  NA    5047~ 2438~ 4.7   2536~ 12.7
## # ... with 6 more variables: ...12 <chr>, ...13 <chr>, ...14 <chr>,
## #   ...15 <chr>, ...16 <chr>, ...17 <chr>
```

```
# total monthly red meat and poultry production from January 1983 to April 2021
meat <- meat[464:5, c(1,17)] # changes data from reverse to chronological order
colnames(meat) <- c("Month","MeatProd")
head(meat)
```

```
## # A tibble: 6 x 2
##   Month    MeatProd
##   <chr>    <chr>
## 1 Jan-1983 4236.146
## 2 Feb-1983 3789.626
## 3 Mar-1983 4489.982
## 4 Apr-1983 4207.824
## 5 May-1983 4376.39
## 6 Jun-1983 4587.926
```

```
dim(meat) # 460 months = 38 1/3 years
```

```
## [1] 460   2
```

```r
class(meat) # data frame
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

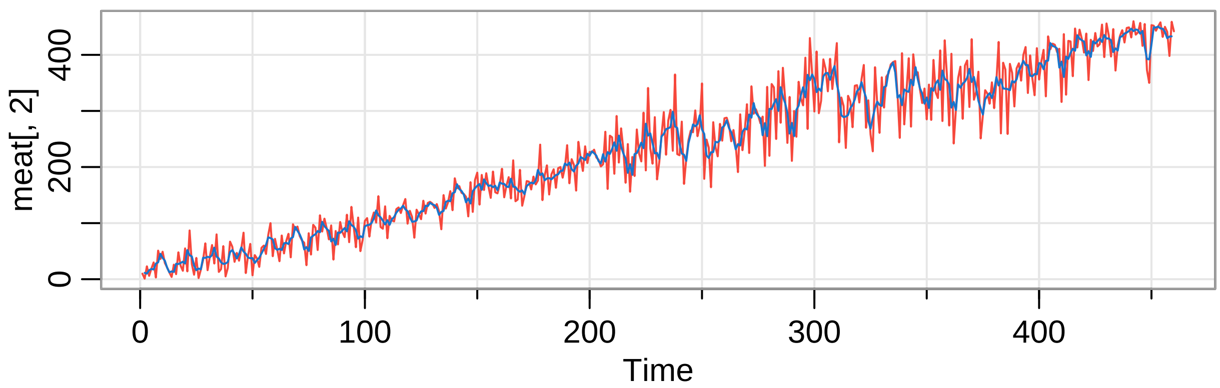
```r
meat = ts(meat) # change data frame to time series
class(meat) # time series
```

```
## [1] "mts"    "ts"     "matrix"
```

Exploratory Data Analysis

Upward trend shows that the raw data is not stationary.

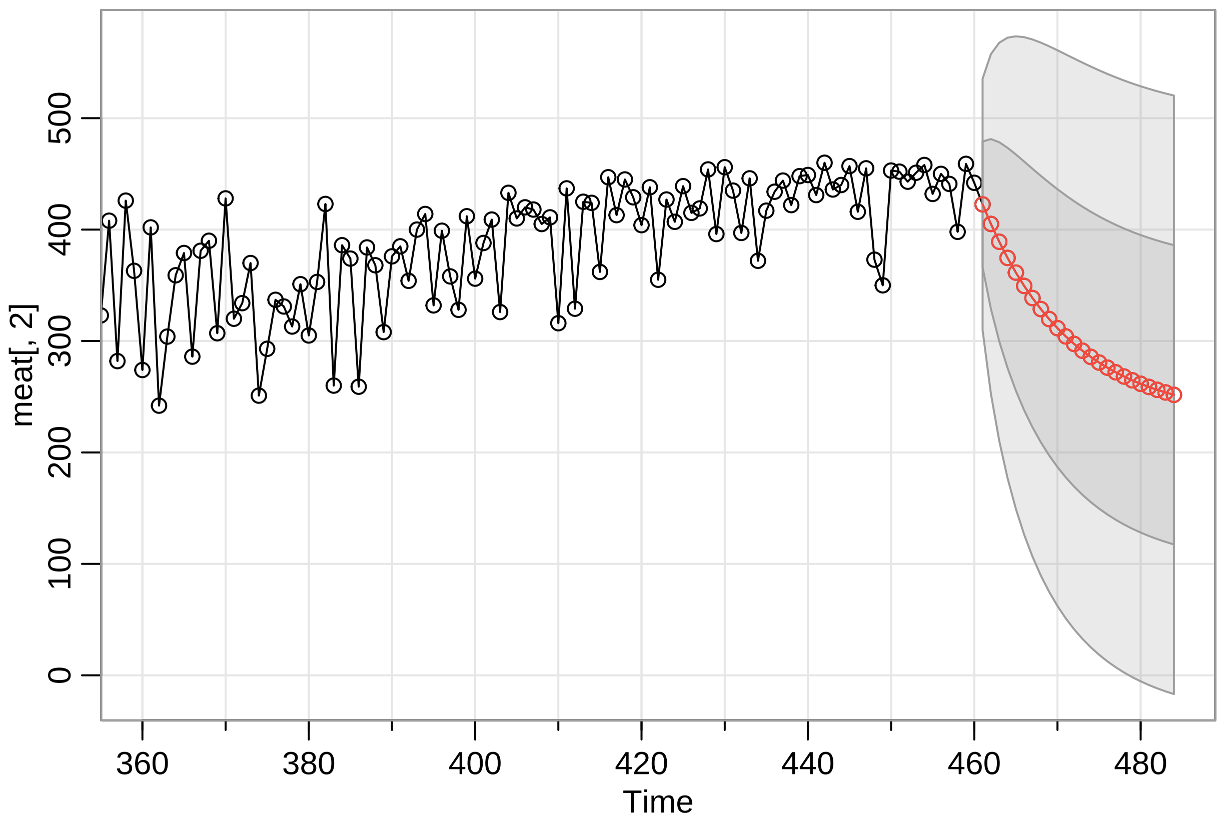
```r
# 3-term moving average
meat_ma = stats::filter(meat[,2], sides = 2, filter = rep(1/3,3))
par(mfrow = c(2,1))
tsplot(meat[,2], col=2)
lines(meat_ma, col=4)
```



Forecasting

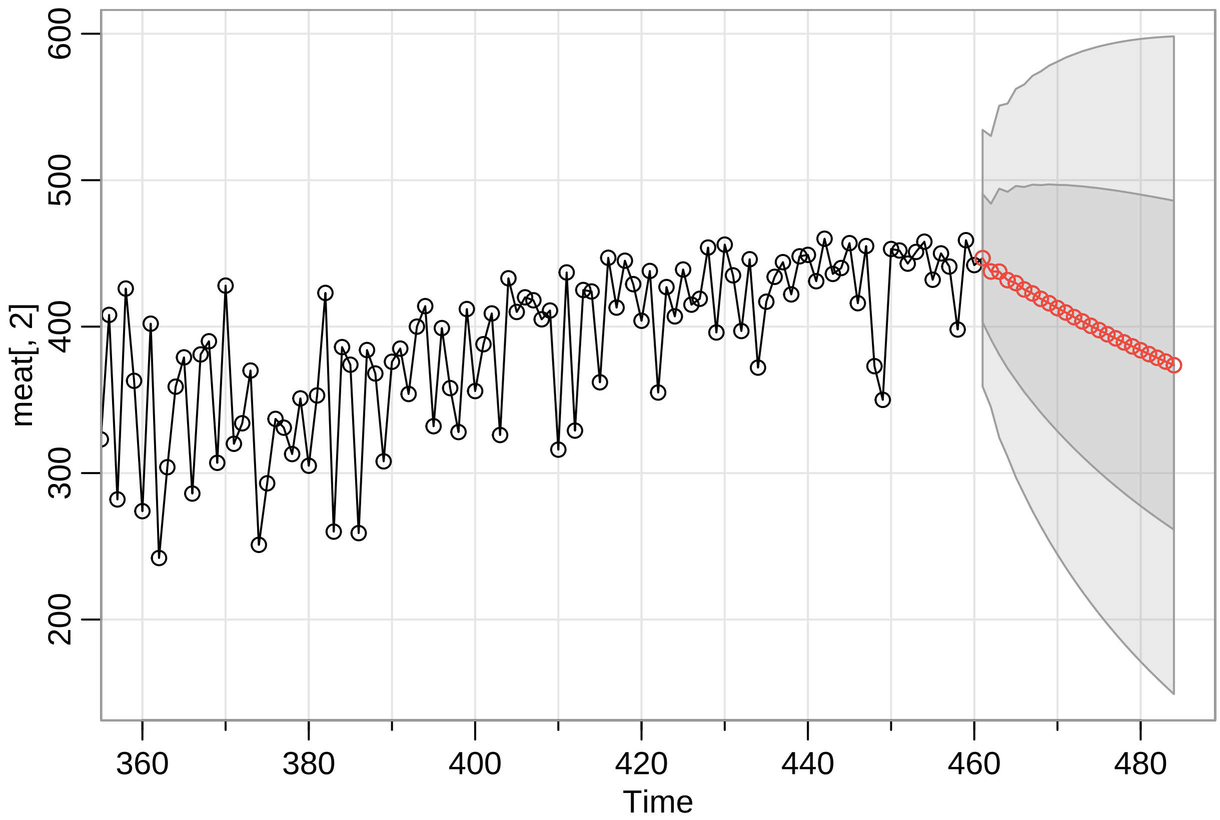
No Transformation AR models seem to work best, but the data is nonstationary.

```r
sarima.for(meat[,2], n.ahead=24, p=1, d=0, q=0) # AR(1)
```



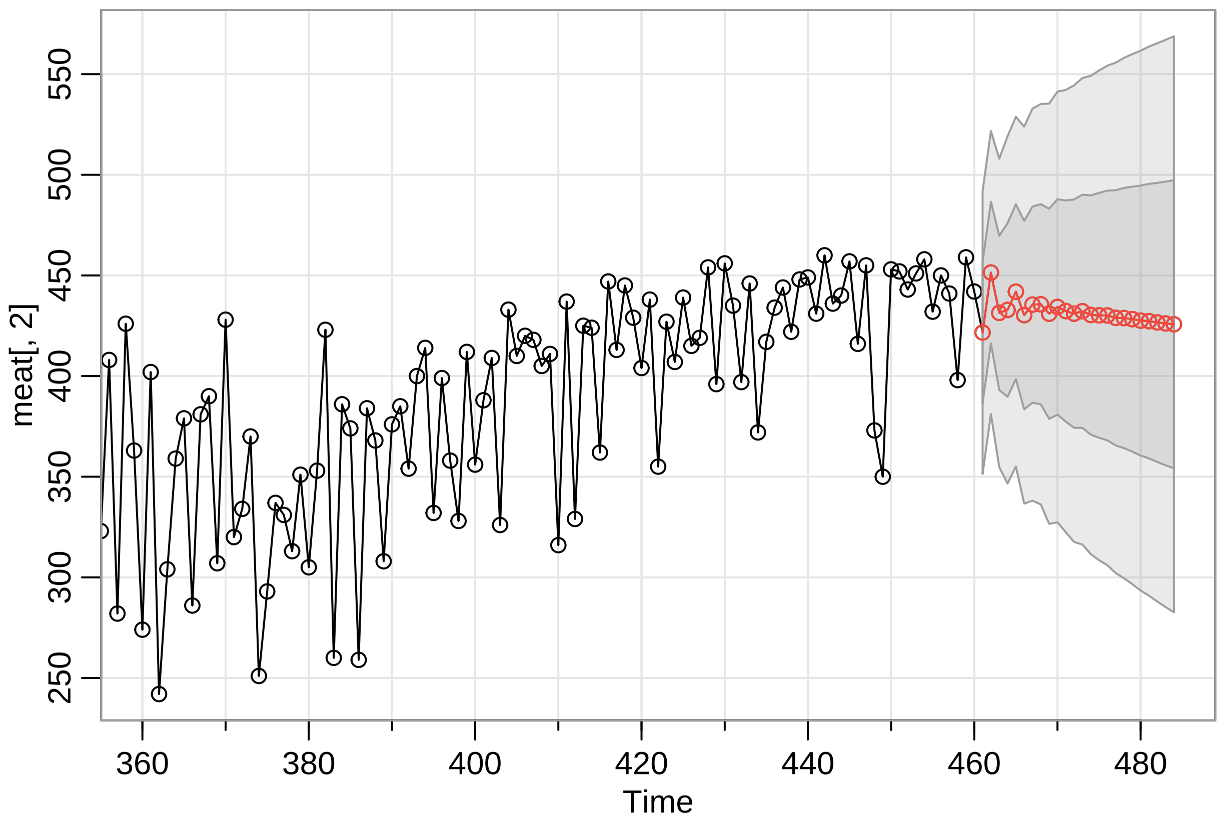

```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 422.6482 405.0683 389.0978 374.5896 361.4096 349.4364 338.5594 328.6783
##  [9] 319.7018 311.5471 304.1391 297.4094 291.2957 285.7418 280.6965 276.1130
## [17] 271.9492 268.1666 264.7303 261.6087 258.7728 256.1966 253.8562 251.7302
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1]  56.41769  76.22176  89.31729  98.82619 106.03329 111.63119 116.04776
##  [8] 119.56983 122.40018 124.68761 126.54421 128.05614 129.29058 130.30052
## [15] 131.12813 131.80722 132.36503 132.82361 133.20087 133.51141 133.76715
## [22] 133.97783 134.15145 134.29457
```

```
sarima.for(meat[,2], n.ahead=24, p=2, d=0, q=0) # AR(2)
```



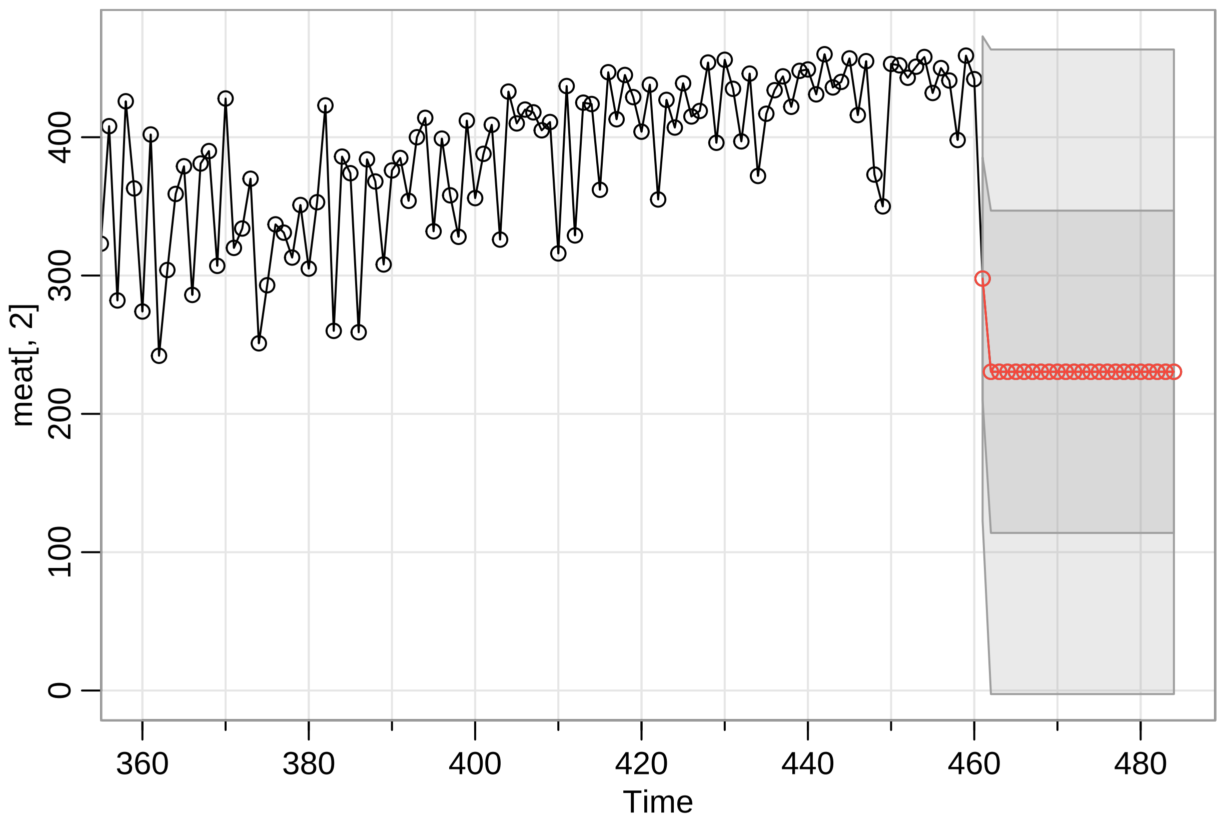

```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 446.7550 437.6426 437.5456 431.7617 429.7340 425.3943 422.6393 418.9638
##  [9] 415.9755 412.6398 409.6198 406.4879 403.5172 400.5306 397.6404 394.7729
## [17] 391.9739 389.2127 386.5074 383.8450 381.2326 378.6641 376.1421 373.6637
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1]  43.81088  46.27355  56.66937  60.27080  66.29932  69.96287  74.30526
##  [8]  77.66915  81.14163  84.13787  87.06508  89.71778  92.25577  94.60976
## [15]  96.84640  98.94530 100.93687 102.81785 104.60394 106.29750 107.90785
## [22] 109.43887 110.89685 112.28574
```

```
sarima.for(meat[,2], n.ahead=24, p=3, d=0, q=0) # AR(3)
```



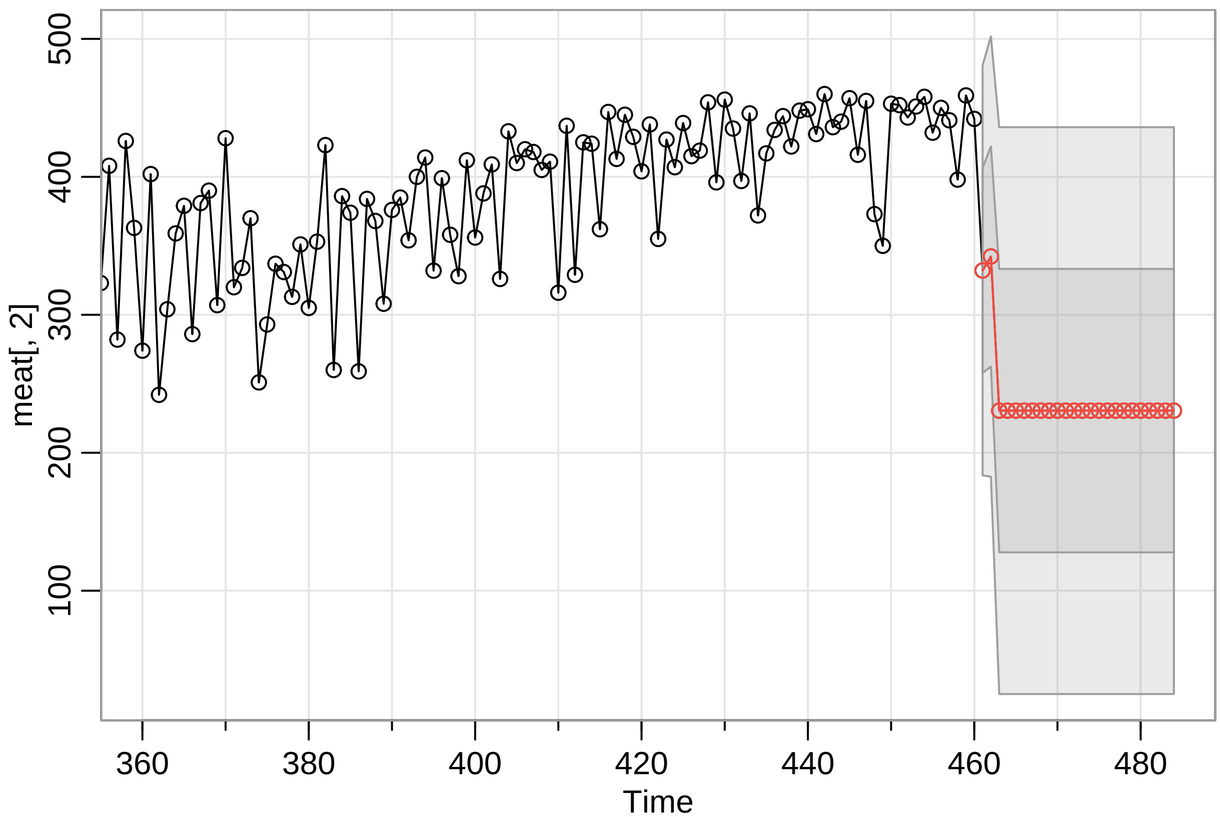

```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 421.6302 451.4472 431.3951 432.8628 441.9406 430.2703 435.5006 435.6824
##  [9] 430.9699 434.3445 432.2917 431.0129 432.1873 430.3659 430.1773 430.0981
## [17] 428.9322 428.8285 428.2811 427.5605 427.2886 426.6604 426.1360 425.7215
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 35.12410 35.14800 38.30532 43.14048 43.43808 46.82331 48.70152 49.73592
##  [9] 52.20084 53.51176 54.93875 56.71705 57.94868 59.40602 60.82575 62.04980
## [17] 63.39714 64.64098 65.83357 67.05773 68.20439 69.34025 70.46416 71.54236
```

```
sarima.for(meat[,2], n.ahead=24, p=0, d=0, q=1) # MA(1)
```



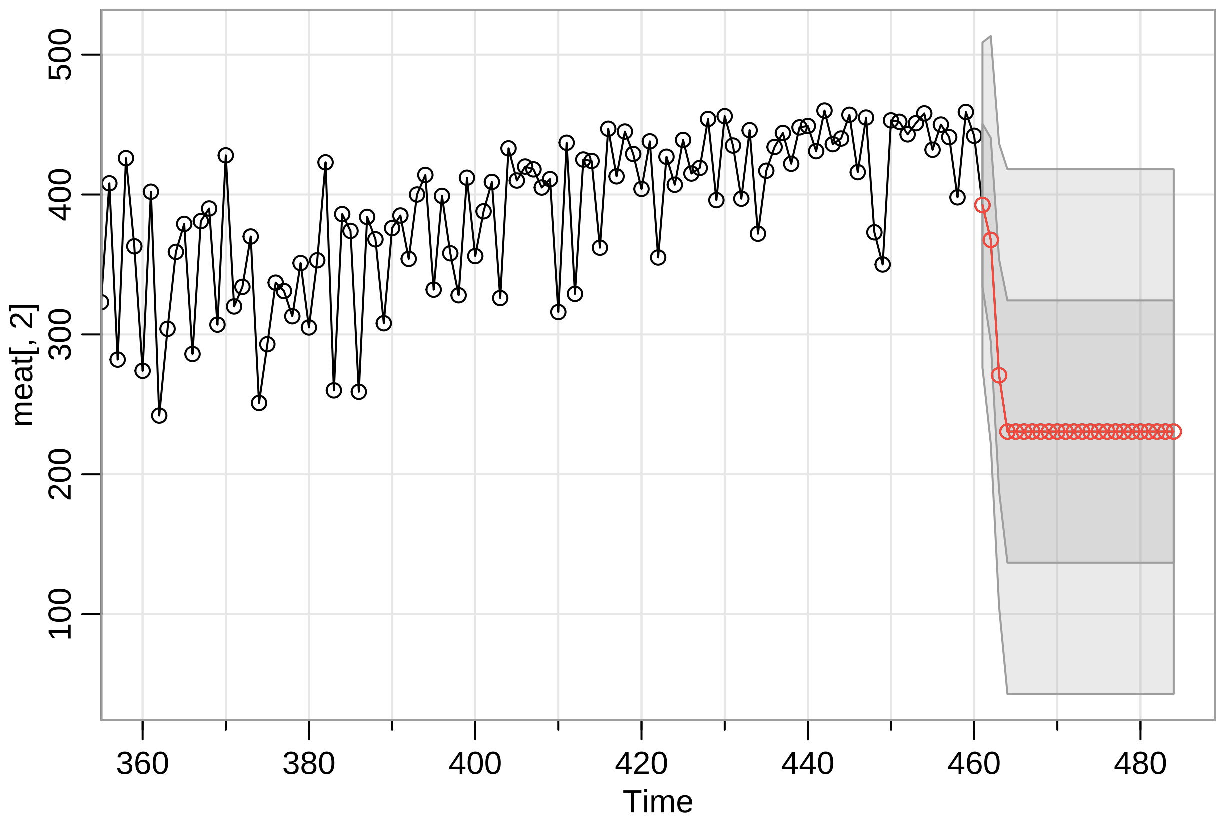

```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 297.7934 230.4252 230.4252 230.4252 230.4252 230.4252 230.4252 230.4252
##  [9] 230.4252 230.4252 230.4252 230.4252 230.4252 230.4252 230.4252 230.4252
## [17] 230.4252 230.4252 230.4252 230.4252 230.4252 230.4252 230.4252 230.4252
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1]  87.58598 116.50247 116.50247 116.50247 116.50247 116.50247 116.50247
##  [8] 116.50247 116.50247 116.50247 116.50247 116.50247 116.50247 116.50247
## [15] 116.50247 116.50247 116.50247 116.50247 116.50247 116.50247 116.50247
## [22] 116.50247 116.50247 116.50247
```

```
sarima.for(meat[,2], n.ahead=24, p=0, d=0, q=2) # MA(2)
```



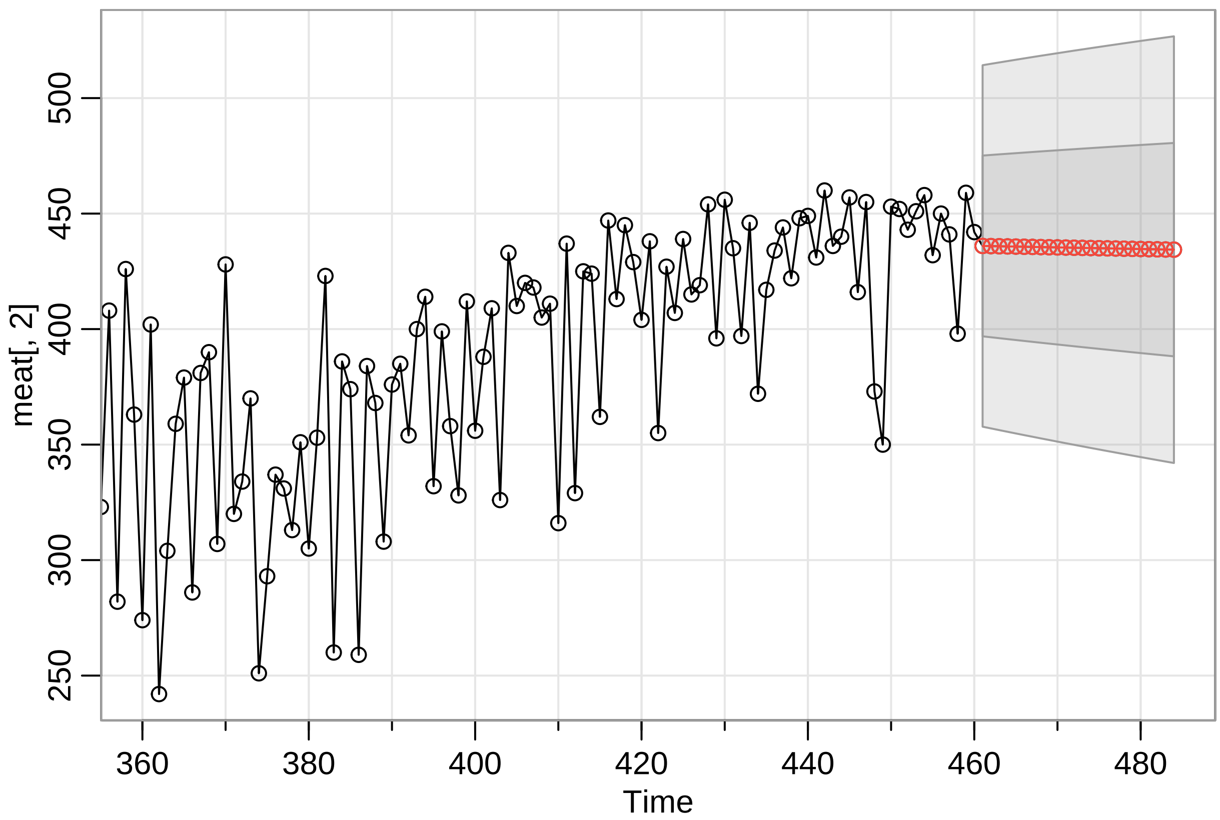

```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 332.1006 342.2846 230.5283 230.5283 230.5283 230.5283 230.5283 230.5283
##  [9] 230.5283 230.5283 230.5283 230.5283 230.5283 230.5283 230.5283 230.5283
## [17] 230.5283 230.5283 230.5283 230.5283 230.5283 230.5283 230.5283 230.5283
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1]  74.27672  79.79195 102.72696 102.72696 102.72696 102.72696 102.72696
##  [8] 102.72696 102.72696 102.72696 102.72696 102.72696 102.72696 102.72696
## [15] 102.72696 102.72696 102.72696 102.72696 102.72696 102.72696 102.72696
## [22] 102.72696 102.72696 102.72696
```

```r
sarima.for(meat[,2], n.ahead=24, p=0, d=0, q=3) # MA(3)
```



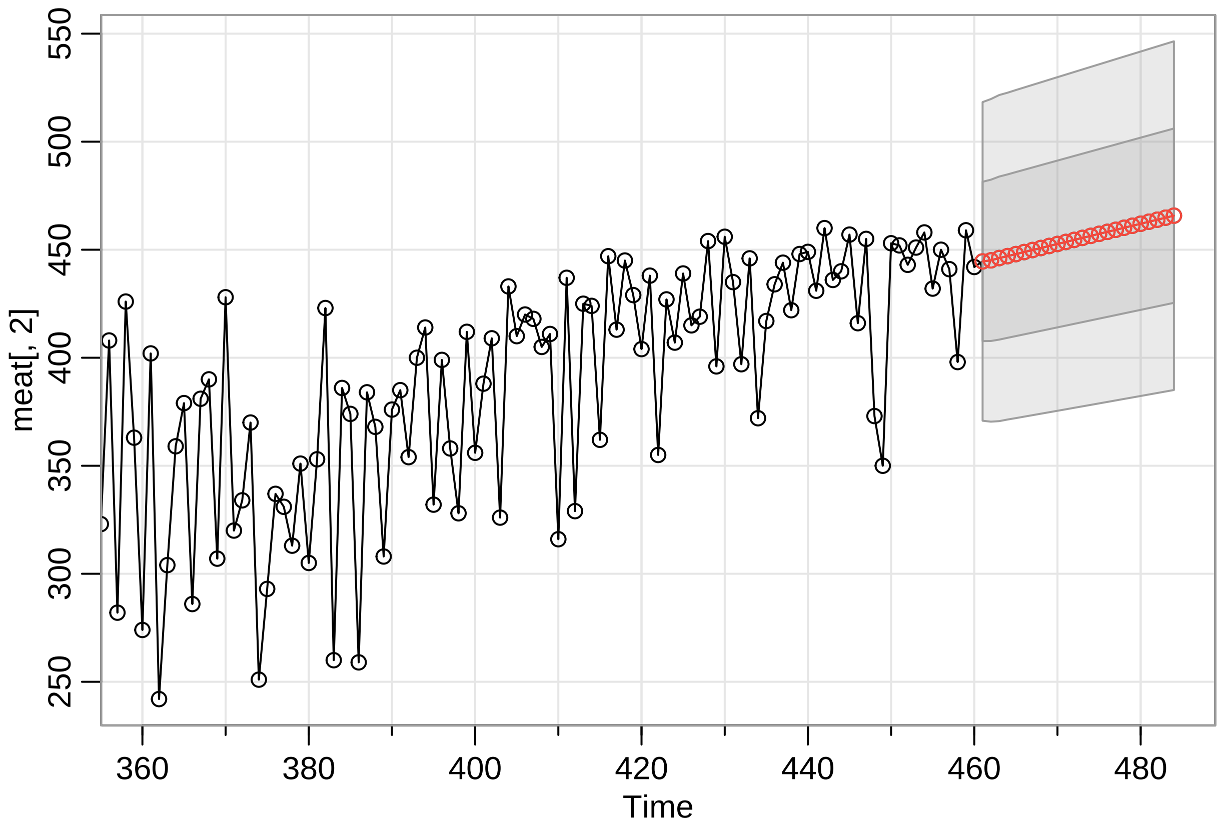

```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 392.5160 367.6075 270.7920 230.5637 230.5637 230.5637 230.5637 230.5637
##  [9] 230.5637 230.5637 230.5637 230.5637 230.5637 230.5637 230.5637 230.5637
## [17] 230.5637 230.5637 230.5637 230.5637 230.5637 230.5637 230.5637 230.5637
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 58.10443 72.83191 82.81926 93.74574 93.74574 93.74574 93.74574 93.74574
##  [9] 93.74574 93.74574 93.74574 93.74574 93.74574 93.74574 93.74574 93.74574
## [17] 93.74574 93.74574 93.74574 93.74574 93.74574 93.74574 93.74574 93.74574
```

```
sarima.for(meat[,2], n.ahead=24, p=1, d=0, q=1) # ARMA(1,1)
```



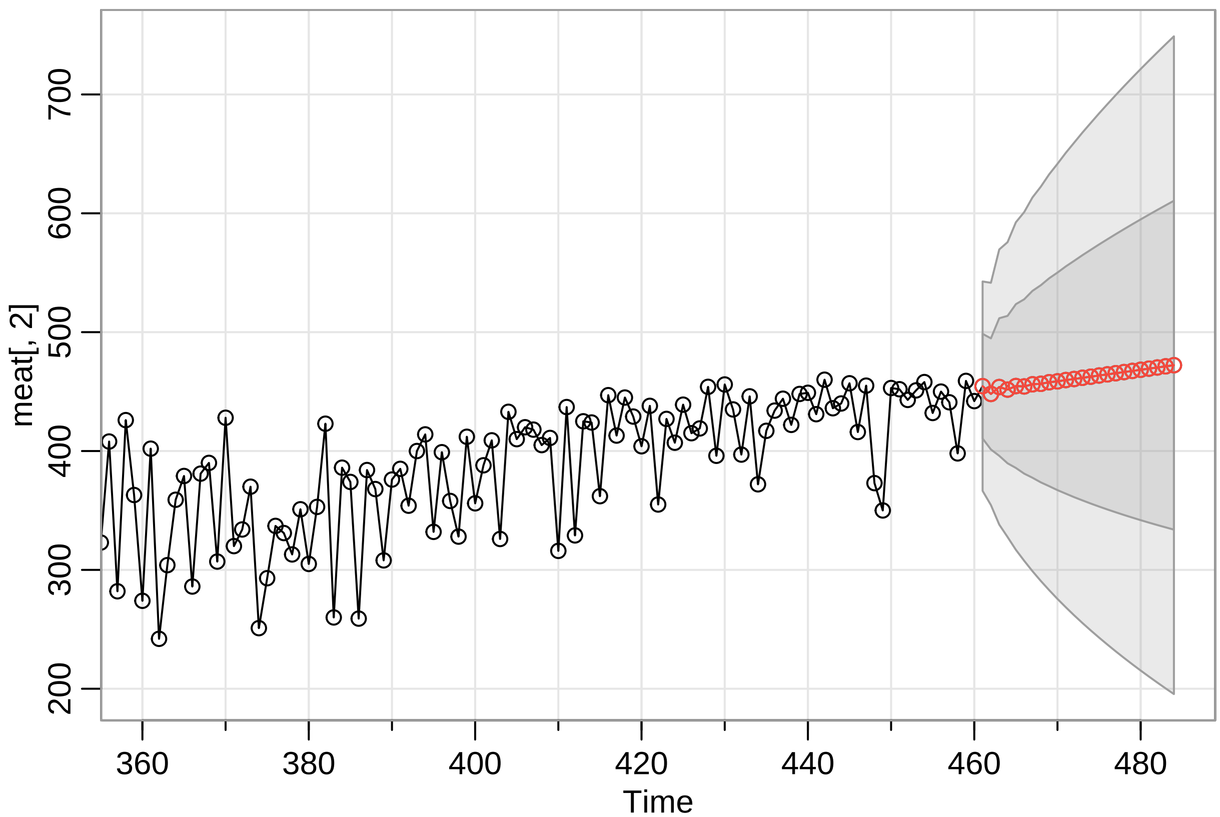

```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 435.9969 435.9267 435.8566 435.7864 435.7163 435.6462 435.5761 435.5060
##  [9] 435.4360 435.3660 435.2960 435.2260 435.1561 435.0861 435.0163 434.9464
## [17] 434.8765 434.8067 434.7369 434.6671 434.5974 434.5277 434.4580 434.3883
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 39.12054 39.45620 39.78881 40.11843 40.44515 40.76902 41.09013 41.40853
##  [9] 41.72429 42.03746 42.34810 42.65627 42.96203 43.26541 43.56648 43.86528
## [17] 44.16186 44.45625 44.74852 45.03868 45.32680 45.61290 45.89702 46.17920
```

```
sarima.for(meat[,2], n.ahead=24, p=1, d=1, q=1) # ARIMA(1,1,1)
```




```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 444.5789 445.0792 446.1308 447.0362 447.9803 448.9142 449.8507 450.7866
##  [9] 451.7226 452.6586 453.5947 454.5307 455.4667 456.4027 457.3387 458.2747
## [17] 459.2107 460.1467 461.0827 462.0187 462.9547 463.8908 464.8268 465.7628
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 36.87696 37.33152 37.73626 37.81964 37.96388 38.08983 38.21994 38.34840
##  [9] 38.47674 38.60458 38.73201 38.85903 38.98562 39.11181 39.23760 39.36298
## [17] 39.48796 39.61255 39.73675 39.86056 39.98399 40.10703 40.22971 40.35200
```

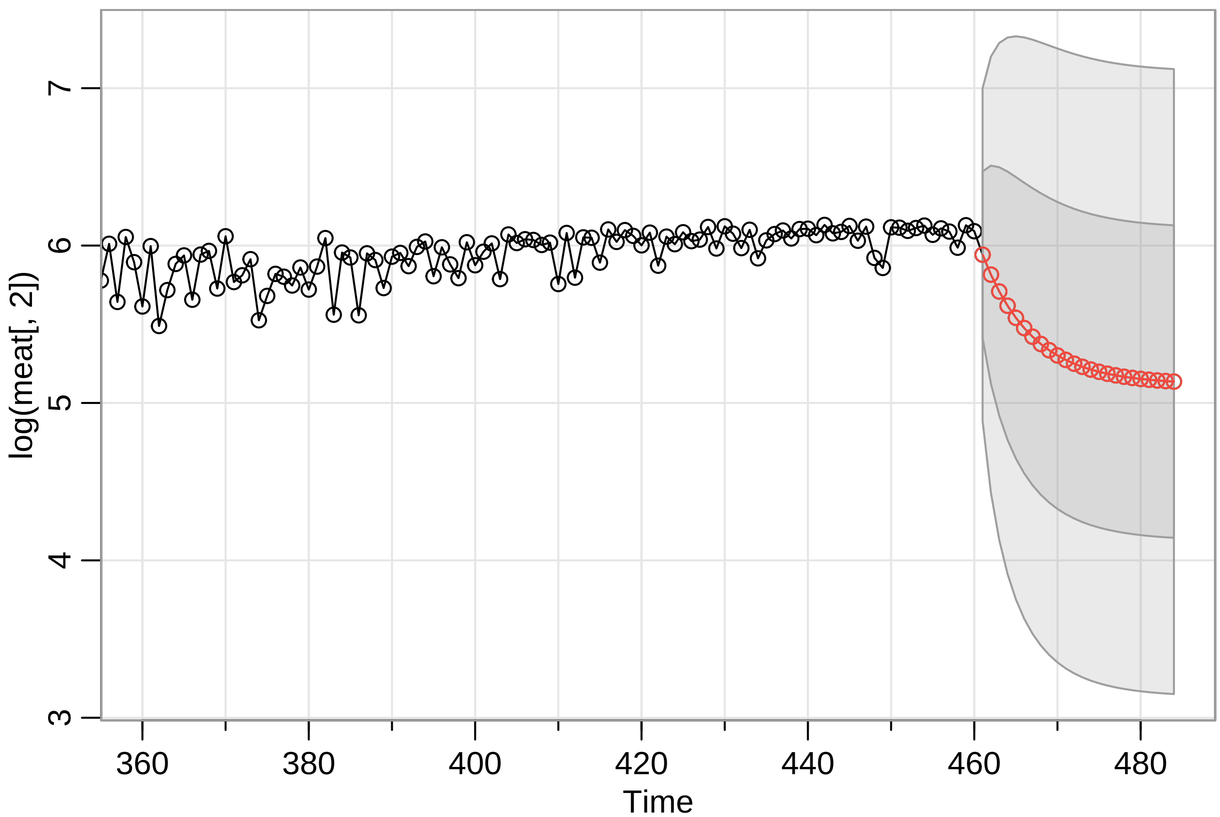
```r
sarima.for(meat[,2], n.ahead=24, p=1, d=2, q=1) # ARIMA(1,2,1)
```




```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 454.5717 448.0381 453.8478 451.6829 454.6701 454.3287 456.1378 456.5576
##  [9] 457.8750 458.6124 459.7245 460.5946 461.6210 462.5464 463.5371 464.4856
## [17] 465.4613 466.4195 467.3890 468.3512 469.3181 470.2820 471.2478 472.2124
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1]  44.03909  46.74806  57.86268  62.00960  68.89252  73.33046  78.63046
##  [8]  82.93455  87.45805  91.51832  95.58137  99.39501 103.14460 106.73845
## [15] 110.25358 113.65851 116.98728 120.23085 123.40561 126.51065 129.55483
## [22] 132.53990 135.47132 138.35152
```

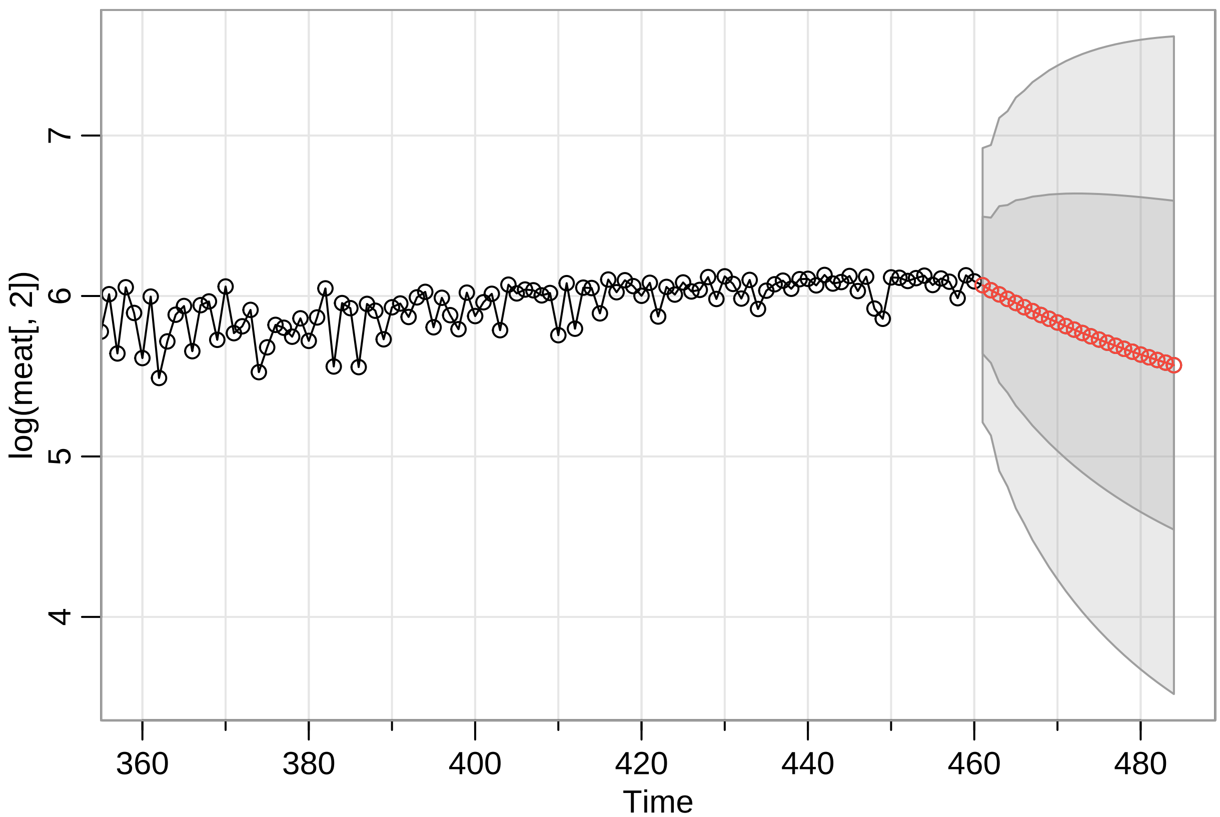
Log Transformation The log transformation removes the trend, resulting in stationarity. Again, the AR models seem to perform best.

```
sarima.for(log(meat[,2]), n.ahead=24, p=1, d=0, q=0) # AR(1)
```



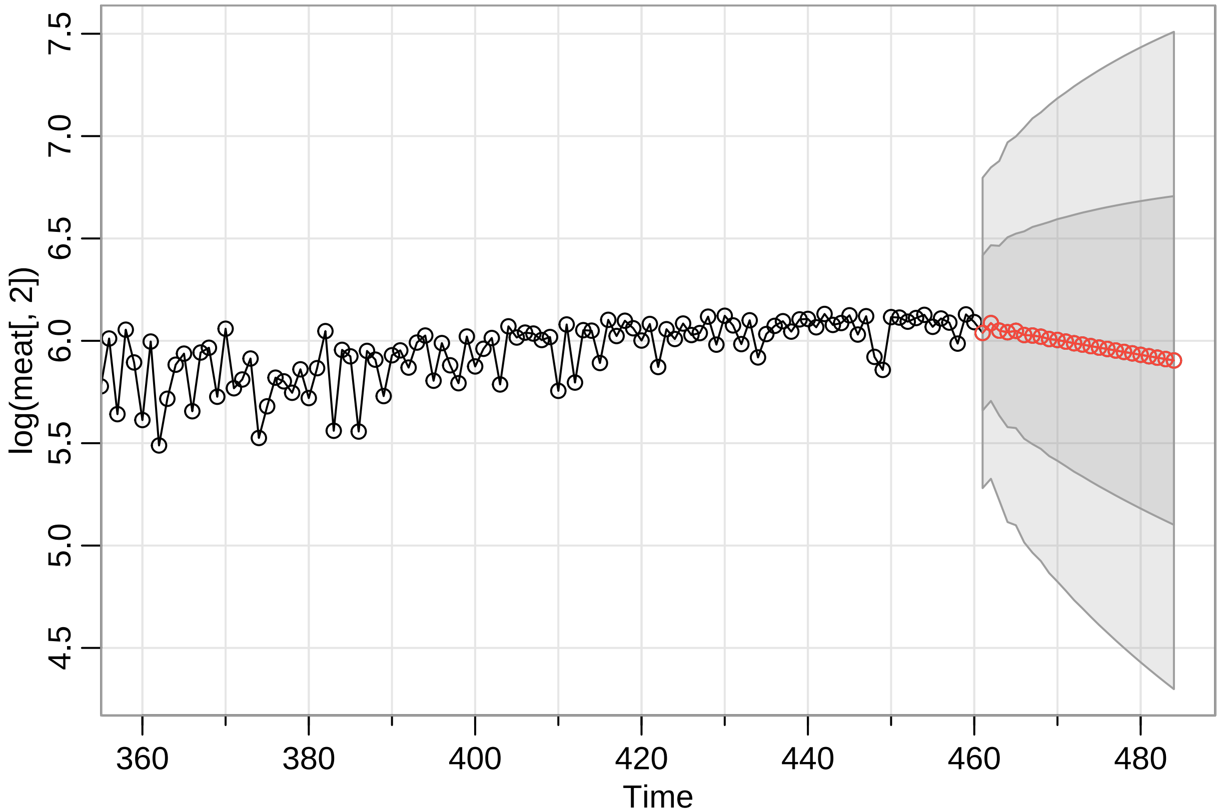

```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 5.941850 5.815343 5.708265 5.617630 5.540915 5.475981 5.421019 5.374498
##  [9] 5.335121 5.301792 5.273581 5.249702 5.229491 5.212383 5.197903 5.185646
## [17] 5.175272 5.166491 5.159058 5.152767 5.147443 5.142935 5.139120 5.135891
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 0.5287434 0.6927222 0.7895332 0.8521565 0.8943301 0.9233612 0.9436113
##  [8] 0.9578560 0.9679325 0.9750878 0.9801819 0.9838154 0.9864103 0.9882652
## [15] 0.9895919 0.9905414 0.9912211 0.9917077 0.9920563 0.9923059 0.9924847
## [22] 0.9926127 0.9927045 0.9927702
```

```
sarima.for(log(meat[,2]), n.ahead=24, p=2, d=0, q=0) # AR(2)
```



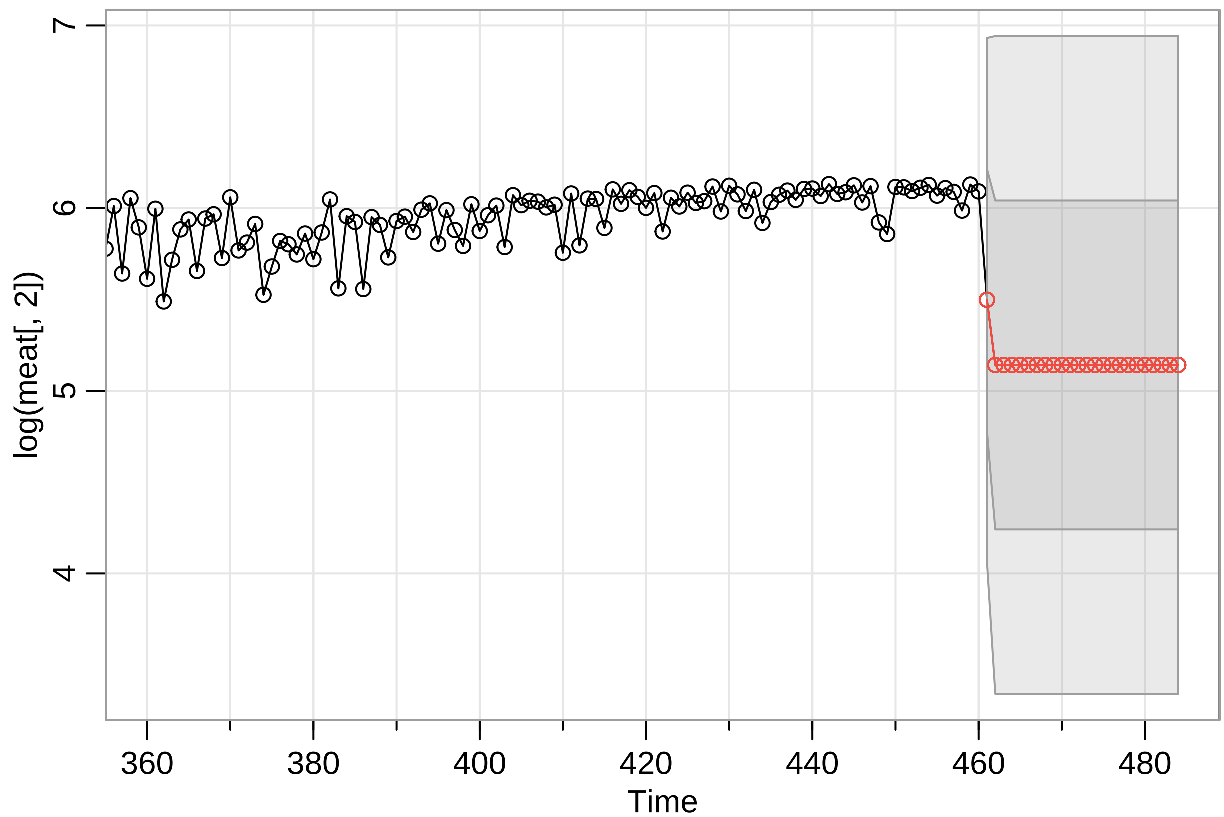

```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 6.067145 6.035712 6.010030 5.981917 5.956467 5.930460 5.905883 5.881464
##  [9] 5.857973 5.834901 5.812541 5.790684 5.769437 5.748710 5.728534 5.708868
## [17] 5.689716 5.671054 5.652875 5.635164 5.617911 5.601101 5.584726 5.568773
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 0.4274212 0.4525659 0.5499819 0.5850492 0.6402521 0.6745069 0.7134581
##  [8] 0.7438299 0.7742957 0.8005116 0.8255822 0.8481200 0.8693095 0.8887502
## [15] 0.9069368 0.9237939 0.9395564 0.9542496 0.9680041 0.9808704 0.9929332
## [22] 1.0042443 1.0148651 1.0248420
```

```
sarima.for(log(meat[,2]), n.ahead=24, p=3, d=0, q=0) # AR(3)
```



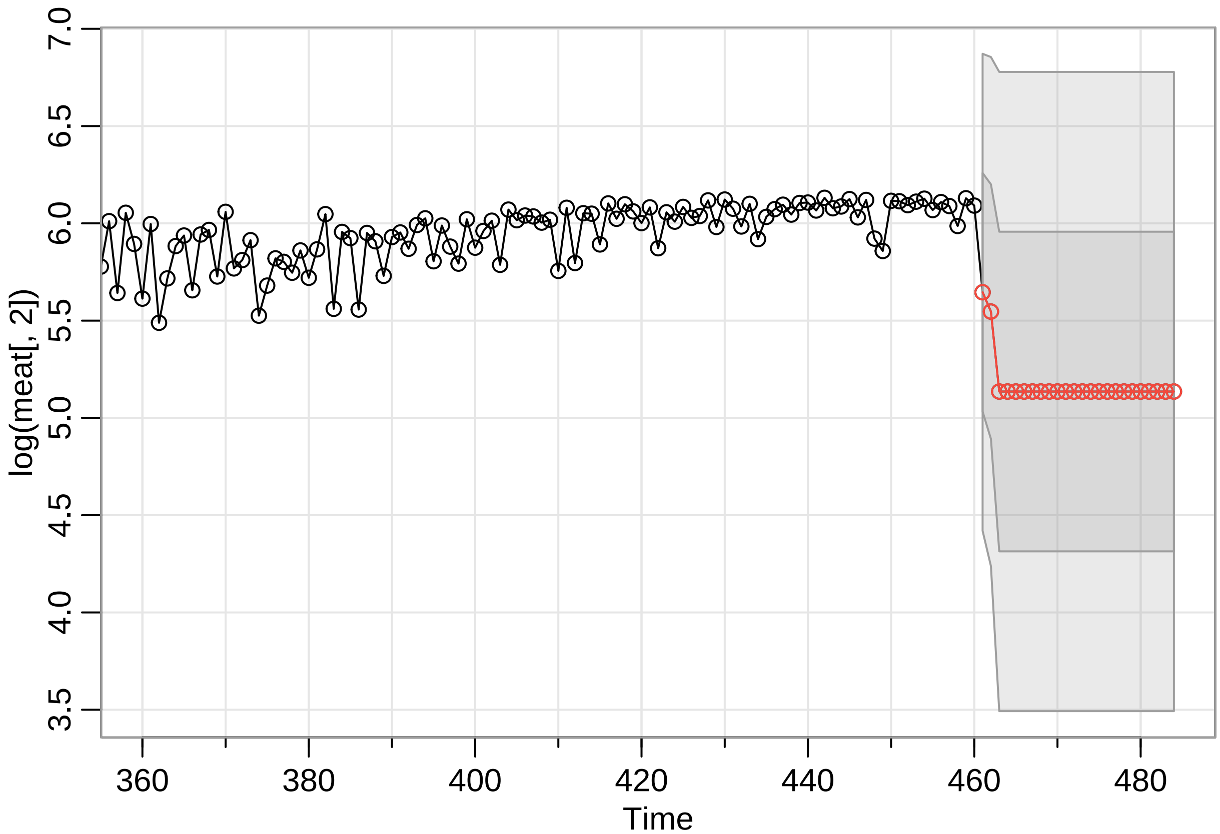

```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 6.038501 6.086649 6.049768 6.042045 6.048781 6.028293 6.025826 6.020098
##  [9] 6.008739 6.004194 5.996218 5.988176 5.981936 5.974163 5.967007 5.960110
## [17] 5.952761 5.945783 5.938765 5.931687 5.924765 5.917810 5.910890 5.904034
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 0.3790059 0.3802168 0.4143180 0.4637167 0.4747307 0.5064260 0.5304078
##  [8] 0.5477970 0.5715814 0.5902857 0.6084137 0.6274247 0.6442280 0.6610359
## [15] 0.6774131 0.6928556 0.7081052 0.7228152 0.7370295 0.7509346 0.7644034
## [22] 0.7775173 0.7903108 0.8027599
```

```r
sarima.for(log(meat[,2]), n.ahead=24, p=0, d=0, q=1) # MA(1)
```



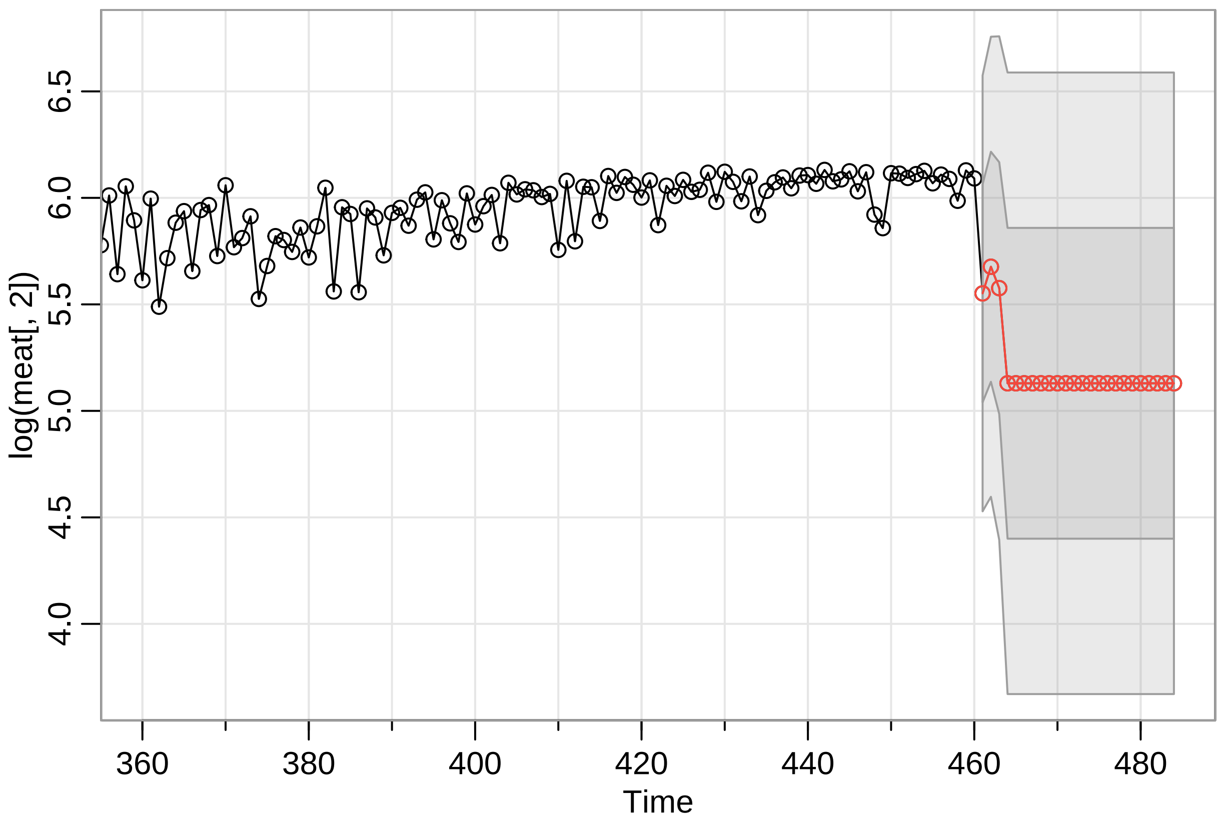

```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 5.498636 5.141346 5.141346 5.141346 5.141346 5.141346 5.141346 5.141346
##  [9] 5.141346 5.141346 5.141346 5.141346 5.141346 5.141346 5.141346 5.141346
## [17] 5.141346 5.141346 5.141346 5.141346 5.141346 5.141346 5.141346 5.141346
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 0.7163762 0.9003035 0.9003035 0.9003035 0.9003035 0.9003035 0.9003035
##  [8] 0.9003035 0.9003035 0.9003035 0.9003035 0.9003035 0.9003035 0.9003035
## [15] 0.9003035 0.9003035 0.9003035 0.9003035 0.9003035 0.9003035 0.9003035
## [22] 0.9003035 0.9003035 0.9003035
```

```
sarima.for(log(meat[,2]), n.ahead=24, p=0, d=0, q=2) # MA(2)
```




```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 5.645782 5.546558 5.135576 5.135576 5.135576 5.135576 5.135576 5.135576
##  [9] 5.135576 5.135576 5.135576 5.135576 5.135576 5.135576 5.135576 5.135576
## [17] 5.135576 5.135576 5.135576 5.135576 5.135576 5.135576 5.135576 5.135576
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 0.6129696 0.6542054 0.8214904 0.8214904 0.8214904 0.8214904 0.8214904
##  [8] 0.8214904 0.8214904 0.8214904 0.8214904 0.8214904 0.8214904 0.8214904
## [15] 0.8214904 0.8214904 0.8214904 0.8214904 0.8214904 0.8214904 0.8214904
## [22] 0.8214904 0.8214904 0.8214904
```

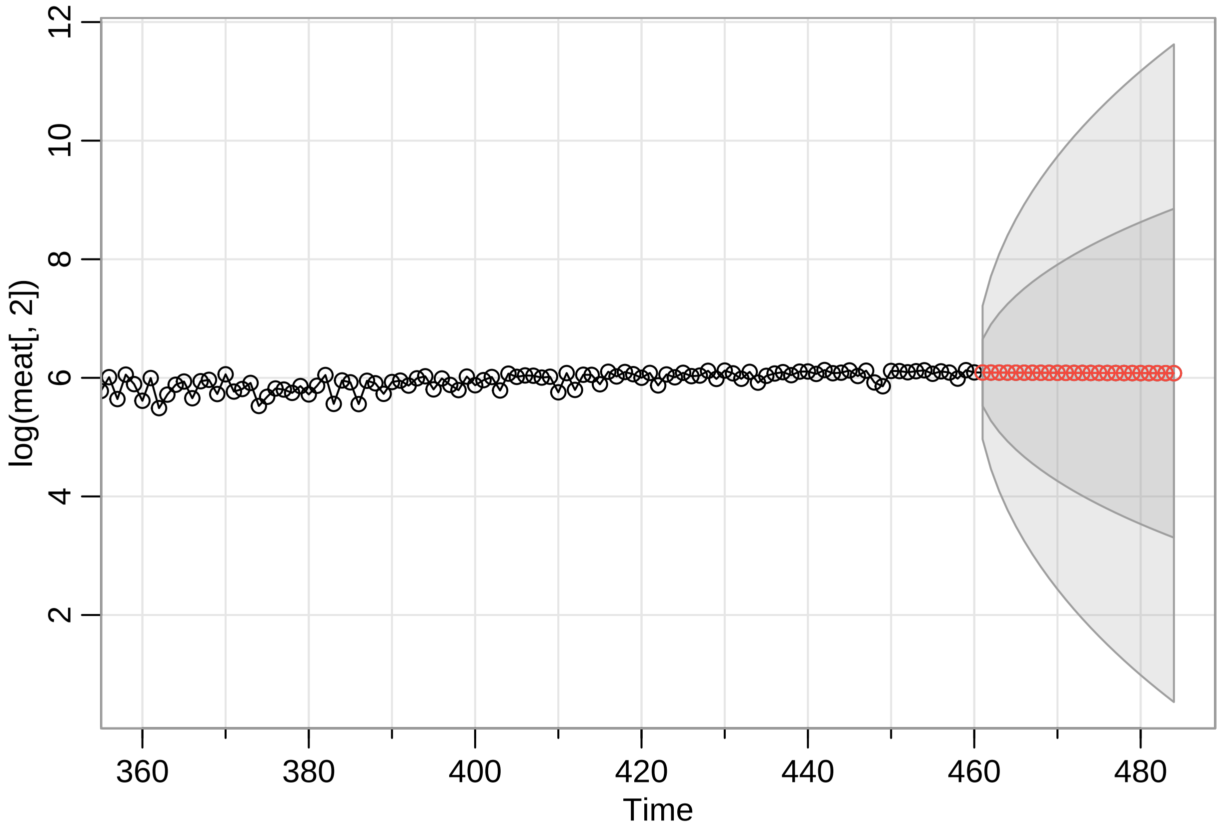
```r
sarima.for(log(meat[,2]), n.ahead=24, p=0, d=0, q=3) # MA(3)
```




```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 5.551603 5.676610 5.576208 5.129480 5.129480 5.129480 5.129480 5.129480
##  [9] 5.129480 5.129480 5.129480 5.129480 5.129480 5.129480 5.129480 5.129480
## [17] 5.129480 5.129480 5.129480 5.129480 5.129480 5.129480 5.129480 5.129480
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 0.5115948 0.5400251 0.5911559 0.7295526 0.7295526 0.7295526 0.7295526
##  [8] 0.7295526 0.7295526 0.7295526 0.7295526 0.7295526 0.7295526 0.7295526
## [15] 0.7295526 0.7295526 0.7295526 0.7295526 0.7295526 0.7295526 0.7295526
## [22] 0.7295526 0.7295526 0.7295526
```

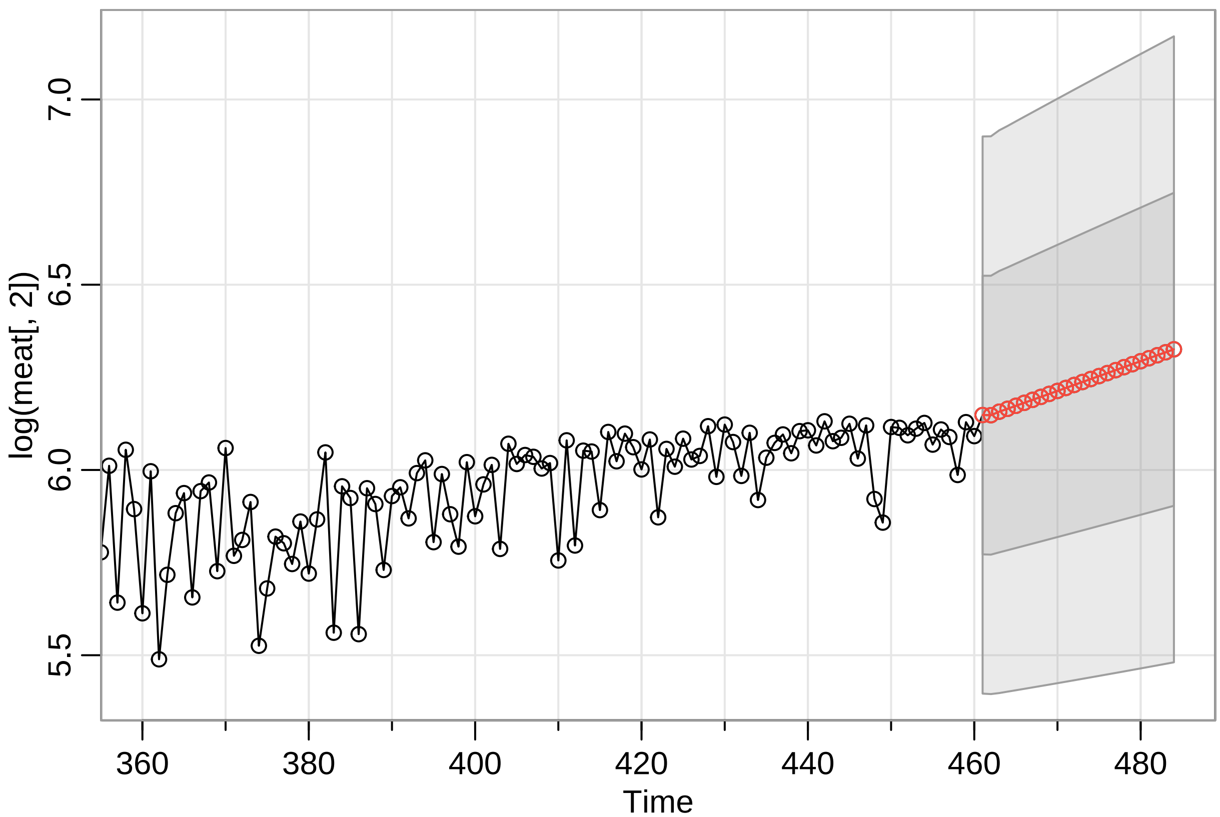
```
sarima.for(log(meat[,2]), n.ahead=24, p=1, d=0, q=1) # ARMA(1,1)
```

```
## Warning in stats::arima(xdata, order = c(p, d, q), seasonal = list(order =
## c(P, : possible convergence problem: optim gave code = 1
```



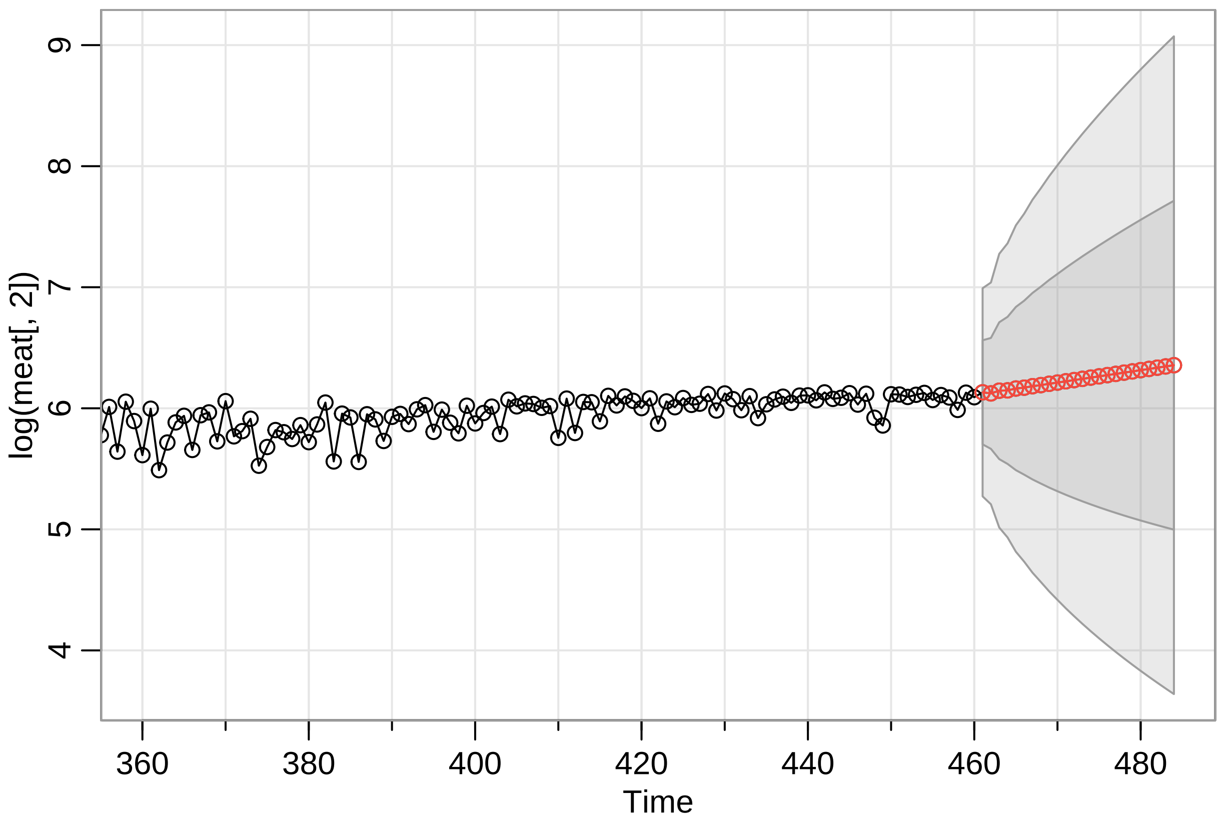

```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 6.088896 6.088417 6.087938 6.087461 6.086986 6.086512 6.086040 6.085569
##  [9] 6.085100 6.084632 6.084165 6.083700 6.083237 6.082775 6.082314 6.081855
## [17] 6.081397 6.080941 6.080486 6.080033 6.079581 6.079131 6.078682 6.078234
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 0.5629673 0.8126350 1.0008272 1.1579224 1.2952902 1.4186775 1.5314715
##  [8] 1.6358690 1.7333916 1.8251453 1.9119641 1.9944960 2.0732559 2.1486618
## [15] 2.2210580 2.2907328 2.3579301 2.4228586 2.4856985 2.5466070 2.6057217
## [22] 2.6631645 2.7190436 2.7734555
```

```
sarima.for(log(meat[,2]), n.ahead=24, p=1, d=1, q=1) # ARIMA(1,1,1)
```




```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 6.148223 6.147818 6.157299 6.165075 6.173146 6.181165 6.189194 6.197221
##  [9] 6.205248 6.213275 6.221302 6.229329 6.237356 6.245383 6.253410 6.261437
## [17] 6.269464 6.277492 6.285519 6.293546 6.301573 6.309600 6.317627 6.325654
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 0.3759662 0.3763701 0.3797326 0.3817014 0.3838691 0.3859880 0.3881016
##  [8] 0.3902027 0.3922927 0.3943716 0.3964396 0.3984969 0.4005436 0.4025800
## [15] 0.4046060 0.4066220 0.4086280 0.4106242 0.4126108 0.4145878 0.4165555
## [22] 0.4185139 0.4204632 0.4224035
```

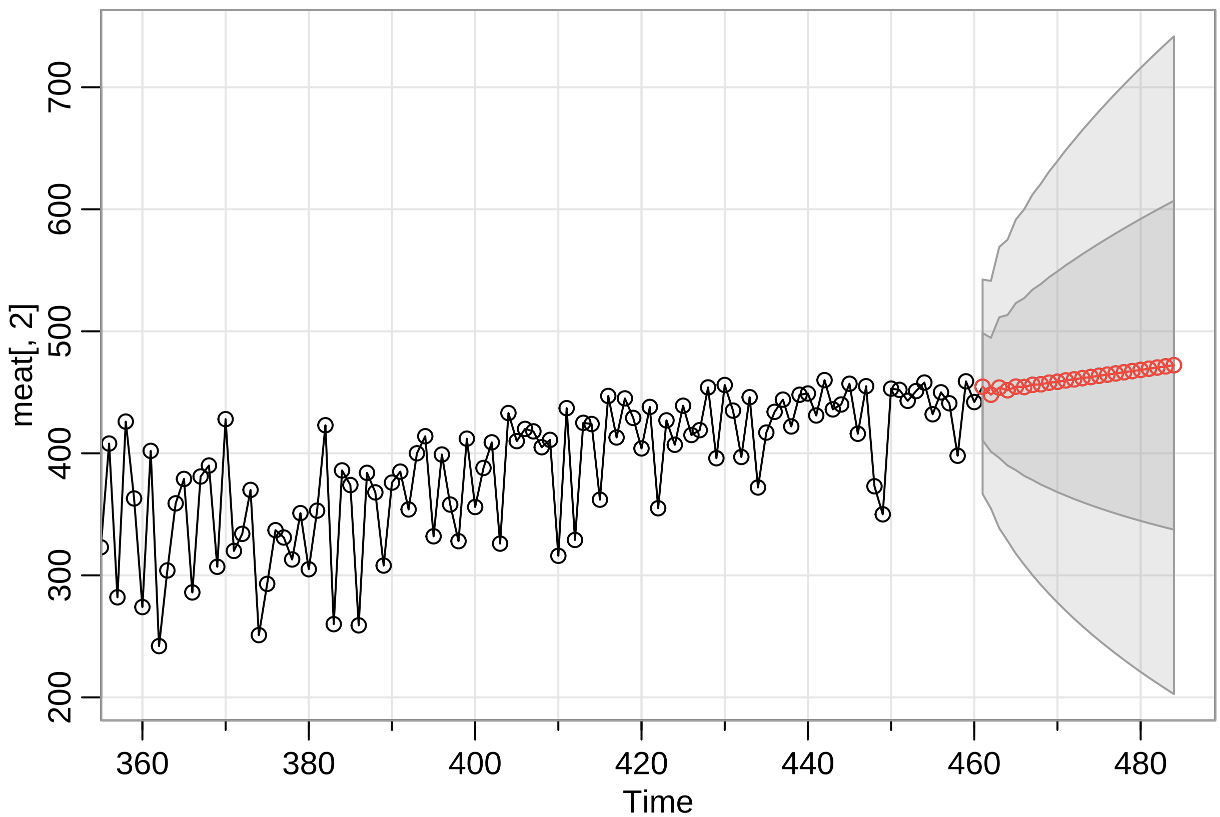
```
sarima.for(log(meat[,2]), n.ahead=24, p=1, d=2, q=1) # ARIMA(1,2,1)
```




```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 6.132016 6.122941 6.145456 6.147924 6.163114 6.170231 6.182471 6.191460
##  [9] 6.202512 6.212255 6.222829 6.232875 6.243256 6.253425 6.263728 6.273946
## [17] 6.284218 6.294456 6.304716 6.314961 6.325216 6.335465 6.345717 6.355968
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 0.4298091 0.4579267 0.5652740 0.6074114 0.6741579 0.7186249 0.7703294
##  [8] 0.8130992 0.8574165 0.8975779 0.9374792 0.9751077 1.0119736 1.0473867
## [15] 1.0819647 1.1154925 1.1482430 1.1801690 1.2114050 1.2419601 1.2719101
## [22] 1.3012801 1.3301190 1.3584540
```

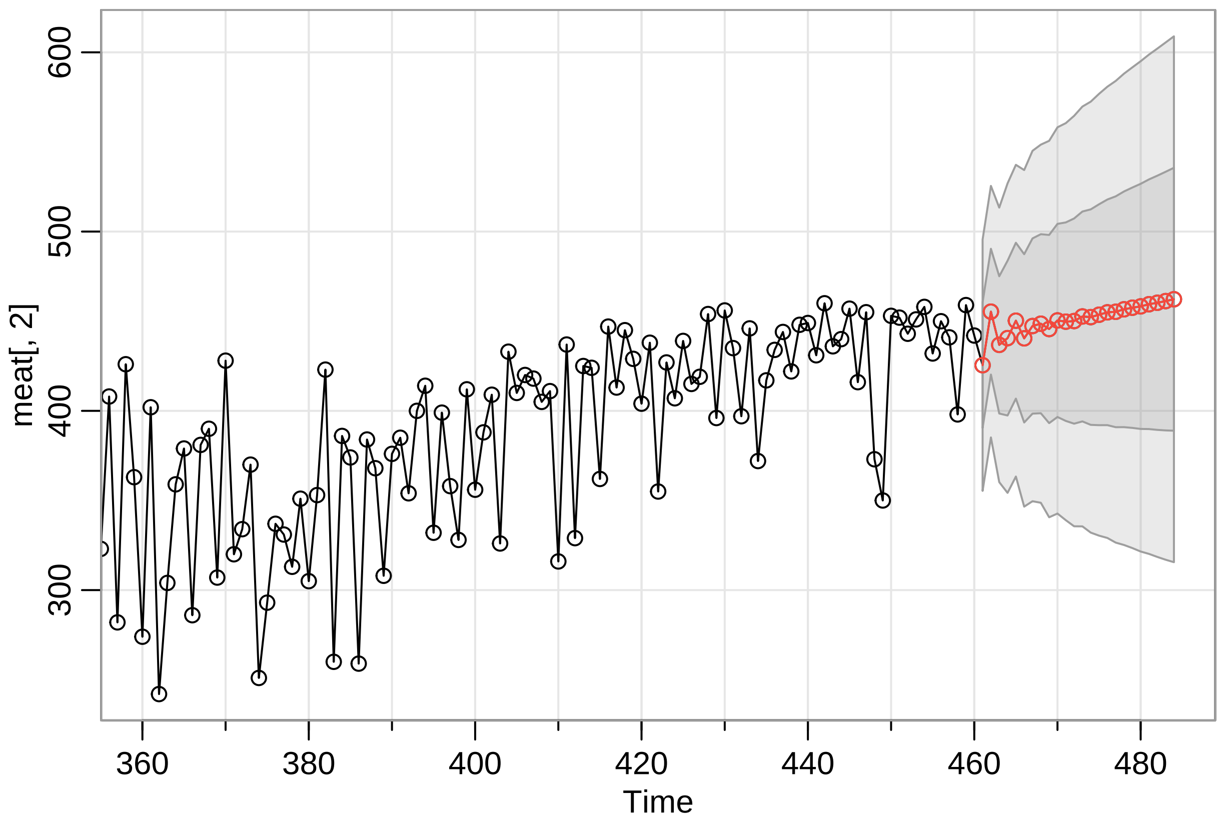
First Order Differencing First order differencing doesn't remove the trend as well as the log transformation.

```
sarima.for(meat[,2], n.ahead=24, p=1, d=1, q=0) # AR(1)
```



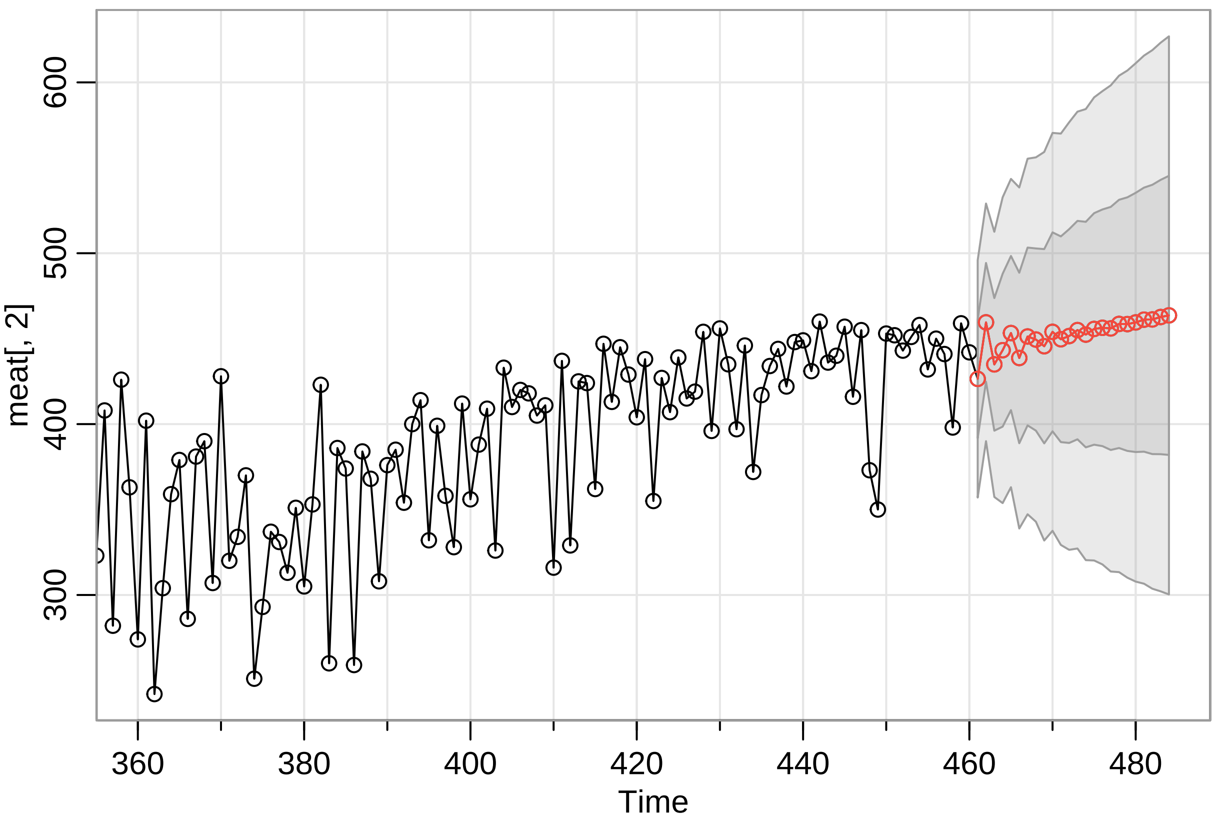

```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 454.5830 448.0348 453.8587 451.6816 454.6787 454.3297 456.1445 456.5600
##  [9] 457.8805 458.6158 459.7295 460.5984 461.6257 462.5505 463.5416 464.4899
## [17] 465.4658 466.4238 467.3935 468.3556 469.3226 470.2864 471.2523 472.2168
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1]  43.94308  46.60505  57.63193  61.69476  68.47314  72.80334  77.98342
##  [8]  82.16225  86.55179  90.47227  94.38849  98.04953 101.64039 105.07002
## [15] 108.41538 111.64536 114.79404 117.85261 120.83758 123.74817 126.59338
## [22] 129.37508 132.09881 134.76713
```

```
sarima.for(meat[,2], n.ahead=24, p=2, d=1, q=0) # AR(2)
```



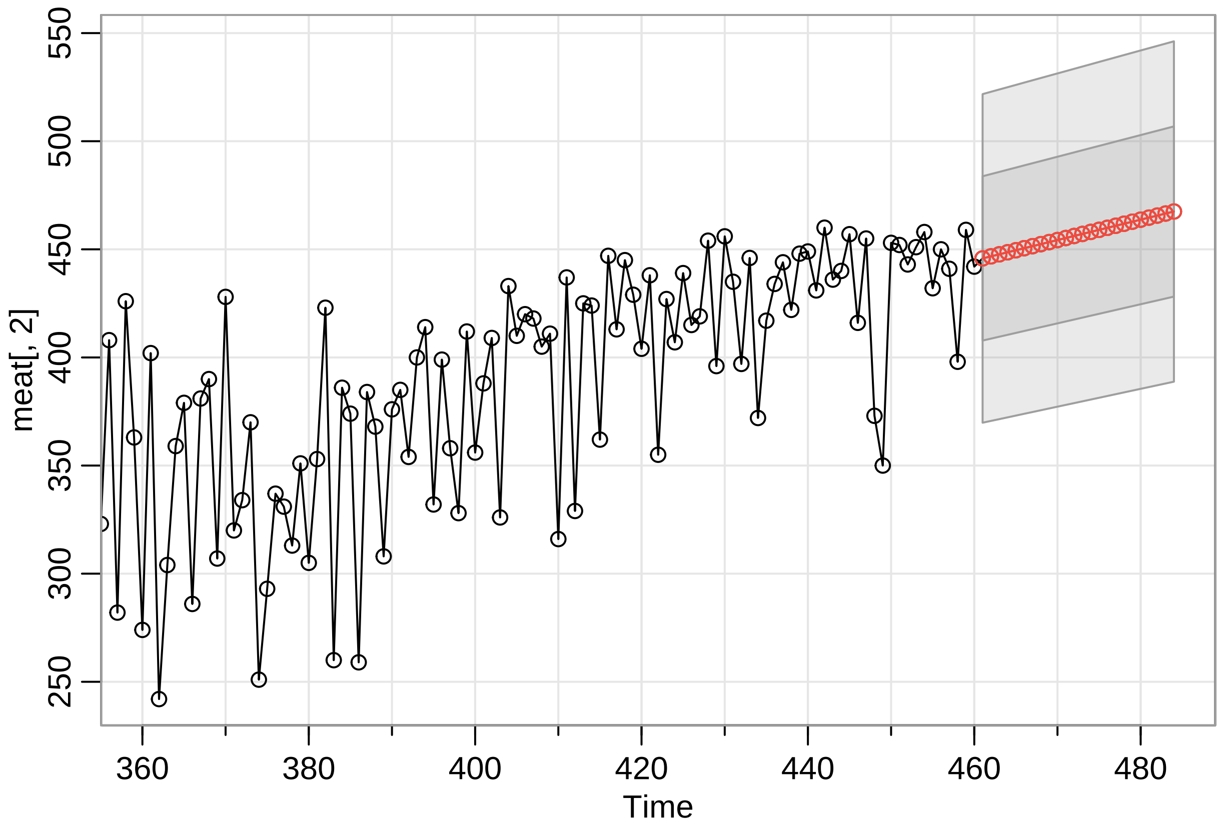

```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 425.4632 455.3336 436.7939 440.5668 450.2984 440.4372 447.3149 448.6094
##  [9] 445.6325 450.4419 449.7433 450.0768 452.6509 452.2807 453.6173 454.9535
## [17] 455.2642 456.6388 457.5264 458.2795 459.4648 460.2827 461.2219 462.2562
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 35.05276 35.07695 38.28164 43.16040 43.46798 46.93981 48.86329 49.94195
##  [9] 52.49919 53.86662 55.37000 57.24012 58.54637 60.09937 61.61486 62.93176
## [17] 64.38576 65.73354 67.03382 68.37277 69.63328 70.88817 72.13459 73.33644
```

```
sarima.for(meat[,2], n.ahead=24, p=3, d=1, q=0) # AR(3)
```



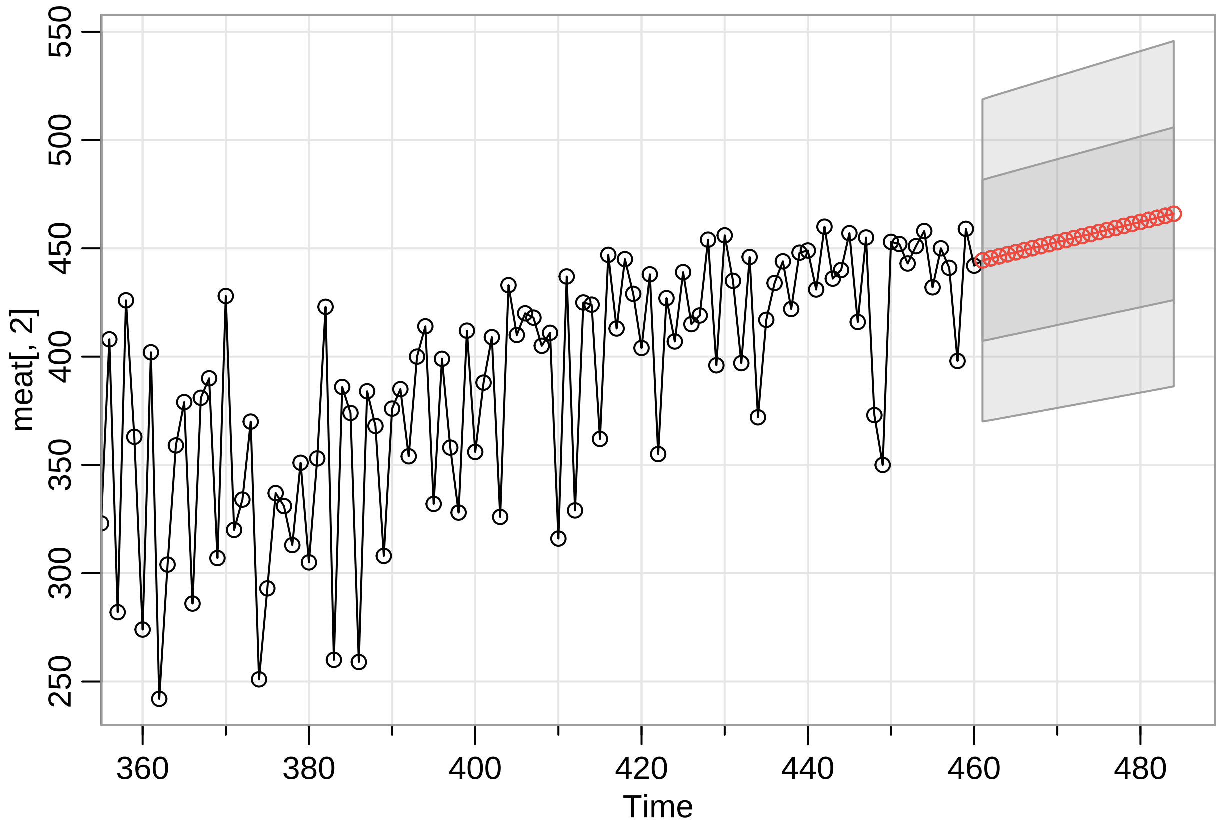

```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 426.4668 459.5554 434.9491 443.2501 453.2904 438.6872 451.2829 449.4879
##  [9] 445.5671 453.9940 449.6550 451.5329 455.0278 452.3834 455.7010 456.3715
## [17] 455.9914 458.6426 458.5235 459.5116 461.1293 461.2600 462.6711 463.6296
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 34.72378 34.75966 38.80255 44.73386 45.10309 49.89410 52.02549 53.30871
##  [9] 56.84930 58.21380 60.18100 62.55767 63.91577 65.99471 67.75233 69.24759
## [17] 71.12762 72.63473 74.19755 75.84117 77.26578 78.79808 80.27086 81.66217
```

```
sarima.for(meat[,2], n.ahead=24, p=0, d=1, q=1) # MA(1)
```



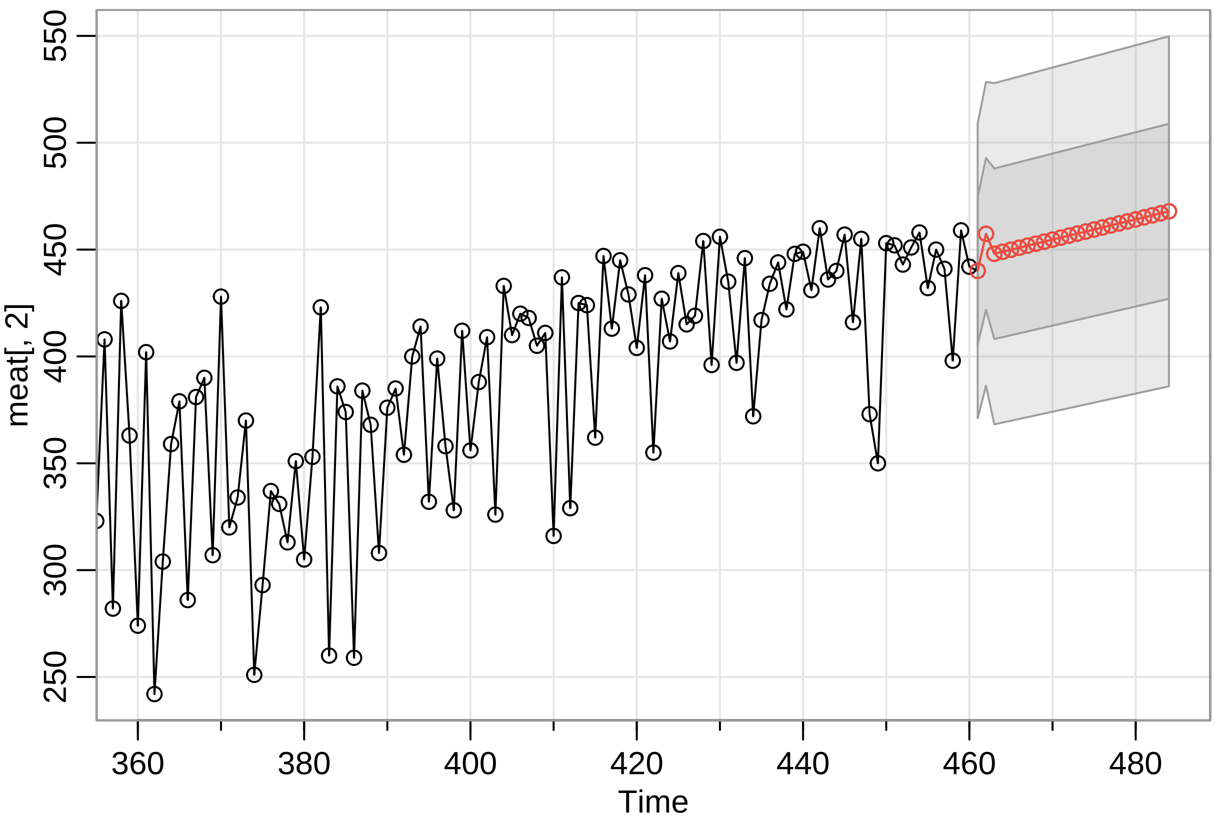

```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 445.8002 446.7434 447.6865 448.6297 449.5728 450.5160 451.4591 452.4023
##  [9] 453.3454 454.2886 455.2317 456.1749 457.1180 458.0612 459.0044 459.9475
## [17] 460.8907 461.8338 462.7770 463.7201 464.6633 465.6064 466.5496 467.4927
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 37.99206 38.05231 38.11246 38.17252 38.23248 38.29235 38.35212 38.41180
##  [9] 38.47139 38.53089 38.59030 38.64961 38.70883 38.76797 38.82701 38.88596
## [17] 38.94483 39.00360 39.06229 39.12089 39.17940 39.23782 39.29616 39.35441
```

```
sarima.for(meat[,2], n.ahead=24, p=0, d=1, q=2) # MA(2)
```



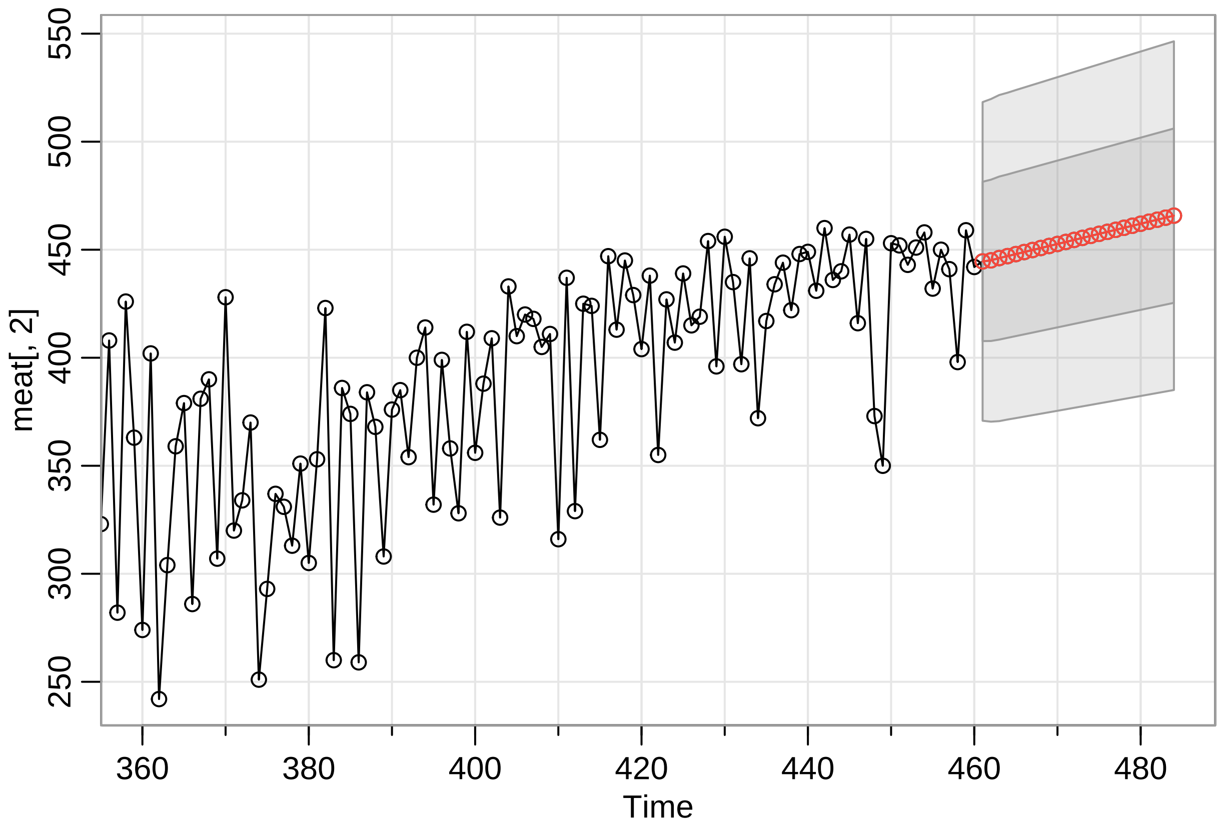

```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 444.4255 445.3740 446.3105 447.2471 448.1836 449.1202 450.0567 450.9933
##  [9] 451.9298 452.8664 453.8029 454.7395 455.6760 456.6126 457.5491 458.4857
## [17] 459.4222 460.3588 461.2953 462.2318 463.1684 464.1049 465.0415 465.9780
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 37.16786 37.34681 37.46470 37.58221 37.69936 37.81614 37.93256 38.04863
##  [9] 38.16435 38.27971 38.39473 38.50941 38.62374 38.73774 38.85140 38.96473
## [17] 39.07773 39.19041 39.30277 39.41480 39.52651 39.63792 39.74900 39.85978
```

```
sarima.for(meat[,2], n.ahead=24, p=0, d=1, q=3) # MA(3)
```




```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 440.0643 457.3672 448.0460 448.9919 449.9378 450.8837 451.8296 452.7755
##  [9] 453.7214 454.6673 455.6132 456.5591 457.5050 458.4509 459.3968 460.3427
## [17] 461.2886 462.2345 463.1804 464.1263 465.0722 466.0181 466.9640 467.9099
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 34.49266 35.53703 39.91605 39.96579 40.01547 40.06508 40.11463 40.16412
##  [9] 40.21356 40.26292 40.31223 40.36148 40.41067 40.45980 40.50887 40.55788
## [17] 40.60683 40.65572 40.70456 40.75333 40.80205 40.85071 40.89931 40.94785
```

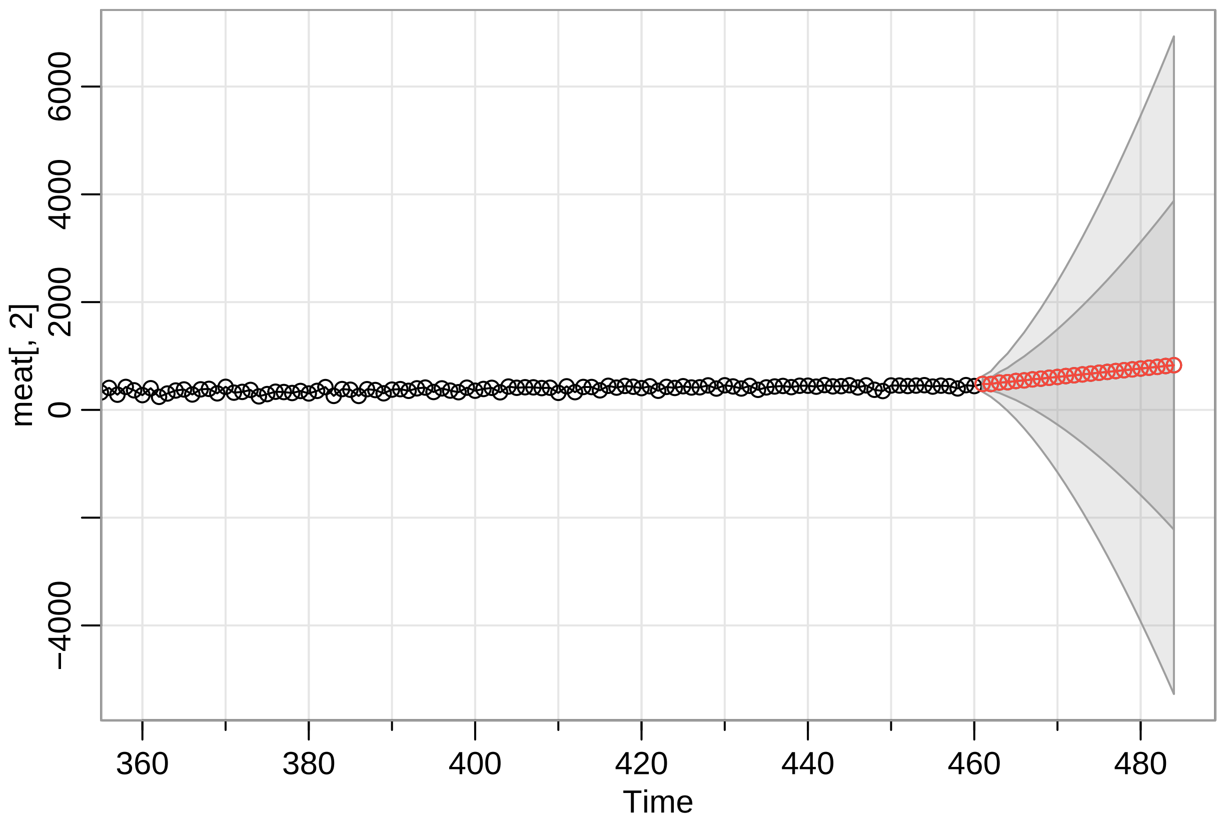
```
sarima.for(meat[,2], n.ahead=24, p=1, d=1, q=1) # ARIMA(1,1,1)
```




```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 444.5789 445.0792 446.1308 447.0362 447.9803 448.9142 449.8507 450.7866
##  [9] 451.7226 452.6586 453.5947 454.5307 455.4667 456.4027 457.3387 458.2747
## [17] 459.2107 460.1467 461.0827 462.0187 462.9547 463.8908 464.8268 465.7628
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 36.87696 37.33152 37.73626 37.81964 37.96388 38.08983 38.21994 38.34840
##  [9] 38.47674 38.60458 38.73201 38.85903 38.98562 39.11181 39.23760 39.36298
## [17] 39.48796 39.61255 39.73675 39.86056 39.98399 40.10703 40.22971 40.35200
```

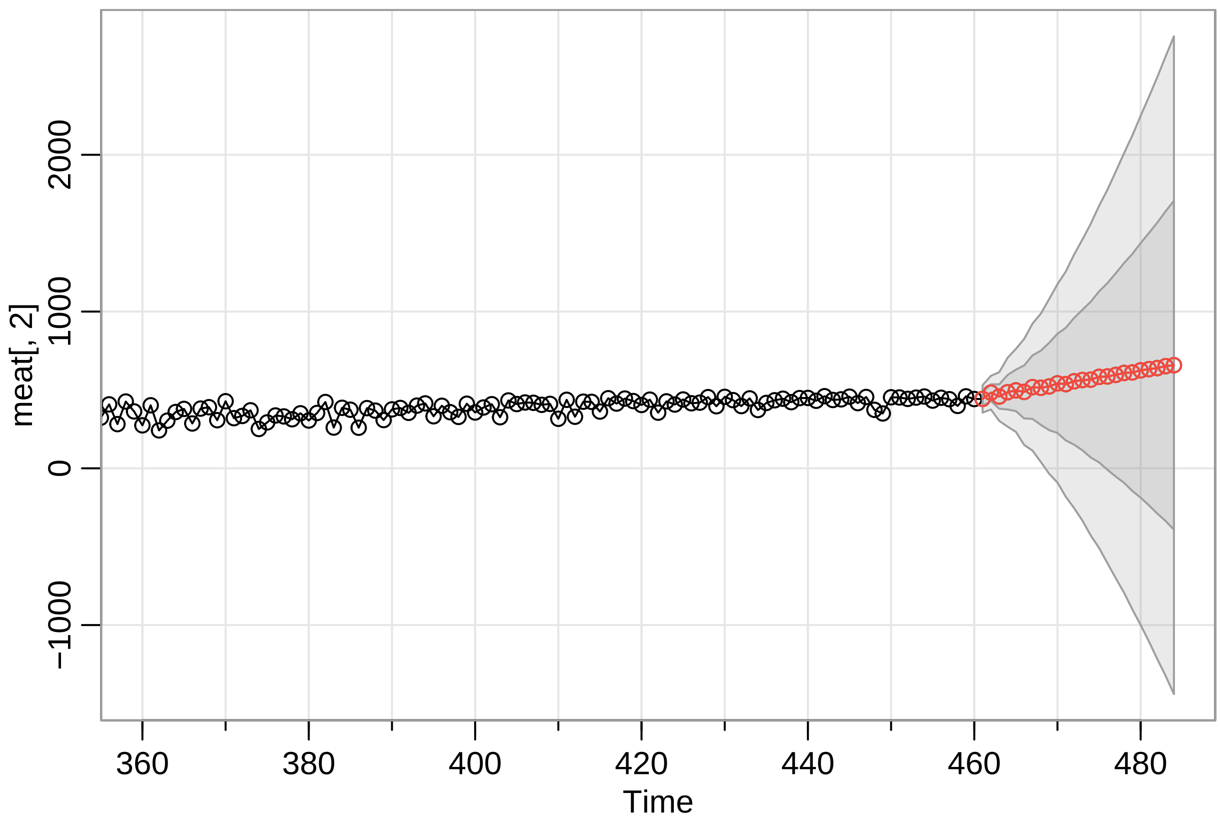
Second Order Differencing Second order differencing creates too much stationarity for AR models, and not enough for MA models.

```
sarima.for(meat[,2], n.ahead=24, p=1, d=2, q=0) # AR(1)
```



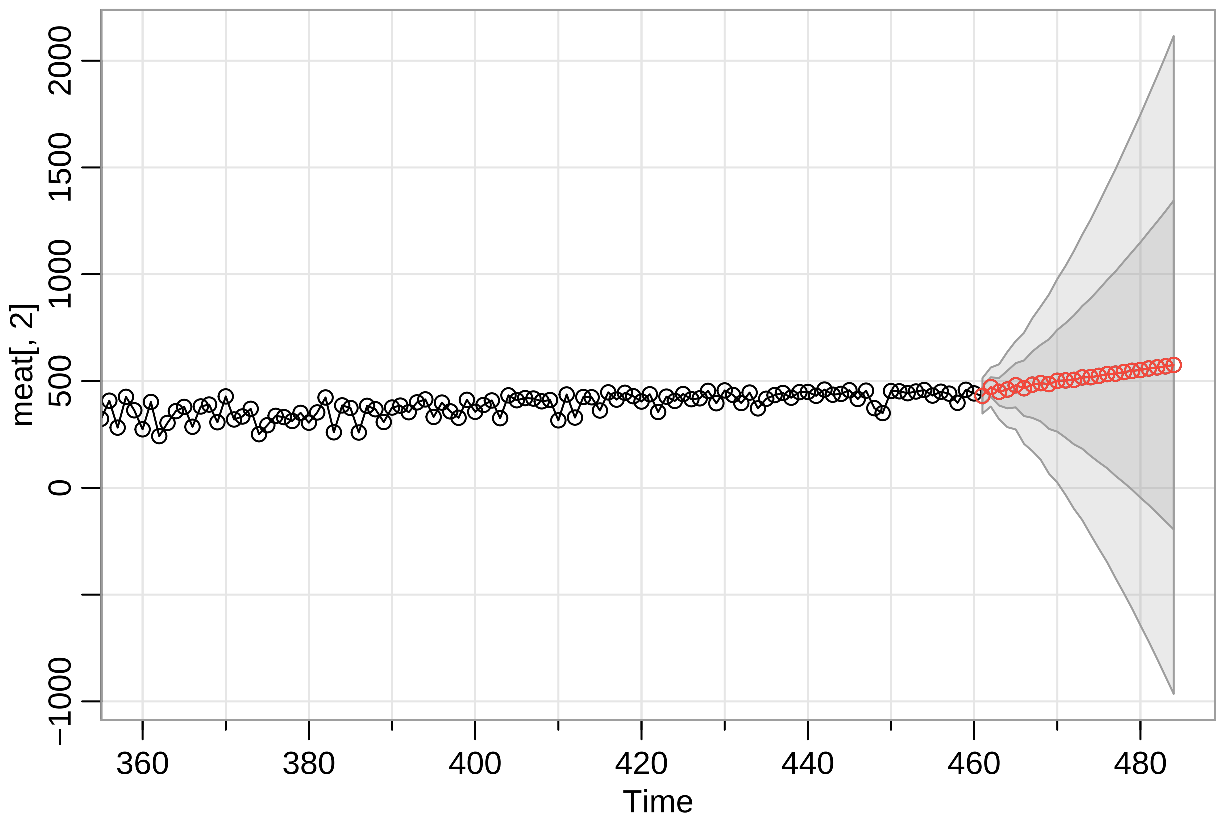

```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 480.9273 479.7538 507.3332 514.2963 536.0416 547.1879 565.9338 579.2307
##  [9] 596.4346 610.8371 627.2483 642.2192 658.2228 673.4859 689.2800 704.6934
## [17] 720.3797 735.8704 751.5013 767.0317 782.6341 798.1849 813.7727 829.3340
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1]   72.97454  118.70532  192.69673  265.78764  353.86340  445.31294
##  [7]  546.60931  652.59813  766.02937  884.46016 1009.05286 1138.58241
## [13] 1273.51188 1413.18781 1557.76239 1706.86050 1860.49772 2018.43970
## [19] 2180.64454 2346.95227 2517.29946 2691.56758 2869.68773 3051.56606
```

```
sarima.for(meat[,2], n.ahead=24, p=2, d=2, q=0) # AR(2)
```



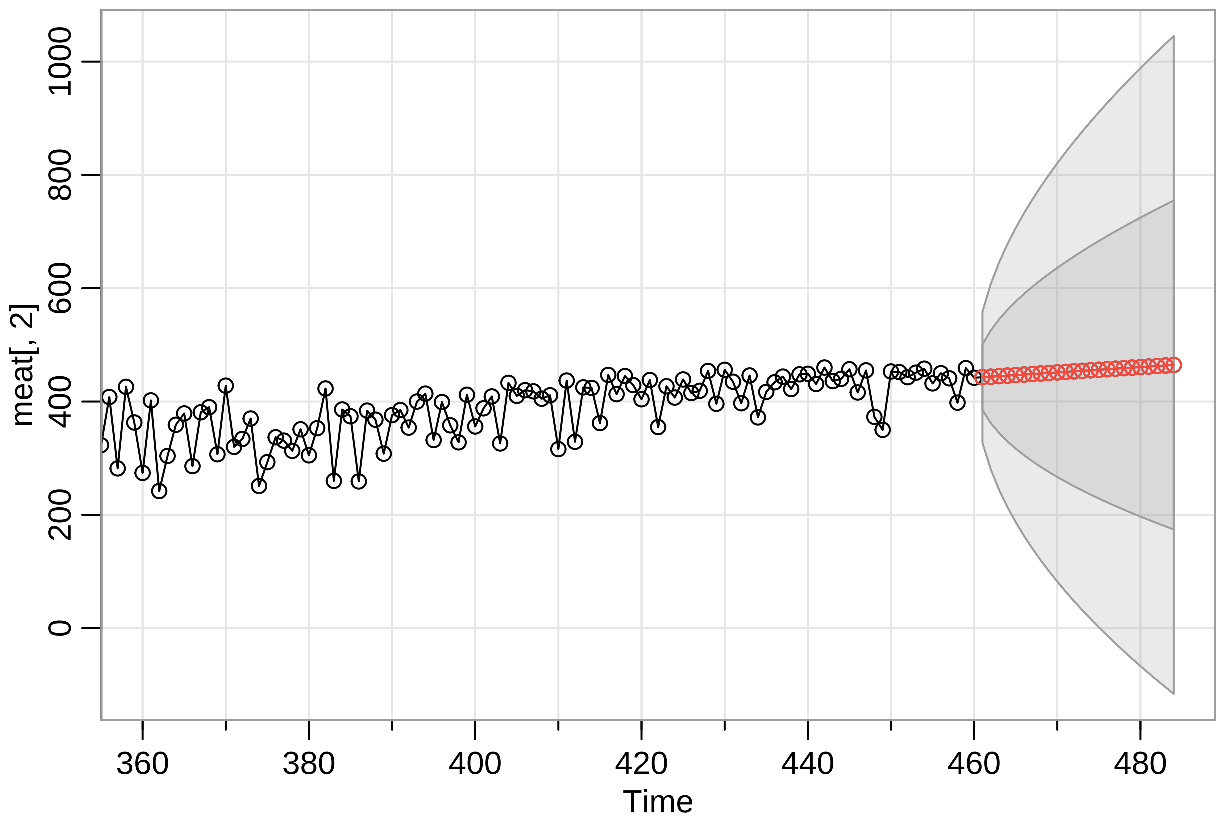

```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 442.6450 482.7676 457.8755 485.3629 497.1511 487.3962 517.9641 513.6984
##  [9] 522.1980 542.0247 537.0506 555.0357 563.1895 565.7064 583.3413 585.9691
## [17] 595.8952 608.3805 611.7399 624.8314 632.6489 639.5048 651.8099 657.8517
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1]   43.74075   53.63922   77.73301  109.83713  132.26547  169.15467
##  [7]  202.50567  236.48643  278.57879  316.38179  359.23641  404.38405
## [13]  448.20287  497.02649  545.46035  595.32700  648.25471  700.70580
## [19]  755.75511  812.10046  868.94384  928.18387  987.94262 1049.05241
```

```
sarima.for(meat[,2], n.ahead=24, p=3, d=2, q=0) # AR(3)
```



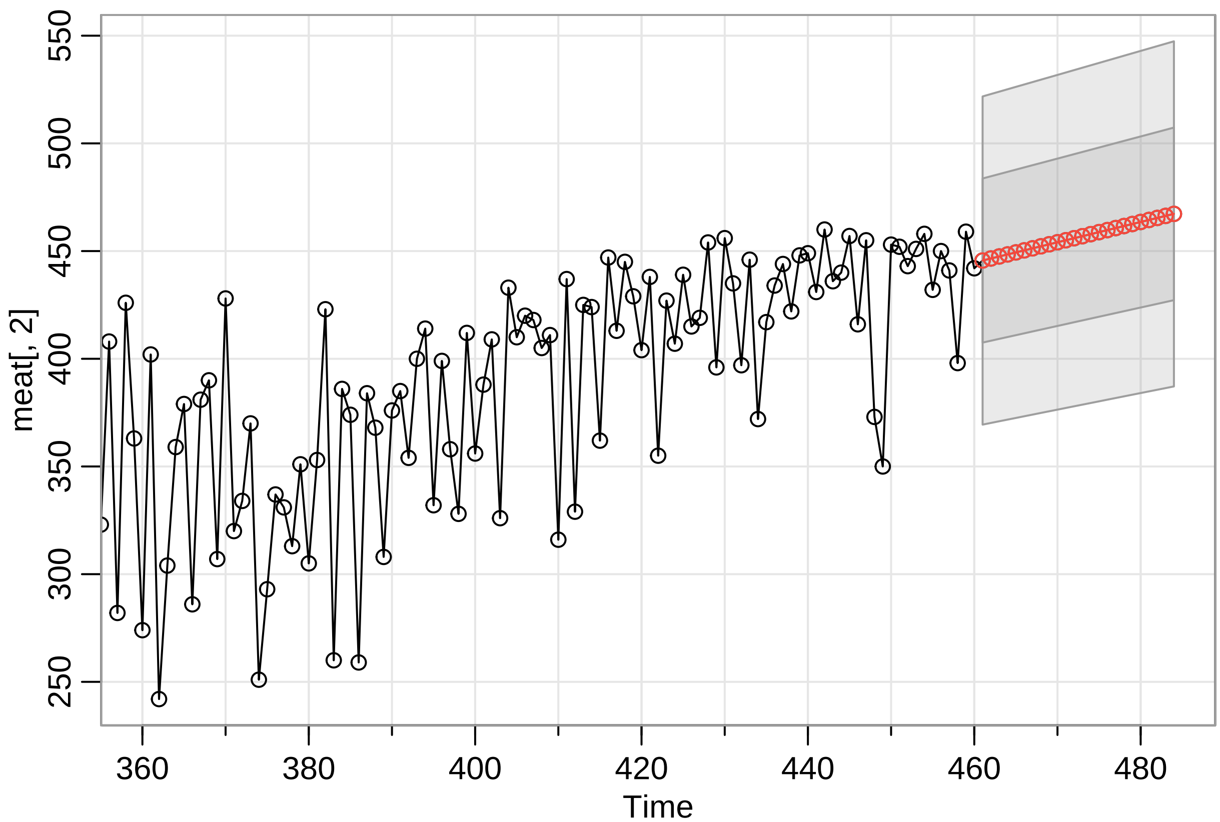

```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 430.7231 471.9202 449.7713 460.3284 480.2668 466.2253 483.0297 490.2397
##  [9] 485.7612 501.1854 503.0012 505.5036 517.0907 518.1099 524.2715 532.3784
## [17] 534.6081 541.9458 547.8731 551.6681 558.8474 563.8177 568.7883 575.3569
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1]  41.51519  45.66419  64.28097  88.19988 103.43365 130.08786 155.38540
##  [8] 179.15462 209.76115 238.23048 268.21723 301.63944 333.88053 368.41095
## [15] 404.50475 440.48297 478.54048 517.36788 556.78936 597.81833 639.40999
## [22] 681.92916 725.64273 769.95215
```

```
sarima.for(meat[,2], n.ahead=24, p=0, d=2, q=1) # MA(1)
```



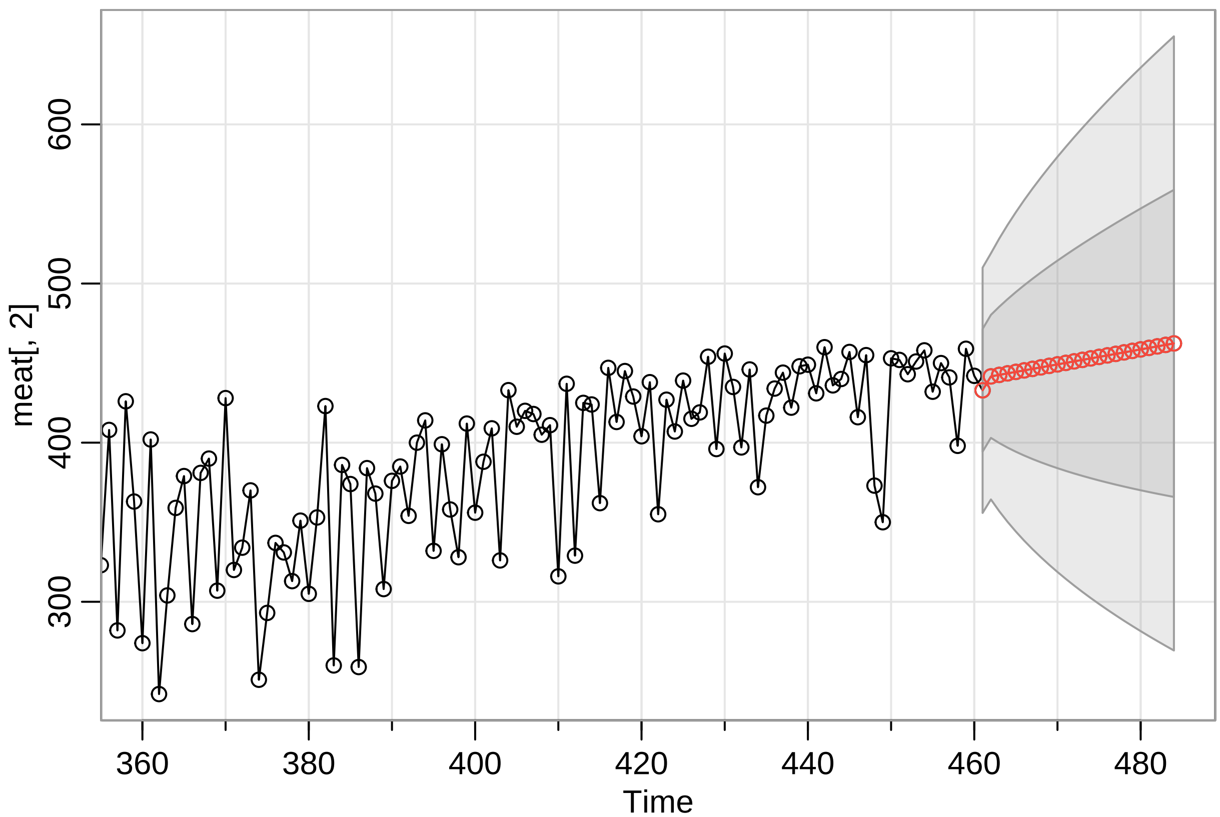

```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 442.9412 443.8824 444.8235 445.7647 446.7059 447.6471 448.5882 449.5294
##  [9] 450.4706 451.4118 452.3529 453.2941 454.2353 455.1765 456.1176 457.0588
## [17] 458.0000 458.9412 459.8824 460.8235 461.7647 462.7059 463.6471 464.5882
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1]  57.82167  81.86102 100.36755 116.01982 129.85411 142.40125 153.97623
##  [8] 164.78404 174.96690 184.62824 193.84606 202.68088 211.18078 219.38468
## [15] 227.32464 235.02741 242.51558 249.80847 256.92269 263.87270 270.67116
## [22] 277.32919 283.85670 290.26248
```

```
sarima.for(meat[,2], n.ahead=24, p=0, d=2, q=2) # MA(2)
```



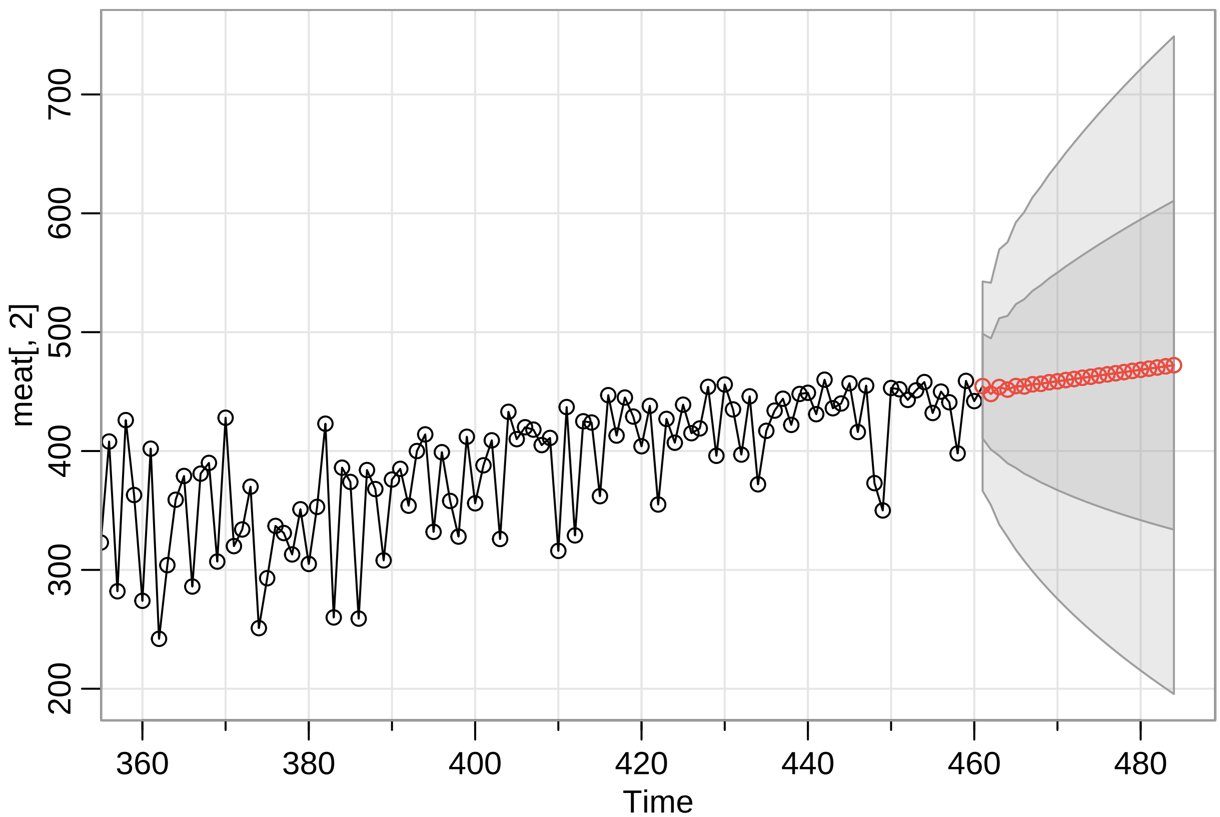

```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 445.6146 446.5562 447.4978 448.4394 449.3811 450.3227 451.2643 452.2060
##  [9] 453.1476 454.0892 455.0308 455.9725 456.9141 457.8557 458.7974 459.7390
## [17] 460.6806 461.6222 462.5639 463.5055 464.4471 465.3888 466.3304 467.2720
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 38.09113 38.17480 38.25865 38.34268 38.42689 38.51128 38.59584 38.68058
##  [9] 38.76549 38.85058 38.93584 39.02127 39.10688 39.19265 39.27859 39.36469
## [17] 39.45096 39.53740 39.62400 39.71076 39.79768 39.88477 39.97201 40.05941
```

```
sarima.for(meat[,2], n.ahead=24, p=0, d=2, q=3) # MA(3)
```




```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 432.9002 441.6485 442.5907 443.5330 444.4752 445.4174 446.3597 447.3019
##  [9] 448.2441 449.1863 450.1286 451.0708 452.0130 452.9553 453.8975 454.8397
## [17] 455.7820 456.7242 457.6664 458.6087 459.5509 460.4931 461.4353 462.3776
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 38.54334 38.65525 42.82702 46.64272 50.18365 53.50439 56.64367 59.63017
##  [9] 62.48578 65.22771 67.86974 70.42310 72.89712 75.29962 77.63724 79.91567
## [17] 82.13983 84.31403 86.44203 88.52716 90.57238 92.58035 94.55343 96.49376
```

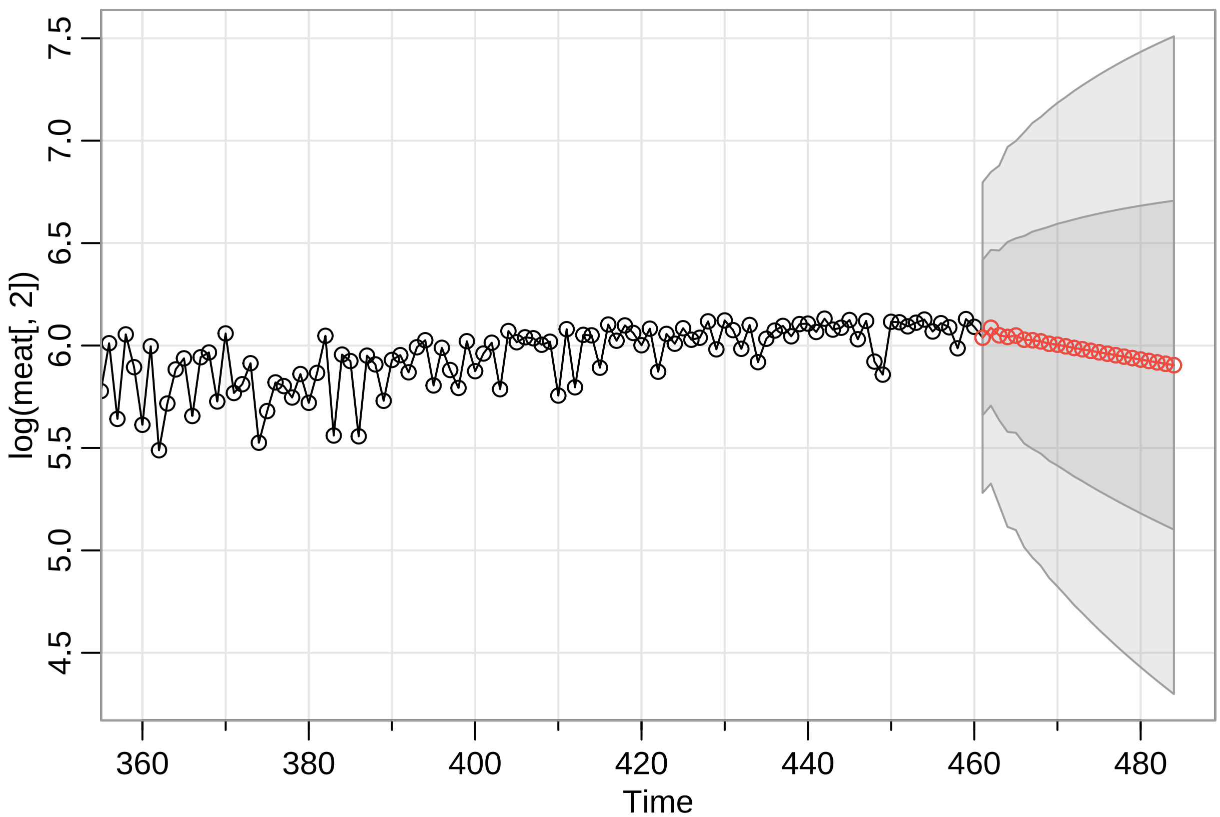
```
sarima.for(meat[,2], n.ahead=24, p=1, d=2, q=1) # ARIMA(1,1,1)
```




```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 454.5717 448.0381 453.8478 451.6829 454.6701 454.3287 456.1378 456.5576
##  [9] 457.8750 458.6124 459.7245 460.5946 461.6210 462.5464 463.5371 464.4856
## [17] 465.4613 466.4195 467.3890 468.3512 469.3181 470.2820 471.2478 472.2124
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1]  44.03909  46.74806  57.86268  62.00960  68.89252  73.33046  78.63046
##  [8]  82.93455  87.45805  91.51832  95.58137  99.39501 103.14460 106.73845
## [15] 110.25358 113.65851 116.98728 120.23085 123.40561 126.51065 129.55483
## [22] 132.53990 135.47132 138.35152
```

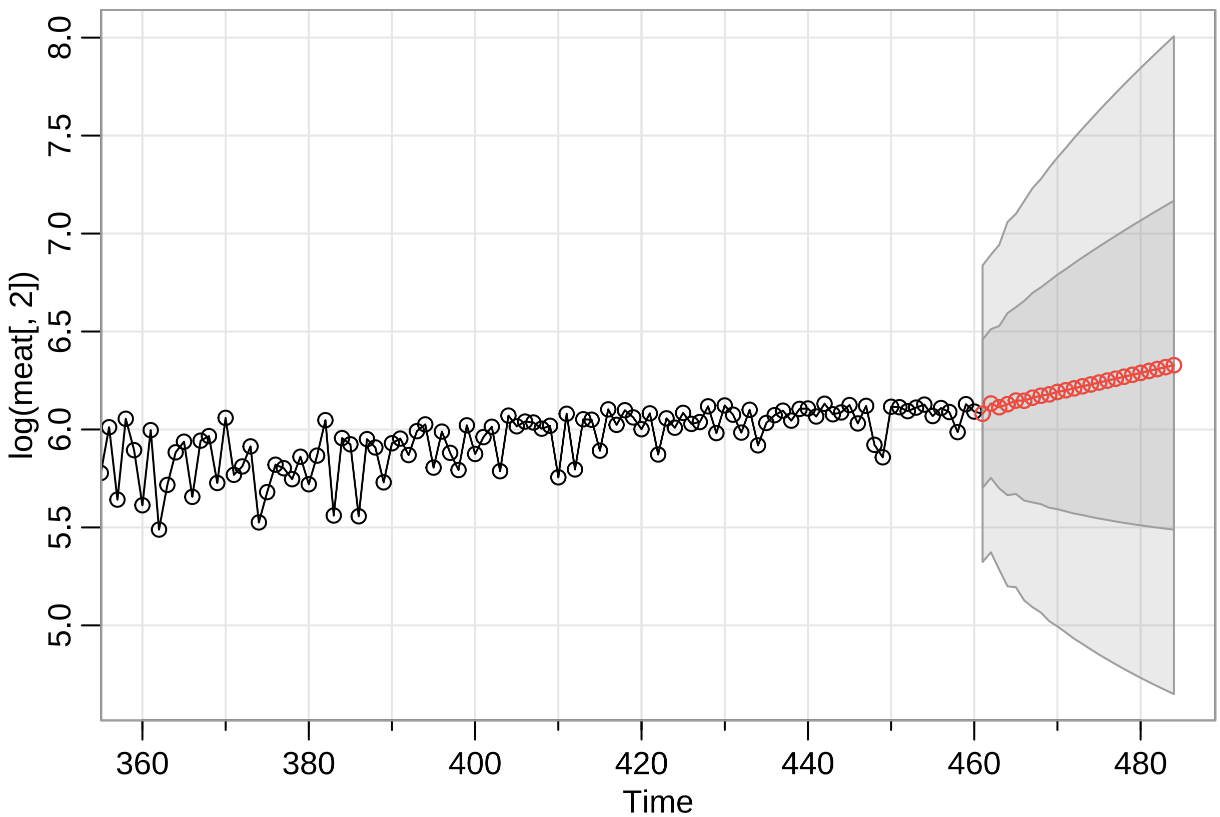
Differencing for AR(3) Differencing doesn't appear to improve the stationarity or forecast. AR(3) on log transformed data seems to work the best of the models investigated so far.

```
sarima.for(log(meat[,2]), n.ahead=24, p=3, d=0, q=0) # AR(1)
```



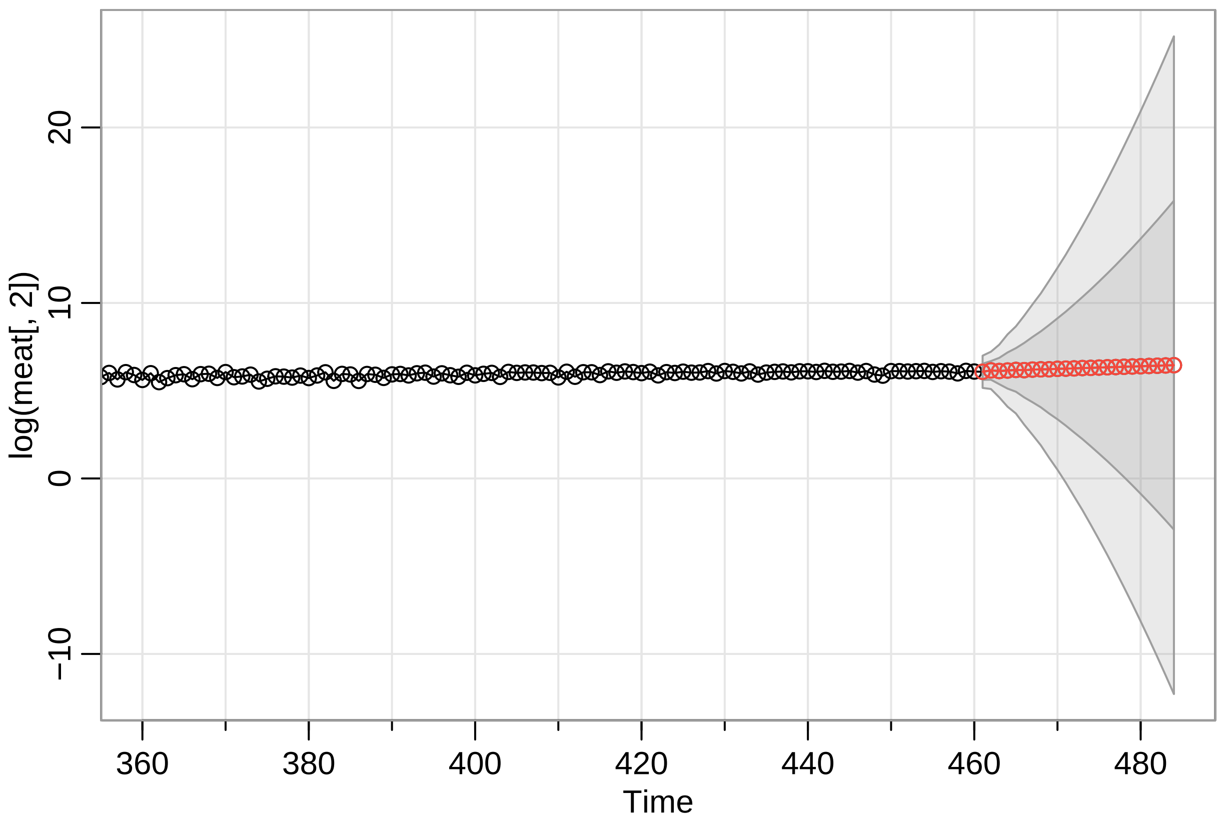

```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 6.038501 6.086649 6.049768 6.042045 6.048781 6.028293 6.025826 6.020098
##  [9] 6.008739 6.004194 5.996218 5.988176 5.981936 5.974163 5.967007 5.960110
## [17] 5.952761 5.945783 5.938765 5.931687 5.924765 5.917810 5.910890 5.904034
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 0.3790059 0.3802168 0.4143180 0.4637167 0.4747307 0.5064260 0.5304078
##  [8] 0.5477970 0.5715814 0.5902857 0.6084137 0.6274247 0.6442280 0.6610359
## [15] 0.6774131 0.6928556 0.7081052 0.7228152 0.7370295 0.7509346 0.7644034
## [22] 0.7775173 0.7903108 0.8027599
```

```
sarima.for(log(meat[,2]), n.ahead=24, p=2, d=1, q=0) # AR(2)
```




```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 6.080271 6.132424 6.113473 6.129163 6.147594 6.146649 6.162188 6.171996
##  [9] 6.179054 6.191430 6.200253 6.209756 6.220362 6.229623 6.239584 6.249556
## [17] 6.259177 6.269116 6.278933 6.288708 6.298580 6.308384 6.318203 6.328041
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 0.3785023 0.3797235 0.4145641 0.4650430 0.4764831 0.5095378 0.5346660
##  [8] 0.5531241 0.5784770 0.5985719 0.6182293 0.6389621 0.6574424 0.6760701
## [15] 0.6943448 0.7117162 0.7289957 0.7457867 0.7621372 0.7782490 0.7939725
## [22] 0.8093969 0.8245560 0.8394175
```

```
sarima.for(log(meat[,2]), n.ahead=24, p=3, d=2, q=0) # AR(3)
```




```
## $pred
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 6.077672 6.157767 6.121770 6.149084 6.183692 6.169956 6.202766 6.217291
##  [9] 6.221299 6.248204 6.257589 6.270840 6.291247 6.301585 6.318000 6.334440
## [17] 6.346950 6.363611 6.378396 6.392503 6.408534 6.422960 6.437852 6.453281
##
## $se
## Time Series:
## Start = 461
## End = 484
## Frequency = 1
##  [1] 0.4593158 0.5311106 0.7499263 1.0305711 1.2401777 1.5508092 1.8586427
##  [8] 2.1640320 2.5231487 2.8763608 3.2480934 3.6465705 4.0472159 4.4684796
## [15] 4.9047617 5.3494393 5.8115279 6.2847142 6.7690635 7.2674751 7.7761016
## [22] 8.2963922 8.8285058 9.3707318
```

Forecasting using data sets provided by Jocelyn Hunyadi Jocelyn's code for data loading and cleaning

```
# Load Data
FI_long <- readRDS("FI_long.rds") %>% group_by(Time) %>%
  summarize(Production_capita = sum(Production_capita, na.rm = TRUE))
FI_long_rmp <- readRDS("FI_long_rmp.rds")
```

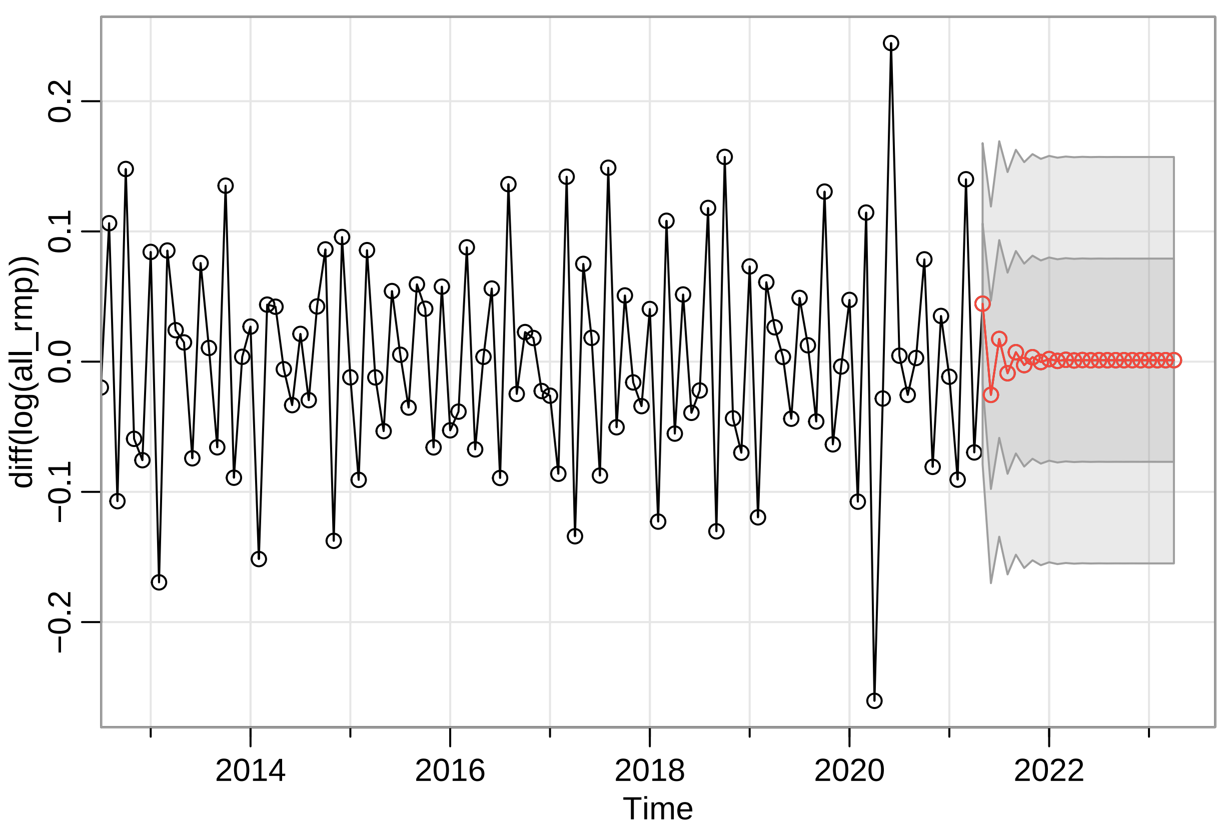
```r
FI_long_big <- readRDS("FI_long_big.rds")
FI_long_small <- readRDS("FI_long_small.rds")

# Time Series Objects:
FI_long_rm <- FI_long_rmp %>%
  dplyr::select(-Time) %>% arrange(Time2, decreasing = TRUE) %>%
  filter(Meat_Type == "TotalRedMeat")
all_rmp <-ts(FI_long_rm$Production, start=c(1983,1), end=c(2021,4), deltat=1/12)

FI_long_p <- FI_long_rmp %>% filter(Meat_Type != "TotalRedMeat") %>%
  arrange(Time2, decreasing = TRUE)
all_p <- ts(FI_long_p$Production, start=c(1983,1), end=c(2021,4), deltat=1/12)
```

24-Month Forecast: All Red Meat and Poultry

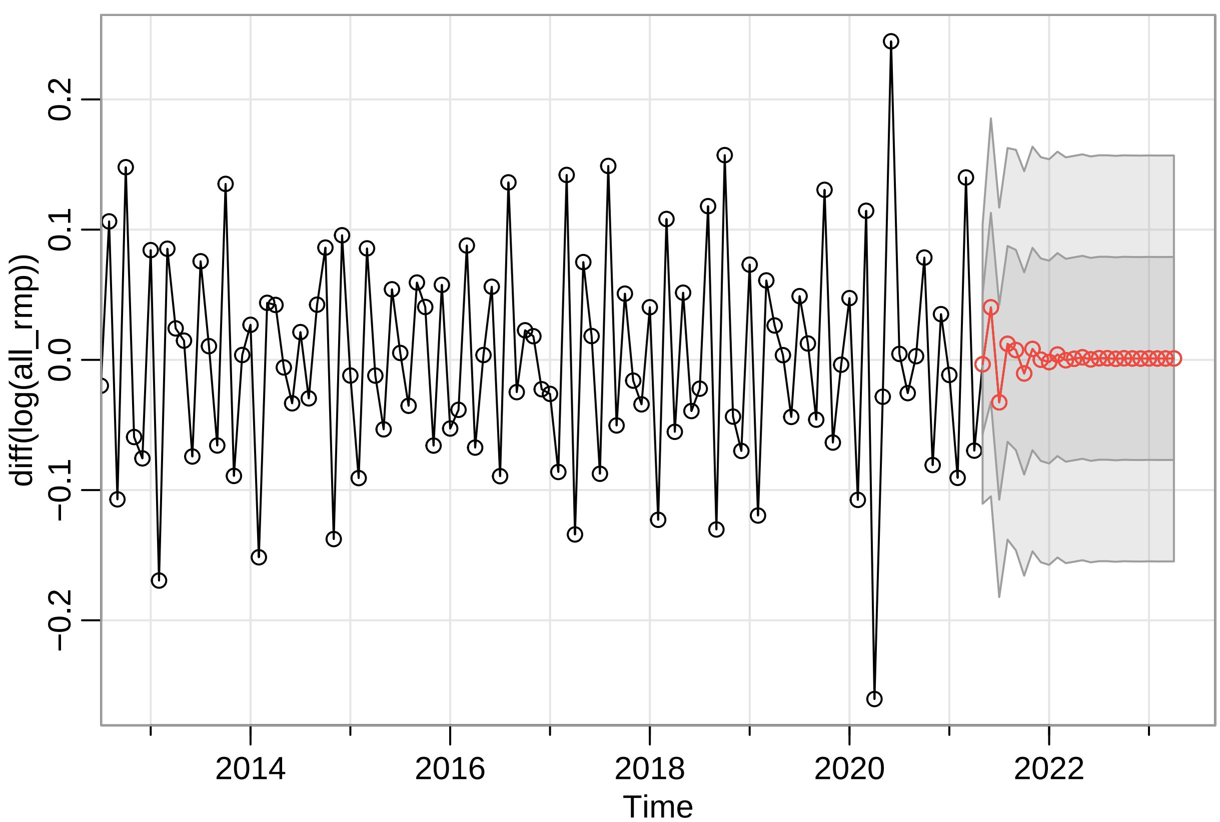
```r
sarima.for(diff(log(all_rmp)), n.ahead=24, p=1, d=0, q=0) # AR(1)
```




```
## $pred
##               Jan           Feb           Mar           Apr           May
## 2021                                                        0.0445167266
## 2022  0.0019720463  0.0005767208  0.0014320013  0.0009077475  0.0012290948
## 2023  0.0011094093  0.0011054840  0.0011078900  0.0011064152
##               Jun           Jul           Aug           Sep           Oct
## 2021 -0.0255015206  0.0174169534 -0.0088904091  0.0072349854 -0.0026492570
## 2022  0.0010321214  0.0011528585  0.0010788513  0.0011242148  0.0010964087
## 2023
##               Nov           Dec
```

```
## 2021  0.0034094008 -0.0003043218
## 2022  0.0011134528  0.0011030054
## 2023
##
## $se
##             Jan        Feb        Mar        Apr        May        Jun
## 2021                                              0.06164717 0.07230668
## 2022 0.07801742 0.07802105 0.07802242 0.07802293 0.07802313 0.07802320
## 2023 0.07802324 0.07802324 0.07802324 0.07802324
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.07592590 0.07724191 0.07773060 0.07791342 0.07798200 0.07800775
## 2022 0.07802323 0.07802324 0.07802324 0.07802324 0.07802324 0.07802324
## 2023
```

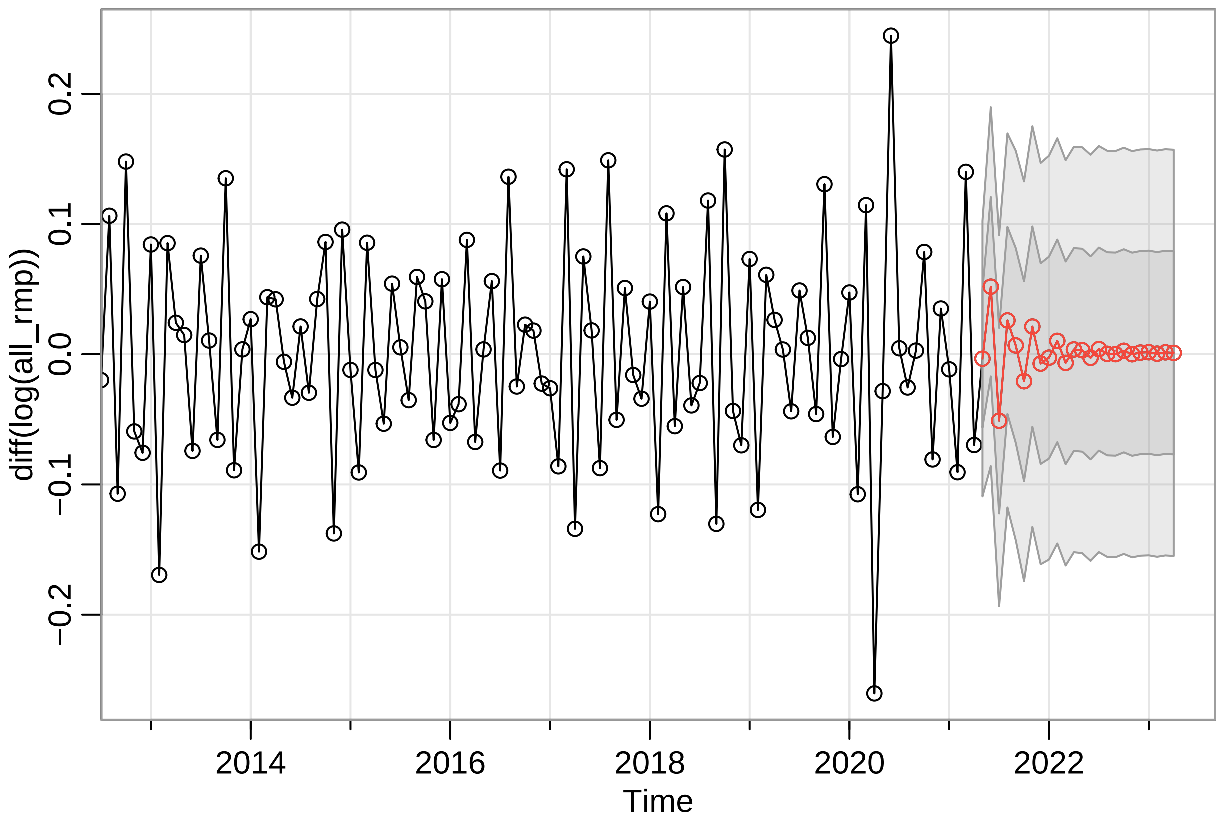
```
sarima.for(diff(log(all_rmp)), n.ahead=24, p=2, d=0, q=0) # AR(2)
```




```
## $pred
##               Jan           Feb           Mar           Apr           May
## 2021                                                          -0.0033952524
## 2022 -0.0017064994  0.0040555146 -0.0003059913  0.0008142561  0.0019596802
## 2023  0.0011222038  0.0010325536  0.0010498504  0.0010786309
##               Jun           Jul           Aug           Sep           Oct
## 2021  0.0403268439 -0.0326148905  0.0123019035  0.0075300829 -0.0104486403
## 2022  0.0003556984  0.0012519107  0.0012306052  0.0008043378  0.0012045053
## 2023
##               Nov           Dec
## 2021  0.0083556419  0.0001120474
```

```
## 2022  0.0010507950  0.0009922460
## 2023
##
## $se
##             Jan        Feb        Mar        Apr        May        Jun
## 2021                                                 0.05355144 0.07254660
## 2022 0.07787826 0.07790664 0.07790693 0.07791682 0.07792361 0.07792427
## 2023 0.07792583 0.07792583 0.07792583 0.07792583
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.07476895 0.07517524 0.07688674 0.07765015 0.07767346 0.07775979
## 2022 0.07792457 0.07792540 0.07792571 0.07792571 0.07792576 0.07792582
## 2023
```

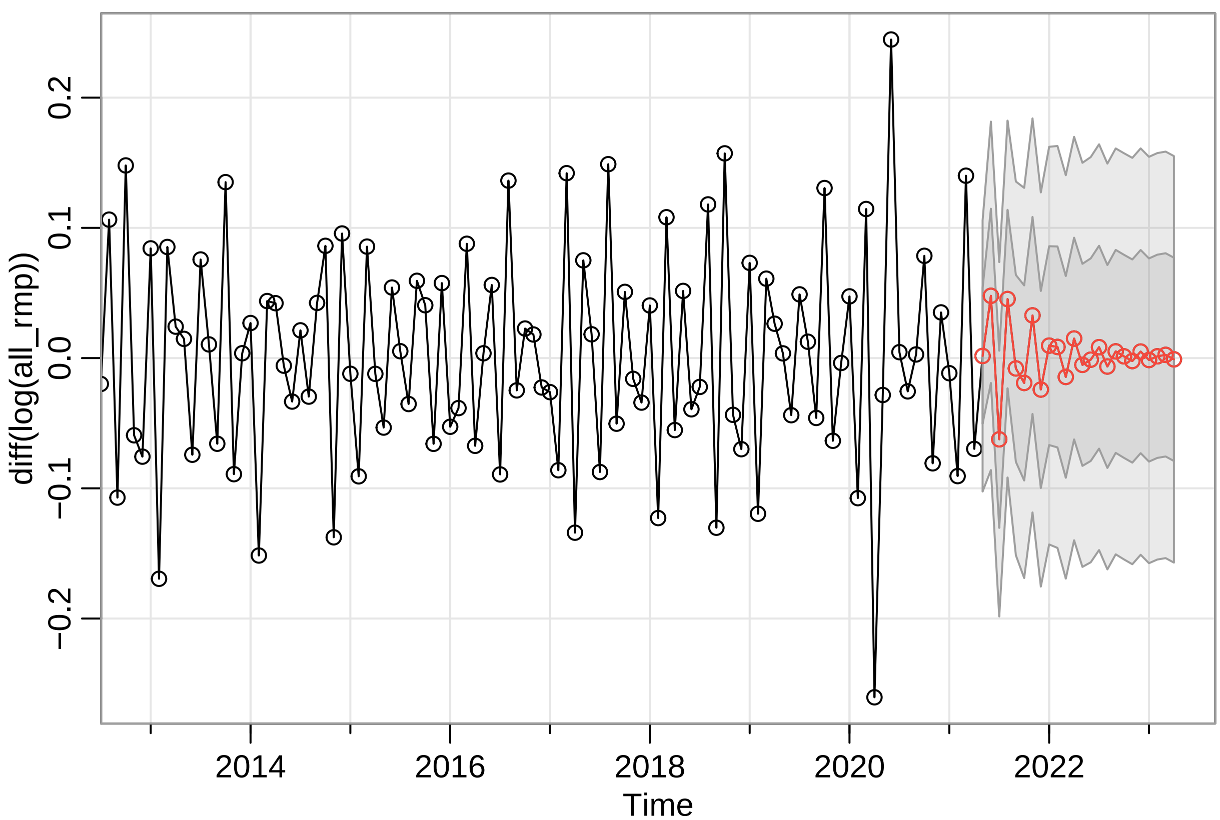
```
sarima.for(diff(log(all_rmp)), n.ahead=24, p=3, d=0, q=0) # AR(3)
```




```
## $pred
##               Jan           Feb           Mar           Apr           May
## 2021                                                        -3.422033e-03
## 2022 -2.618696e-03  1.026906e-02 -6.623685e-03  3.665345e-03  3.079398e-03
## 2023  1.542160e-03  4.413936e-04  1.458360e-03  1.043053e-03
##               Jun           Jul           Aug           Sep           Oct
## 2021  5.188732e-02 -5.104473e-02  2.595948e-02  6.744712e-03 -2.073493e-02
## 2022 -2.746498e-03  3.962194e-03  3.206584e-04  6.948640e-05  2.630476e-03
## 2023
##               Nov           Dec
## 2021  2.123880e-02 -7.181501e-03
## 2022  1.017267e-06  1.253958e-03
```

```
## 2023
##
## $se
##              Jan        Feb        Mar        Apr        May        Jun
## 2021                                                 0.05287449 0.06890385
## 2022 0.07752796 0.07779466 0.07781108 0.07784518 0.07792433 0.07796049
## 2023 0.07798998 0.07799058 0.07799058 0.07799087
##              Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.07127896 0.07181401 0.07475723 0.07670204 0.07690746 0.07704879
## 2022 0.07796141 0.07796889 0.07798169 0.07798643 0.07798644 0.07798797
## 2023
```

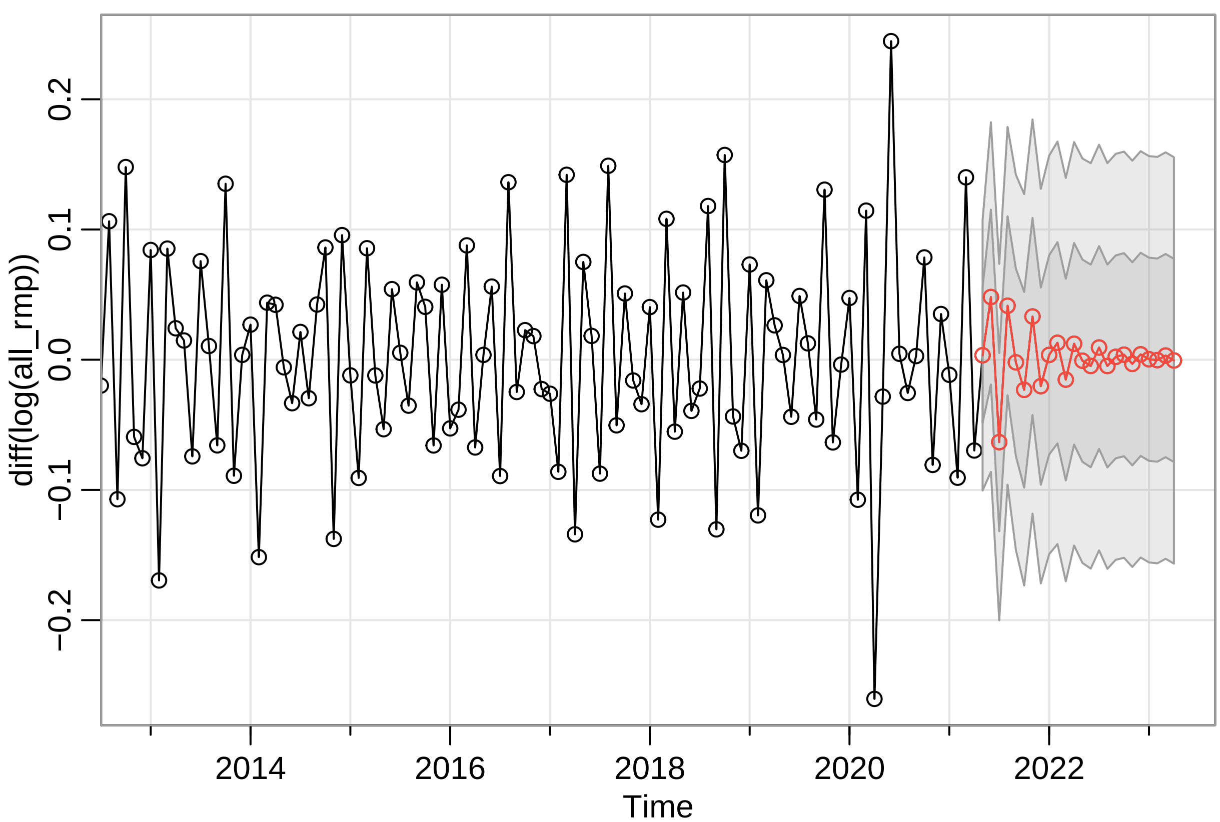
```
sarima.for(diff(log(all_rmp)), n.ahead=24, p=4, d=0, q=0) # AR(4)
```




```
## $pred
##                Jan           Feb           Mar           Apr           May
## 2021                                                                0.0016336539
## 2022  0.0095764396  0.0085736467 -0.0144690002  0.0150207866 -0.0051988766
## 2023 -0.0014736610  0.0013640643  0.0024859732 -0.0009606225
##                Jun           Jul           Aug           Sep           Oct
## 2021  0.0478282533 -0.0623662547  0.0453584467 -0.0079415412 -0.0190999424
## 2022 -0.0012114411  0.0083866454 -0.0064719576  0.0051627590  0.0013458846
## 2023
##                Nov           Dec
## 2021  0.0327968891 -0.0241511302
## 2022 -0.0022754658  0.0049979931
## 2023
```

```
##
## $se
##             Jan        Feb        Mar        Apr        May        Jun
## 2021                                                  0.05204540 0.06689625
## 2022 0.07634058 0.07715834 0.07744837 0.07744884 0.07757835 0.07779407
## 2023 0.07801856 0.07803154 0.07804197 0.07804403
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.06802514 0.06850655 0.07176210 0.07490319 0.07564190 0.07567055
## 2022 0.07789690 0.07790124 0.07792341 0.07797783 0.07801166 0.07801545
## 2023
```

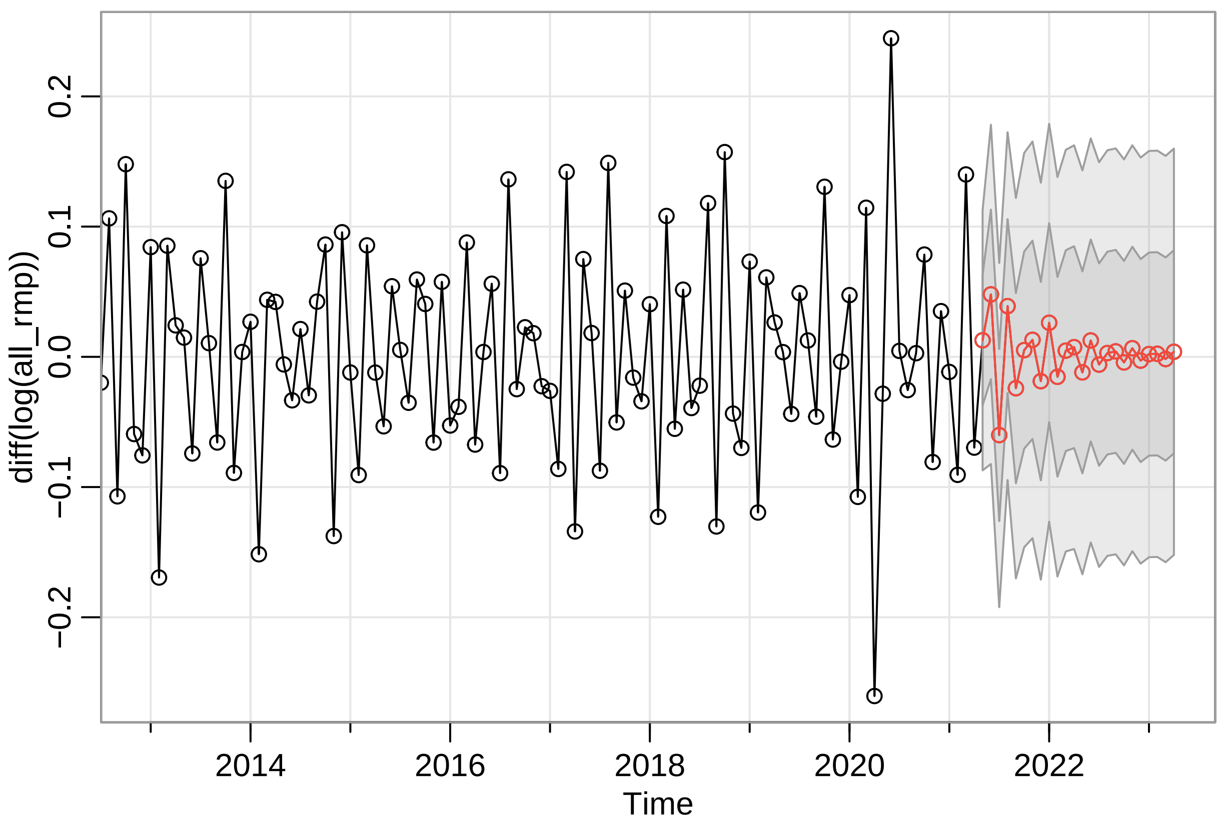
```
sarima.for(diff(log(all_rmp)), n.ahead=24, p=5, d=0, q=0) # AR(5)
```




```
## $pred
##                Jan           Feb           Mar           Apr           May
## 2021                                                          0.0034259591
## 2022  0.0036748273  0.0129894804 -0.0152946225  0.0122530562 -0.0007592687
## 2023  0.0003491386 -0.0003126774  0.0031892613 -0.0005641353
##                Jun           Jul           Aug           Sep           Oct
## 2021  0.0481356438 -0.0633071219  0.0414274636 -0.0020387473 -0.0231201337
## 2022 -0.0047718537  0.0093623304 -0.0048315497  0.0021947956  0.0038678437
## 2023
##                Nov           Dec
## 2021  0.0331948703 -0.0203025811
## 2022 -0.0031753821  0.0041288994
## 2023
##
```

```
## $se
##             Jan        Feb        Mar        Apr        May        Jun
## 2021                                                0.05193810 0.06712605
## 2022 0.07651181 0.07726583 0.07743241 0.07744708 0.07763646 0.07783142
## 2023 0.07800688 0.07802041 0.07802443 0.07802447
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.06843208 0.06869217 0.07201984 0.07513831 0.07567189 0.07574972
## 2022 0.07788006 0.07788252 0.07792955 0.07798088 0.07799489 0.07799526
## 2023
```

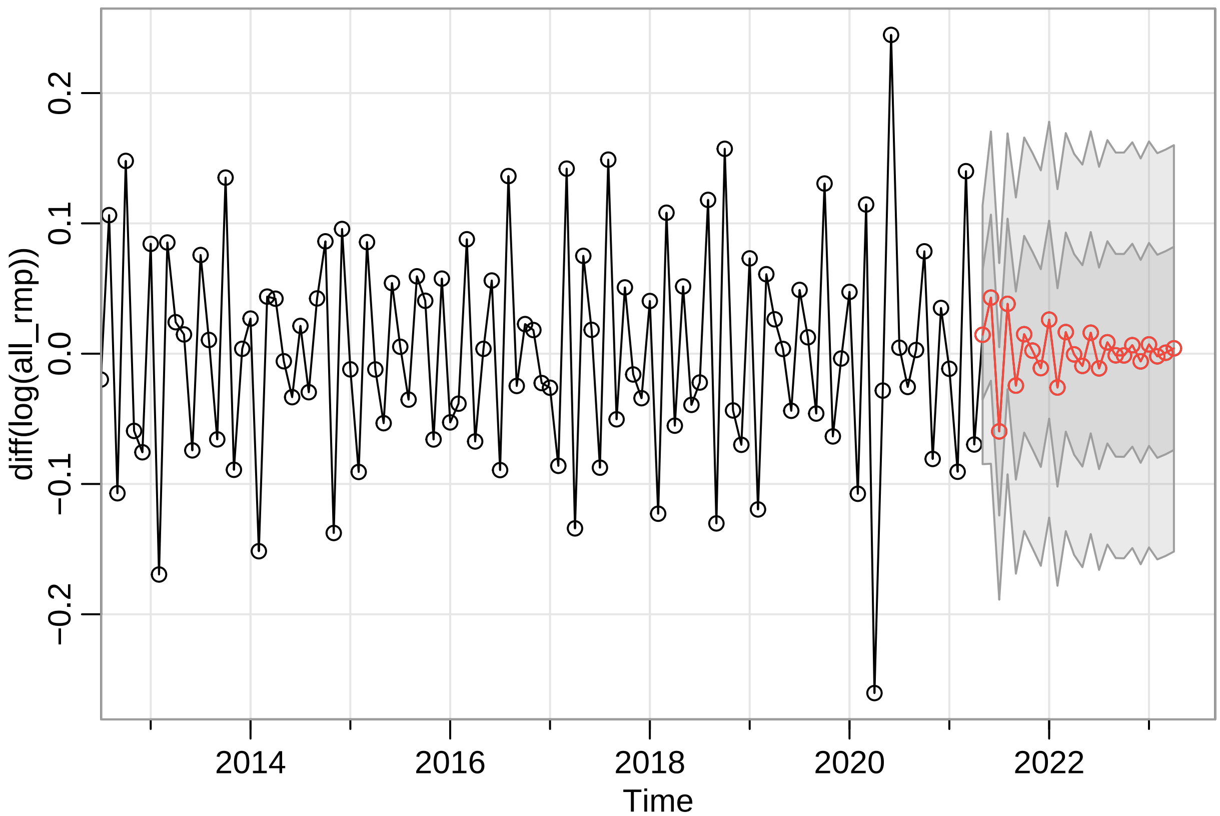
```
sarima.for(diff(log(all_rmp)), n.ahead=24, p=6, d=0, q=0) # AR(6)
```




```
## $pred
##              Jan          Feb          Mar          Apr          May
## 2021                                                      0.012709705
## 2022  0.026154362 -0.015335259  0.004754233  0.007425063 -0.011937521
## 2023  0.002101034  0.002340529 -0.001694503  0.003905617
##              Jun          Jul          Aug          Sep          Oct
## 2021  0.047955163 -0.060085985  0.038926691 -0.024083192  0.005155970
## 2022  0.012569279 -0.005962300  0.002897919  0.004197088 -0.004292060
## 2023
##              Nov          Dec
## 2021  0.013057541 -0.018684206
## 2022  0.006603148 -0.002894404
## 2023
##
## $se
```

```
##              Jan        Feb        Mar        Apr        May        Jun
## 2021                                            0.04995407 0.06512118
## 2022 0.07639100 0.07668117 0.07711008 0.07750239 0.07753779 0.07755276
## 2023 0.07798835 0.07800621 0.07800890 0.07800981
##              Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.06607441 0.06676630 0.07303487 0.07569913 0.07614398 0.07619876
## 2022 0.07768796 0.07786247 0.07791984 0.07792682 0.07793247 0.07796087
## 2023
```

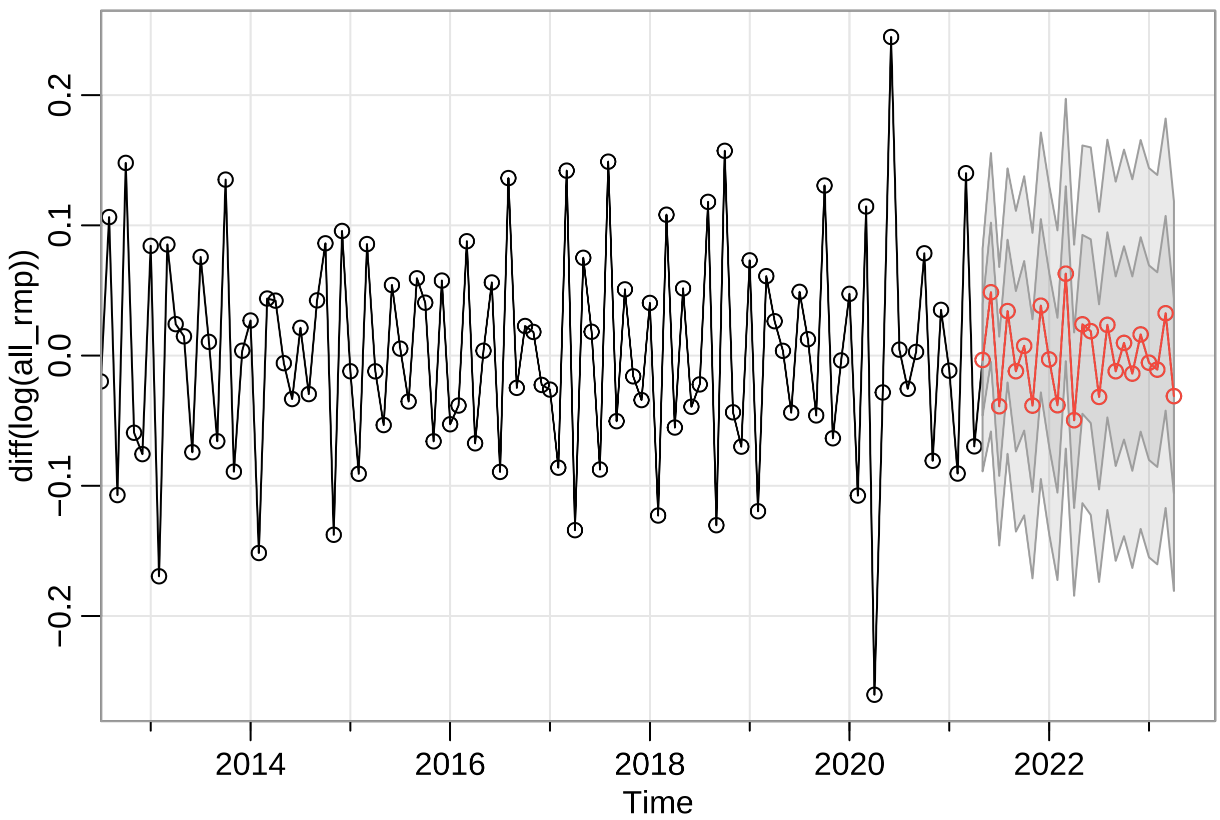
```
sarima.for(diff(log(all_rmp)), n.ahead=24, p=7, d=0, q=0) # AR(7)
```




```
## $pred
##                 Jan           Feb           Mar           Apr           May
## 2021                                                             0.0144792298
## 2022  0.0261214815 -0.0259534336  0.0165558428 -0.0005379099 -0.0093926766
## 2023  0.0071044750 -0.0020295310  0.0007009912  0.0040778742
##                 Jun           Jul           Aug           Sep           Oct
## 2021  0.0430133376 -0.0596536426  0.0382276640 -0.0244977389  0.0149241207
## 2022  0.0160943647 -0.0112829252  0.0087108769 -0.0012625074 -0.0013464494
## 2023
##                 Nov           Dec
## 2021  0.0023418155 -0.0110827420
## 2022  0.0065271183 -0.0059276664
## 2023
##
## $se
##              Jan        Feb        Mar        Apr        May        Jun
```

```
## 2021                                        0.04963994 0.06374859
## 2022 0.07594250 0.07610442 0.07637735 0.07695441 0.07724700 0.07726045
## 2023 0.07789575 0.07793231 0.07796205 0.07797116
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.06456782 0.06540205 0.07217628 0.07549277 0.07583328 0.07586574
## 2022 0.07733470 0.07760344 0.07779954 0.07785203 0.07785370 0.07786533
## 2023
```

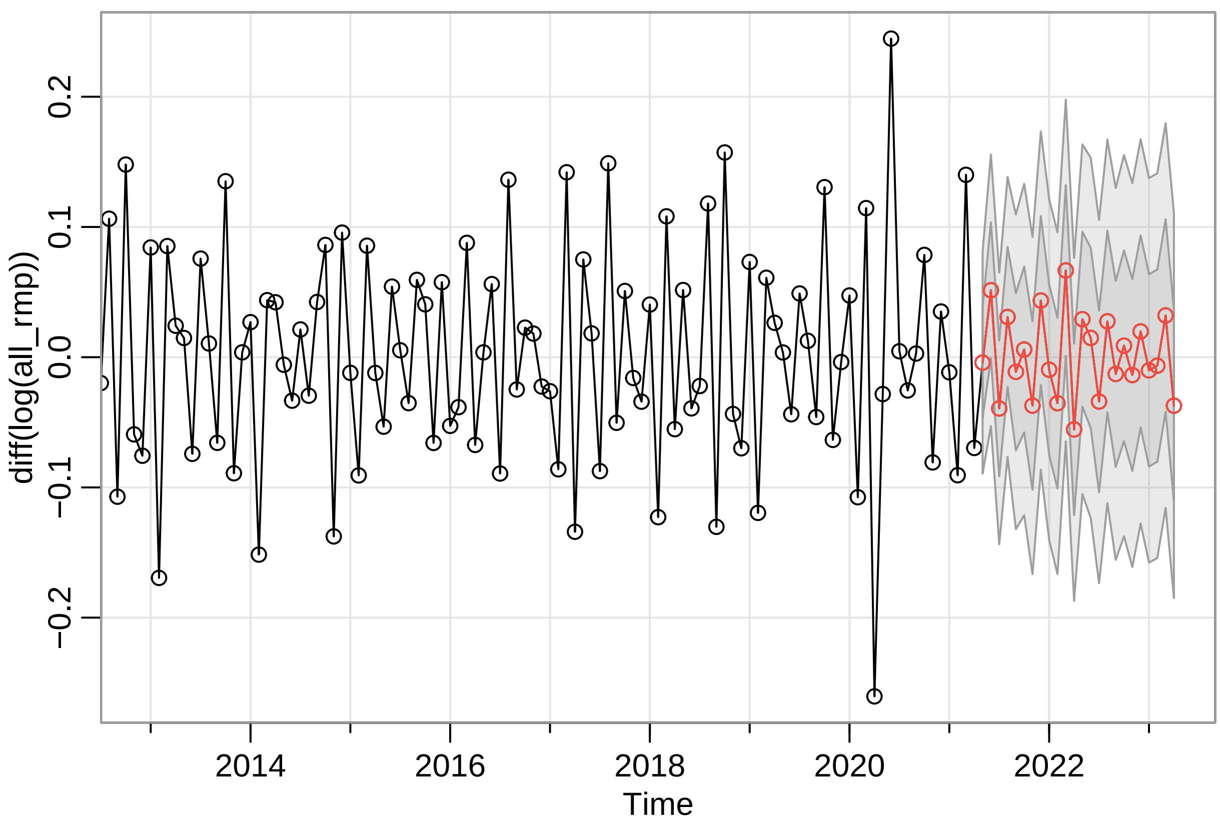
```
sarima.for(diff(log(all_rmp)), n.ahead=24, p=8, d=0, q=0) # AR(8)
```




```
## $pred
##               Jan          Feb          Mar          Apr          May
## 2021                                                      -0.003377495
## 2022 -0.002956033 -0.038124447  0.062869148 -0.049639614  0.023986968
## 2023 -0.005453579 -0.010807799  0.032482361 -0.031152803
##               Jun          Jul          Aug          Sep          Oct
## 2021  0.048598241 -0.038908726  0.034145947 -0.012012800  0.007464844
## 2022  0.018747341 -0.031720485  0.023588757 -0.012020141  0.009712295
## 2023
##               Nov          Dec
## 2021 -0.038429834  0.038275368
## 2022 -0.013850272  0.016222982
## 2023
##
## $se
##             Jan        Feb        Mar        Apr        May        Jun
## 2021                                        0.04277541 0.05345157
```

```
## 2022 0.06693803 0.06712837 0.06715099 0.06739466 0.06867404 0.07055490
## 2023 0.07472638 0.07474686 0.07476404 0.07479595
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.05345982 0.05480072 0.06156685 0.06510698 0.06632253 0.06651693
## 2022 0.07106048 0.07108378 0.07279704 0.07421526 0.07462016 0.07465879
## 2023
```

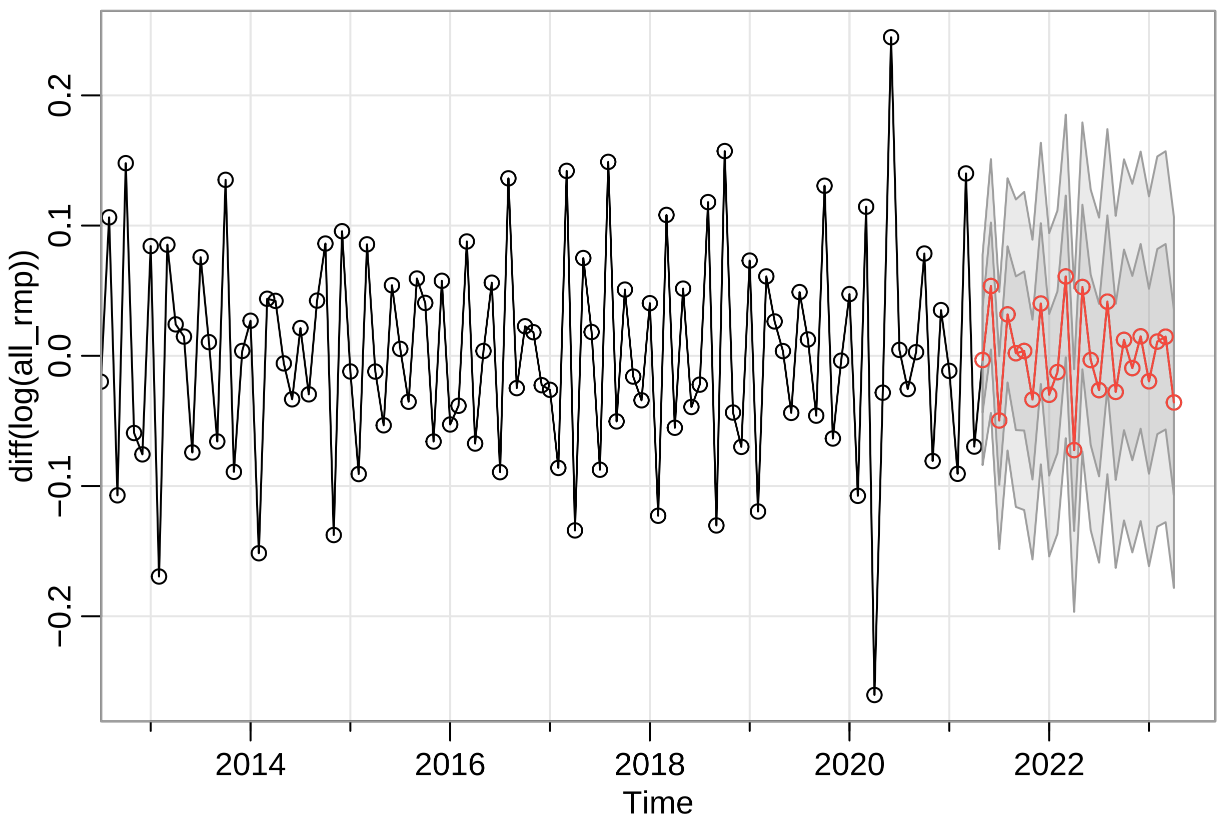
```
sarima.for(diff(log(all_rmp)), n.ahead=24, p=9, d=0, q=0) # AR(9)
```




```
## $pred
##               Jan          Feb          Mar          Apr          May
## 2021                                                      -0.004144047
## 2022 -0.009603886 -0.035344606  0.066613027 -0.055455117  0.029147602
## 2023 -0.010108908 -0.006509592  0.032043084 -0.037243772
##               Jun          Jul          Aug          Sep          Oct
## 2021  0.051411946 -0.039376235  0.030873907 -0.011278960  0.005913052
## 2022  0.014766959 -0.034034277  0.027523509 -0.012864458  0.008849245
## 2023
##               Nov          Dec
## 2021 -0.037187846  0.043515009
## 2022 -0.013738006  0.019774768
## 2023
##
## $se
##             Jan        Feb        Mar        Apr        May        Jun
## 2021                                            0.04261248 0.05213641
## 2022 0.06544148 0.06560616 0.06560760 0.06583654 0.06714046 0.06916044
```

```
## 2023 0.07381766 0.07385091 0.07386657 0.07389505
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.05214079 0.05372009 0.06040715 0.06361034 0.06472095 0.06490318
## 2022 0.06976601 0.06977190 0.07139339 0.07318289 0.07370862 0.07375598
## 2023
```

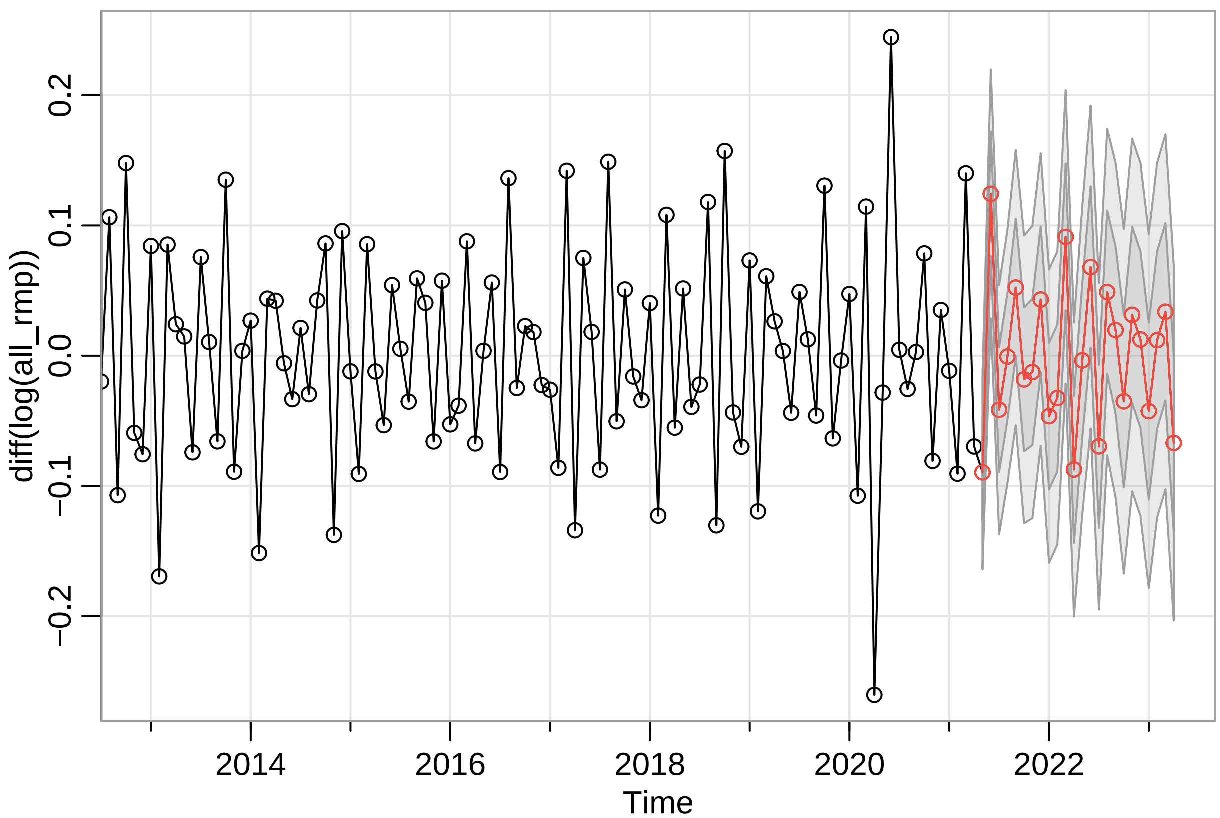
```
sarima.for(diff(log(all_rmp)), n.ahead=24, p=10, d=0, q=0) # AR(10)
```




```
## $pred
##               Jan          Feb          Mar          Apr          May
## 2021                                                      -0.003214204
## 2022 -0.029908061 -0.012528980  0.060955046 -0.072413034  0.052826685
## 2023 -0.019558444  0.010928188  0.014665144 -0.035815543
##               Jun          Jul          Aug          Sep          Oct
## 2021  0.053596959 -0.049664911  0.031812847  0.001999276  0.003712382
## 2022 -0.003231638 -0.026342817  0.041582681 -0.027723260  0.012245527
## 2023
##               Nov          Dec
## 2021 -0.033593002  0.040120059
## 2022 -0.009437482  0.014950441
## 2023
##
## $se
##             Jan        Feb        Mar        Apr        May        Jun
## 2021                                            0.04034426 0.04873141
## 2022 0.06205798 0.06207557 0.06211329 0.06211532 0.06318881 0.06538860
## 2023 0.07100045 0.07107439 0.07121417 0.07122292
```

```
##              Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.04938090 0.05227327 0.05902199 0.06107407 0.06137649 0.06173277
## 2022 0.06623649 0.06625256 0.06758011 0.06932771 0.07075954 0.07092844
## 2023
```

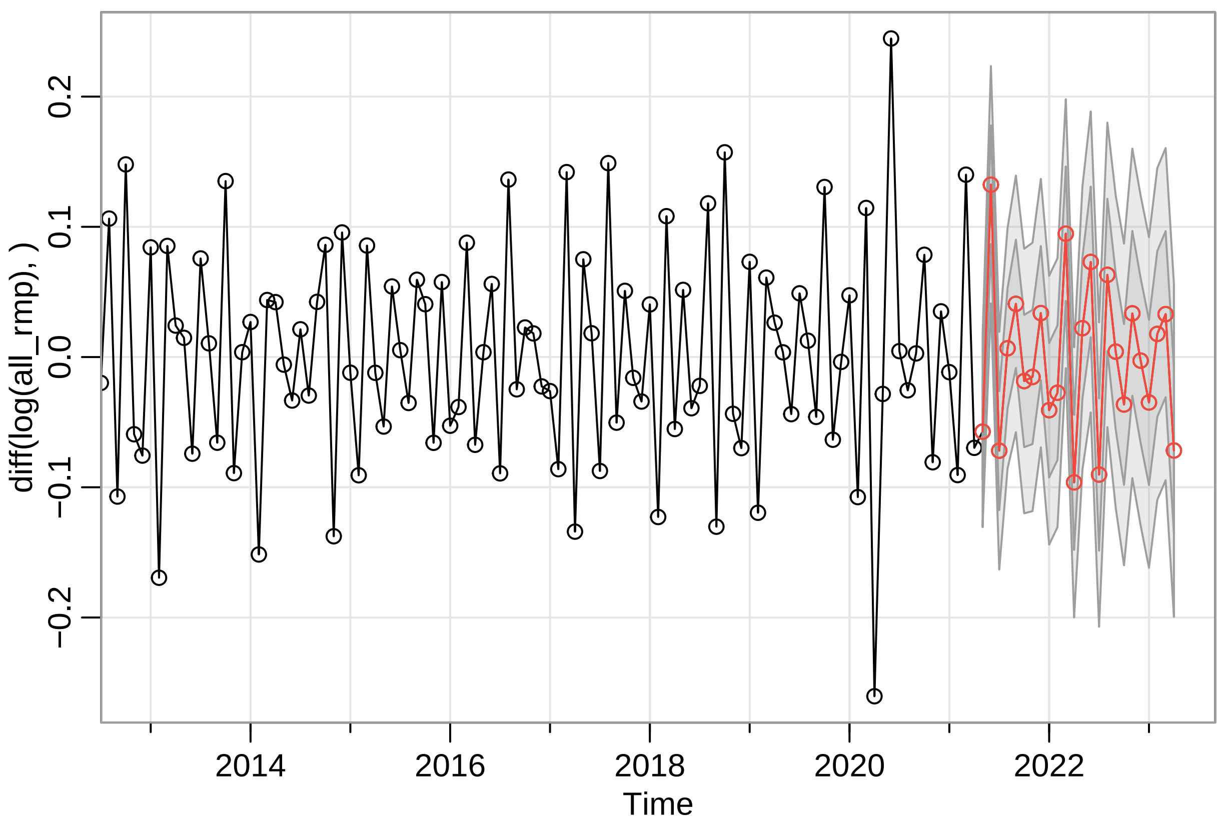
```
sarima.for(diff(log(all_rmp)), n.ahead=24, p=11, d=0, q=0) # AR(11)
```




```
## $pred
##                Jan           Feb           Mar           Apr           May
## 2021                                                        -0.0895919925
## 2022 -0.0464386914 -0.0324664941  0.0912252577 -0.0874125685 -0.0035869123
## 2023 -0.0425792829  0.0119025126  0.0337105143 -0.0669552464
##                Jun           Jul           Aug           Sep           Oct
## 2021  0.1242942873 -0.0414745408 -0.0006750937  0.0522845450 -0.0182037947
## 2022  0.0680365777 -0.0696298133  0.0488527896  0.0196979192 -0.0350515553
## 2023
##                Nov           Dec
## 2021 -0.0125354365  0.0431039374
## 2022  0.0313617190  0.0124342074
## 2023
##
## $se
##              Jan        Feb        Mar        Apr        May        Jun
## 2021                                             0.03718665 0.04776889
## 2022 0.05630966 0.05631574 0.05634288 0.05646373 0.05911515 0.06198914
## 2023 0.06791901 0.06809887 0.06813828 0.06820734
##              Jul        Aug        Sep        Oct        Nov        Dec
```

```
## 2021 0.04783583 0.04863614 0.05286985 0.05521460 0.05612290 0.05612342
## 2022 0.06262954 0.06264689 0.06423810 0.06614260 0.06768132 0.06773557
## 2023
```

```
sarima.for(diff(log(all_rmp), ), n.ahead=24, p=12, d=0, q=0) # AR(12)
```



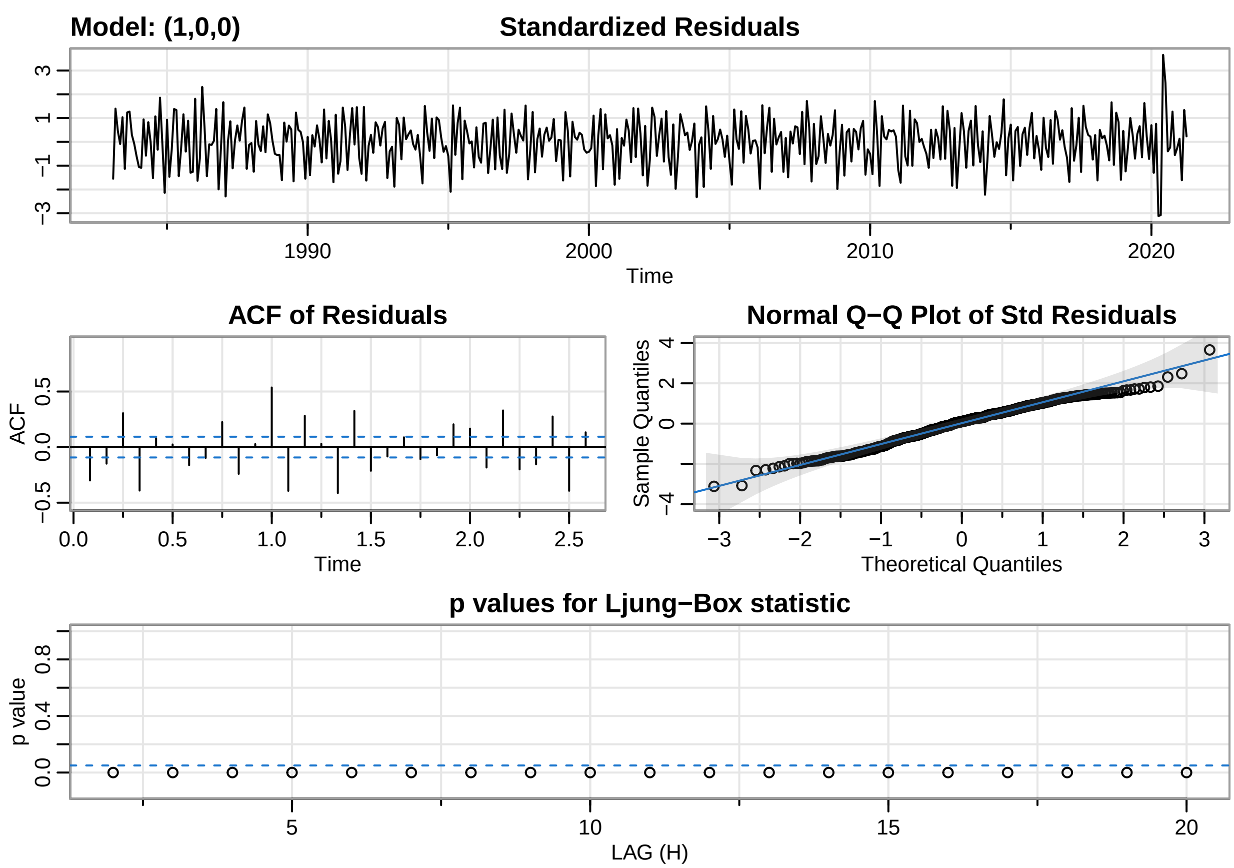

```
## $pred
##                Jan          Feb          Mar          Apr          May
## 2021                                                      -0.057279953
## 2022 -0.040809497 -0.027404471  0.094758735 -0.096233761  0.022150614
## 2023 -0.034882425  0.017760964  0.032953828 -0.071755084
##                Jun          Jul          Aug          Sep          Oct
## 2021  0.132342247 -0.071897230  0.006753334  0.040872153 -0.018446909
## 2022  0.073052388 -0.090260435  0.063142355  0.004154285 -0.036436358
## 2023
##                Nov          Dec
## 2021 -0.015294621  0.033735385
## 2022  0.033555488 -0.002759931
## 2023
##
## $se
##             Jan        Feb        Mar        Apr        May        Jun
## 2021                                              0.03667642 0.04557007
## 2022 0.05162028 0.05162142 0.05162879 0.05185977 0.05431121 0.05778279
## 2023 0.06348096 0.06368164 0.06377308 0.06388253
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.04564367 0.04613179 0.04926954 0.05078371 0.05153252 0.05153275
```

```
## 2022 0.05843953 0.05844253 0.05995576 0.06175625 0.06325433 0.06332524
## 2023
```

Checking the Model Conditions: All Red Meat and Poultry

```
sarima(diff(log(all_rmp)), p=1, d=0, q=0)# AR(1)
```

```
## initial  value -2.553084
## iter   2 value -2.787849
## iter   3 value -2.787860
## iter   4 value -2.787861
## iter   5 value -2.787861
## iter   5 value -2.787861
## iter   5 value -2.787861
## final  value -2.787861
## converged
## initial  value -2.785810
## iter   2 value -2.785814
## iter   3 value -2.785815
## iter   3 value -2.785815
## iter   3 value -2.785815
## final  value -2.785815
## converged
```

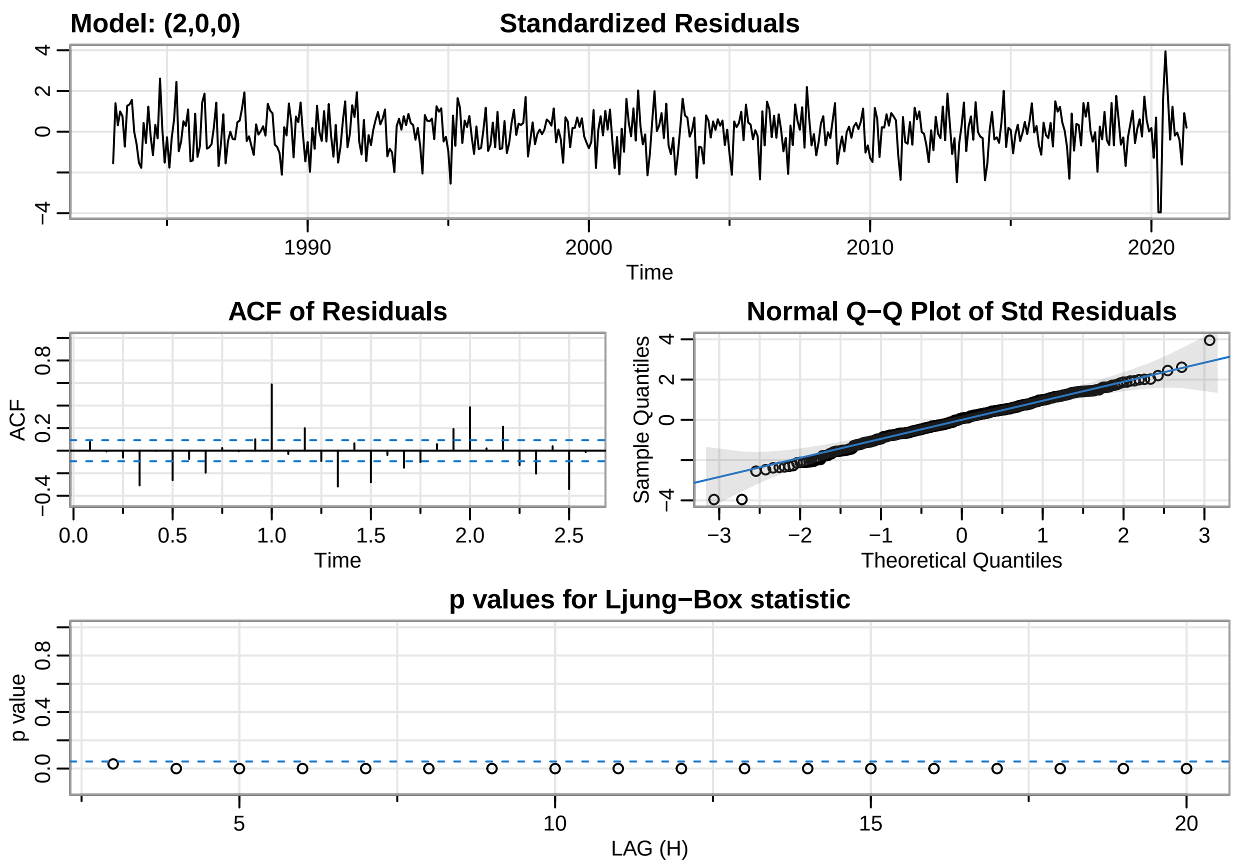



```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
```

```
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1   xmean
##       -0.613  0.0011
## s.e.   0.037  0.0018
##
## sigma^2 estimated as 0.0038:  log likelihood = 627.4,  aic = -1248.79
##
## $degrees_of_freedom
## [1] 457
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1    -0.6130 0.0370 -16.5785  0.0000
## xmean   0.0011 0.0018   0.6198  0.5357
##
## $AIC
## [1] -2.72068
##
## $AICc
## [1] -2.720623
##
## $BIC
## [1] -2.693693
```

```
sarima(diff(log(all_rmp)), p=2, d=0, q=0) # AR(2)
```

```
## initial  value -2.556648
## iter   2 value -2.755918
## iter   3 value -2.915015
## iter   4 value -2.915675
## iter   5 value -2.929618
## iter   6 value -2.929683
## iter   7 value -2.929722
## iter   8 value -2.929724
## iter   9 value -2.929725
## iter  10 value -2.929725
## iter  10 value -2.929725
## iter  10 value -2.929725
## final  value -2.929725
## converged
## initial  value -2.925972
## iter   2 value -2.925978
## iter   3 value -2.925986
## iter   4 value -2.925986
## iter   5 value -2.925986
## iter   5 value -2.925986
## iter   5 value -2.925986
## final  value -2.925986
## converged
```

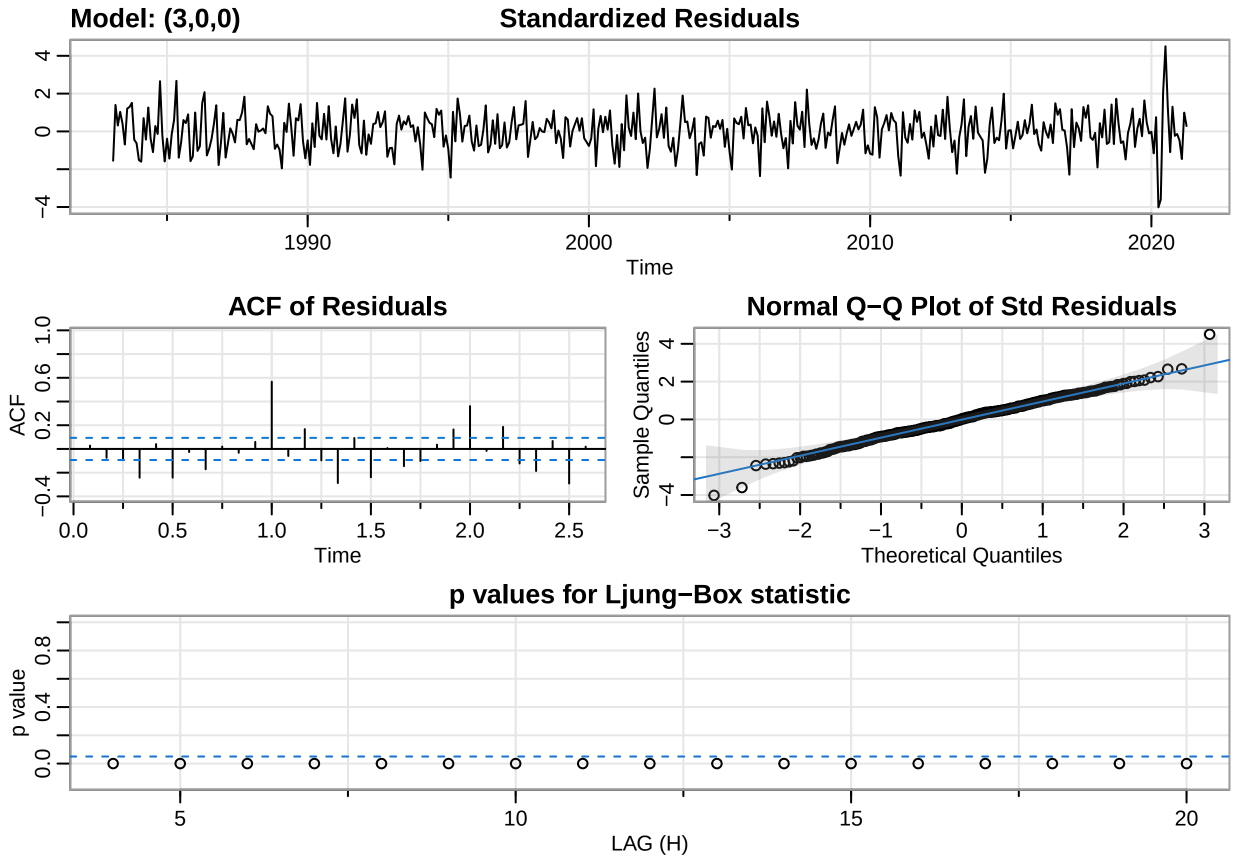



```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2   xmean
##       -0.9139  -0.4974  0.0011
## s.e.   0.0405   0.0408  0.0010
##
## sigma^2 estimated as 0.002868:  log likelihood = 691.73,  aic = -1375.47
##
## $degrees_of_freedom
## [1] 456
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1    -0.9139 0.0405 -22.5581  0.0000
## ar2    -0.4974 0.0408 -12.2024  0.0000
## xmean   0.0011 0.0010   1.0180  0.3092
##
## $AIC
## [1] -2.996665
##
```

```
## $AICc
## [1] -2.99655
##
## $BIC
## [1] -2.960682
```

```
sarima(diff(log(all_rmp)), p=3, d=0, q=0) # AR(3)
```

```
## initial  value -2.556431
## iter   2 value -2.884880
## iter   3 value -2.904106
## iter   4 value -2.929416
## iter   5 value -2.939858
## iter   6 value -2.941141
## iter   7 value -2.941450
## iter   8 value -2.941451
## iter   8 value -2.941451
## iter   8 value -2.941451
## final  value -2.941451
## converged
## initial  value -2.938608
## iter   2 value -2.938614
## iter   3 value -2.938621
## iter   4 value -2.938621
## iter   5 value -2.938621
## iter   5 value -2.938621
## iter   5 value -2.938621
## final  value -2.938621
## converged
```

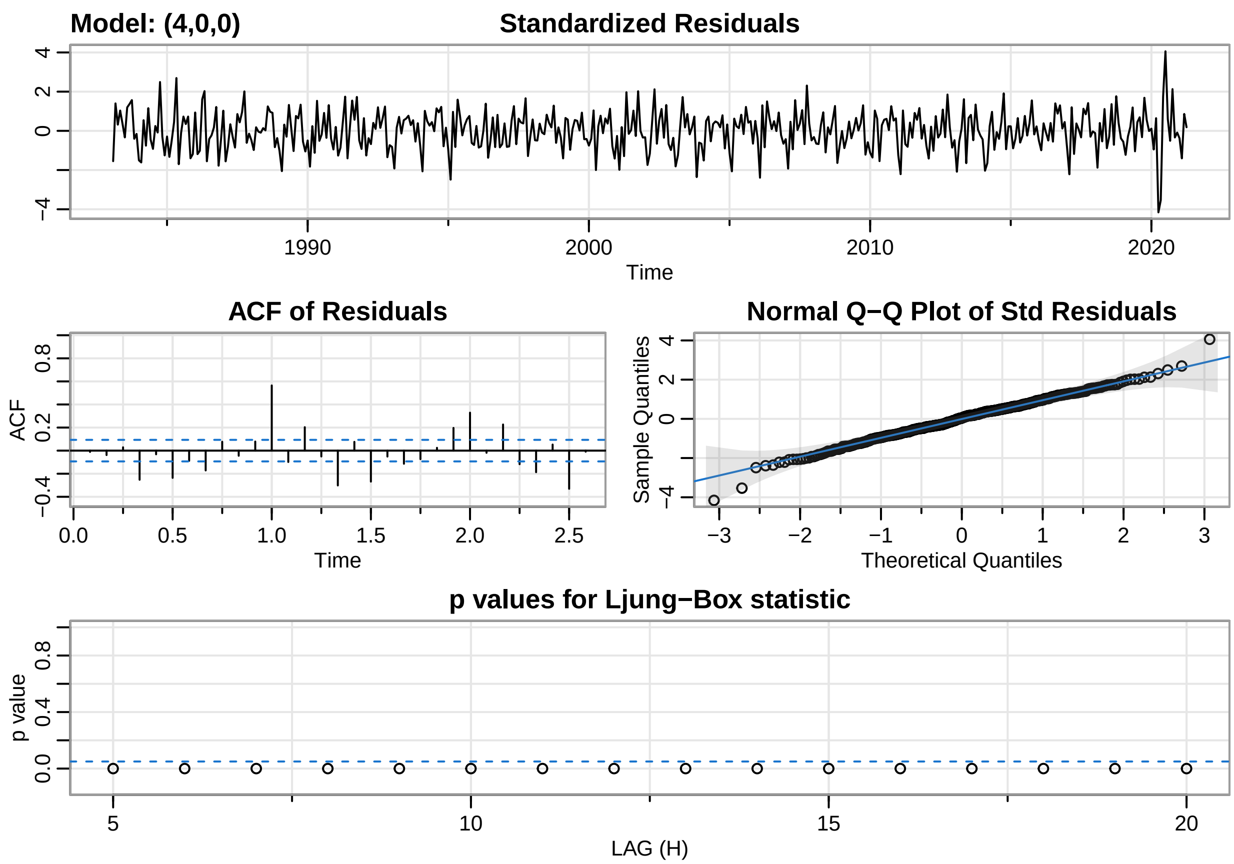



```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2     ar3   xmean
##       -0.8356  -0.3531  0.1588  0.0011
## s.e.   0.0461   0.0582  0.0463  0.0012
##
## sigma^2 estimated as 0.002796:  log likelihood = 697.53,  aic = -1385.07
##
## $degrees_of_freedom
## [1] 455
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1    -0.8356 0.0461 -18.1438  0.0000
## ar2    -0.3531 0.0582  -6.0651  0.0000
## ar3     0.1588 0.0463   3.4286  0.0007
## xmean   0.0011 0.0012   0.8787  0.3800
##
## $AIC
## [1] -3.017578
```

```
##
## $AICc
## [1] -3.017386
##
## $BIC
## [1] -2.972599
```

```
sarima(diff(log(all_rmp)), p=4, d=0, q=0) # AR(4)
```

```
## initial  value -2.555585
## iter   2 value -2.758273
## iter   3 value -2.904481
## iter   4 value -2.937520
## iter   5 value -2.947497
## iter   6 value -2.954731
## iter   7 value -2.956907
## iter   8 value -2.957322
## iter   9 value -2.957327
## iter  10 value -2.957336
## iter  11 value -2.957338
## iter  12 value -2.957338
## iter  12 value -2.957338
## final  value -2.957338
## converged
## initial  value -2.954266
## iter   2 value -2.954279
## iter   3 value -2.954284
## iter   4 value -2.954284
## iter   5 value -2.954284
## iter   5 value -2.954284
## iter   5 value -2.954284
## final  value -2.954284
## converged
```

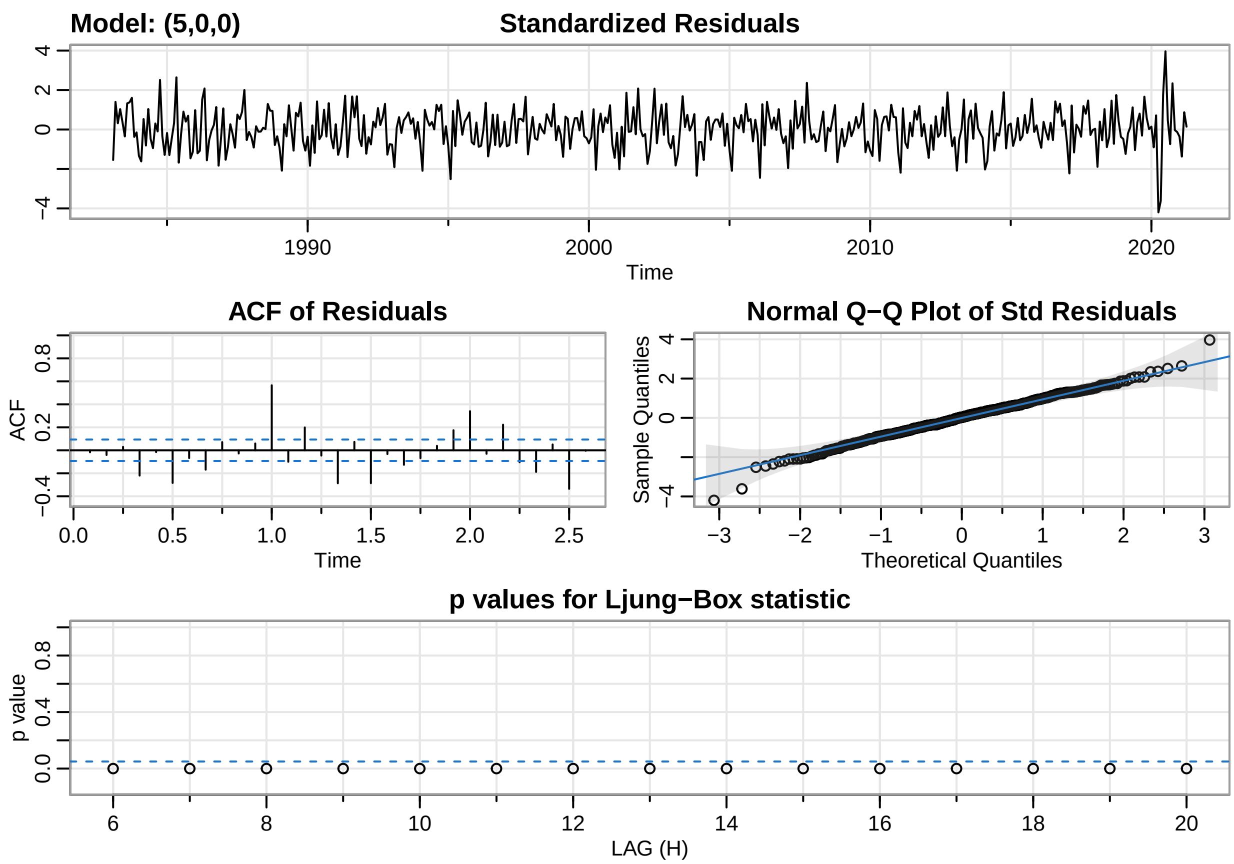



```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2     ar3      ar4   xmean
##       -0.8075  -0.4150  0.0122  -0.1766  0.0011
## s.e.   0.0459   0.0596  0.0596   0.0462  0.0010
##
## sigma^2 estimated as 0.002709:  log likelihood = 704.72,  aic = -1397.45
##
## $degrees_of_freedom
## [1] 454
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1    -0.8075 0.0459 -17.5835  0.0000
## ar2    -0.4150 0.0596  -6.9646  0.0000
## ar3     0.0122 0.0596   0.2039  0.8385
## ar4    -0.1766 0.0462  -3.8226  0.0002
## xmean   0.0011 0.0010   1.0314  0.3029
##
## $AIC
```

```
## [1] -3.044548
##
## $AICc
## [1] -3.044259
##
## $BIC
## [1] -2.990573
```

```r
sarima(diff(log(all_rmp)), p=5, d=0, q=0) # AR(5)
```

```
## initial  value -2.554798
## iter   2 value -2.577322
## iter   3 value -2.900936
## iter   4 value -2.921756
## iter   5 value -2.932988
## iter   6 value -2.949687
## iter   7 value -2.955794
## iter   8 value -2.957711
## iter   9 value -2.958307
## iter  10 value -2.958445
## iter  11 value -2.958452
## iter  12 value -2.958456
## iter  12 value -2.958456
## iter  12 value -2.958456
## final  value -2.958456
## converged
## initial  value -2.956306
## iter   2 value -2.956320
## iter   3 value -2.956326
## iter   4 value -2.956326
## iter   5 value -2.956327
## iter   5 value -2.956327
## iter   5 value -2.956327
## final  value -2.956327
## converged
```

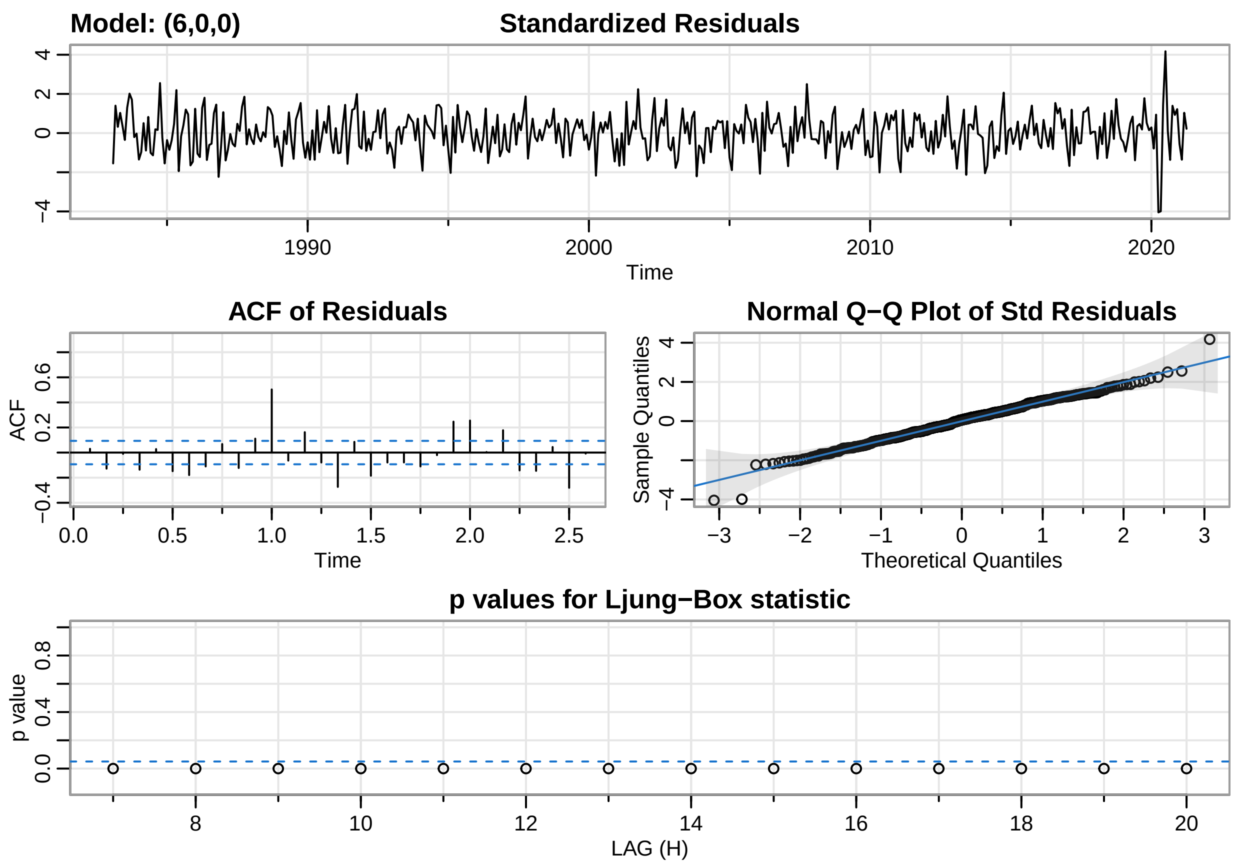



```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5  xmean
##       -0.8188  -0.4142  -0.0144  -0.2282  -0.0642  0.001
## s.e.   0.0466   0.0595   0.0626   0.0595   0.0468  0.001
##
## sigma^2 estimated as 0.002698:  log likelihood = 705.66,  aic = -1397.32
##
## $degrees_of_freedom
## [1] 453
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1    -0.8188 0.0466 -17.5849  0.0000
## ar2    -0.4142 0.0595  -6.9642  0.0000
## ar3    -0.0144 0.0626  -0.2298  0.8184
## ar4    -0.2282 0.0595  -3.8341  0.0001
## ar5    -0.0642 0.0468  -1.3706  0.1712
## xmean   0.0010 0.0010   1.0935  0.2747
##
```

```
## $AIC
## [1] -3.044275
##
## $AICc
## [1] -3.04387
##
## $BIC
## [1] -2.981305
```

```
sarima(diff(log(all_rmp)), p=6, d=0, q=0) # AR(6)
```

```
## initial  value -2.555352
## iter   2 value -2.707763
## iter   3 value -2.874728
## iter   4 value -2.890036
## iter   5 value -2.949570
## iter   6 value -2.974339
## iter   7 value -2.989233
## iter   8 value -2.994296
## iter   9 value -2.994595
## iter  10 value -2.995595
## iter  11 value -2.996427
## iter  12 value -2.996439
## iter  13 value -2.996441
## iter  14 value -2.996441
## iter  14 value -2.996441
## iter  14 value -2.996441
## final  value -2.996441
## converged
## initial  value -2.994739
## iter   2 value -2.994752
## iter   3 value -2.994756
## iter   4 value -2.994761
## iter   5 value -2.994768
## iter   6 value -2.994769
## iter   7 value -2.994769
## iter   7 value -2.994769
## iter   7 value -2.994769
## final  value -2.994769
## converged
```

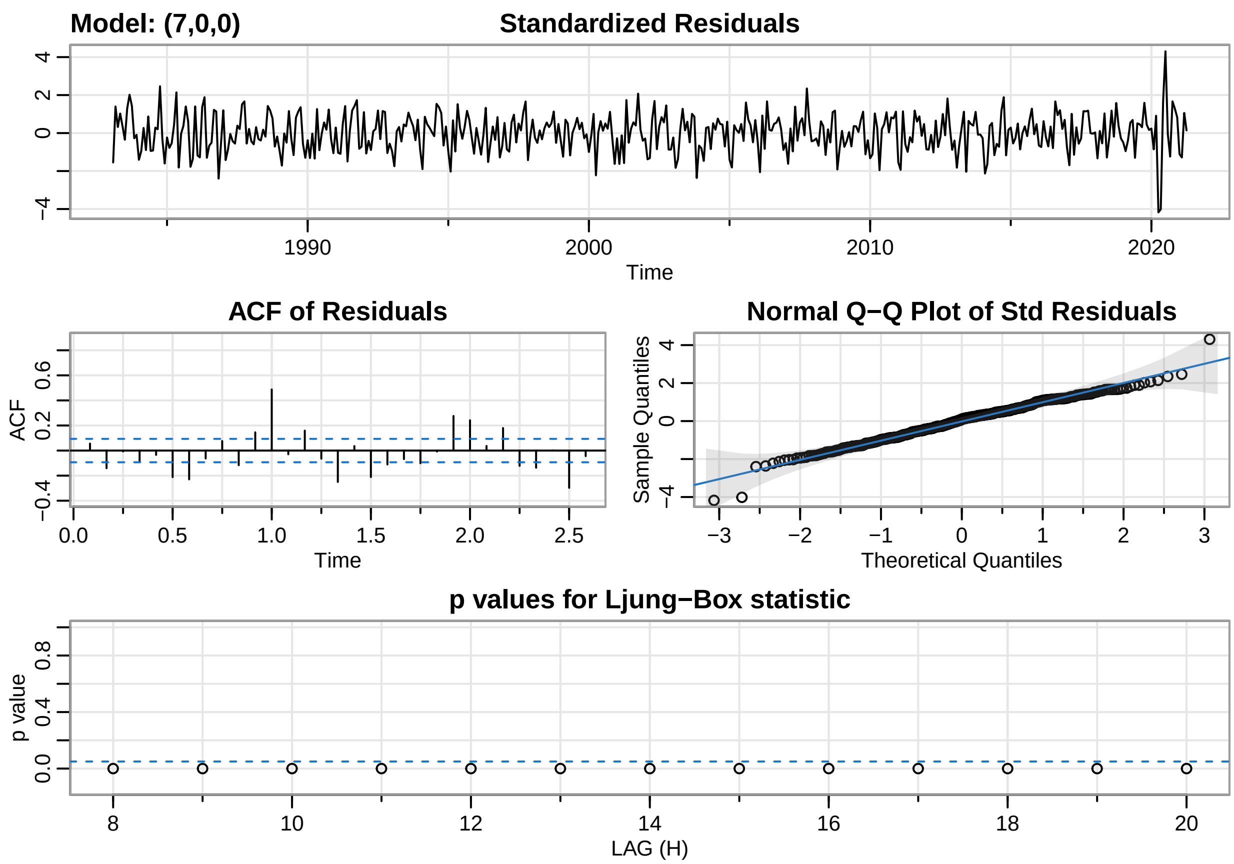



```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5      ar6  xmean
##       -0.8363  -0.4756  -0.0186  -0.3412  -0.2870  -0.2732  1e-03
## s.e.   0.0449   0.0581   0.0602   0.0602   0.0581   0.0451  7e-04
##
## sigma^2 estimated as 0.002495:  log likelihood = 723.31,  aic = -1430.61
##
## $degrees_of_freedom
## [1] 452
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1    -0.8363 0.0449 -18.6433  0.0000
## ar2    -0.4756 0.0581  -8.1889  0.0000
## ar3    -0.0186 0.0602  -0.3085  0.7578
## ar4    -0.3412 0.0602  -5.6701  0.0000
## ar5    -0.2870 0.0581  -4.9397  0.0000
## ar6    -0.2732 0.0451  -6.0628  0.0000
## xmean   0.0010 0.0007   1.4387  0.1509
```

```
##
## $AIC
## [1] -3.116802
##
## $AICc
## [1] -3.116261
##
## $BIC
## [1] -3.044836
```

```r
sarima(diff(log(all_rmp)), p=7, d=0, q=0) # AR(7)
```

```
## initial  value -2.557589
## iter   2 value -2.710865
## iter   3 value -2.900994
## iter   4 value -2.928243
## iter   5 value -2.931114
## iter   6 value -2.987438
## iter   7 value -2.995125
## iter   8 value -3.001252
## iter   9 value -3.003258
## iter  10 value -3.003472
## iter  11 value -3.003528
## iter  12 value -3.003547
## iter  13 value -3.003569
## iter  14 value -3.003570
## iter  15 value -3.003570
## iter  15 value -3.003570
## iter  15 value -3.003570
## final  value -3.003570
## converged
## initial  value -3.000871
## iter   2 value -3.000917
## iter   3 value -3.000971
## iter   4 value -3.000973
## iter   5 value -3.000979
## iter   6 value -3.000981
## iter   7 value -3.000981
## iter   7 value -3.000981
## iter   7 value -3.000981
## final  value -3.000981
## converged
```

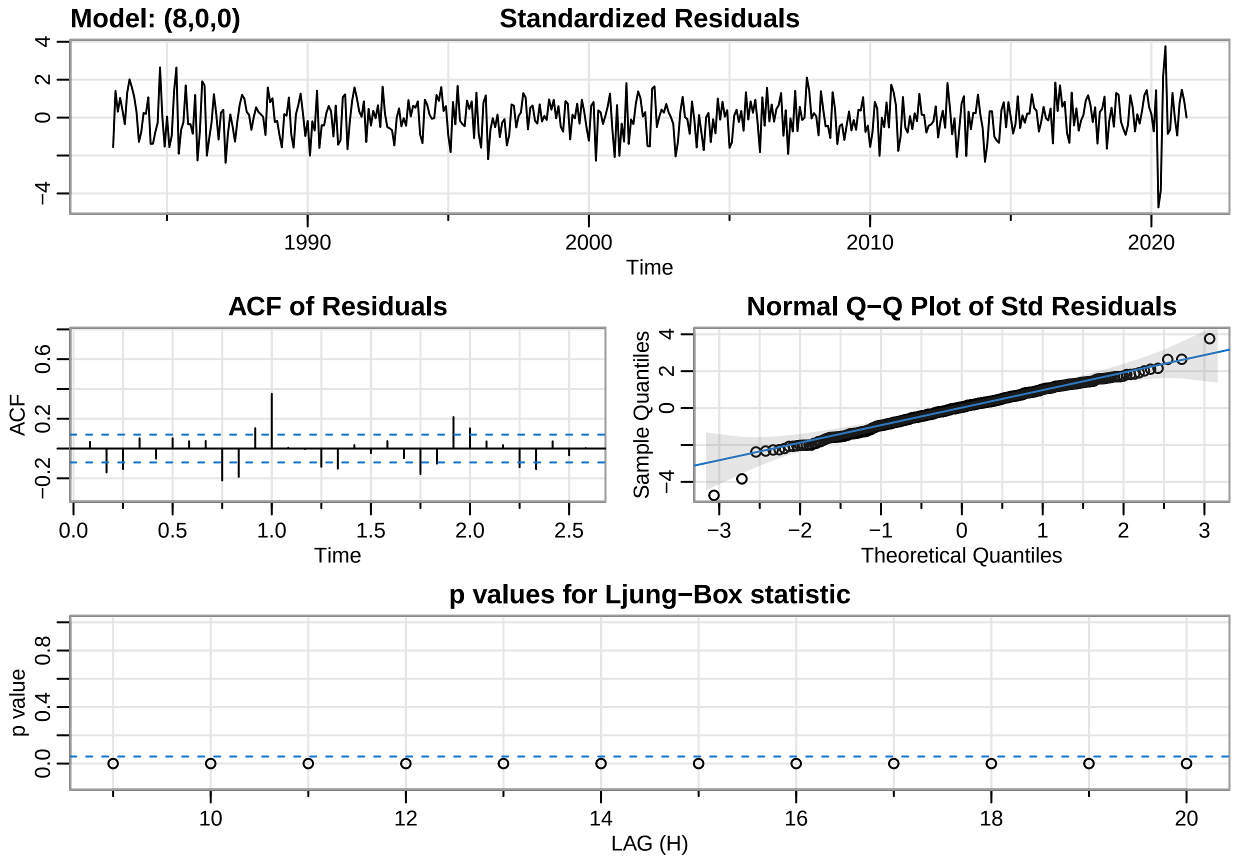



```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2     ar3      ar4      ar5      ar6     ar7   xmean
##       -0.8057  -0.4427  0.0195  -0.3388  -0.2339  -0.1802  0.1117  0.0011
## s.e.   0.0464   0.0593  0.0619   0.0598   0.0619   0.0593  0.0466  0.0008
##
## sigma^2 estimated as 0.002464:  log likelihood = 726.16,  aic = -1434.32
##
## $degrees_of_freedom
## [1] 451
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1    -0.8057 0.0464 -17.3794  0.0000
## ar2    -0.4427 0.0593  -7.4625  0.0000
## ar3     0.0195 0.0619   0.3150  0.7529
## ar4    -0.3388 0.0598  -5.6680  0.0000
## ar5    -0.2339 0.0619  -3.7807  0.0002
## ar6    -0.1802 0.0593  -3.0414  0.0025
## ar7     0.1117 0.0466   2.3965  0.0170
```

```
## xmean   0.0011 0.0008   1.2978  0.1950
##
## $AIC
## [1] -3.12487
##
## $AICc
## [1] -3.124173
##
## $BIC
## [1] -3.043908
```

```r
sarima(diff(log(all_rmp)), p=8, d=0, q=0) # AR(8)
```

```
## initial  value -2.556485
## iter   2 value -2.635598
## iter   3 value -3.026915
## iter   4 value -3.048763
## iter   5 value -3.072725
## iter   6 value -3.119837
## iter   7 value -3.128842
## iter   8 value -3.152746
## iter   9 value -3.153003
## iter  10 value -3.155731
## iter  11 value -3.155852
## iter  12 value -3.155868
## iter  13 value -3.155880
## iter  14 value -3.155881
## iter  14 value -3.155881
## iter  14 value -3.155881
## final  value -3.155881
## converged
## initial  value -3.147151
## iter   2 value -3.147172
## iter   3 value -3.147195
## iter   4 value -3.147210
## iter   5 value -3.147225
## iter   6 value -3.147230
## iter   7 value -3.147234
## iter   8 value -3.147236
## iter   9 value -3.147236
## iter  10 value -3.147236
## iter  10 value -3.147236
## iter  10 value -3.147236
## final  value -3.147236
## converged
```

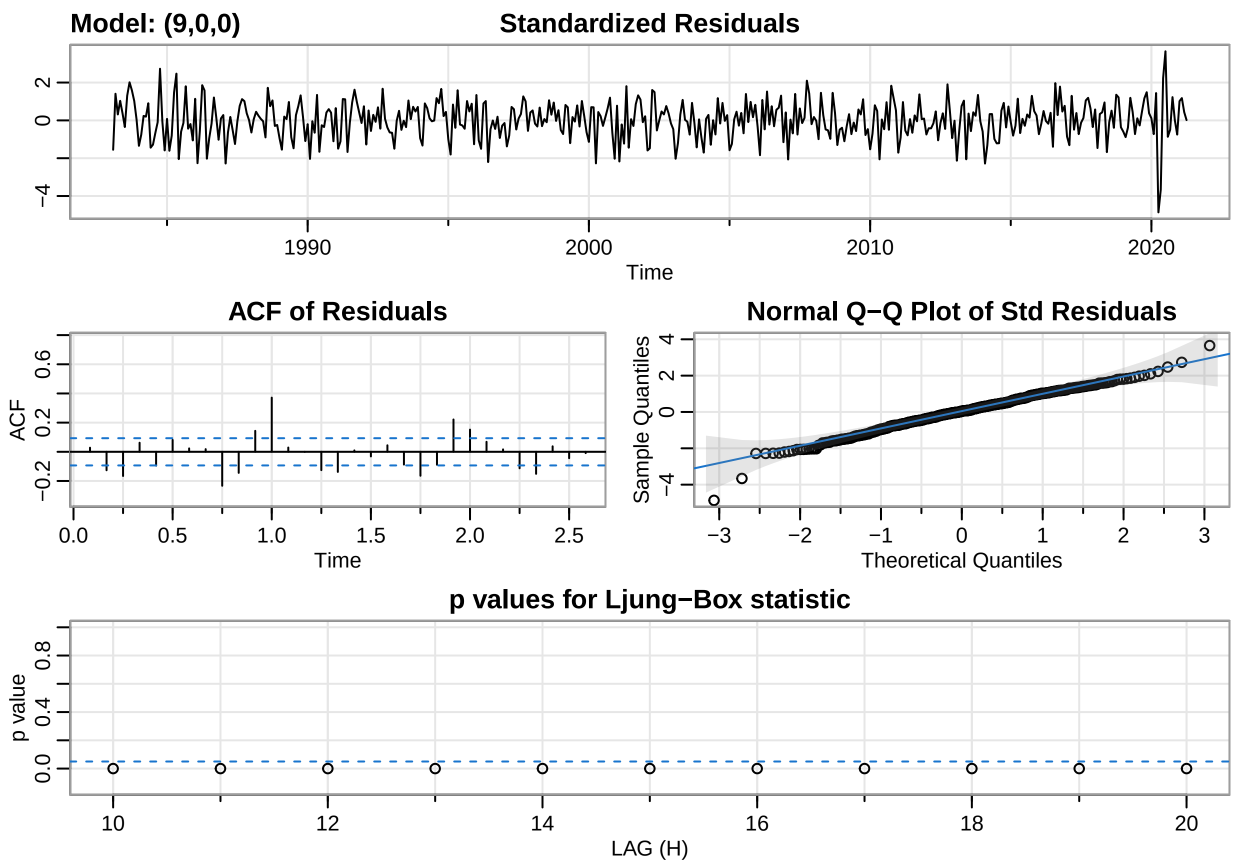



```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ar2      ar3      ar4      ar5      ar6      ar7      ar8
##      -0.7493  -0.5395  -0.1061  -0.5126  -0.2263  -0.4068  -0.2982  -0.5077
## s.e.   0.0401   0.0516   0.0542   0.0535   0.0533   0.0541   0.0517   0.0404
##        xmean
##        1e-03
## s.e.   5e-04
##
## sigma^2 estimated as 0.00183:  log likelihood = 793.29,  aic = -1566.58
##
## $degrees_of_freedom
## [1] 450
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1    -0.7493 0.0401 -18.6776  0.0000
## ar2    -0.5395 0.0516 -10.4501  0.0000
## ar3    -0.1061 0.0542  -1.9568  0.0510
## ar4    -0.5126 0.0535  -9.5856  0.0000
```

```
## ar5    -0.2263 0.0533  -4.2482  0.0000
## ar6    -0.4068 0.0541  -7.5257  0.0000
## ar7    -0.2982 0.0517  -5.7654  0.0000
## ar8    -0.5077 0.0404 -12.5733  0.0000
## xmean   0.0010 0.0005   2.1285  0.0338
##
## $AIC
## [1] -3.413023
##
## $AICc
## [1] -3.412149
##
## $BIC
## [1] -3.323065
```

```
sarima(diff(log(all_rmp)), p=9, d=0, q=0) # AR(9)
```

```
## initial  value -2.555453
## iter   2 value -2.723946
## iter   3 value -2.908137
## iter   4 value -2.935821
## iter   5 value -3.064612
## iter   6 value -3.089706
## iter   7 value -3.116873
## iter   8 value -3.134916
## iter   9 value -3.148688
## iter  10 value -3.160605
## iter  11 value -3.161086
## iter  12 value -3.161264
## iter  13 value -3.161524
## iter  14 value -3.161547
## iter  15 value -3.161548
## iter  16 value -3.161548
## iter  16 value -3.161548
## iter  16 value -3.161548
## final  value -3.161548
## converged
## initial  value -3.150780
## iter   2 value -3.150828
## iter   3 value -3.150899
## iter   4 value -3.150930
## iter   5 value -3.150963
## iter   6 value -3.150977
## iter   7 value -3.150982
## iter   8 value -3.150983
## iter   9 value -3.150984
## iter  10 value -3.150984
## iter  11 value -3.150984
## iter  11 value -3.150984
## iter  11 value -3.150984
## final  value -3.150984
## converged
```

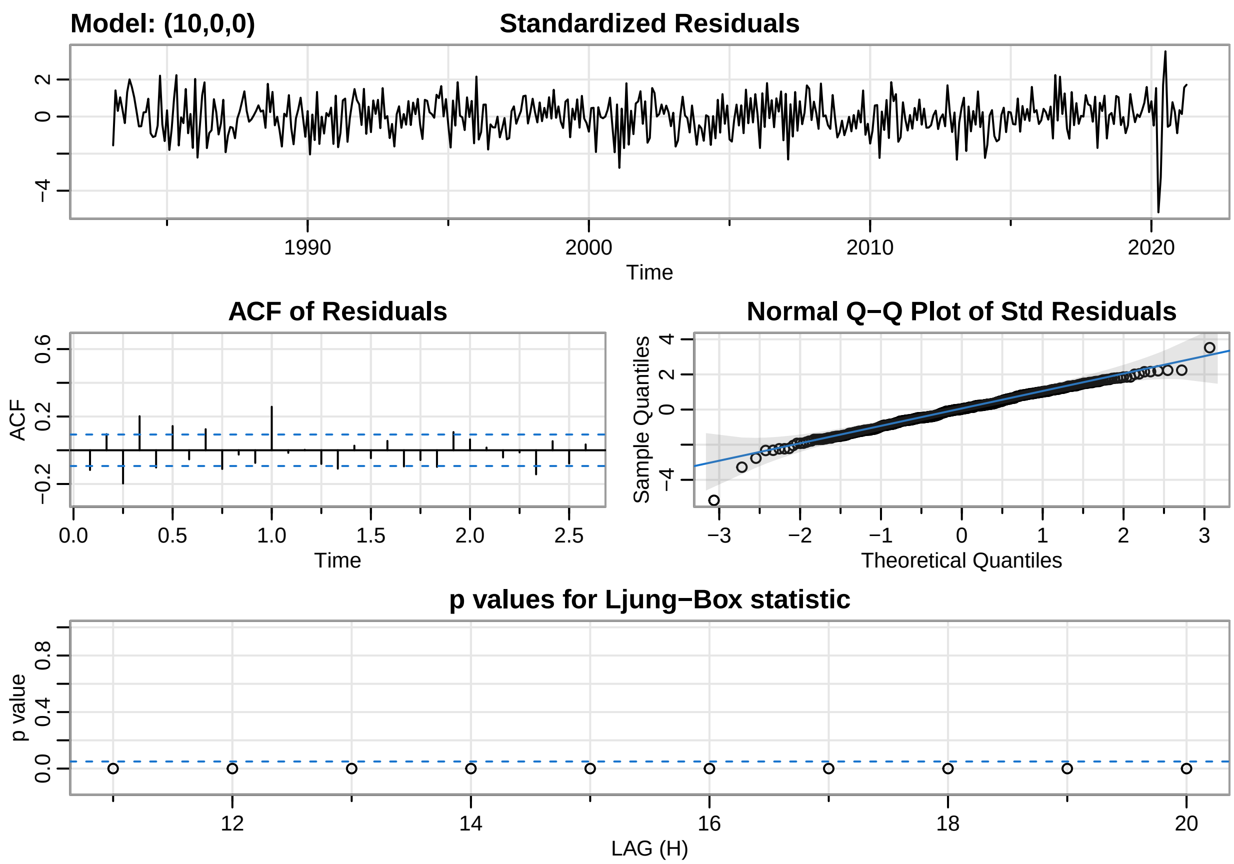



```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5      ar6      ar7      ar8
##       -0.7049  -0.5128  -0.0693  -0.4914  -0.1812  -0.3974  -0.2505  -0.4416
## s.e.   0.0465   0.0534   0.0576   0.0545   0.0584   0.0541   0.0576   0.0537
##          ar9   xmean
##       0.0874   1e-03
## s.e.  0.0470   5e-04
##
## sigma^2 estimated as 0.001816:  log likelihood = 795.01,  aic = -1568.02
##
## $degrees_of_freedom
## [1] 449
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1    -0.7049 0.0465 -15.1519  0.0000
## ar2    -0.5128 0.0534  -9.6046  0.0000
## ar3    -0.0693 0.0576  -1.2030  0.2296
## ar4    -0.4914 0.0545  -9.0188  0.0000
```

```
## ar5    -0.1812 0.0584  -3.1045  0.0020
## ar6    -0.3974 0.0541  -7.3430  0.0000
## ar7    -0.2505 0.0576  -4.3485  0.0000
## ar8    -0.4416 0.0537  -8.2167  0.0000
## ar9     0.0874 0.0470   1.8588  0.0637
## xmean   0.0010 0.0005   1.9676  0.0497
##
## $AIC
## [1] -3.416161
##
## $AICc
## [1] -3.415091
##
## $BIC
## [1] -3.317208
```

```
sarima(diff(log(all_rmp)), p=10, d=0, q=0) # AR(10)
```

```
## initial  value -2.554396
## iter   2 value -2.731118
## iter   3 value -2.923997
## iter   4 value -3.013507
## iter   5 value -3.093885
## iter   6 value -3.114130
## iter   7 value -3.131892
## iter   8 value -3.150052
## iter   9 value -3.177600
## iter  10 value -3.200871
## iter  11 value -3.204509
## iter  12 value -3.215594
## iter  13 value -3.216370
## iter  14 value -3.216414
## iter  15 value -3.216425
## iter  16 value -3.216426
## iter  17 value -3.216426
## iter  17 value -3.216426
## iter  17 value -3.216426
## final  value -3.216426
## converged
## initial  value -3.204221
## iter   2 value -3.204314
## iter   3 value -3.204386
## iter   4 value -3.204410
## iter   5 value -3.204420
## iter   6 value -3.204463
## iter   7 value -3.204471
## iter   8 value -3.204479
## iter   9 value -3.204481
## iter  10 value -3.204482
## iter  10 value -3.204482
## iter  10 value -3.204482
## final  value -3.204482
## converged
```



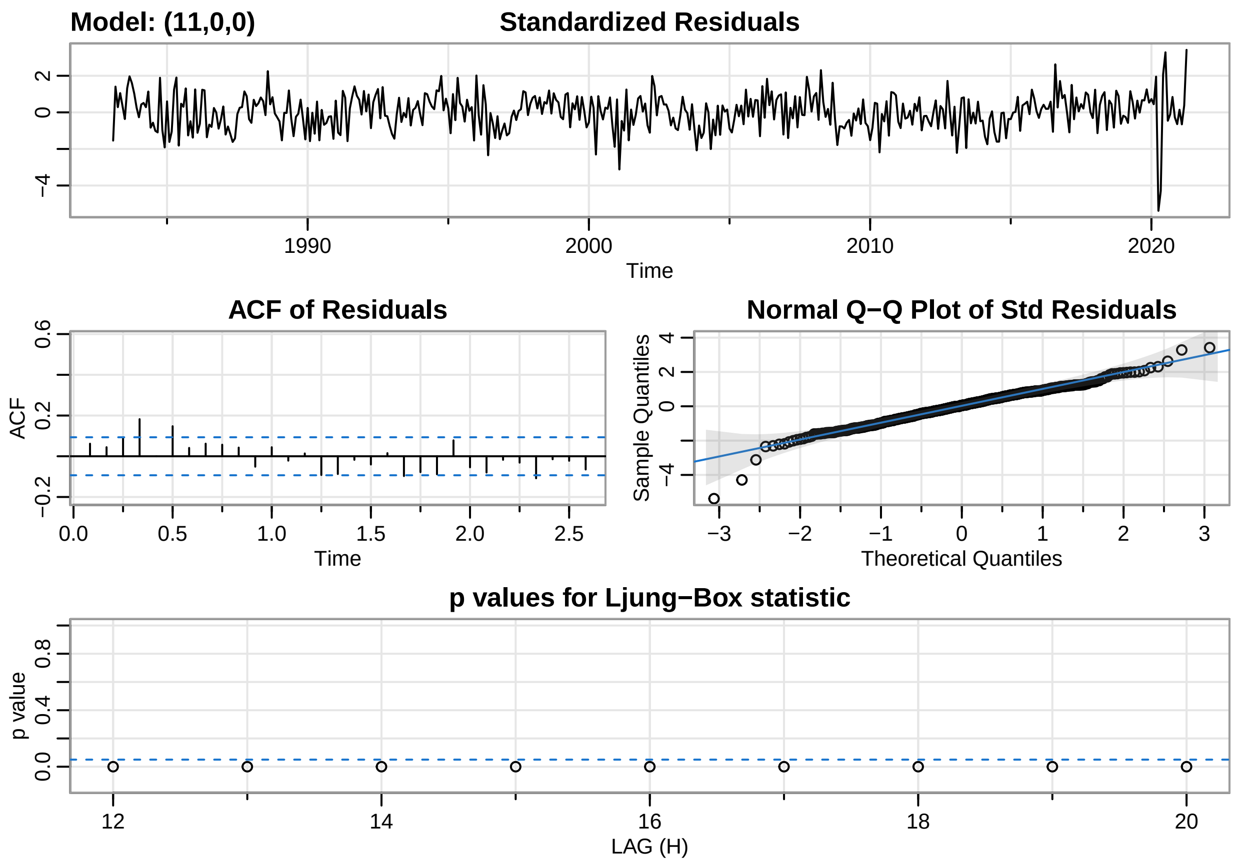

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5      ar6      ar7      ar8
##       -0.6775  -0.6569  -0.1541  -0.6257  -0.2463  -0.5586  -0.2767  -0.6114
## s.e.   0.0442   0.0543   0.0557   0.0548   0.0561   0.0559   0.0546   0.0559
##           ar9     ar10  xmean
##       -0.1433  -0.3220  1e-03
## s.e.   0.0548   0.0446  4e-04
##
## sigma^2 estimated as 0.001628:  log likelihood = 819.56,  aic = -1615.13
##
## $degrees_of_freedom
## [1] 448
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1    -0.6775 0.0442 -15.3319  0.0000
## ar2    -0.6569 0.0543 -12.0886  0.0000
## ar3    -0.1541 0.0557  -2.7637  0.0059
## ar4    -0.6257 0.0548 -11.4118  0.0000
```

```
## ar5    -0.2463 0.0561  -4.3894  0.0000
## ar6    -0.5586 0.0559  -9.9995  0.0000
## ar7    -0.2767 0.0546  -5.0648  0.0000
## ar8    -0.6114 0.0559 -10.9280  0.0000
## ar9    -0.1433 0.0548  -2.6151  0.0092
## ar10   -0.3220 0.0446  -7.2194  0.0000
## xmean   0.0010 0.0004   2.6595  0.0081
##
## $AIC
## [1] -3.518799
##
## $AICc
## [1] -3.517512
##
## $BIC
## [1] -3.41085
```

```
sarima(diff(log(all_rmp)), p=11, d=0, q=0) # AR(11)
```

```
## initial  value -2.553408
## iter   2 value -2.725246
## iter   3 value -2.876035
## iter   4 value -2.980879
## iter   5 value -3.108342
## iter   6 value -3.131789
## iter   7 value -3.168506
## iter   8 value -3.179240
## iter   9 value -3.260190
## iter  10 value -3.289425
## iter  11 value -3.289929
## iter  12 value -3.295660
## iter  13 value -3.296360
## iter  14 value -3.296822
## iter  15 value -3.296825
## iter  16 value -3.296825
## iter  16 value -3.296825
## iter  16 value -3.296825
## final  value -3.296825
## converged
## initial  value -3.283429
## iter   2 value -3.283480
## iter   3 value -3.283531
## iter   4 value -3.283590
## iter   5 value -3.283709
## iter   6 value -3.283743
## iter   7 value -3.283772
## iter   8 value -3.283806
## iter   9 value -3.283814
## iter  10 value -3.283824
## iter  11 value -3.283827
## iter  12 value -3.283827
## iter  13 value -3.283828
## iter  13 value -3.283828
## iter  13 value -3.283828
## final  value -3.283828
```

```
## converged
```



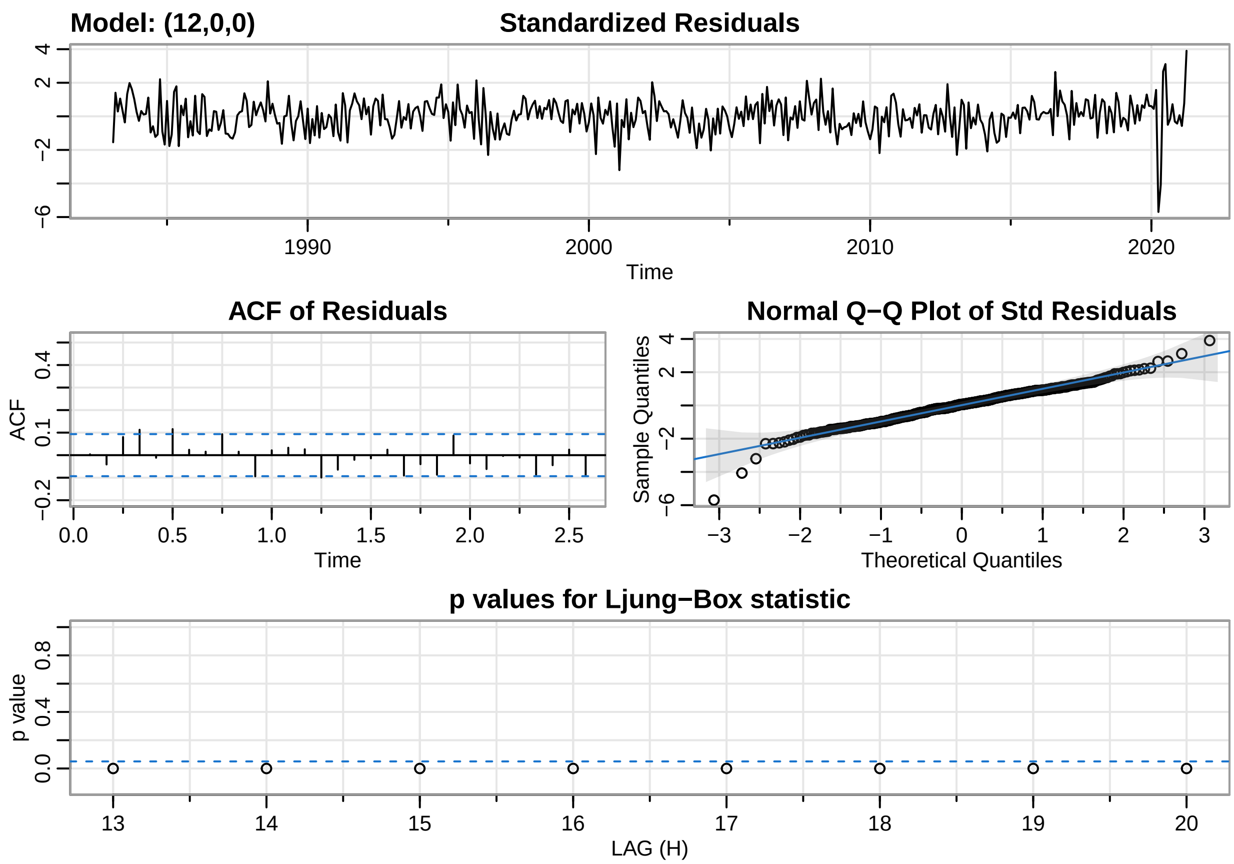

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5      ar6      ar7      ar8
##       -0.8063  -0.7182  -0.3976  -0.7364  -0.4725  -0.6627  -0.5312  -0.6785
## s.e.   0.0431   0.0504   0.0581   0.0519   0.0576   0.0528   0.0577   0.0519
##           ar9     ar10     ar11  xmean
##       -0.4100  -0.5967  -0.3994  9e-04
## s.e.   0.0585   0.0511   0.0447  2e-04
##
## sigma^2 estimated as 0.001383:  log likelihood = 855.98,  aic = -1685.97
##
## $degrees_of_freedom
## [1] 447
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1    -0.8063 0.0431 -18.6917   0e+00
## ar2    -0.7182 0.0504 -14.2452   0e+00
```

```
## ar3    -0.3976 0.0581  -6.8440   0e+00
## ar4    -0.7364 0.0519 -14.1821   0e+00
## ar5    -0.4725 0.0576  -8.1995   0e+00
## ar6    -0.6627 0.0528 -12.5596   0e+00
## ar7    -0.5312 0.0577  -9.2054   0e+00
## ar8    -0.6785 0.0519 -13.0767   0e+00
## ar9    -0.4100 0.0585  -7.0098   0e+00
## ar10   -0.5967 0.0511 -11.6834   0e+00
## ar11   -0.3994 0.0447  -8.9353   0e+00
## xmean   0.0009 0.0002   3.8108   2e-04
##
## $AIC
## [1] -3.673134
##
## $AICc
## [1] -3.67161
##
## $BIC
## [1] -3.556189
```

```
sarima(diff(log(all_rmp)), p=12, d=0, q=0) # AR(12)
```

```
## initial  value -2.552728
## iter   2 value -2.845003
## iter   3 value -3.109176
## iter   4 value -3.155985
## iter   5 value -3.191147
## iter   6 value -3.220362
## iter   7 value -3.231858
## iter   8 value -3.242799
## iter   9 value -3.255816
## iter  10 value -3.271244
## iter  11 value -3.278037
## iter  12 value -3.298541
## iter  13 value -3.301628
## iter  14 value -3.309140
## iter  15 value -3.309682
## iter  16 value -3.309709
## iter  17 value -3.309710
## iter  17 value -3.309710
## iter  17 value -3.309710
## final  value -3.309710
## converged
## initial  value -3.296998
## iter   2 value -3.297064
## iter   3 value -3.297092
## iter   4 value -3.297143
## iter   5 value -3.297283
## iter   6 value -3.297289
## iter   7 value -3.297307
## iter   8 value -3.297325
## iter   9 value -3.297340
## iter  10 value -3.297346
## iter  11 value -3.297351
## iter  12 value -3.297356
```

```
## iter  13 value -3.297357
## iter  14 value -3.297357
## iter  14 value -3.297357
## iter  14 value -3.297357
## final  value -3.297357
## converged
```



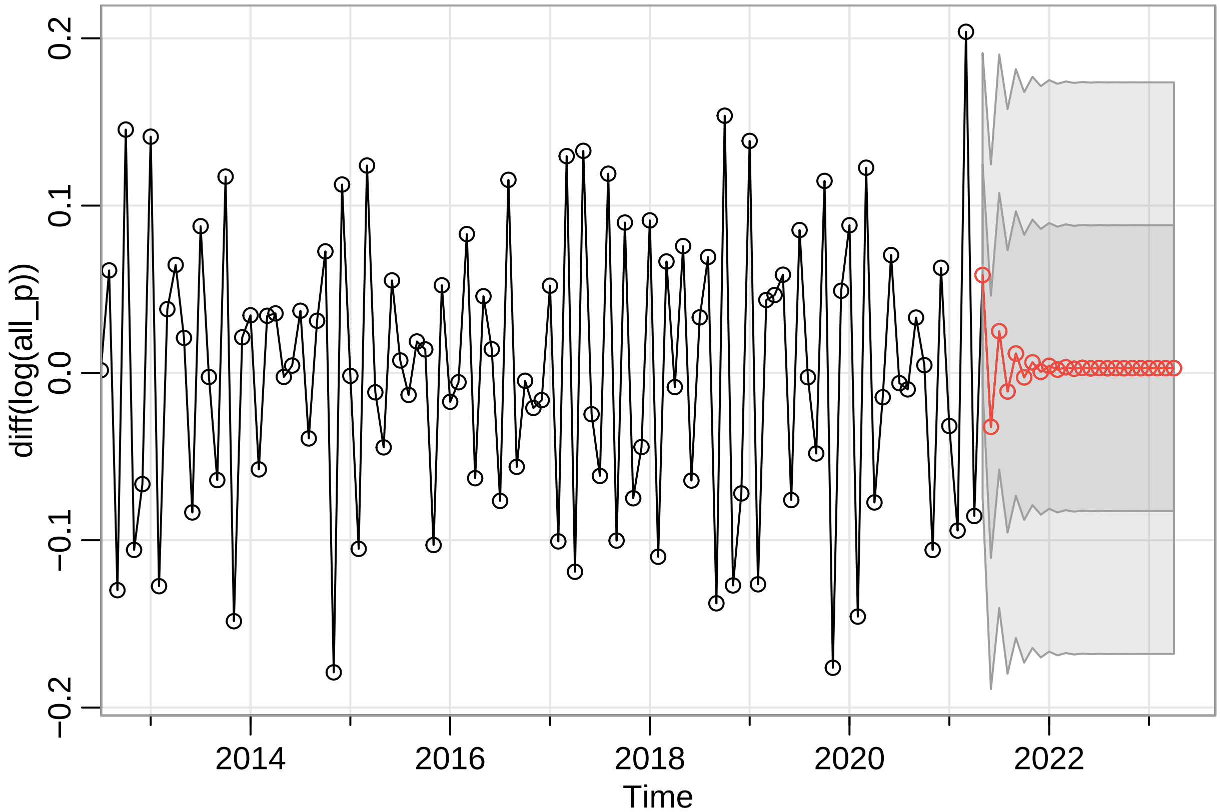

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5      ar6      ar7      ar8
##       -0.7374  -0.6144  -0.3227  -0.6185  -0.3791  -0.5454  -0.4423  -0.5468
## s.e.   0.0467   0.0577   0.0611   0.0611   0.0626   0.0616   0.0622   0.0633
##           ar9     ar10     ar11    ar12  xmean
##       -0.3348  -0.4641  -0.2484  0.1751  9e-04
## s.e.   0.0615   0.0628   0.0613  0.0493  3e-04
##
## sigma^2 estimated as 0.001345:  log likelihood = 862.19,  aic = -1696.39
##
## $degrees_of_freedom
## [1] 446
```

```
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1    -0.7374 0.0467 -15.7764  0.0000
## ar2    -0.6144 0.0577 -10.6527  0.0000
## ar3    -0.3227 0.0611  -5.2832  0.0000
## ar4    -0.6185 0.0611 -10.1308  0.0000
## ar5    -0.3791 0.0626  -6.0518  0.0000
## ar6    -0.5454 0.0616  -8.8527  0.0000
## ar7    -0.4423 0.0622  -7.1073  0.0000
## ar8    -0.5468 0.0633  -8.6393  0.0000
## ar9    -0.3348 0.0615  -5.4402  0.0000
## ar10   -0.4641 0.0628  -7.3872  0.0000
## ar11   -0.2484 0.0613  -4.0488  0.0001
## ar12    0.1751 0.0493   3.5515  0.0004
## xmean   0.0009 0.0003   3.2810  0.0011
##
## $AIC
## [1] -3.695835
##
## $AICc
## [1] -3.694053
##
## $BIC
## [1] -3.569895
```

Autocorrelation displayed by ACF of Residuals is lowest for AR(12).

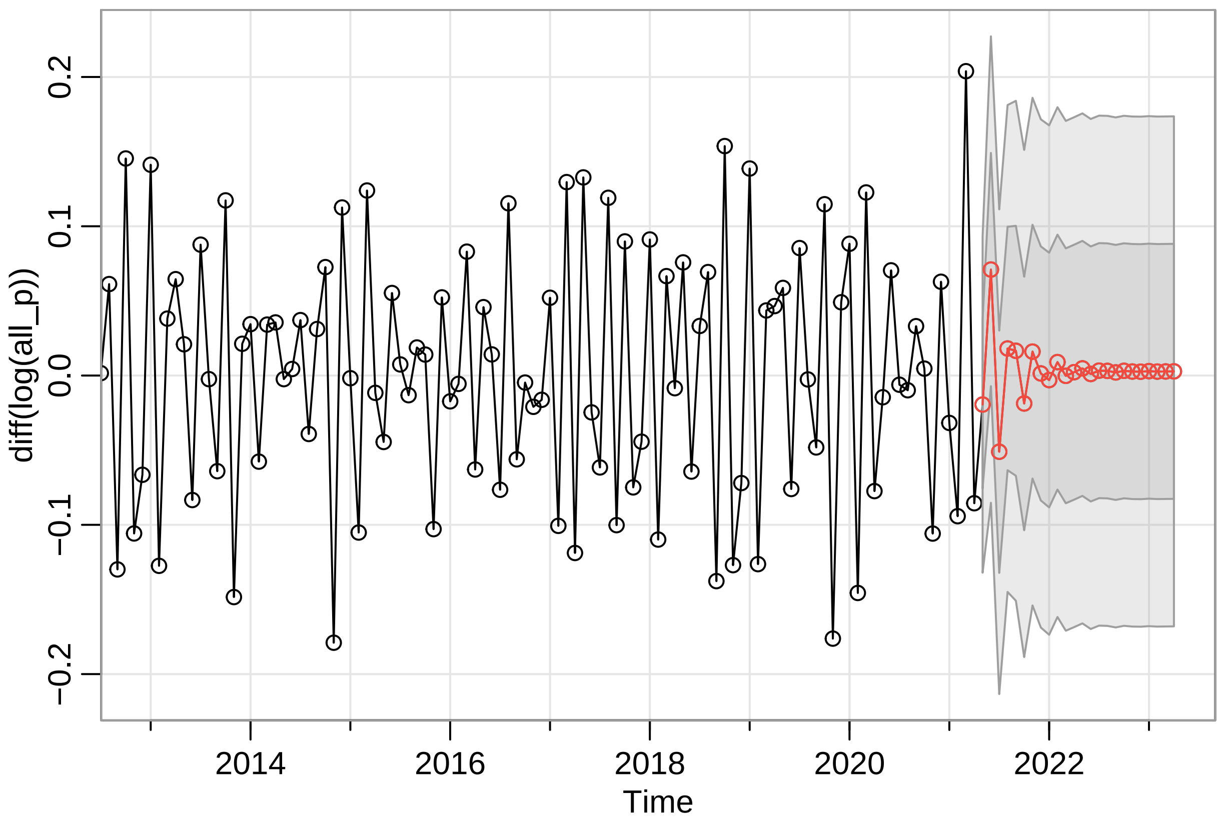
24-Month Forecast: All Poultry

```
sarima.for(diff(log(all_p)), n.ahead=24, p=1, d=0, q=0) # AR(1)
```



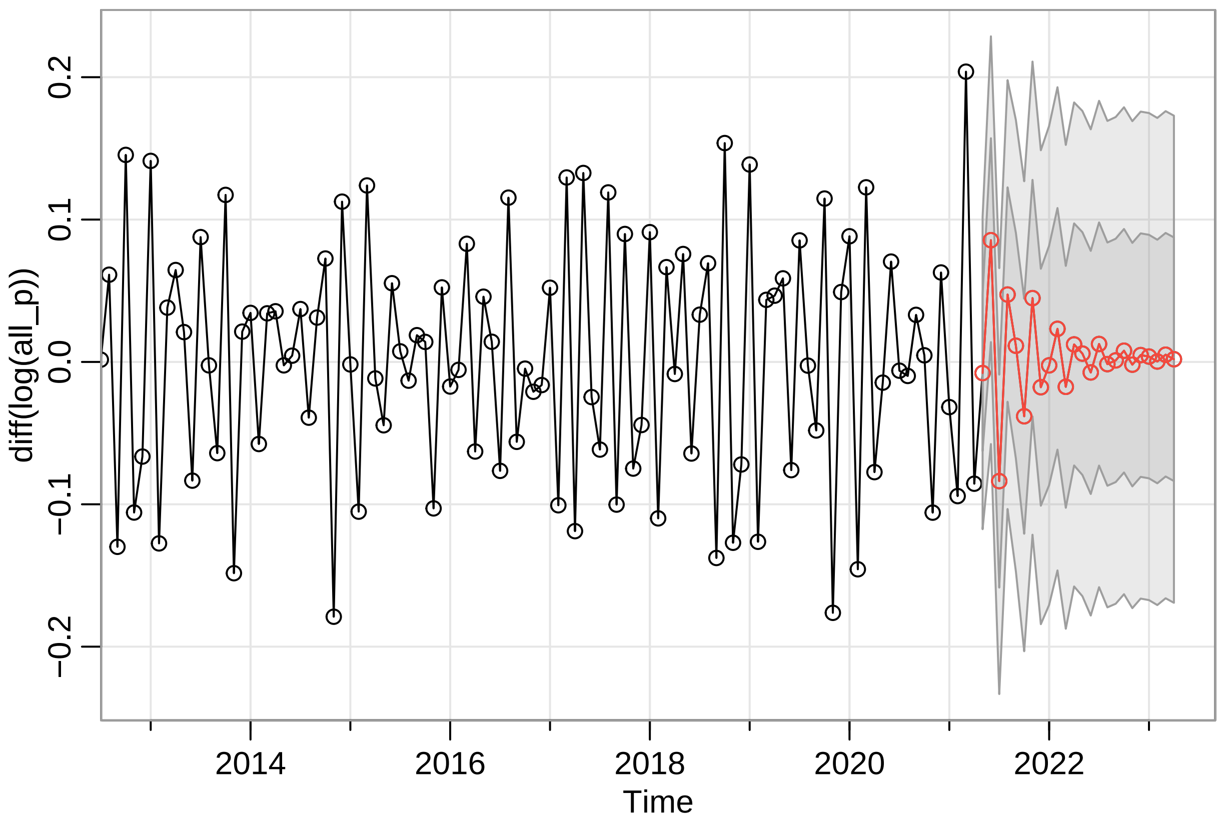

```
## $pred
##                Jan           Feb           Mar           Apr           May
## 2021                                                           0.0585146289
## 2022  0.0042260413  0.0019775394  0.0033938524  0.0025017284  0.0030636700
## 2023  0.0028518825  0.0028431108  0.0028486360  0.0028451557
##                Jun           Jul           Aug           Sep           Oct
## 2021 -0.0322183991  0.0249335950 -0.0110659798  0.0116098587 -0.0026734669
## 2022  0.0027097076  0.0029326656  0.0027922262  0.0028806878  0.0028249666
## 2023
##                Nov           Dec
## 2021  0.0063234833  0.0006563778
## 2022  0.0028600649  0.0028379568
## 2023
##
## $se
##             Jan        Feb        Mar        Apr        May        Jun
## 2021                                             0.06633516 0.07839805
## 2022 0.08539789 0.08540417 0.08540666 0.08540765 0.08540804 0.08540820
## 2023 0.08540830 0.08540830 0.08540830 0.08540830
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.08269803 0.08434339 0.08498738 0.08524154 0.08534217 0.08538207
## 2022 0.08540826 0.08540828 0.08540829 0.08540830 0.08540830 0.08540830
## 2023
```

```
sarima.for(diff(log(all_p)), n.ahead=24, p=2, d=0, q=0) # AR(2)
```



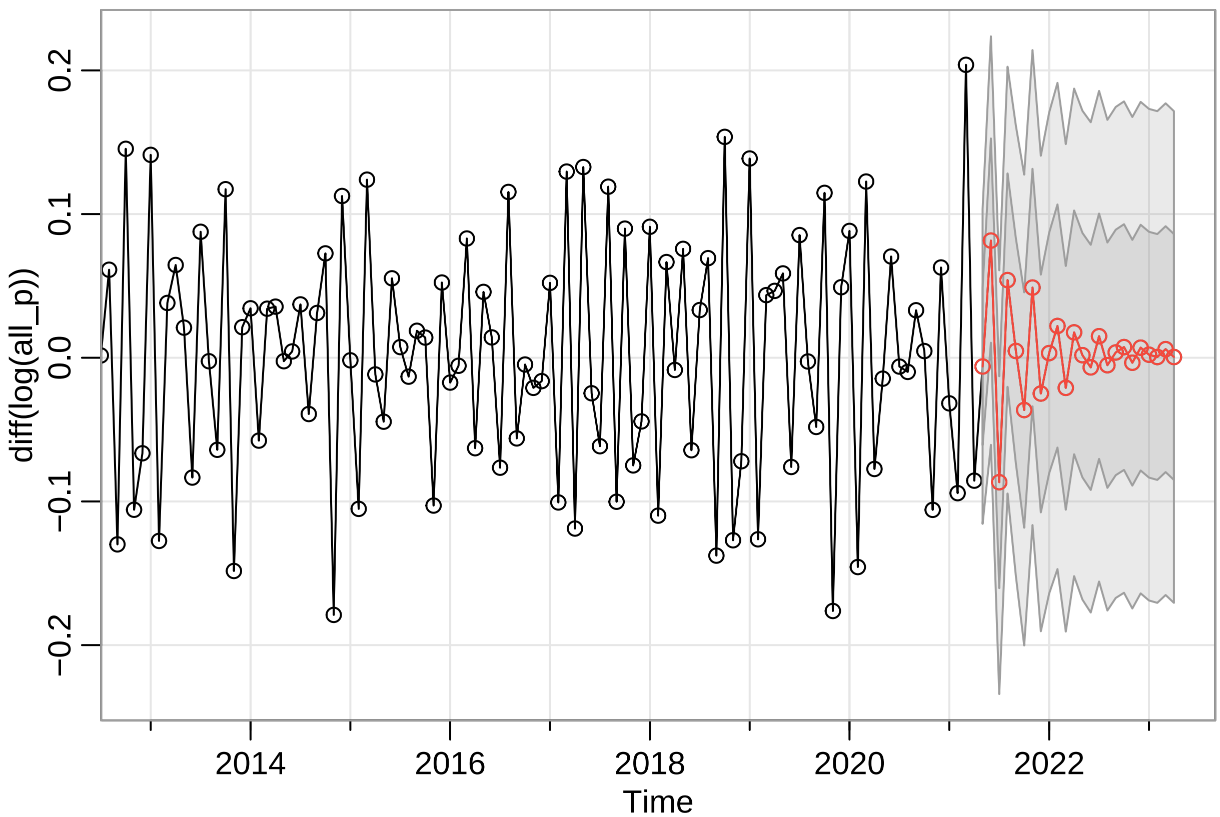

```
## $pred
##               Jan           Feb           Mar           Apr           May
## 2021                                                          -0.0194133021
## 2022 -0.0030306447  0.0090130206 -0.0001900122  0.0022461610  0.0048005932
## 2023  0.0029454318  0.0026538469  0.0027411526  0.0028123639
##               Jun           Jul           Aug           Sep           Oct
## 2021  0.0710034781 -0.0510359426  0.0181257873  0.0165885005 -0.0187215490
## 2022  0.0010505204  0.0032948551  0.0031331796  0.0020947563  0.0031784587
## 2023
##               Nov           Dec
## 2021  0.0160216971  0.0014220946
## 2022  0.0026895840  0.0025828700
## 2023
##
## $se
##             Jan        Feb        Mar        Apr        May        Jun
## 2021                                             0.05633751 0.07812691
## 2022 0.08530842 0.08537793 0.08537812 0.08539427 0.08541101 0.08541422
## 2023 0.08541799 0.08541799 0.08541800 0.08541802
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.08117629 0.08153266 0.08372619 0.08492896 0.08500342 0.08510505
## 2022 0.08541442 0.08541635 0.08541755 0.08541765 0.08541773 0.08541792
## 2023
```

```r
sarima.for(diff(log(all_p)), n.ahead=24, p=3, d=0, q=0) # AR(3)
```



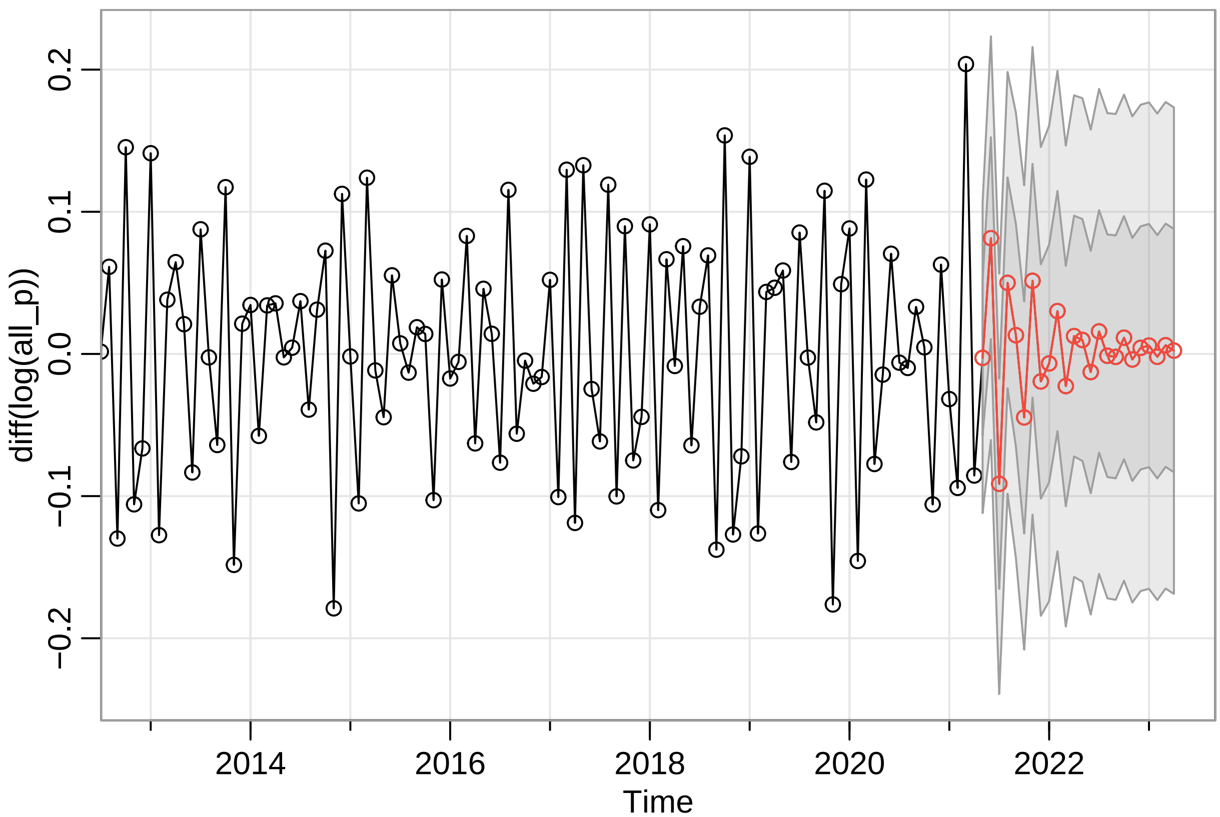

```
## $pred
##                Jan           Feb           Mar           Apr           May
## 2021                                                      -0.0078084452
## 2022 -0.0024051788  0.0232612439 -0.0174911309  0.0122996707  0.0058583738
## 2023  0.0037701761  0.0002931854  0.0050807402  0.0019029355
##                Jun           Jul           Aug           Sep           Oct
## 2021  0.0855285230 -0.0837204202  0.0473295431  0.0113693012 -0.0381578018
## 2022 -0.0073810006  0.0126053535 -0.0015506936  0.0010765388  0.0078750184
## 2023
##                Nov           Dec
## 2021  0.0448139667 -0.0177674828
## 2022 -0.0019135260  0.0048003181
## 2023
##
## $se
##             Jan        Feb        Mar        Apr        May        Jun
## 2021                                            0.05484052 0.07159989
## 2022 0.08412578 0.08484959 0.08496211 0.08500182 0.08521961 0.08538513
## 2023 0.08553117 0.08553989 0.08554079 0.08554169
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.07479518 0.07530891 0.07925725 0.08252804 0.08308675 0.08322314
## 2022 0.08540790 0.08541934 0.08547242 0.08551043 0.08551499 0.08551824
## 2023
```

```
sarima.for(diff(log(all_p)), n.ahead=24, p=4, d=0, q=0) # AR(4)
```



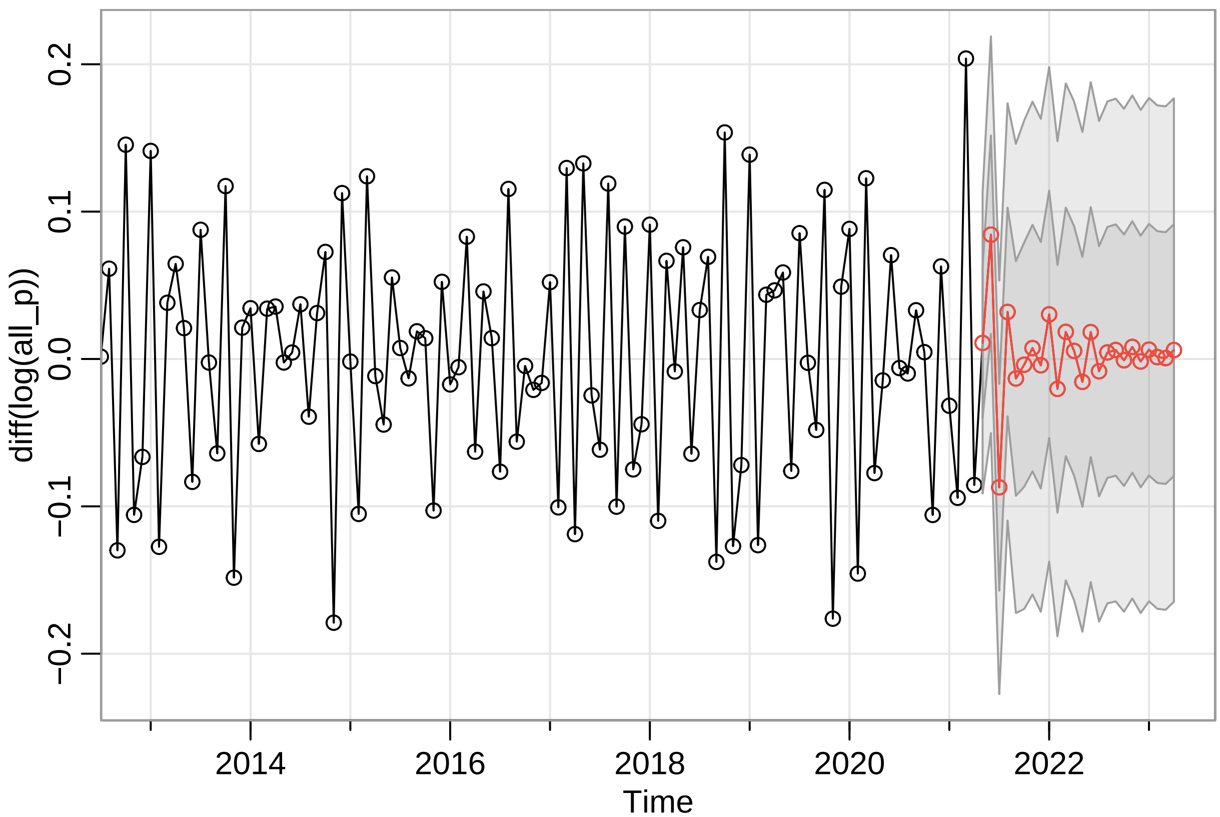

```
## $pred
##                Jan           Feb           Mar           Apr           May
## 2021                                                          -0.0060740795
## 2022  0.0032434937  0.0221132779 -0.0209417057  0.0176658784  0.0018104493
## 2023  0.0021351524  0.0005308815  0.0059985175  0.0005269905
##                Jun           Jul           Aug           Sep           Oct
## 2021  0.0815660575 -0.0866054885  0.0540532540  0.0047216201 -0.0363796448
## 2022 -0.0067023885  0.0149938734 -0.0051763880  0.0036797783  0.0074336073
## 2023
##                Nov           Dec
## 2021  0.0488559544 -0.0248856896
## 2022 -0.0034723136  0.0070436757
## 2023
##
## $se
##             Jan        Feb        Mar        Apr        May        Jun
## 2021                                            0.05476439 0.07108179
## 2022 0.08368096 0.08459941 0.08481615 0.08483097 0.08506248 0.08530733
## 2023 0.08553066 0.08554821 0.08555385 0.08555387
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.07373391 0.07425557 0.07835600 0.08191404 0.08264780 0.08273668
## 2022 0.08537195 0.08537416 0.08543147 0.08549708 0.08551624 0.08551650
## 2023
```

```
sarima.for(diff(log(all_p)), n.ahead=24, p=5, d=0, q=0) # AR(5)
```



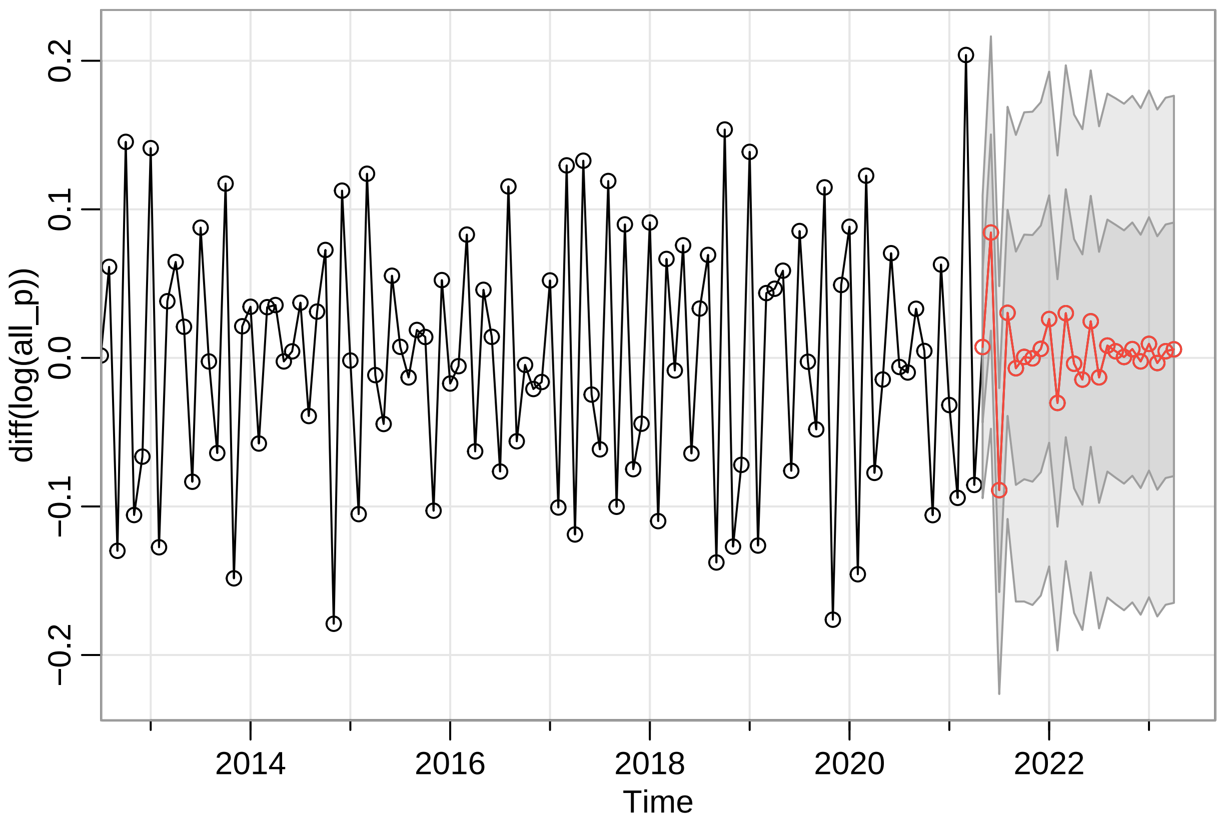

```
## $pred
##               Jan          Feb          Mar          Apr          May
## 2021                                                       -0.002760065
## 2022 -0.006656821  0.030137196 -0.022605163  0.012558853  0.009827633
## 2023  0.005886360 -0.002007599  0.006111314  0.002334095
##               Jun          Jul          Aug          Sep          Oct
## 2021  0.081440996 -0.091351494  0.050011964  0.013187667 -0.044659582
## 2022 -0.012732326  0.015861621 -0.001286328 -0.002054341  0.011427780
## 2023
##               Nov          Dec
## 2021  0.051462221 -0.019351699
## 2022 -0.003896938  0.004272817
## 2023
##
## $se
##             Jan        Feb        Mar        Apr        May        Jun
## 2021                                            0.05458912 0.07099859
## 2022 0.08360273 0.08448324 0.08458353 0.08468057 0.08504063 0.08526842
## 2023 0.08554409 0.08555829 0.08555835 0.08556453
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.07395690 0.07417339 0.07826338 0.08167439 0.08223169 0.08242721
## 2022 0.08528358 0.08532545 0.08543527 0.08549302 0.08549464 0.08551123
## 2023
```

```
sarima.for(diff(log(all_p)), n.ahead=24, p=6, d=0, q=0) # AR(6)
```



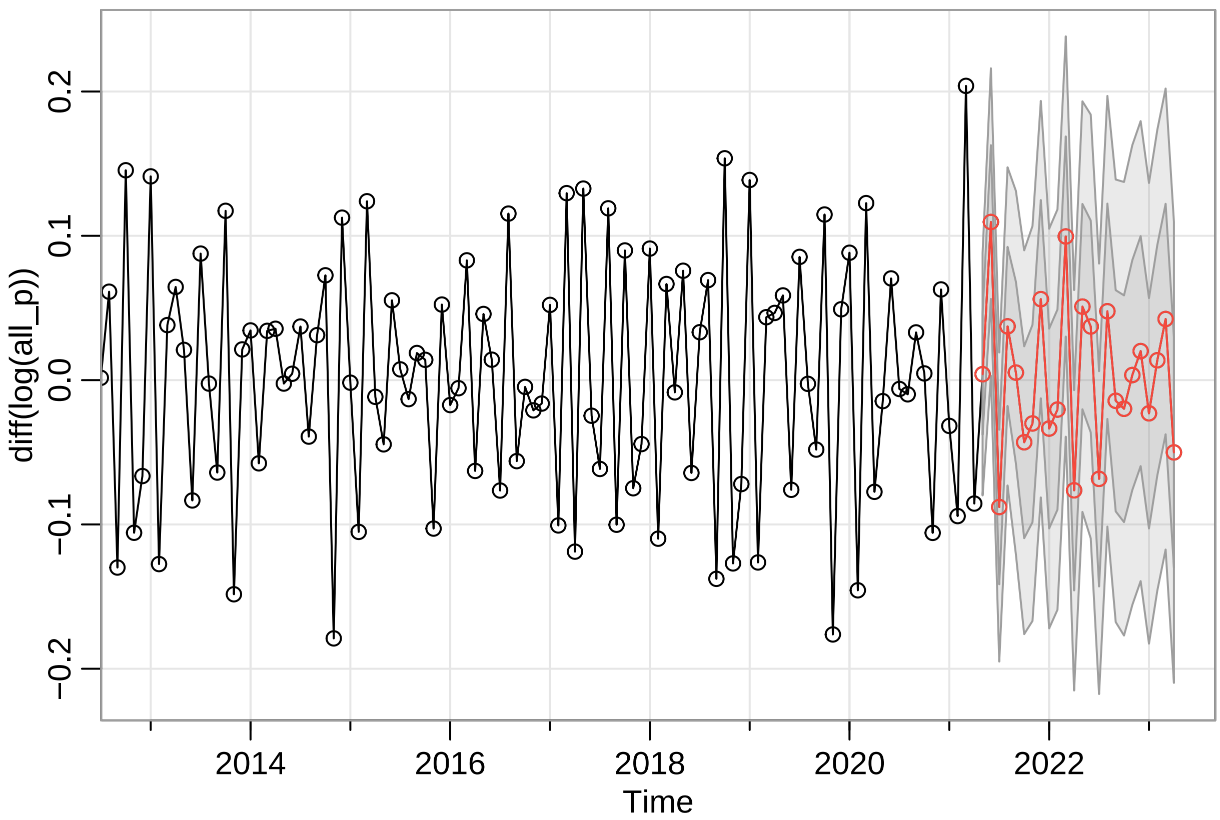

```
## $pred
##               Jan           Feb           Mar           Apr           May
## 2021                                                         0.0108717832
## 2022  0.0302593103 -0.0202750207  0.0183920951  0.0055006648 -0.0155267111
## 2023  0.0063428329  0.0012964025  0.0006130902  0.0060622542
##               Jun           Jul           Aug           Sep           Oct
## 2021  0.0843498218 -0.0870917533  0.0319328092 -0.0132388756 -0.0038820219
## 2022  0.0182395138 -0.0083300868  0.0044387641  0.0060986442 -0.0008569296
## 2023
##               Nov           Dec
## 2021  0.0074004062 -0.0042860358
## 2022  0.0081752792 -0.0017068179
## 2023
##
## $se
##             Jan        Feb        Mar        Apr        May        Jun
## 2021                                            0.05106625 0.06731524
## 2022 0.08393167 0.08399891 0.08429797 0.08460907 0.08478772 0.08482183
## 2023 0.08538268 0.08540123 0.08540835 0.08541124
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.07015801 0.07079276 0.07959489 0.08287554 0.08362732 0.08362847
## 2022 0.08493279 0.08514139 0.08529839 0.08533415 0.08533934 0.08535588
## 2023
```

```
sarima.for(diff(log(all_p)), n.ahead=24, p=7, d=0, q=0) # AR(7)
```



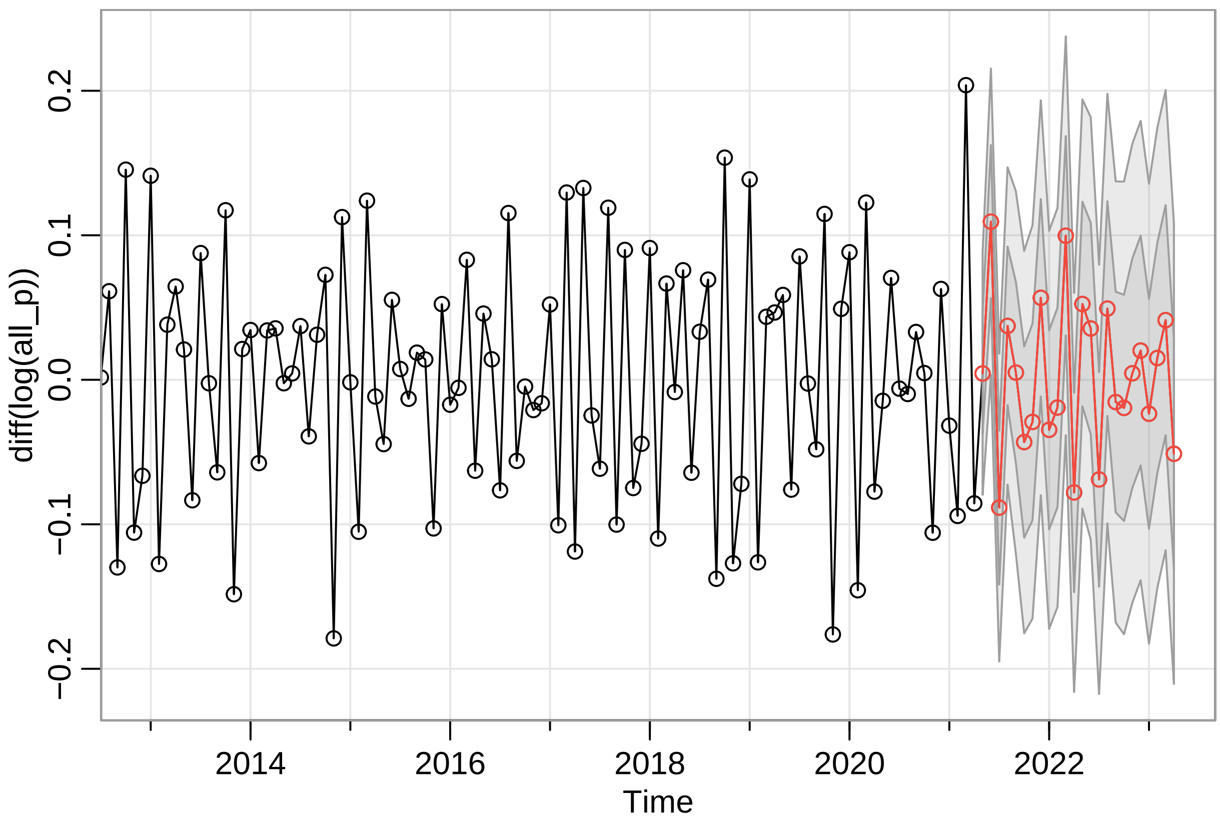

```
## $pred
##               Jan           Feb           Mar           Apr           May
## 2021                                                          0.0072986555
## 2022  0.0261429453 -0.0303390528  0.0300829220 -0.0039272593 -0.0146024875
## 2023  0.0094469481 -0.0034590705  0.0044804410  0.0057643034
##               Jun           Jul           Aug           Sep           Oct
## 2021  0.0843842473 -0.0890232761  0.0303293300 -0.0069693788  0.0006689785
## 2022  0.0246381172 -0.0131057823  0.0082879256  0.0044371498  0.0006047622
## 2023
##               Nov           Dec
## 2021 -0.0003087408  0.0060709524
## 2022  0.0059035794 -0.0023627348
## 2023
##
## $se
##             Jan        Feb        Mar        Apr        May        Jun
## 2021                                              0.05087033 0.06604198
## 2022 0.08326303 0.08329392 0.08344594 0.08385412 0.08427421 0.08446192
## 2023 0.08528188 0.08530577 0.08532522 0.08534407
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.06864025 0.06935316 0.07852925 0.08232069 0.08301402 0.08301500
## 2022 0.08449235 0.08479651 0.08511867 0.08524072 0.08524121 0.08525732
## 2023
```

```
sarima.for(diff(log(all_p)), n.ahead=24, p=8, d=0, q=0) # AR(8)
```



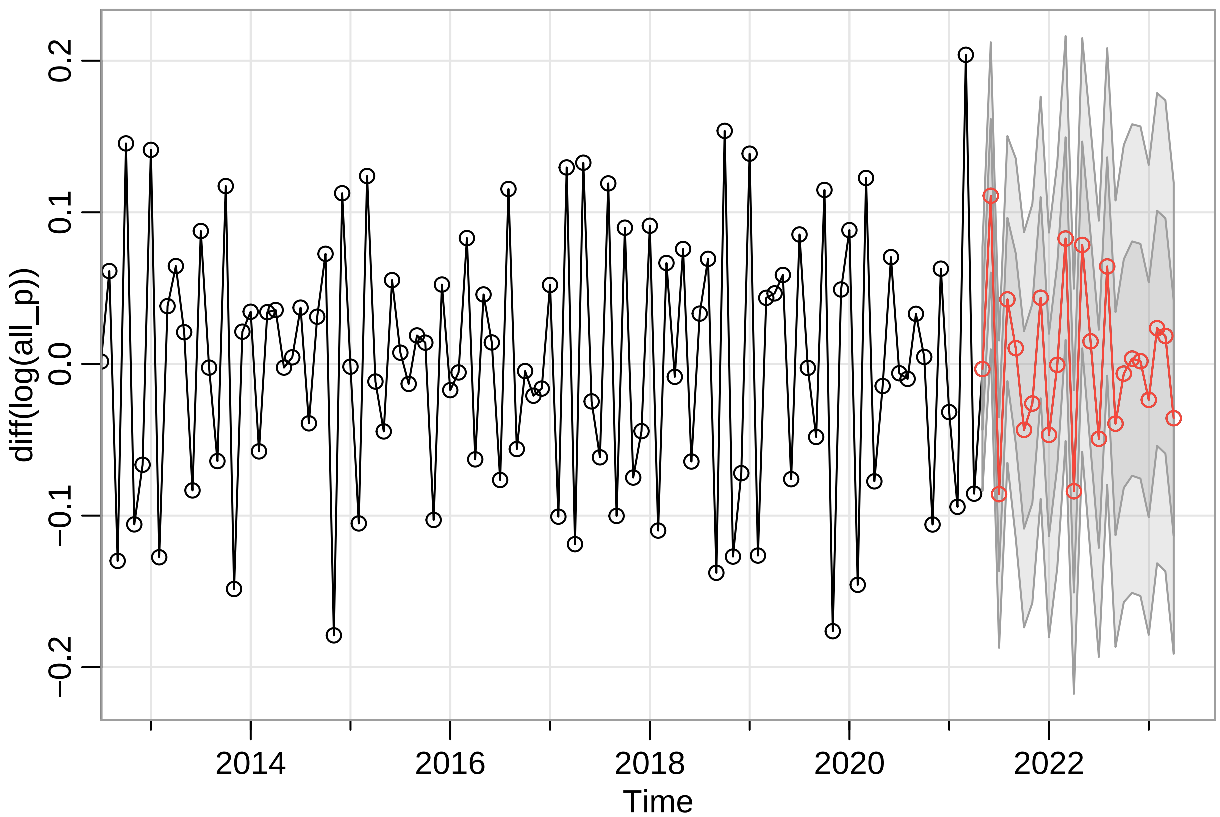

```
## $pred
##               Jan          Feb          Mar          Apr          May
## 2021                                                      0.004090777
## 2022 -0.033559884 -0.020402804  0.099539463 -0.076360597  0.050949795
## 2023 -0.022997366  0.013753246  0.042342306 -0.050080668
##               Jun          Jul          Aug          Sep          Oct
## 2021  0.109574880 -0.087945430  0.037257438  0.005233874 -0.043056418
## 2022  0.037169951 -0.068411971  0.047722903 -0.014324035 -0.019845070
## 2023
##               Nov          Dec
## 2021 -0.030031604  0.056122073
## 2022  0.003521584  0.020114489
## 2023
##
## $se
##             Jan        Feb        Mar        Apr        May        Jun
## 2021                                             0.04200099 0.05326750
## 2022 0.06923073 0.06935844 0.06935844 0.06937317 0.07113284 0.07341067
## 2023 0.07980957 0.07982838 0.07986142 0.07986920
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.05354404 0.05512081 0.06296342 0.06649143 0.06841782 0.06868971
## 2022 0.07457787 0.07461070 0.07665951 0.07857692 0.07965425 0.07969126
## 2023
```

```
sarima.for(diff(log(all_p)), n.ahead=24, p=9, d=0, q=0) # AR(9)
```



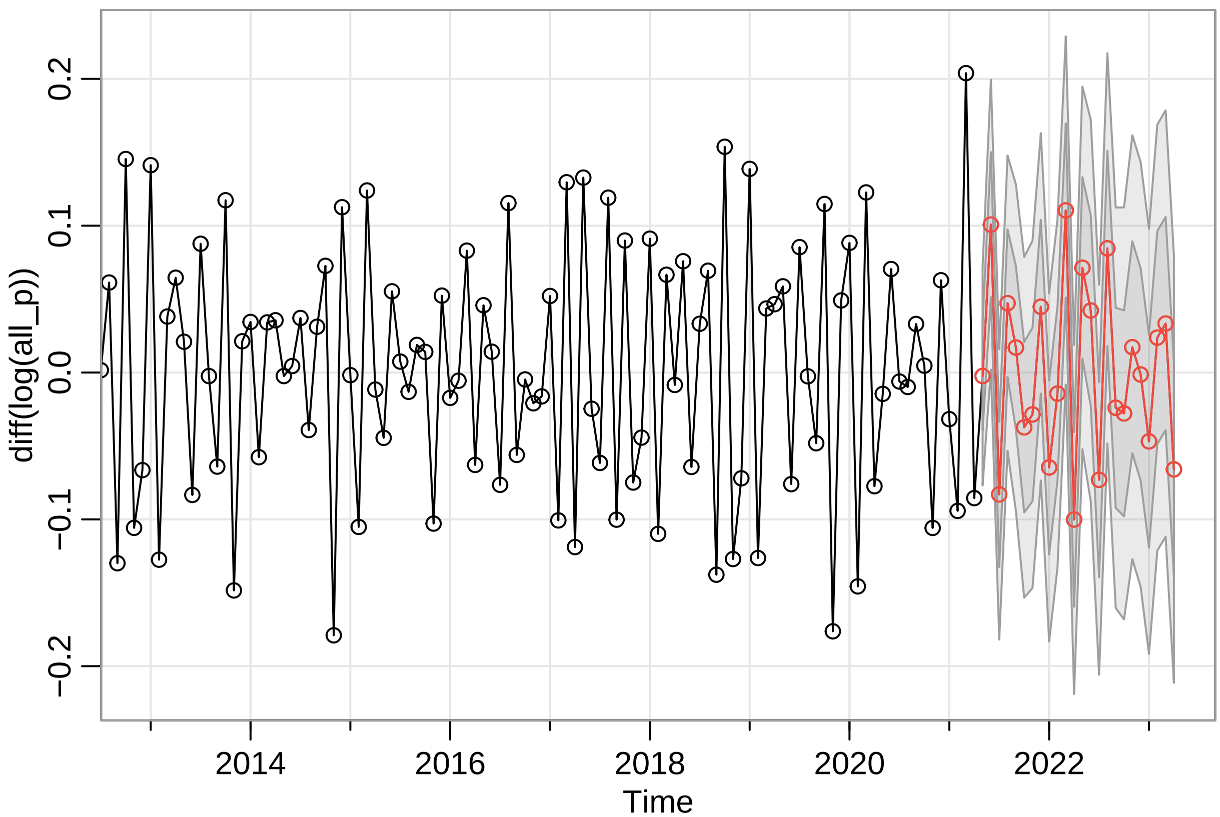

```
## $pred
##               Jan          Feb          Mar          Apr          May
## 2021                                                       0.004365606
## 2022 -0.034612057 -0.019206319  0.099687228 -0.077983767  0.052428844
## 2023 -0.023512285  0.015051492  0.041289826 -0.051210440
##               Jun          Jul          Aug          Sep          Oct
## 2021  0.109435685 -0.088450074  0.037277089  0.004981152 -0.043134087
## 2022  0.035485454 -0.068965191  0.049277622 -0.015415192 -0.019469691
## 2023
##               Nov          Dec
## 2021 -0.029184141  0.056751538
## 2022  0.004483031  0.020208357
## 2023
##
## $se
##             Jan        Feb        Mar        Apr        May        Jun
## 2021                                            0.04199428 0.05299815
## 2022 0.06887979 0.06900468 0.06900487 0.06901810 0.07078751 0.07309686
## 2023 0.07957664 0.07959650 0.07962918 0.07963634
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.05324000 0.05488810 0.06273370 0.06616025 0.06804974 0.06832264
## 2022 0.07427161 0.07430919 0.07633318 0.07830779 0.07941604 0.07945711
## 2023
```

```
sarima.for(diff(log(all_p)), n.ahead=24, p=10, d=0, q=0) # AR(10)
```



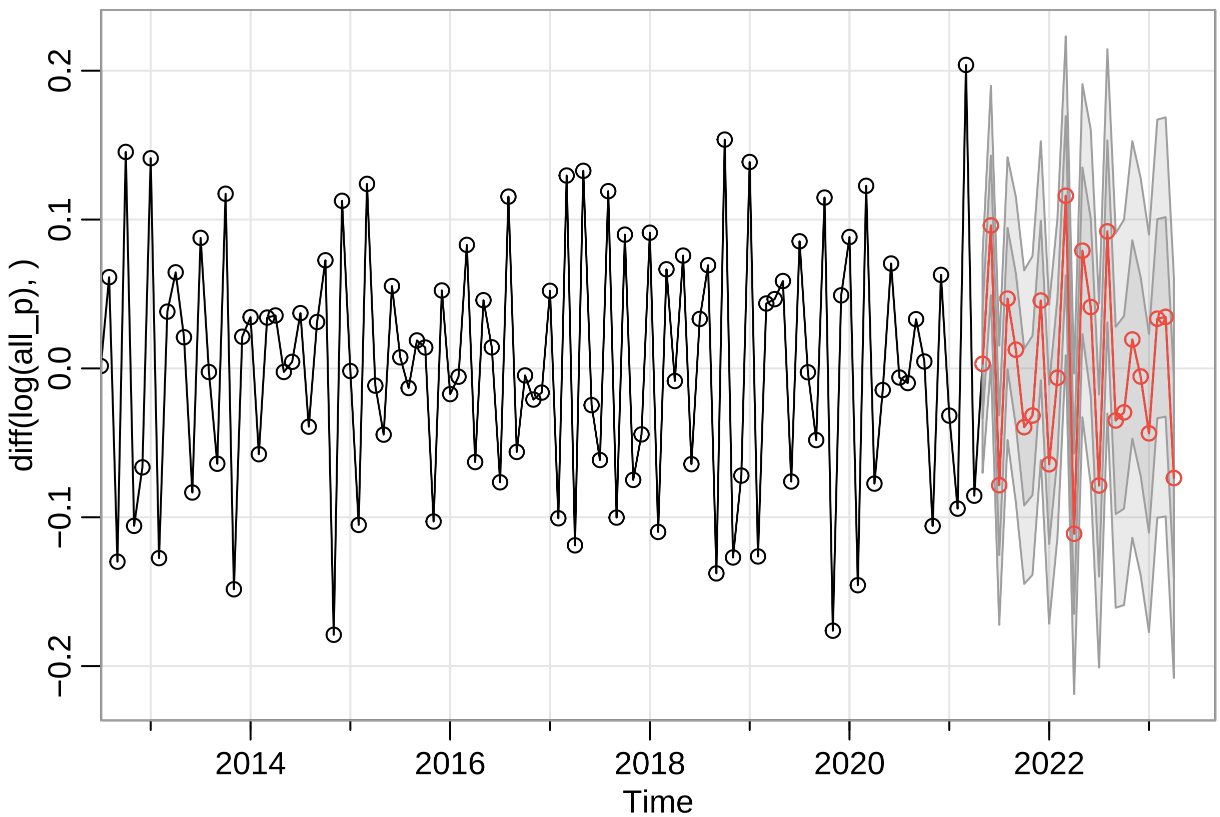

```
## $pred
##               Jan           Feb           Mar           Apr           May
## 2021                                                          -0.0033514473
## 2022 -0.0467705531 -0.0005995298  0.0826980517 -0.0839573203  0.0785014782
## 2023 -0.0236598045  0.0235989526  0.0184682280 -0.0357829336
##               Jun           Jul           Aug           Sep           Oct
## 2021  0.1109152109 -0.0858634891  0.0425737243  0.0104396587 -0.0434570811
## 2022  0.0149380910 -0.0493396473  0.0642893899 -0.0393727932 -0.0063425988
## 2023
##               Nov           Dec
## 2021 -0.0260875110  0.0436592611
## 2022  0.0035537876  0.0018244444
## 2023
##
## $se
##             Jan        Feb        Mar        Apr        May        Jun
## 2021                                              0.04022200 0.05061803
## 2022 0.06669952 0.06675795 0.06676883 0.06678706 0.06818161 0.07050281
## 2023 0.07747115 0.07752622 0.07764621 0.07765425
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.05065101 0.05385535 0.06258086 0.06514868 0.06573474 0.06631307
## 2022 0.07191295 0.07197957 0.07359578 0.07536504 0.07726297 0.07740637
## 2023
```

```
sarima.for(diff(log(all_p)), n.ahead=24, p=11, d=0, q=0) # AR(11)
```




```
## $pred
##               Jan          Feb          Mar          Apr          May
## 2021                                                       -0.002373511
## 2022 -0.064640861 -0.014357606  0.110420880 -0.100104874  0.071305895
## 2023 -0.046876441  0.023792573  0.033350947 -0.065942637
##               Jun          Jul          Aug          Sep          Oct
## 2021  0.100739157 -0.083007873  0.047209517  0.017009041 -0.037319479
## 2022  0.042233530 -0.073041526  0.084587356 -0.024017169 -0.027852751
## 2023
##               Nov          Dec
## 2021 -0.028534849  0.044786769
## 2022  0.017233581 -0.001362971
## 2023
##
## $se
##             Jan        Feb        Mar        Apr        May        Jun
## 2021                                              0.03726194 0.04938943
## 2022 0.05926919 0.05926929 0.05927803 0.05944570 0.06174053 0.06496328
## 2023 0.07238035 0.07253053 0.07260545 0.07270982
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.04943454 0.05027963 0.05558328 0.05801509 0.05914432 0.05914438
## 2022 0.06638471 0.06645595 0.06818521 0.07015082 0.07218729 0.07228468
## 2023
```

```
sarima.for(diff(log(all_p), ), n.ahead=24, p=12, d=0, q=0) # AR(12)
```

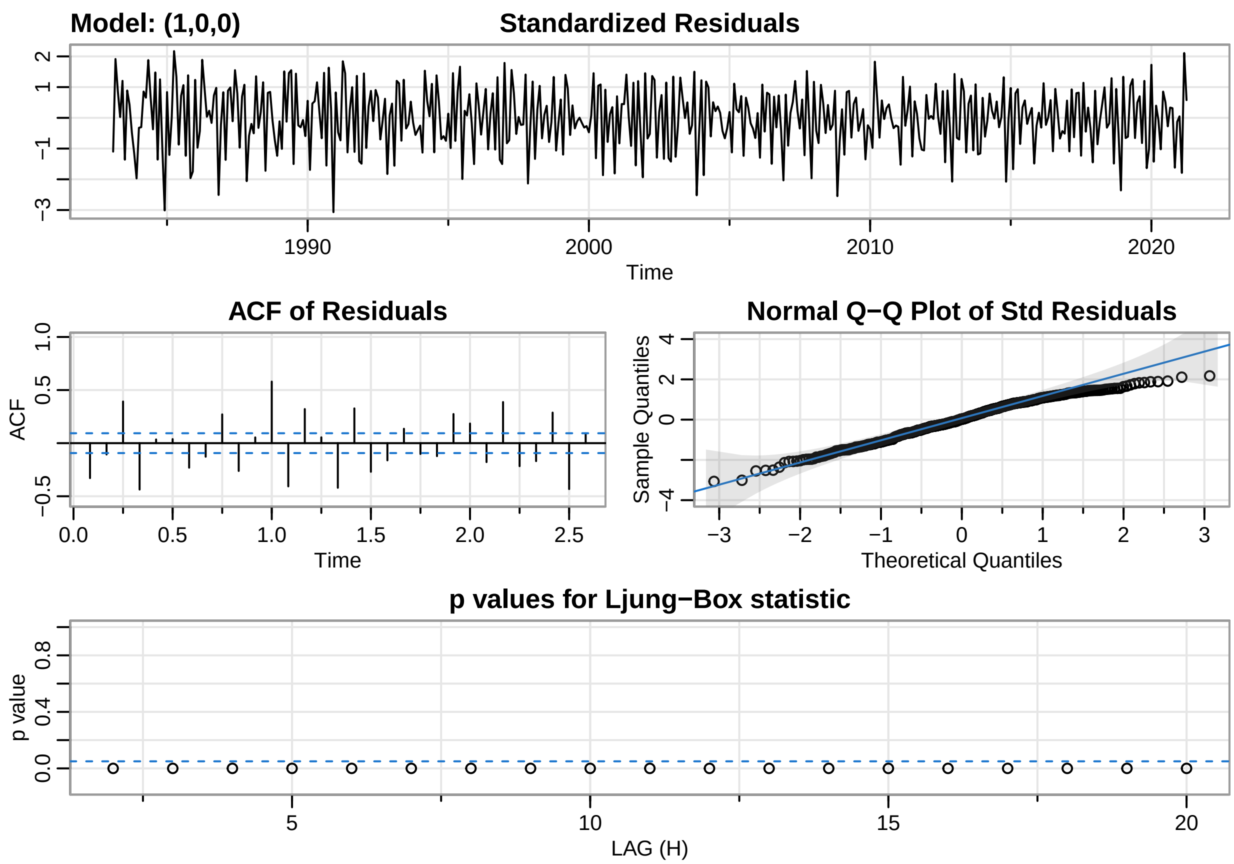



```
## $pred
##               Jan          Feb          Mar          Apr          May
## 2021                                                      0.003054730
## 2022 -0.064386499 -0.006247674  0.115995011 -0.111101022  0.079072958
## 2023 -0.043624549  0.033391930  0.034616240 -0.073660460
##               Jun          Jul          Aug          Sep          Oct
## 2021  0.096037396 -0.078472542  0.046892765  0.012557702 -0.039464957
## 2022  0.041243346 -0.078673766  0.092052846 -0.034972862 -0.029552056
## 2023
##               Nov          Dec
## 2021 -0.031654604  0.045557116
## 2022  0.019444550 -0.005442959
## 2023
##
## $se
##             Jan        Feb        Mar        Apr        May        Jun
## 2021                                              0.03660896 0.04688029
## 2022 0.05355666 0.05356083 0.05356120 0.05385451 0.05595170 0.05963166
## 2023 0.06676030 0.06687985 0.06701896 0.06715002
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.04690973 0.04750461 0.05129102 0.05267326 0.05353520 0.05353652
## 2022 0.06115542 0.06119357 0.06292297 0.06472366 0.06662407 0.06670183
## 2023
```

Checking the Model Conditions: All Poultry

```r
sarima(diff(log(all_p)), p=1, d=0, q=0)# AR(1)
```

```
## initial  value -2.460721
## iter   2 value -2.713258
## iter   3 value -2.713266
## iter   3 value -2.713266
## iter   3 value -2.713266
## final  value -2.713266
## converged
## initial  value -2.712484
## iter   2 value -2.712485
## iter   2 value -2.712485
## iter   2 value -2.712485
## final  value -2.712485
## converged
```

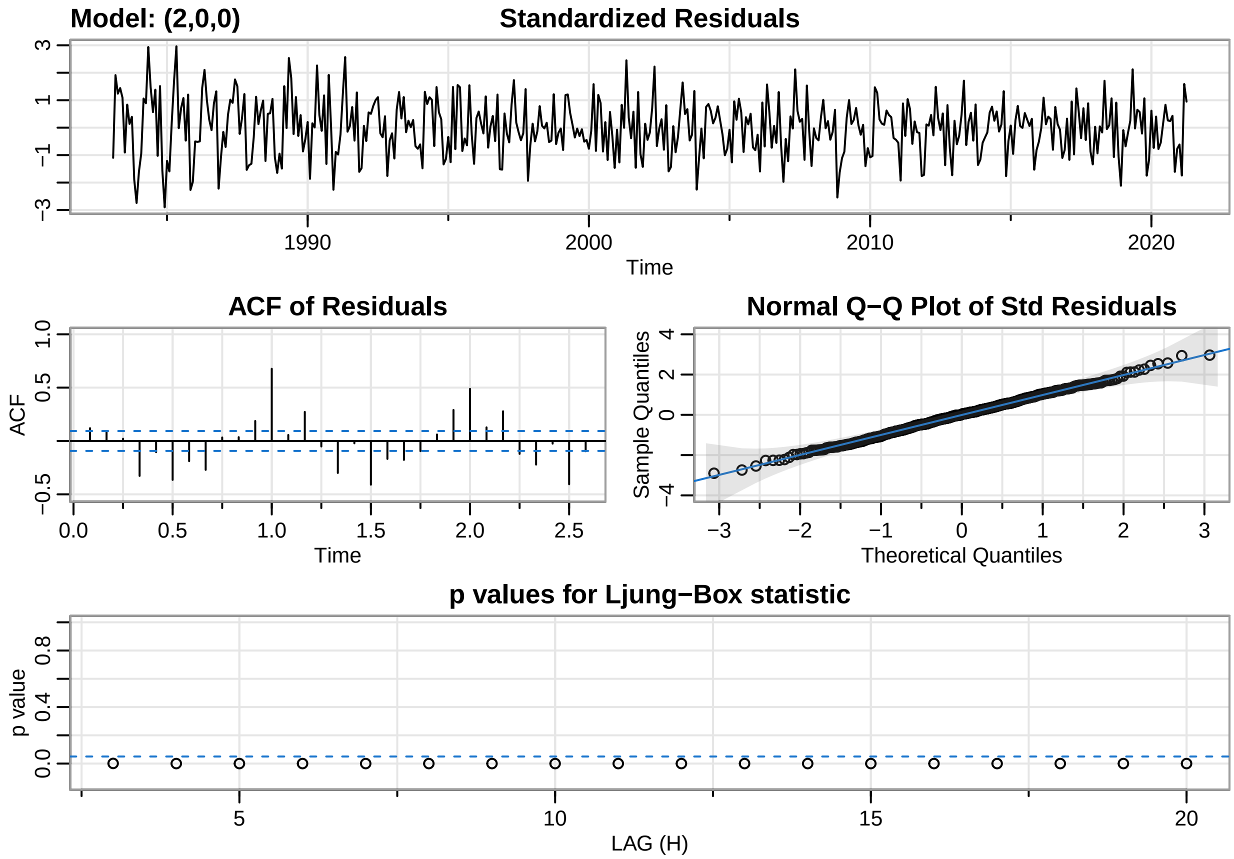



```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1   xmean
##       -0.6299  0.0028
```

```
## s.e.   0.0363  0.0019
##
## sigma^2 estimated as 0.0044:  log likelihood = 593.74,  aic = -1181.48
##
## $degrees_of_freedom
## [1] 457
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1    -0.6299 0.0363 -17.3619  0.0000
## xmean   0.0028 0.0019   1.4967  0.1352
##
## $AIC
## [1] -2.57402
##
## $AICc
## [1] -2.573963
##
## $BIC
## [1] -2.547033
```

```
sarima(diff(log(all_p)), p=2, d=0, q=0) # AR(2)
```

```
## initial  value -2.464856
## iter   2 value -2.672827
## iter   3 value -2.857715
## iter   4 value -2.871723
## iter   5 value -2.879038
## iter   6 value -2.879409
## iter   7 value -2.879555
## iter   8 value -2.879561
## iter   9 value -2.879564
## iter  10 value -2.879564
## iter  11 value -2.879564
## iter  12 value -2.879564
## iter  12 value -2.879564
## iter  12 value -2.879564
## final  value -2.879564
## converged
## initial  value -2.875113
## iter   2 value -2.875115
## iter   3 value -2.875124
## iter   4 value -2.875125
## iter   5 value -2.875126
## iter   5 value -2.875126
## iter   5 value -2.875126
## final  value -2.875126
## converged
```

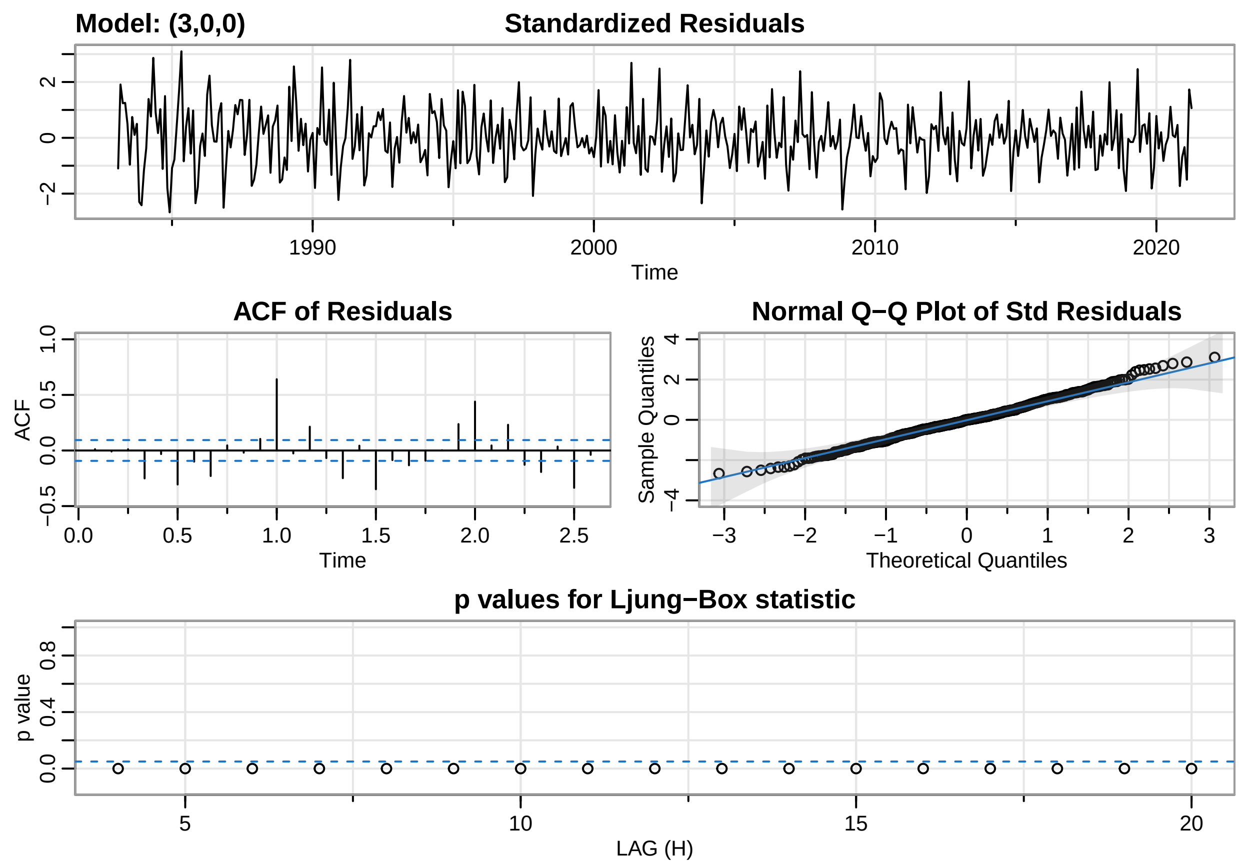



```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2   xmean
##       -0.9608  -0.5319  0.0028
## s.e.   0.0396   0.0400  0.0011
##
## sigma^2 estimated as 0.003174:  log likelihood = 668.39,  aic = -1328.78
##
## $degrees_of_freedom
## [1] 456
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1    -0.9608 0.0396 -24.2530  0.0000
## ar2    -0.5319 0.0400 -13.3045  0.0000
## xmean   0.0028 0.0011   2.6006  0.0096
##
## $AIC
## [1] -2.894945
##
```

```
## $AICc
## [1] -2.89483
##
## $BIC
## [1] -2.858962
```

```r
sarima(diff(log(all_p)), p=3, d=0, q=0) # AR(3)
```

```
## initial  value -2.464292
## iter   2 value -2.851924
## iter   3 value -2.876119
## iter   4 value -2.894698
## iter   5 value -2.904788
## iter   6 value -2.907010
## iter   7 value -2.907085
## iter   8 value -2.907086
## iter   9 value -2.907086
## iter   9 value -2.907086
## iter   9 value -2.907086
## final  value -2.907086
## converged
## initial  value -2.901851
## iter   2 value -2.901865
## iter   3 value -2.901873
## iter   4 value -2.901874
## iter   5 value -2.901875
## iter   5 value -2.901875
## iter   5 value -2.901875
## final  value -2.901875
## converged
```

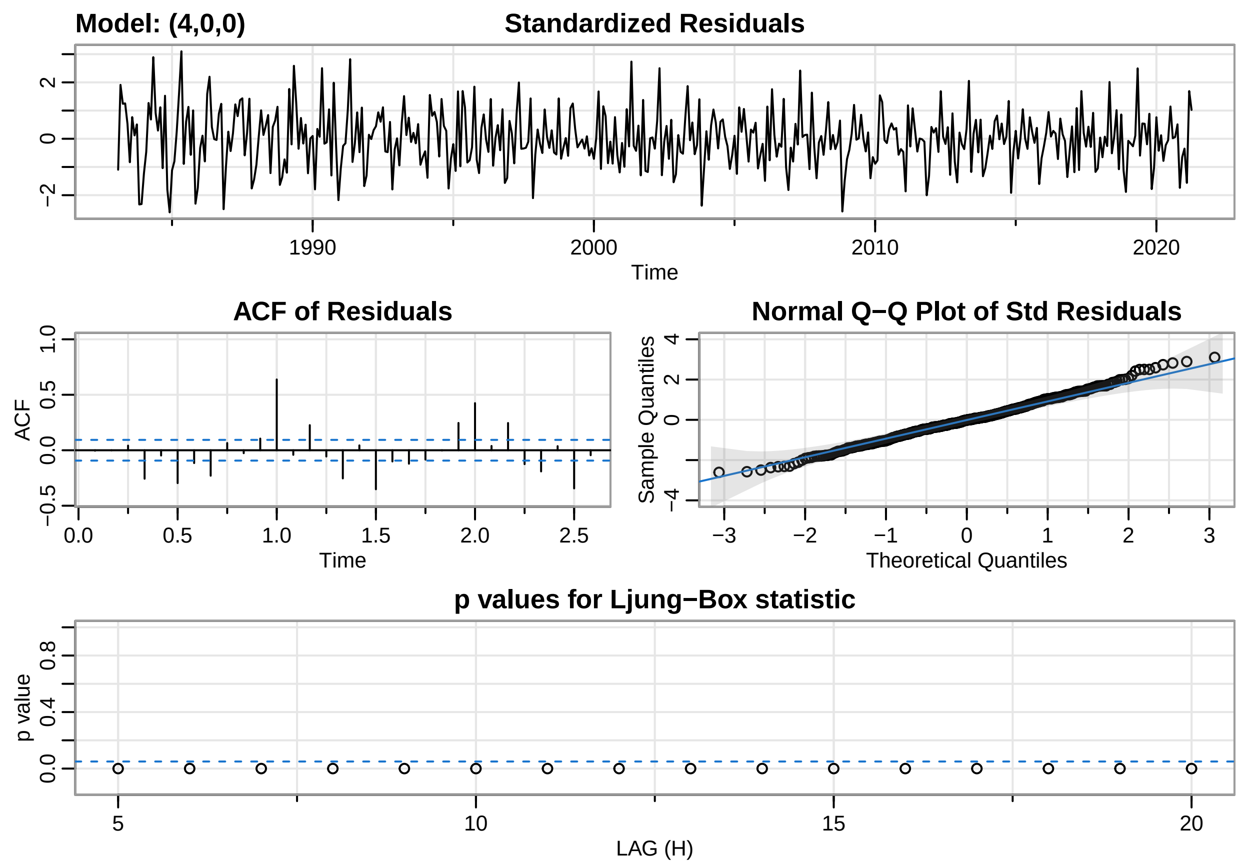



```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2     ar3   xmean
##       -0.8394  -0.3102  0.2308  0.0028
## s.e.   0.0455   0.0588  0.0459  0.0013
##
## sigma^2 estimated as 0.003007:  log likelihood = 680.67,  aic = -1351.34
##
## $degrees_of_freedom
## [1] 455
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1    -0.8394 0.0455 -18.4553  0.0000
## ar2    -0.3102 0.0588  -5.2739  0.0000
## ar3     0.2308 0.0459   5.0260  0.0000
## xmean   0.0028 0.0013   2.1018  0.0361
##
## $AIC
## [1] -2.944087
```

```
##
## $AICc
## [1] -2.943895
##
## $BIC
## [1] -2.899108
```

```r
sarima(diff(log(all_p)), p=4, d=0, q=0) # AR(4)
```

```
## initial  value -2.463428
## iter   2 value -2.673666
## iter   3 value -2.848804
## iter   4 value -2.889797
## iter   5 value -2.903160
## iter   6 value -2.907549
## iter   7 value -2.908573
## iter   8 value -2.909134
## iter   9 value -2.909143
## iter   9 value -2.909143
## iter   9 value -2.909143
## final  value -2.909143
## converged
## initial  value -2.903206
## iter   2 value -2.903207
## iter   3 value -2.903251
## iter   4 value -2.903251
## iter   5 value -2.903251
## iter   6 value -2.903251
## iter   6 value -2.903251
## iter   6 value -2.903251
## final  value -2.903251
## converged
```

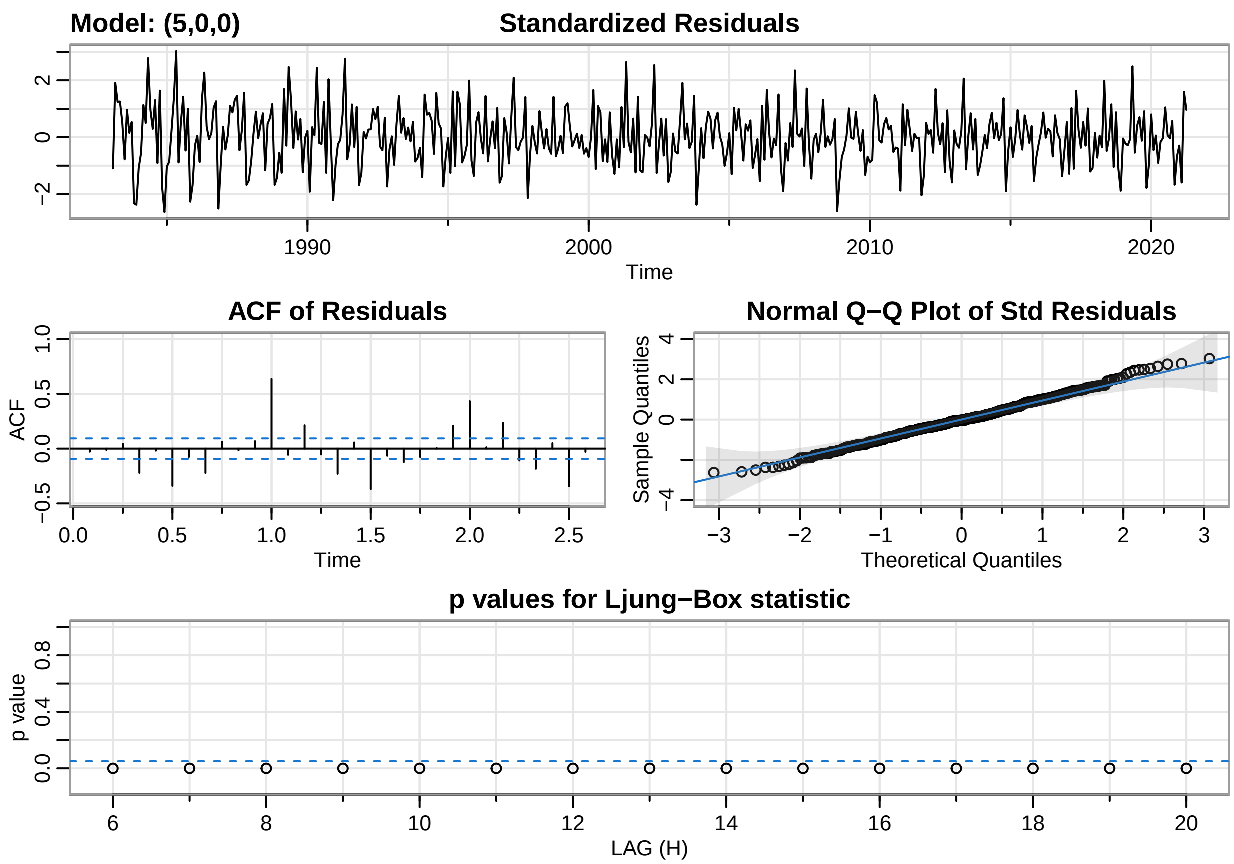



```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2     ar3      ar4   xmean
##       -0.8275  -0.3268  0.1861  -0.0531  0.0028
## s.e.   0.0467   0.0606  0.0607   0.0472  0.0013
##
## sigma^2 estimated as 0.002999:  log likelihood = 681.3,  aic = -1350.6
##
## $degrees_of_freedom
## [1] 454
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1    -0.8275 0.0467 -17.7357  0.0000
## ar2    -0.3268 0.0606  -5.3966  0.0000
## ar3     0.1861 0.0607   3.0679  0.0023
## ar4    -0.0531 0.0472  -1.1248  0.2613
## xmean   0.0028 0.0013   2.2021  0.0282
##
## $AIC
```

```
## [1] -2.942482
##
## $AICc
## [1] -2.942194
##
## $BIC
## [1] -2.888508
```

```
sarima(diff(log(all_p)), p=5, d=0, q=0) # AR(5)
```

```
## initial  value -2.462793
## iter   2 value -2.478506
## iter   3 value -2.854306
## iter   4 value -2.873560
## iter   5 value -2.875993
## iter   6 value -2.899286
## iter   7 value -2.907045
## iter   8 value -2.910995
## iter   9 value -2.911386
## iter  10 value -2.911543
## iter  11 value -2.911559
## iter  12 value -2.911575
## iter  13 value -2.911576
## iter  13 value -2.911576
## iter  13 value -2.911576
## final  value -2.911576
## converged
## initial  value -2.906356
## iter   2 value -2.906394
## iter   3 value -2.906420
## iter   4 value -2.906420
## iter   5 value -2.906420
## iter   6 value -2.906421
## iter   6 value -2.906421
## iter   6 value -2.906421
## final  value -2.906421
## converged
```

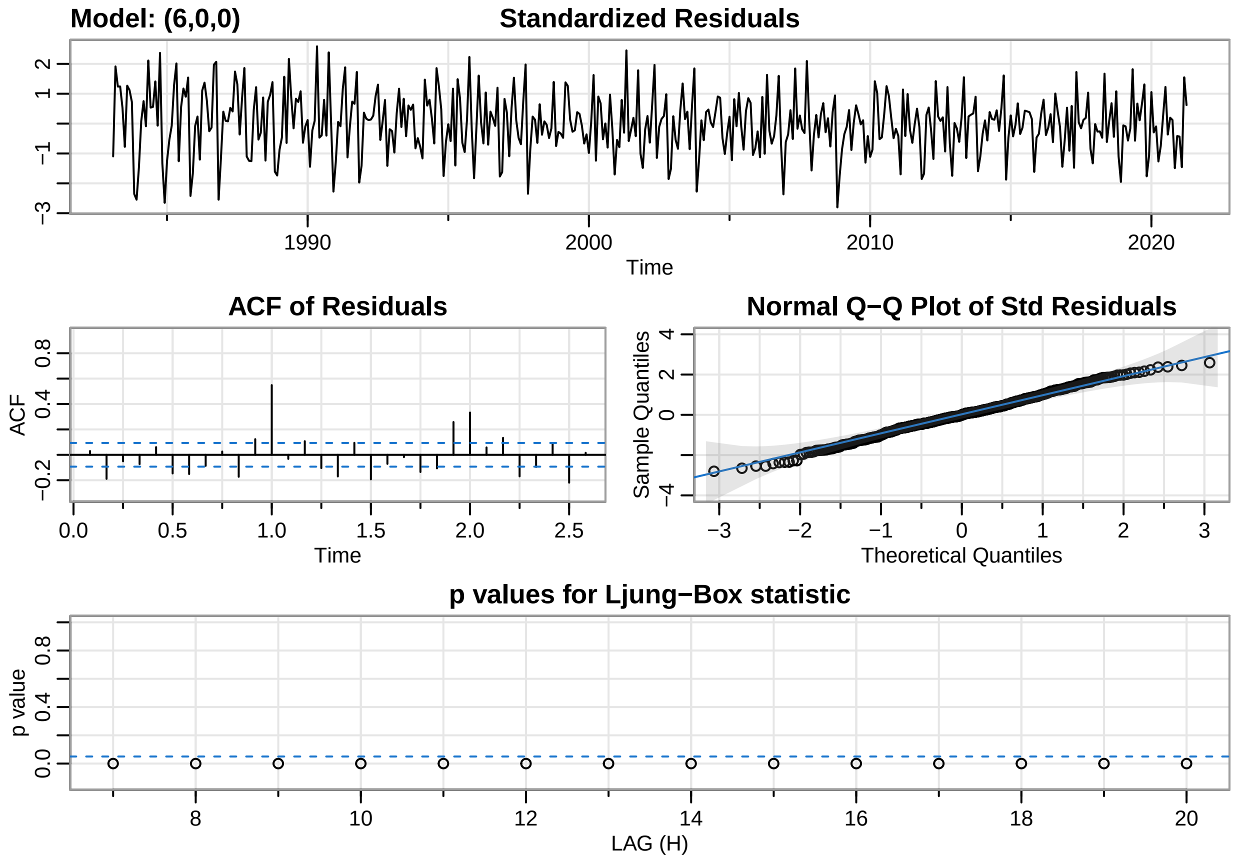



```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2     ar3      ar4      ar5   xmean
##       -0.8316  -0.3122  0.1596  -0.1200  -0.0804  0.0028
## s.e.   0.0466   0.0610  0.0624   0.0612   0.0471  0.0012
##
## sigma^2 estimated as 0.00298:  log likelihood = 682.75,  aic = -1351.51
##
## $degrees_of_freedom
## [1] 453
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1    -0.8316 0.0466 -17.8591  0.0000
## ar2    -0.3122 0.0610  -5.1194  0.0000
## ar3     0.1596 0.0624   2.5552  0.0109
## ar4    -0.1200 0.0612  -1.9606  0.0505
## ar5    -0.0804 0.0471  -1.7084  0.0882
## xmean   0.0028 0.0012   2.3675  0.0183
##
```

```
## $AIC
## [1] -2.944463
##
## $AICc
## [1] -2.944058
##
## $BIC
## [1] -2.881493
```

```r
sarima(diff(log(all_p)), p=6, d=0, q=0) # AR(6)
```

```
## initial  value -2.464065
## iter   2 value -2.627908
## iter   3 value -2.824735
## iter   4 value -2.865397
## iter   5 value -2.939363
## iter   6 value -2.966124
## iter   7 value -2.974919
## iter   8 value -2.976749
## iter   9 value -2.977162
## iter  10 value -2.977497
## iter  11 value -2.977825
## iter  12 value -2.977843
## iter  13 value -2.977848
## iter  13 value -2.977848
## iter  13 value -2.977848
## final  value -2.977848
## converged
## initial  value -2.972190
## iter   2 value -2.972214
## iter   3 value -2.972246
## iter   4 value -2.972252
## iter   5 value -2.972256
## iter   6 value -2.972257
## iter   7 value -2.972258
## iter   7 value -2.972258
## iter   7 value -2.972258
## final  value -2.972258
## converged
```

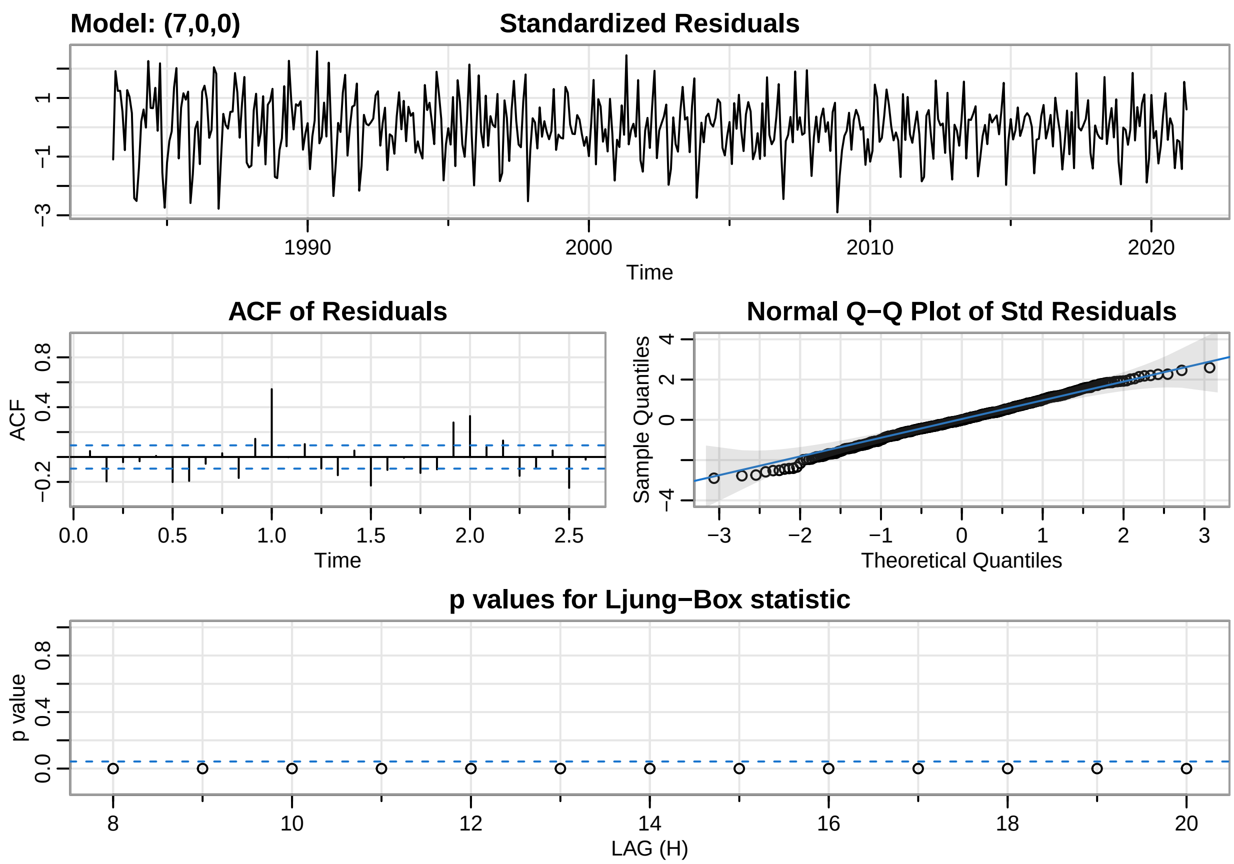



```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2     ar3      ar4      ar5      ar6   xmean
##       -0.8589  -0.3505  0.2167  -0.2316  -0.3777  -0.3554  0.0027
## s.e.   0.0437   0.0572  0.0589   0.0589   0.0574   0.0441  0.0008
##
## sigma^2 estimated as 0.002608:  log likelihood = 712.97,  aic = -1409.95
##
## $degrees_of_freedom
## [1] 452
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1    -0.8589 0.0437 -19.6738   0e+00
## ar2    -0.3505 0.0572  -6.1269   0e+00
## ar3     0.2167 0.0589   3.6770   3e-04
## ar4    -0.2316 0.0589  -3.9343   1e-04
## ar5    -0.3777 0.0574  -6.5759   0e+00
## ar6    -0.3554 0.0441  -8.0533   0e+00
## xmean   0.0027 0.0008   3.3595   8e-04
```

```
##
## $AIC
## [1] -3.07178
##
## $AICc
## [1] -3.071239
##
## $BIC
## [1] -2.999814
```

```
sarima(diff(log(all_p)), p=7, d=0, q=0) # AR(7)
```

```
## initial  value -2.465872
## iter   2 value -2.629833
## iter   3 value -2.844550
## iter   4 value -2.863836
## iter   5 value -2.876902
## iter   6 value -2.968669
## iter   7 value -2.975521
## iter   8 value -2.981750
## iter   9 value -2.982166
## iter  10 value -2.982339
## iter  11 value -2.982388
## iter  12 value -2.982402
## iter  13 value -2.982415
## iter  14 value -2.982424
## iter  14 value -2.982424
## iter  14 value -2.982424
## final  value -2.982424
## converged
## initial  value -2.975940
## iter   2 value -2.975999
## iter   3 value -2.976034
## iter   4 value -2.976043
## iter   5 value -2.976046
## iter   6 value -2.976046
## iter   7 value -2.976046
## iter   8 value -2.976046
## iter   8 value -2.976046
## iter   8 value -2.976046
## final  value -2.976046
## converged
```

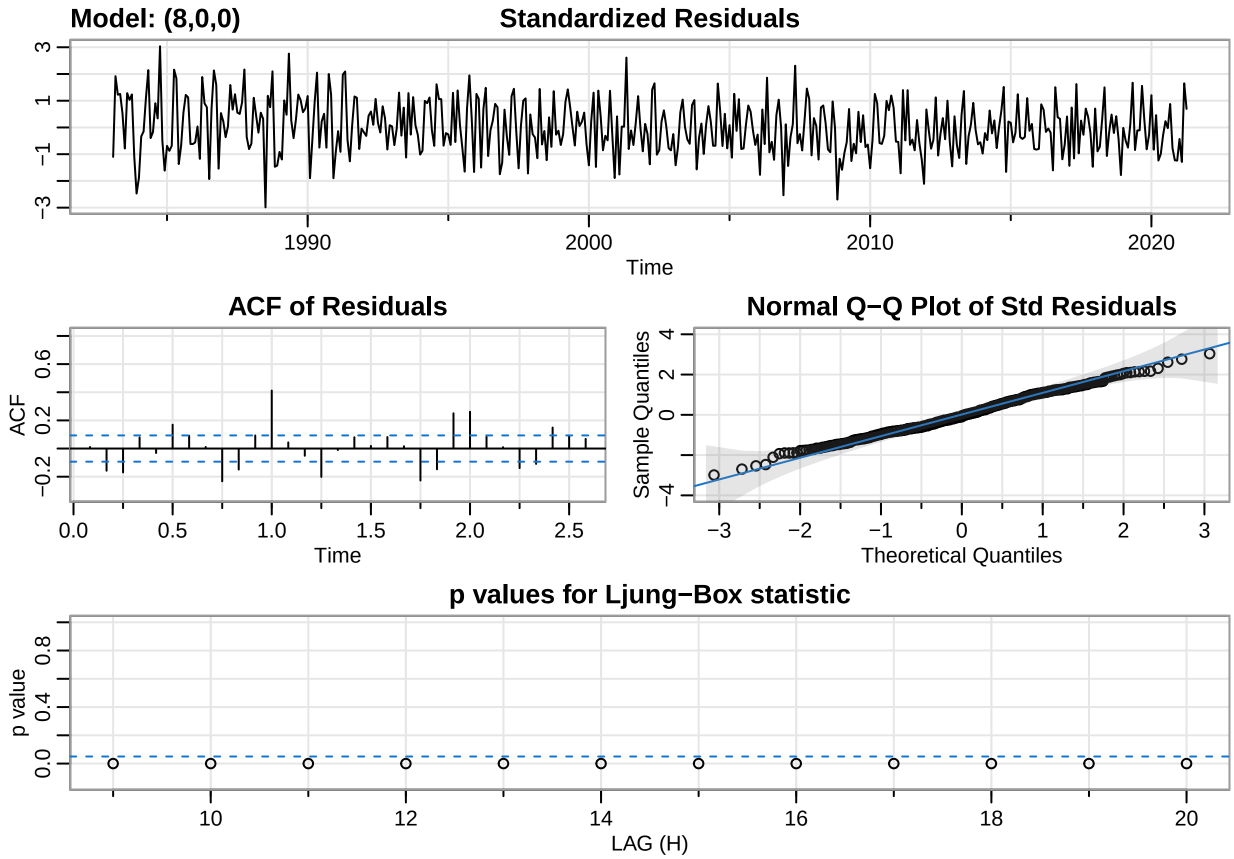



```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2     ar3      ar4      ar5      ar6     ar7   xmean
##       -0.8279  -0.3177  0.2364  -0.2502  -0.3462  -0.2793  0.0881  0.0027
## s.e.   0.0465   0.0596  0.0596   0.0595   0.0597   0.0599  0.0471  0.0009
##
## sigma^2 estimated as 0.002588:  log likelihood = 714.71,  aic = -1411.42
##
## $degrees_of_freedom
## [1] 451
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1    -0.8279 0.0465 -17.7923  0.0000
## ar2    -0.3177 0.0596  -5.3265  0.0000
## ar3     0.2364 0.0596   3.9648  0.0001
## ar4    -0.2502 0.0595  -4.2073  0.0000
## ar5    -0.3462 0.0597  -5.8015  0.0000
## ar6    -0.2793 0.0599  -4.6608  0.0000
## ar7     0.0881 0.0471   1.8689  0.0623
```

```
## xmean   0.0027 0.0009   3.0886  0.0021
##
## $AIC
## [1] -3.075
##
## $AICc
## [1] -3.074303
##
## $BIC
## [1] -2.994038
```

```r
sarima(diff(log(all_p)), p=8, d=0, q=0) # AR(8)
```

```
## initial  value -2.465287
## iter   2 value -2.570959
## iter   3 value -3.007957
## iter   4 value -3.042305
## iter   5 value -3.077042
## iter   6 value -3.101109
## iter   7 value -3.143835
## iter   8 value -3.165972
## iter   9 value -3.168334
## iter  10 value -3.173559
## iter  11 value -3.173994
## iter  12 value -3.174058
## iter  13 value -3.174097
## iter  14 value -3.174098
## iter  14 value -3.174098
## iter  14 value -3.174098
## final  value -3.174098
## converged
## initial  value -3.164190
## iter   2 value -3.164211
## iter   3 value -3.164227
## iter   4 value -3.164237
## iter   5 value -3.164250
## iter   6 value -3.164255
## iter   7 value -3.164262
## iter   8 value -3.164264
## iter   9 value -3.164266
## iter  10 value -3.164266
## iter  11 value -3.164267
## iter  12 value -3.164267
## iter  12 value -3.164267
## iter  12 value -3.164267
## final  value -3.164267
## converged
```

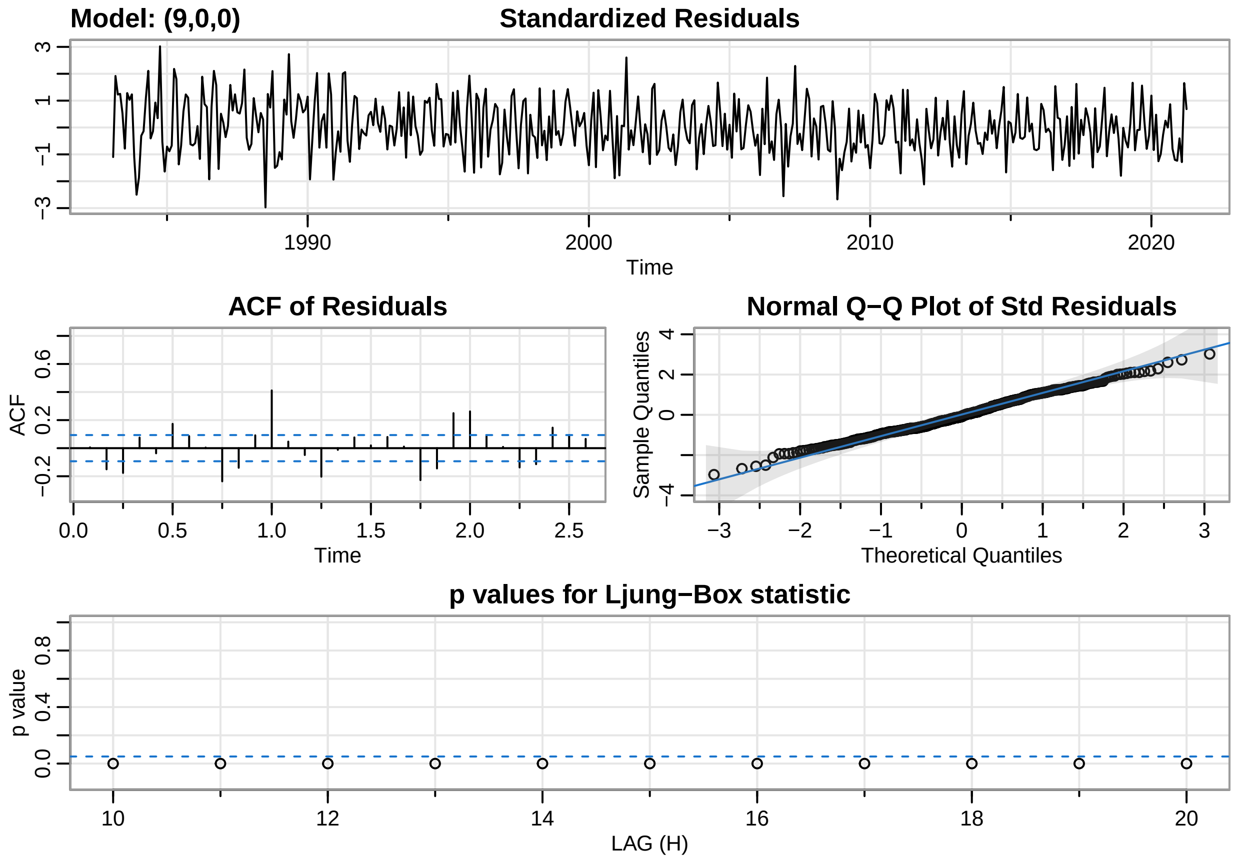



```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2     ar3      ar4      ar5      ar6      ar7      ar8
##       -0.7800  -0.4790  0.0389  -0.3891  -0.2156  -0.4640  -0.3856  -0.5668
## s.e.   0.0384   0.0504  0.0510   0.0502   0.0500   0.0509   0.0506   0.0388
##        xmean
##       0.0027
## s.e.  0.0005
##
## sigma^2 estimated as 0.001764:  log likelihood = 801.11,  aic = -1582.21
##
## $degrees_of_freedom
## [1] 450
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1    -0.7800 0.0384 -20.2901  0.0000
## ar2    -0.4790 0.0504  -9.5134  0.0000
## ar3     0.0389 0.0510   0.7636  0.4455
## ar4    -0.3891 0.0502  -7.7557  0.0000
```

```
## ar5    -0.2156 0.0500  -4.3133  0.0000
## ar6    -0.4640 0.0509  -9.1169  0.0000
## ar7    -0.3856 0.0506  -7.6210  0.0000
## ar8    -0.5668 0.0388 -14.6212  0.0000
## xmean   0.0027 0.0005   5.7115  0.0000
##
## $AIC
## [1] -3.447084
##
## $AICc
## [1] -3.44621
##
## $BIC
## [1] -3.357126
```

```
sarima(diff(log(all_p)), p=9, d=0, q=0) # AR(9)
```

```
## initial  value -2.464181
## iter   2 value -2.672072
## iter   3 value -2.802841
## iter   4 value -2.843741
## iter   5 value -3.038175
## iter   6 value -3.091677
## iter   7 value -3.125938
## iter   8 value -3.133829
## iter   9 value -3.155723
## iter  10 value -3.165121
## iter  11 value -3.169150
## iter  12 value -3.172499
## iter  13 value -3.174652
## iter  14 value -3.174904
## iter  15 value -3.174920
## iter  16 value -3.174920
## iter  16 value -3.174920
## iter  16 value -3.174920
## final  value -3.174920
## converged
## initial  value -3.164284
## iter   2 value -3.164331
## iter   3 value -3.164368
## iter   4 value -3.164384
## iter   5 value -3.164397
## iter   6 value -3.164409
## iter   7 value -3.164421
## iter   8 value -3.164422
## iter   9 value -3.164423
## iter  10 value -3.164424
## iter  11 value -3.164424
## iter  12 value -3.164424
## iter  12 value -3.164424
## iter  12 value -3.164424
## final  value -3.164424
## converged
```



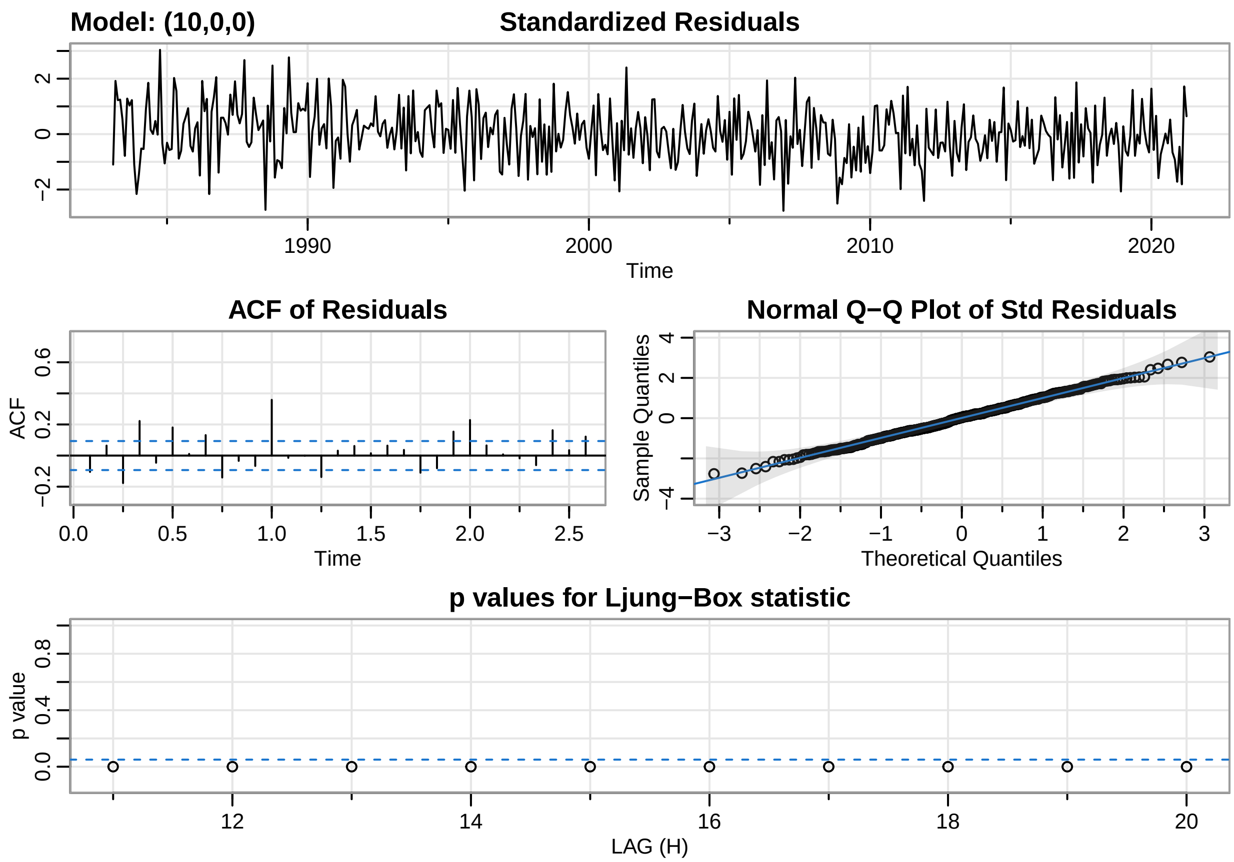

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2     ar3      ar4      ar5      ar6      ar7      ar8
##       -0.7699  -0.4720  0.0474  -0.3852  -0.2087  -0.4646  -0.3768  -0.5526
## s.e.   0.0468   0.0536  0.0556   0.0512   0.0532   0.0509   0.0556   0.0538
##          ar9   xmean
##       0.0180  0.0027
## s.e.  0.0473  0.0005
##
## sigma^2 estimated as 0.001764:  log likelihood = 801.18,  aic = -1580.36
##
## $degrees_of_freedom
## [1] 449
##
## $ttable
##     Estimate     SE  t.value p.value
## ar1  -0.7699 0.0468 -16.4496  0.0000
## ar2  -0.4720 0.0536  -8.8026  0.0000
## ar3   0.0474 0.0556   0.8525  0.3944
## ar4  -0.3852 0.0512  -7.5214  0.0000
```

```
## ar5    -0.2087 0.0532  -3.9233  0.0001
## ar6    -0.4646 0.0509  -9.1254  0.0000
## ar7    -0.3768 0.0556  -6.7771  0.0000
## ar8    -0.5526 0.0538 -10.2680  0.0000
## ar9     0.0180 0.0473   0.3804  0.7038
## xmean   0.0027 0.0005   5.6127  0.0000
##
## $AIC
## [1] -3.443041
##
## $AICc
## [1] -3.441971
##
## $BIC
## [1] -3.344088
```

```
sarima(diff(log(all_p)), p=10, d=0, q=0) # AR(10)
```

```
## initial  value -2.464056
## iter   2 value -2.659213
## iter   3 value -2.883117
## iter   4 value -2.984988
## iter   5 value -3.080437
## iter   6 value -3.102941
## iter   7 value -3.125926
## iter   8 value -3.148332
## iter   9 value -3.181323
## iter  10 value -3.183598
## iter  11 value -3.210294
## iter  12 value -3.217205
## iter  13 value -3.218067
## iter  14 value -3.218292
## iter  15 value -3.218303
## iter  16 value -3.218303
## iter  17 value -3.218304
## iter  17 value -3.218304
## iter  17 value -3.218304
## final  value -3.218304
## converged
## initial  value -3.206385
## iter   2 value -3.206409
## iter   3 value -3.206462
## iter   4 value -3.206517
## iter   5 value -3.206537
## iter   6 value -3.206555
## iter   7 value -3.206570
## iter   8 value -3.206586
## iter   9 value -3.206591
## iter  10 value -3.206593
## iter  11 value -3.206593
## iter  12 value -3.206593
## iter  13 value -3.206593
## iter  13 value -3.206594
## iter  13 value -3.206594
## final  value -3.206594
```

```
## converged
```



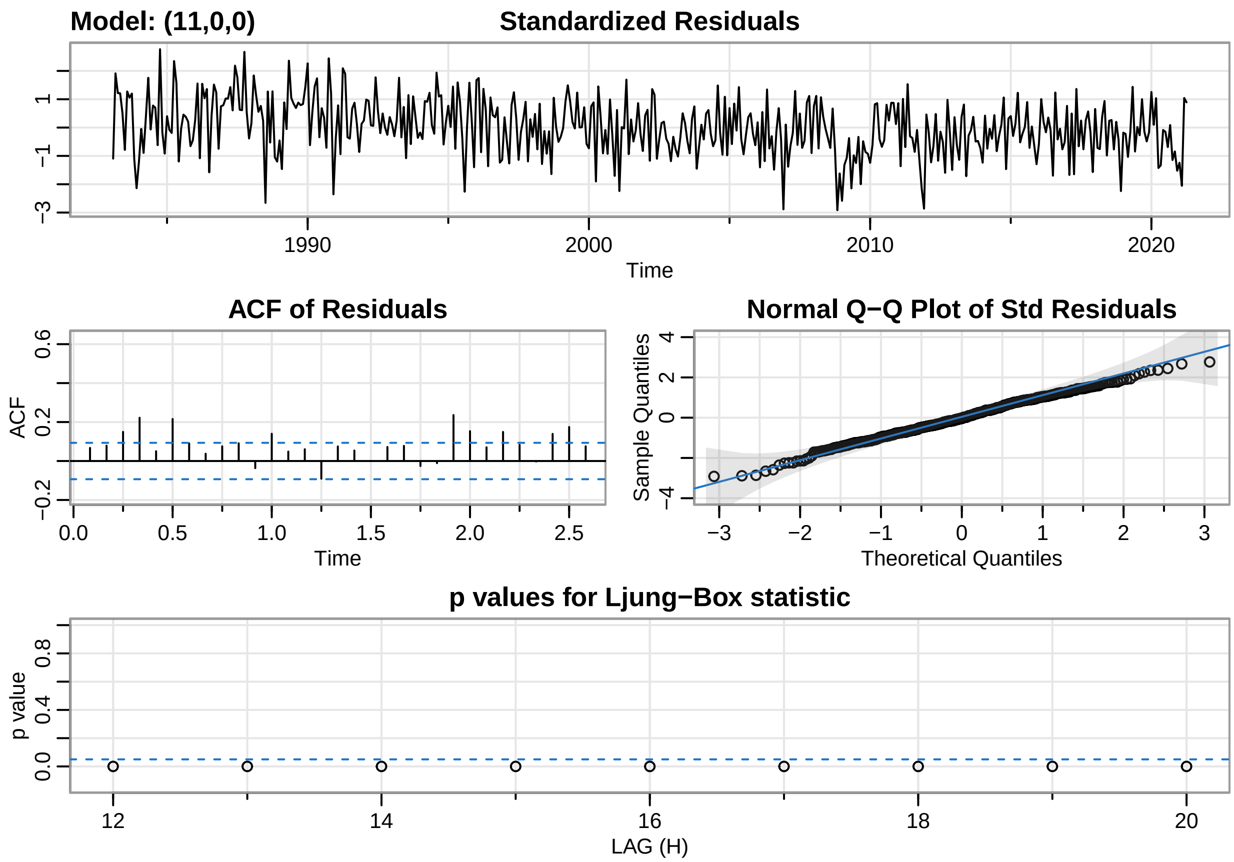

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5      ar6      ar7      ar8
##       -0.7640  -0.6292  -0.0605  -0.5196  -0.2687  -0.5737  -0.3624  -0.6900
## s.e.   0.0448   0.0570   0.0559   0.0534   0.0520   0.0516   0.0533   0.0558
##           ar9     ar10    xmean
##       -0.2058  -0.2885   0.0027
## s.e.   0.0574   0.0453   0.0004
##
## sigma^2 estimated as 0.001618:  log likelihood = 820.53,  aic = -1617.07
##
## $degrees_of_freedom
## [1] 448
##
## $ttable
##        Estimate     SE  t.value p.value
## ar1     -0.7640 0.0448 -17.0463  0.0000
## ar2     -0.6292 0.0570 -11.0433  0.0000
```

```
## ar3    -0.0605 0.0559  -1.0823  0.2797
## ar4    -0.5196 0.0534  -9.7338  0.0000
## ar5    -0.2687 0.0520  -5.1711  0.0000
## ar6    -0.5737 0.0516 -11.1090  0.0000
## ar7    -0.3624 0.0533  -6.8049  0.0000
## ar8    -0.6900 0.0558 -12.3745  0.0000
## ar9    -0.2058 0.0574  -3.5887  0.0004
## ar10   -0.2885 0.0453  -6.3690  0.0000
## xmean   0.0027 0.0004   7.4592  0.0000
##
## $AIC
## [1] -3.523022
##
## $AICc
## [1] -3.521736
##
## $BIC
## [1] -3.415073
```

```
sarima(diff(log(all_p)), p=11, d=0, q=0) # AR(11)
```

```
## initial  value -2.463928
## iter   2 value -2.673834
## iter   3 value -2.815646
## iter   4 value -2.938704
## iter   5 value -3.095876
## iter   6 value -3.120999
## iter   7 value -3.165648
## iter   8 value -3.180609
## iter   9 value -3.236583
## iter  10 value -3.240540
## iter  11 value -3.281513
## iter  12 value -3.296276
## iter  13 value -3.298287
## iter  14 value -3.299164
## iter  15 value -3.299190
## iter  16 value -3.299206
## iter  17 value -3.299207
## iter  18 value -3.299207
## iter  18 value -3.299207
## iter  18 value -3.299207
## final  value -3.299207
## converged
## initial  value -3.280281
## iter   2 value -3.280389
## iter   3 value -3.280485
## iter   4 value -3.280653
## iter   5 value -3.280743
## iter   6 value -3.280770
## iter   7 value -3.280856
## iter   8 value -3.280970
## iter   9 value -3.281014
## iter  10 value -3.281083
## iter  11 value -3.281105
## iter  12 value -3.281109
```

```
## iter  13 value -3.281112
## iter  14 value -3.281113
## iter  15 value -3.281113
## iter  15 value -3.281113
## iter  15 value -3.281113
## final  value -3.281113
## converged
```



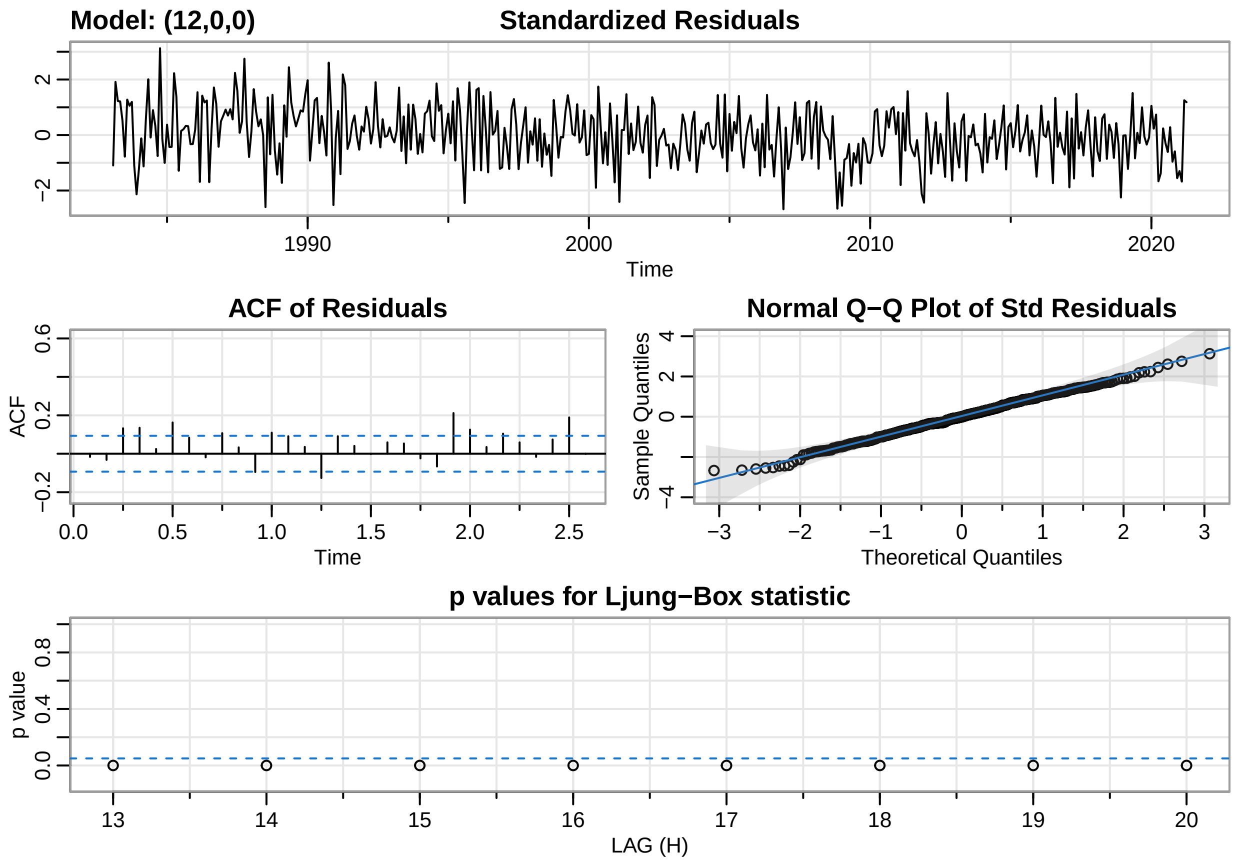

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5      ar6      ar7      ar8
##       -0.8700  -0.7002  -0.3135  -0.6547  -0.4864  -0.6748  -0.5515  -0.7092
## s.e.   0.0432   0.0533   0.0594   0.0518   0.0544   0.0493   0.0538   0.0515
##           ar9     ar10     ar11   xmean
##       -0.4453  -0.5825  -0.3790  0.0027
## s.e.   0.0597   0.0538   0.0439  0.0002
##
## sigma^2 estimated as 0.001388:  log likelihood = 854.74,  aic = -1683.48
##
## $degrees_of_freedom
```

```
## [1] 447
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1    -0.8700 0.0432 -20.1236       0
## ar2    -0.7002 0.0533 -13.1370       0
## ar3    -0.3135 0.0594  -5.2767       0
## ar4    -0.6547 0.0518 -12.6490       0
## ar5    -0.4864 0.0544  -8.9385       0
## ar6    -0.6748 0.0493 -13.6999       0
## ar7    -0.5515 0.0538 -10.2482       0
## ar8    -0.7092 0.0515 -13.7716       0
## ar9    -0.4453 0.0597  -7.4529       0
## ar10   -0.5825 0.0538 -10.8220       0
## ar11   -0.3790 0.0439  -8.6293       0
## xmean   0.0027 0.0002  10.9275       0
##
## $AIC
## [1] -3.667703
##
## $AICc
## [1] -3.666179
##
## $BIC
## [1] -3.550759
```

```
sarima(diff(log(all_p)), p=12, d=0, q=0) # AR(12)
```

```
## initial  value -2.462942
## iter   2 value -2.784242
## iter   3 value -3.100059
## iter   4 value -3.139230
## iter   5 value -3.181578
## iter   6 value -3.218946
## iter   7 value -3.229287
## iter   8 value -3.230789
## iter   9 value -3.255590
## iter  10 value -3.273590
## iter  11 value -3.280609
## iter  12 value -3.294054
## iter  13 value -3.315383
## iter  14 value -3.316446
## iter  15 value -3.317740
## iter  16 value -3.317794
## iter  17 value -3.317797
## iter  18 value -3.317797
## iter  18 value -3.317797
## iter  18 value -3.317797
## final  value -3.317797
## converged
## initial  value -3.297174
## iter   2 value -3.297241
## iter   3 value -3.297365
## iter   4 value -3.297714
## iter   5 value -3.297841
```

```
## iter   6 value -3.297937
## iter   7 value -3.298107
## iter   8 value -3.298137
## iter   9 value -3.298179
## iter  10 value -3.298247
## iter  11 value -3.298308
## iter  12 value -3.298346
## iter  13 value -3.298354
## iter  14 value -3.298356
## iter  15 value -3.298358
## iter  16 value -3.298358
## iter  16 value -3.298358
## iter  16 value -3.298358
## final  value -3.298358
## 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      ar4      ar5      ar6      ar7      ar8
##       -0.7999  -0.5945  -0.2345  -0.5252  -0.3829  -0.5479  -0.4619  -0.5904
```

```
## s.e.   0.0459   0.0586   0.0616   0.0602   0.0594   0.0577   0.0574   0.0587
##           ar9     ar10     ar11     ar12    xmean
##       -0.3865  -0.4476  -0.2118  0.1880  0.0027
## s.e.   0.0606   0.0628   0.0600  0.0468  0.0003
##
## sigma^2 estimated as 0.00134:  log likelihood = 862.65,  aic = -1697.31
##
## $degrees_of_freedom
## [1] 446
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1    -0.7999 0.0459 -17.4274   0e+00
## ar2    -0.5945 0.0586 -10.1410   0e+00
## ar3    -0.2345 0.0616  -3.8065   2e-04
## ar4    -0.5252 0.0602  -8.7197   0e+00
## ar5    -0.3829 0.0594  -6.4504   0e+00
## ar6    -0.5479 0.0577  -9.4971   0e+00
## ar7    -0.4619 0.0574  -8.0456   0e+00
## ar8    -0.5904 0.0587 -10.0566   0e+00
## ar9    -0.3865 0.0606  -6.3819   0e+00
## ar10   -0.4476 0.0628  -7.1330   0e+00
## ar11   -0.2118 0.0600  -3.5283   5e-04
## ar12    0.1880 0.0468   4.0186   1e-04
## xmean   0.0027 0.0003   9.1108   0e+00
##
## $AIC
## [1] -3.697837
##
## $AICc
## [1] -3.696054
##
## $BIC
## [1] -3.571896
```

```
# all red meat
rmp.yw = ar.yw(all_rmp, order=12)
rmp.yw$x.mean # mean estimate
```

Yule-Walker Estimation

```
## [1] 3738.141
```

```
rmp.yw$ar # phi parameter estimates
```

```
##  [1]  0.44075593  0.09600863  0.41482443 -0.43528207  0.38147023 -0.29479448
##  [7]  0.19712510 -0.25245246  0.38504390 -0.30020108  0.15334287  0.19541801
```

```
sqrt(diag(rmp.yw$asy.var.coef)) # their standard errors
```

```
##  [1] 0.04638647 0.05033586 0.04850472 0.04905097 0.05183909 0.05409165
##  [7] 0.05409165 0.05183909 0.04905097 0.04850472 0.05033586 0.04638647
```

```
# all poultry
p.yw = ar.yw(all_p, order=12)
p.yw$x.mean # mean estimate
```

```
## [1] 2957.05
```

```
p.yw$ar # phi parameter estimates
```

```
##  [1]  0.55792085  0.19755941  0.43744646 -0.47996689  0.34886169 -0.25032792
##  [7]  0.20226884 -0.26970524  0.39375872 -0.29757828  0.07303883  0.07698175
```

```
sqrt(diag(p.yw$asy.var.coef)) # their standard errors
```

```
##  [1] 0.04715802 0.05392875 0.05289159 0.05365407 0.05684534 0.05841348
##  [7] 0.05841348 0.05684534 0.05365407 0.05289159 0.05392875 0.04715802
```