Project_2.R

erbolaliev

2020-12-04

```
setwd("~/Downloads")
dat <- read.table("mortgage.txt")
names(dat) <- c("year", 'month','day','morg','ffr')
dat <- dat[-1,]
rownames(dat) <- 1:nrow(dat)
dim(dat)
```

```
## [1] 488   5
```

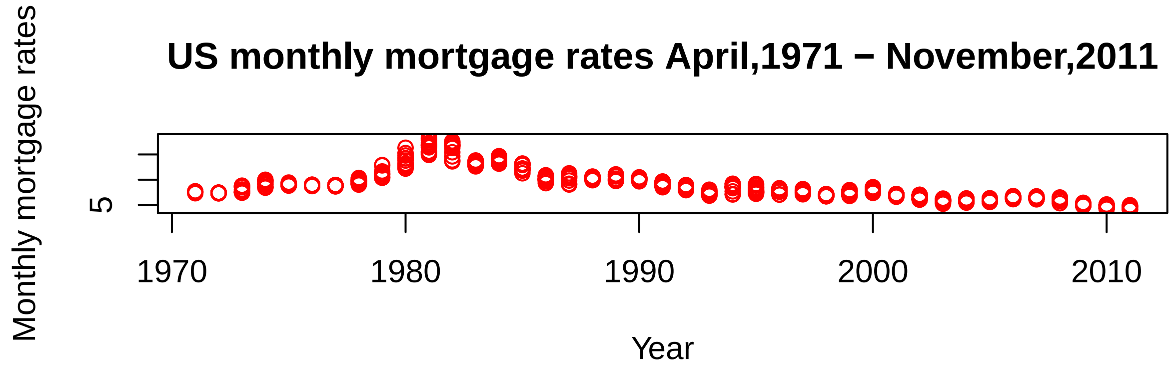
```
#morg = monthly mortgage rate(aka MMR) <- RED COLOR
#ffr = monthly federal funds rate(aka MFFR) <- BLUE COLOR

library(astsa)
library(ggplot2)
library(tseries)
```

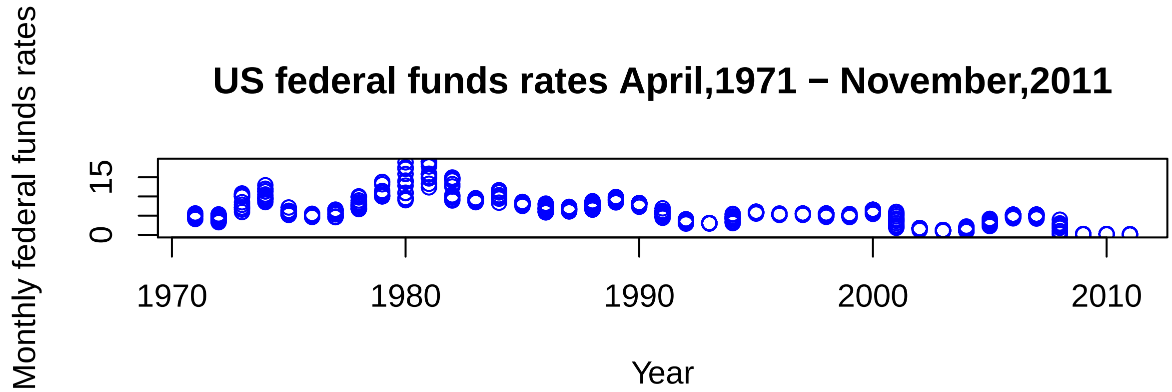
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
#(A) Explain the data, why it is a time series data.
```

Answer: It is a time series data because the same types of data(in our case: mortgage and

```
# federal funds rates) are recorded on a regular basis(monthly). Furthermore the data is represented
# as a collection of random variables indexed according to the order they were obtained in time.

# THIS IS JUST TIME SERIES OF EACH MMR AND MFFR
morg_ts <- ts(as.double(dat$morg), start=c(1971,4), frequency = 12)
ffr_ts <- ts(as.double(dat$ffr), start=c(1971,4), frequency = 12)


#(B) Use exploratory analysis  techniques to describe the data.
#Graph-1 (Scatter Plots of all observations)
par(mfrow=2:1)
plot.ts(dat$year, dat$morg, col = 'red', ylab="Monthly mortgage rates", xlab="Year", main = "US monthly
plot.ts(dat$year, dat$ffr, col = 'blue', ylab="Monthly federal funds rates", xlab="Year", main = "US fe
```

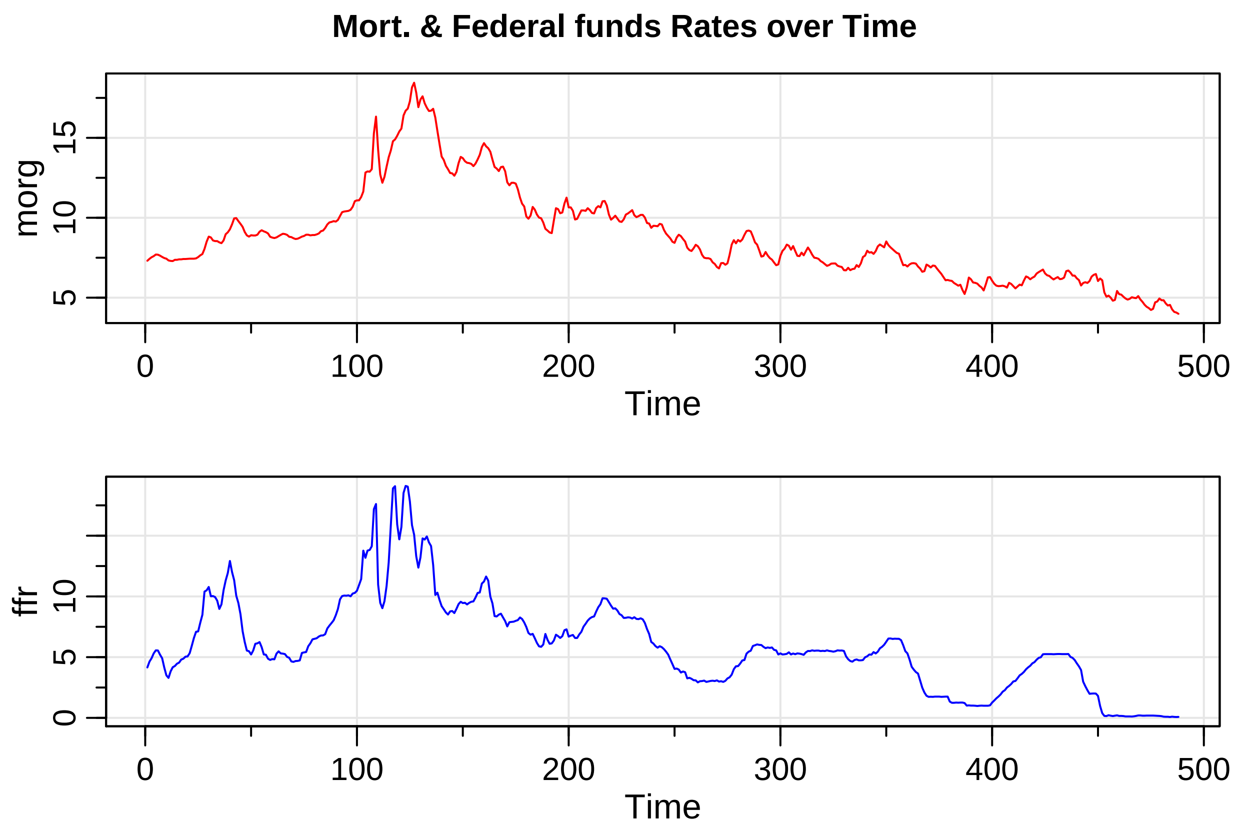




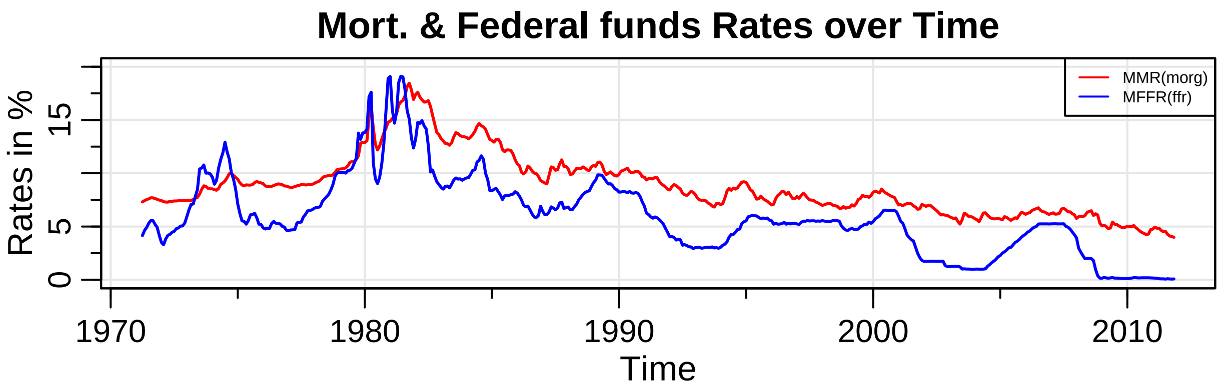

```
#Graph-2 (Multiple type Line Graph)
time_series <- dat[1:488, c("morg", 'ffr')]
tsplot(time_series,
        plot.type = "multiple",
        nc = 1,
        main = "Mort. & Federal funds Rates over Time",
        col=c('red', 'blue')
        )
```

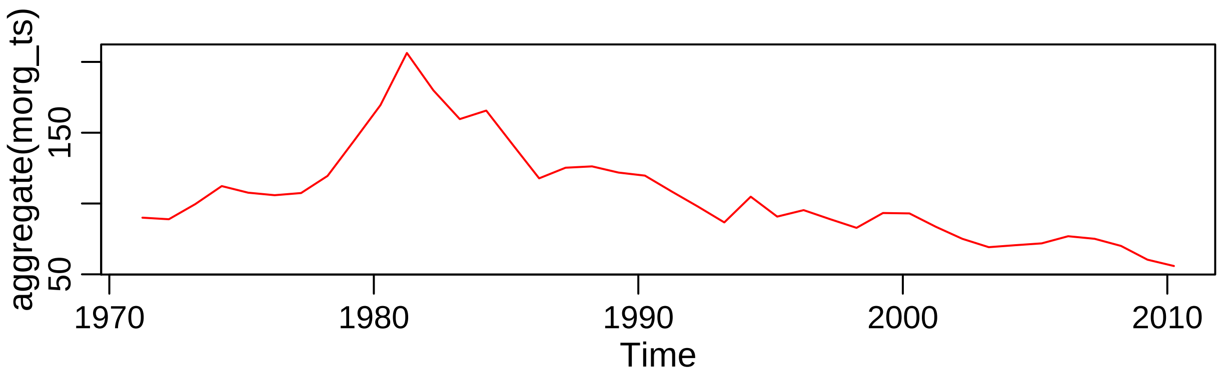
```
## Warning in plot.window(...): "plot.type" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "plot.type" is not a graphical parameter

## Warning in plot.window(...): "plot.type" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "plot.type" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "plot.type" is not
## a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "plot.type" is not
## a graphical parameter

## Warning in box(...): "plot.type" is not a graphical parameter

## Warning in title(...): "plot.type" is not a graphical parameter

## Warning in plot.window(...): "plot.type" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "plot.type" is not a graphical parameter
```

```
## Warning in plot.window(...): "plot.type" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "plot.type" is not a graphical parameter
## Warning in axis(side = side, at = at, labels = labels, ...): "plot.type" is not
## a graphical parameter
## Warning in axis(side = side, at = at, labels = labels, ...): "plot.type" is not
## a graphical parameter
## Warning in box(...): "plot.type" is not a graphical parameter
## Warning in title(...): "plot.type" is not a graphical parameter
```

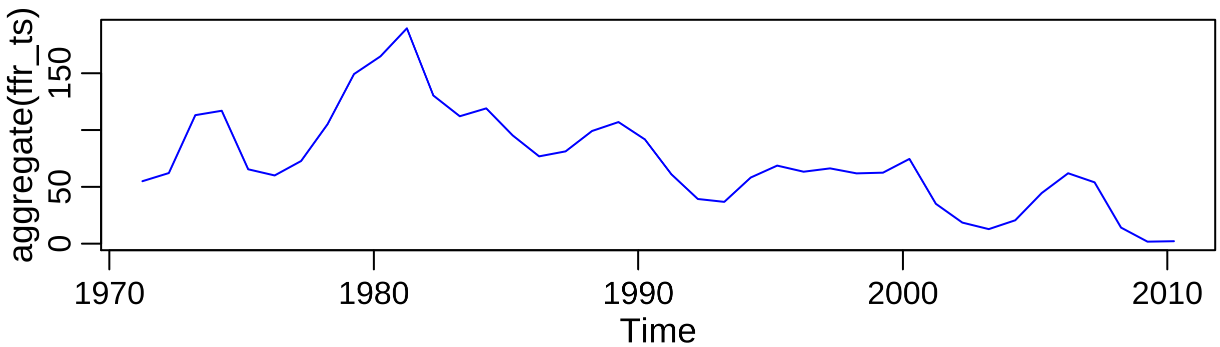


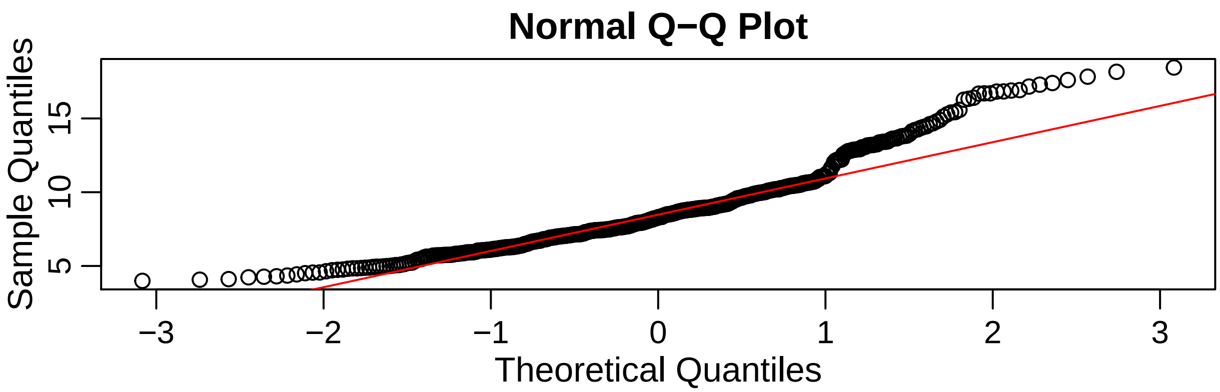

```r
#Graph-3 (Single type Line Graph)
tsplot(morg_ts, ylim=c(0, 20), col = 'red', lwd=1.5, type = 'l', pch = 15, ylab = 'Rates in %', main =
lines(ffr_ts, col = 'blue', lwd =1.5, type = 'l', pch = 15)
legend("topright",
       c("MMR(morg)", "MFFR(ffr)"),
       col=c("red", "blue"),
       lty=1,
       cex = 0.5)
#Graph-4 (Aggregated Data and QQplot)
plot(aggregate(morg_ts), col = 'red') #this is aggregate anual
```






```
plot(aggregate(ffr_ts), col ='blue')
#AND
qqnorm((morg_ts))
qqline((morg_ts), col = 'red')
```



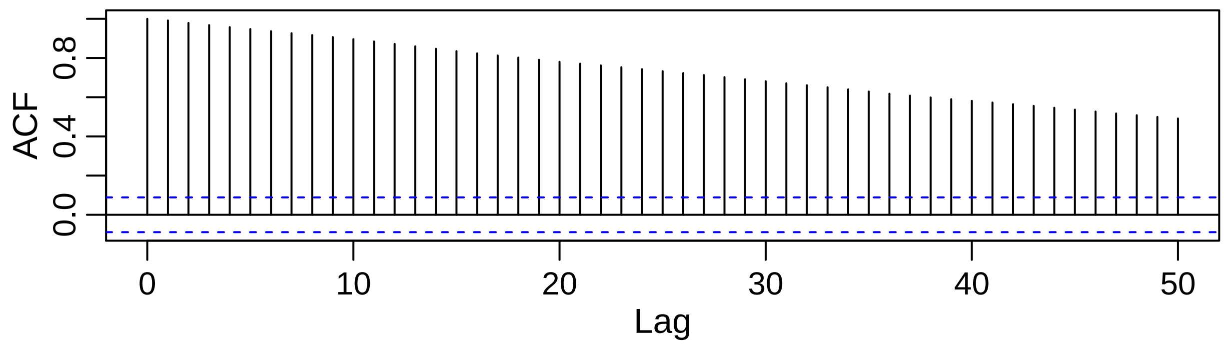


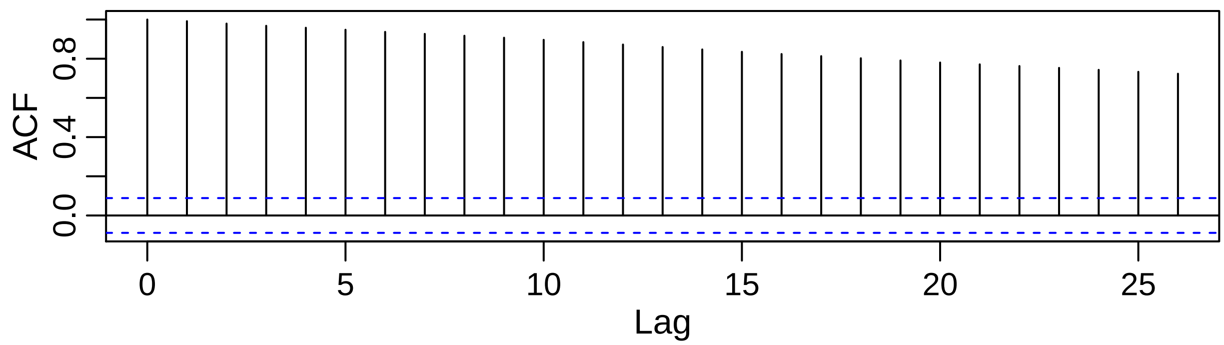

```r
#shows that the data is not normally distributed

#(C) Determine if the monthly mortgage rate series is stationary. If the series
#was not stationary apply appropriate transformation(s) to make the transformed series stationary.
dat_acf <- acf(as.numeric(dat$morg), lag = 50)
data.frame(lag = dat_acf$lag,
           acf = dat_acf$acf)
```

```
##    lag       acf
## 1    0 1.0000000
## 2    1 0.9917260
## 3    2 0.9793506
## 4    3 0.9680841
## 5    4 0.9581258
## 6    5 0.9479082
## 7    6 0.9370049
## 8    7 0.9267950
## 9    8 0.9172375
## 10   9 0.9071628
## 11  10 0.8964723
## 12  11 0.8847726
## 13  12 0.8723937
## 14  13 0.8597532
## 15  14 0.8472817
## 16  15 0.8352684
## 17  16 0.8239484
```

```
## 18   17 0.8131428
## 19   18 0.8023476
## 20   19 0.7911628
## 21   20 0.7805873
## 22   21 0.7711726
## 23   22 0.7626010
## 24   23 0.7533268
## 25   24 0.7431773
## 26   25 0.7331229
## 27   26 0.7233973
## 28   27 0.7133020
## 29   28 0.7027802
## 30   29 0.6919313
## 31   30 0.6816213
## 32   31 0.6712089
## 33   32 0.6611371
## 34   33 0.6509971
## 35   34 0.6403339
## 36   35 0.6291118
## 37   36 0.6179982
## 38   37 0.6080598
## 39   38 0.5989266
## 40   39 0.5900941
## 41   40 0.5817001
## 42   41 0.5731845
## 43   42 0.5647504
## 44   43 0.5557768
## 45   44 0.5462389
## 46   45 0.5365749
## 47   46 0.5269671
## 48   47 0.5171787
## 49   48 0.5081181
## 50   49 0.4995996
## 51   50 0.4915325
```

```r
str(acf(as.numeric(dat$morg)))
```






```
## List of 6
##  $ acf   : num [1:27, 1, 1] 1 0.992 0.979 0.968 0.958 ...
##  $ type  : chr "correlation"
##  $ n.used: int 488
##  $ lag   : num [1:27, 1, 1] 0 1 2 3 4 5 6 7 8 9 ...
##  $ series: chr "as.numeric(dat$morg)"
##  $ snames: NULL
##  - attr(*, "class")= chr "acf"
```

```
adf.test(as.numeric(dat$morg)) # p-val > 0.05 indicates the TS is not stationary
```

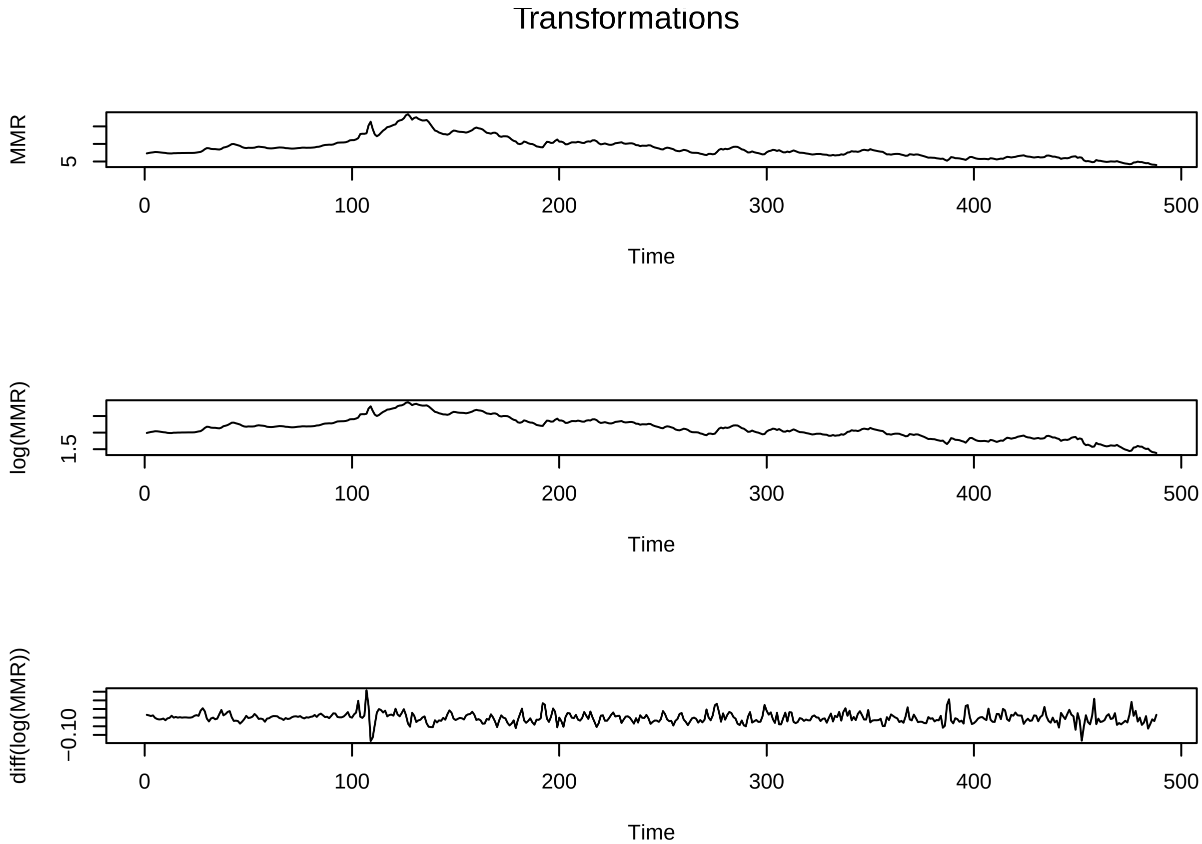
```
##
##  Augmented Dickey-Fuller Test
##
## data:  as.numeric(dat$morg)
## Dickey-Fuller = -2.3071, Lag order = 7, p-value = 0.4482
## alternative hypothesis: stationary
```

Answer: We can see from part (B) that the mean is not constant over time. Also, prolonged or

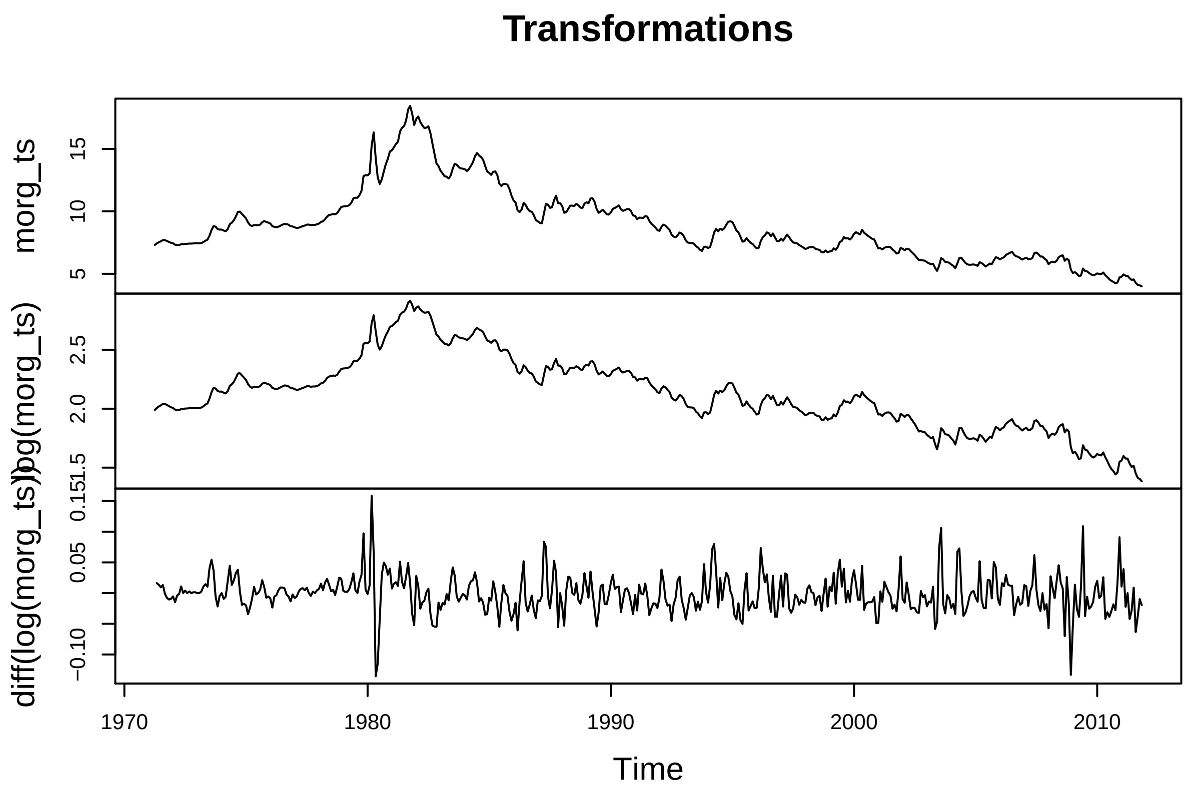
```
# slowly decaying ACF indicates that MMR is not stationary. We can confirm that by performing
# Augmented Dickey-Fuller Test, which resulted in p-val > 0.05 confirming that MMR is NOT stationary.
x <- as.numeric(dat$morg)
dat_more <- cbind(x, log(x), diff(log(x)))
```

```
## Warning in cbind(x, log(x), diff(log(x))): number of rows of result is not a
## multiple of vector length (arg 3)
```

```r
par(mfrow=c(3,1))
plot(dat_more[,1], type = 'l', xlab = "Time", ylab = 'MMR')
plot(dat_more[,2], type = 'l', xlab = "Time", ylab = 'log(MMR)')
plot(dat_more[,3], type = 'l', xlab = "Time", ylab = 'diff(log(MMR))')
mtext("Transformations", side = 3, line = -1, outer = TRUE)
```



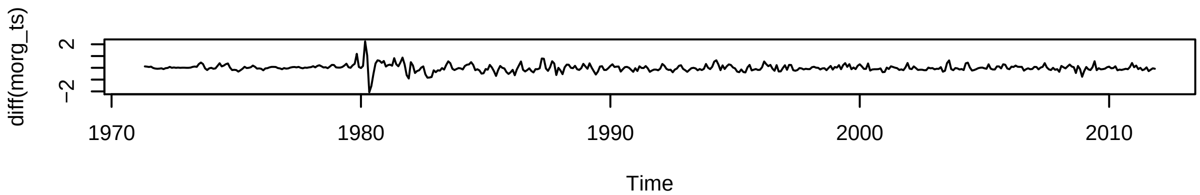

```r
#OR
plot(cbind(morg_ts, log(morg_ts), diff(log(morg_ts))), main="Transformations")
```

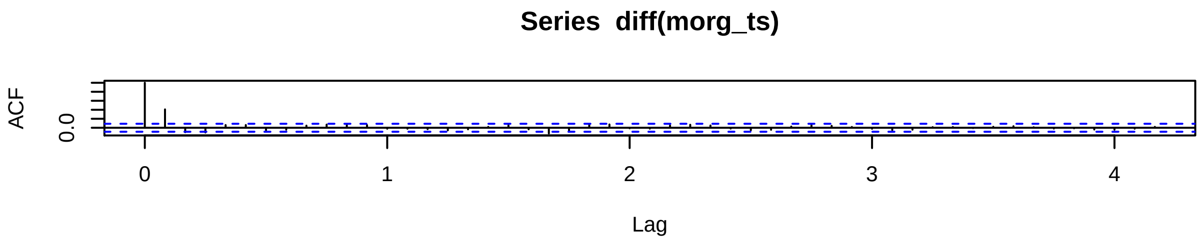


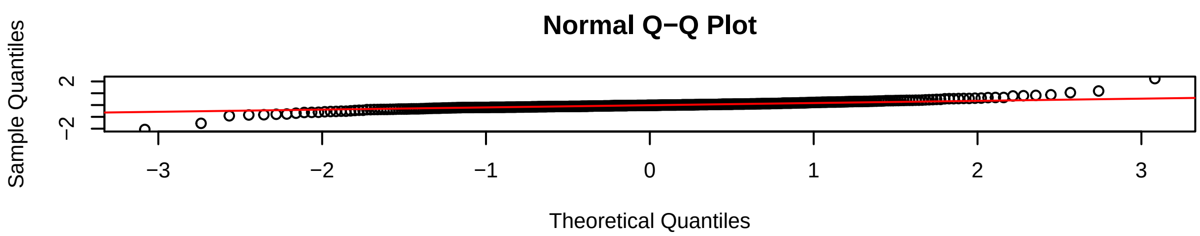

```r
#AND to TRANSFORM
plot(diff(morg_ts), type="l")
mean(diff(morg_ts)) # <- drift estimate = -0.007
```

```
## [1] -0.006817248
```

```r
acf(diff(morg_ts), 50)
qqnorm(diff(morg_ts))
qqline(diff(morg_ts), col = 'red')
```

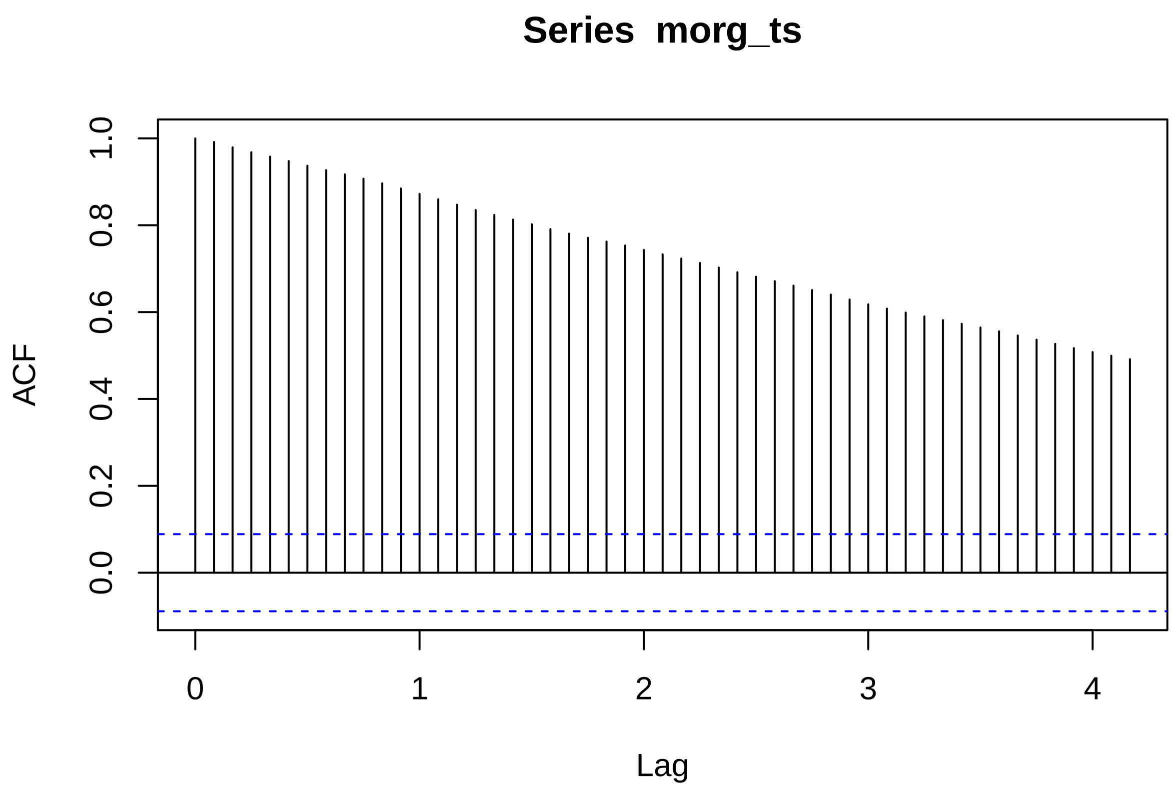






Answer: Differencing technique was used to coerce the MMR series into stationarity.

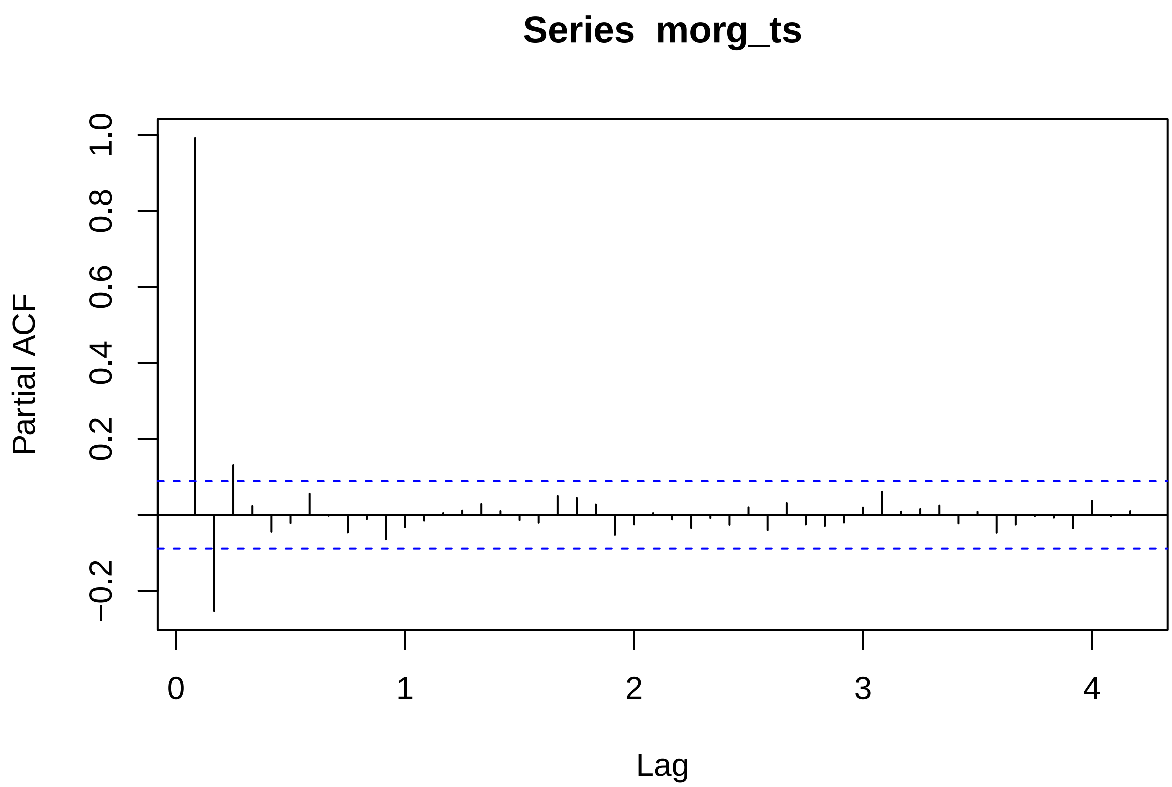
```
# The 1st order differencing makes it look like stationary.We see that ACF has changed,
# it shows exponential decay. Which is an indicator of a stationary series.
# We can also argue that the approximation to normality is improved by the transformation.

#(D) Compute sample ACF and PACF functions for the monthly mortgage rate(MMR)
# and explain their meanings. Is there any model suggested by
# the autocorrelation or partial autocorrelation functions?
acf(morg_ts, 50)
```



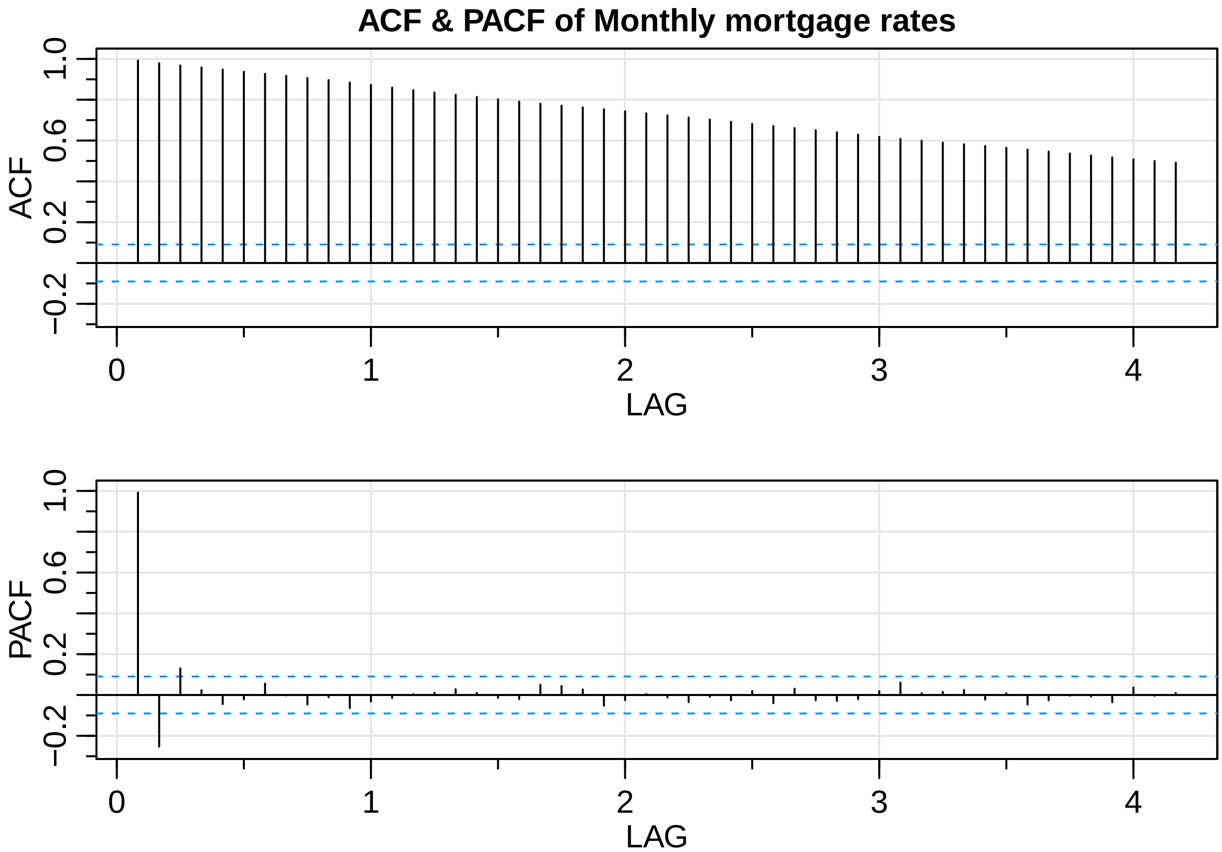

```
r <- acf1(morg_ts, 50, plot=F) #sample ACF values
head(r,10)
```

```
## [1] 0.9917260 0.9793506 0.9680841 0.9581258 0.9479082 0.9370049 0.9267950
## [8] 0.9172375 0.9071628 0.8964723
```

```
pacf(morg_ts,50)
```




```
#OR we can obtain it both by:
acf2(morg_ts, 50, main = "ACF & PACF of Monthly mortgage rates")
```




```
##       [,1]  [,2] [,3] [,4]  [,5]  [,6] [,7] [,8]  [,9] [,10] [,11] [,12] [,13]
## ACF  0.99  0.98 0.97 0.96  0.95  0.94 0.93 0.92  0.91  0.90  0.88  0.87  0.86
## PACF 0.99 -0.25 0.13 0.02 -0.04 -0.02 0.06 0.00 -0.05 -0.01 -0.06 -0.03 -0.02
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF   0.85  0.84  0.82  0.81  0.80  0.79  0.78  0.77  0.76  0.75  0.74  0.73
## PACF  0.00  0.01  0.03  0.01 -0.01 -0.02  0.05  0.04  0.03 -0.05 -0.03  0.00
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
## ACF   0.72  0.71  0.70  0.69  0.68  0.67  0.66  0.65  0.64  0.63  0.62  0.61
## PACF -0.01 -0.03 -0.01 -0.03  0.02 -0.04  0.03 -0.03 -0.03 -0.02  0.02  0.06
##      [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48] [,49]
## ACF   0.60  0.59  0.58  0.57  0.56  0.56  0.55  0.54  0.53  0.52  0.51   0.5
## PACF  0.01  0.02  0.02 -0.02  0.01 -0.05 -0.03  0.00 -0.01 -0.04  0.04   0.0
##      [,50]
## ACF   0.49
## PACF  0.01
```

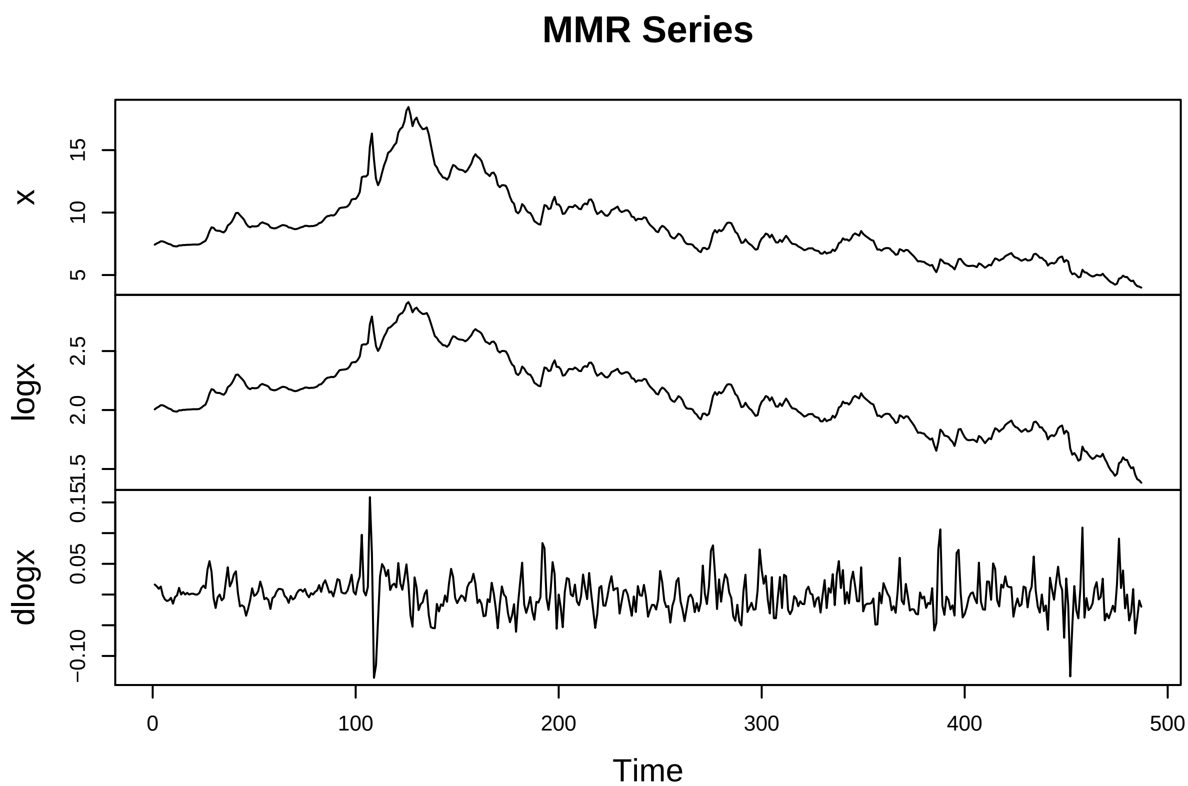
Answer: ACF tails off meaning that it is an AR model, we can also see PACF cuts of after

```
# lag (2) and then  is esentially 0 for higher lags. These results suggest a second-order
# autoregressive model might provide a good fit, i.e. AR(2) process has been indicated.


#(E) Build an ARIMA model for the MMR. Perform model checking and write down the fitted
# model. If there are competing models that fit the data, determine your model selection criterion.
arima(morg_ts)
```

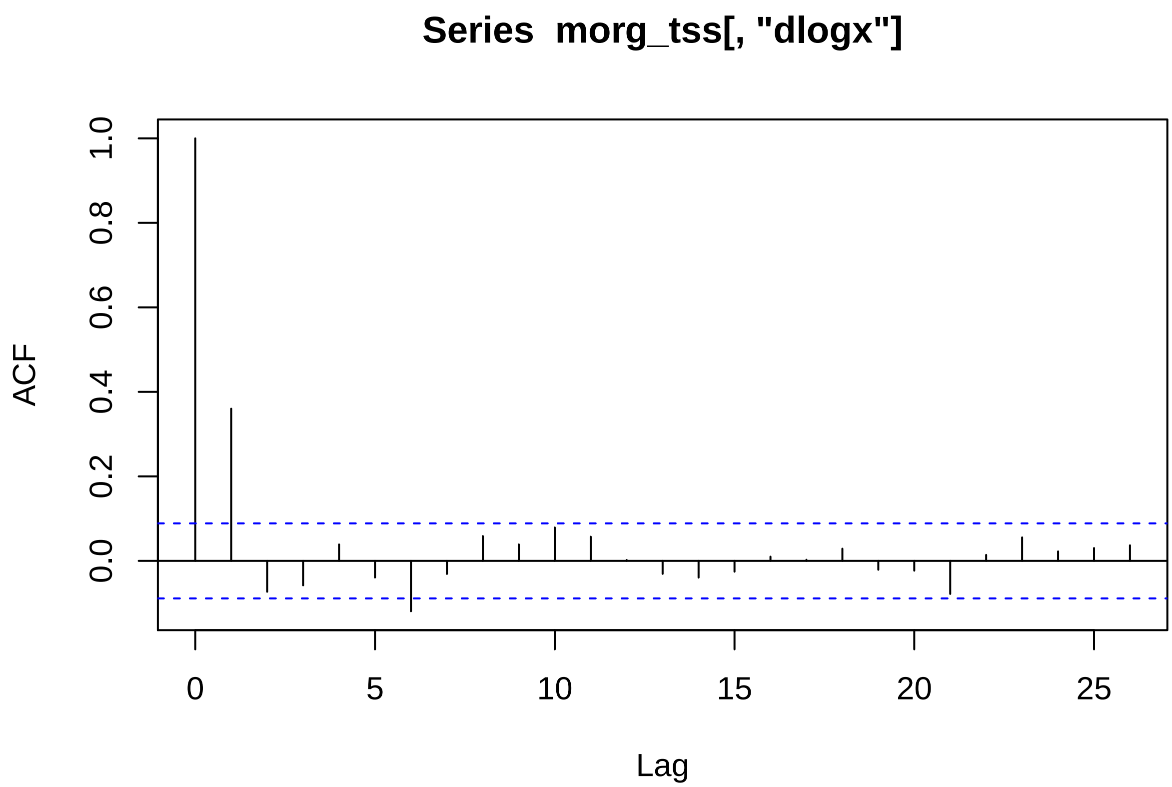
```
##
```

```
## Call:
## arima(x = morg_ts)
##
## Coefficients:
##       intercept
##          8.7996
## s.e.     0.1313
##
## sigma^2 estimated as 8.414:  log likelihood = -1212.13,  aic = 2428.25
```

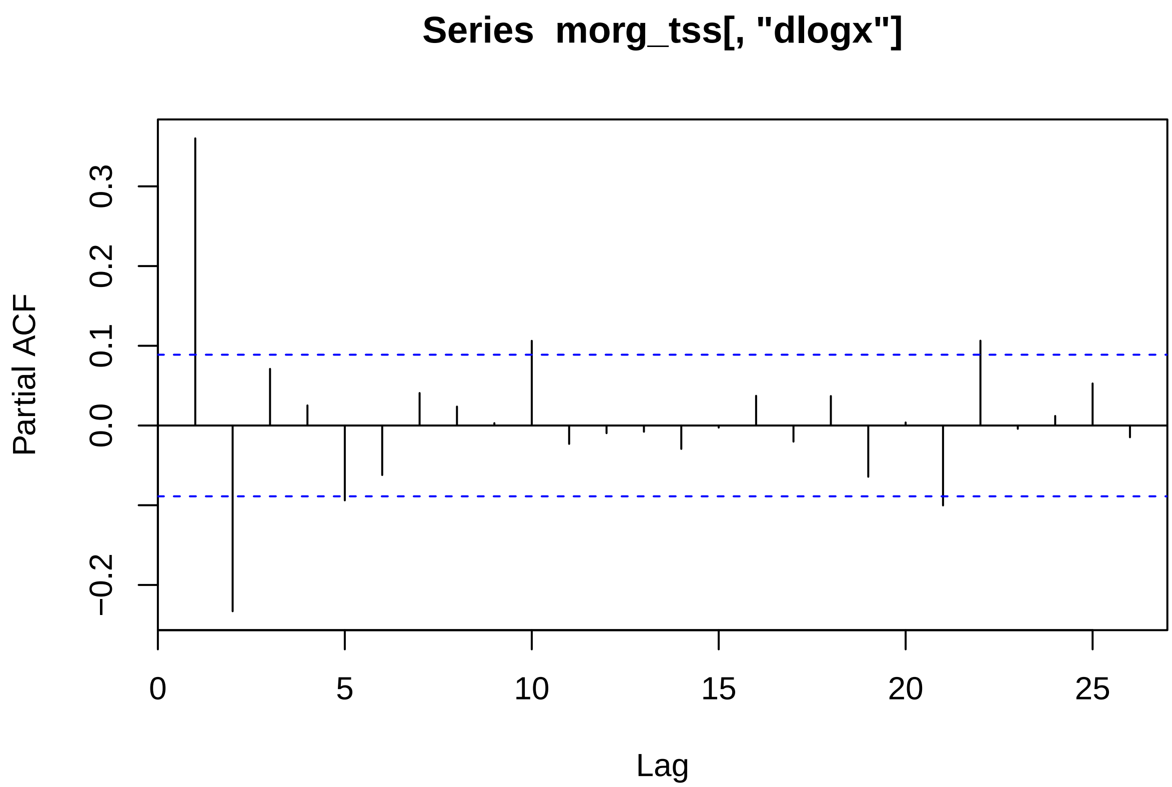
```
morg_tss <- ts(data.frame(x = morg_ts[-1],
                          logx = log(morg_ts)[-1],
                          dlogx = diff(log(morg_ts))))
plot(morg_tss, main = "MMR Series")
```



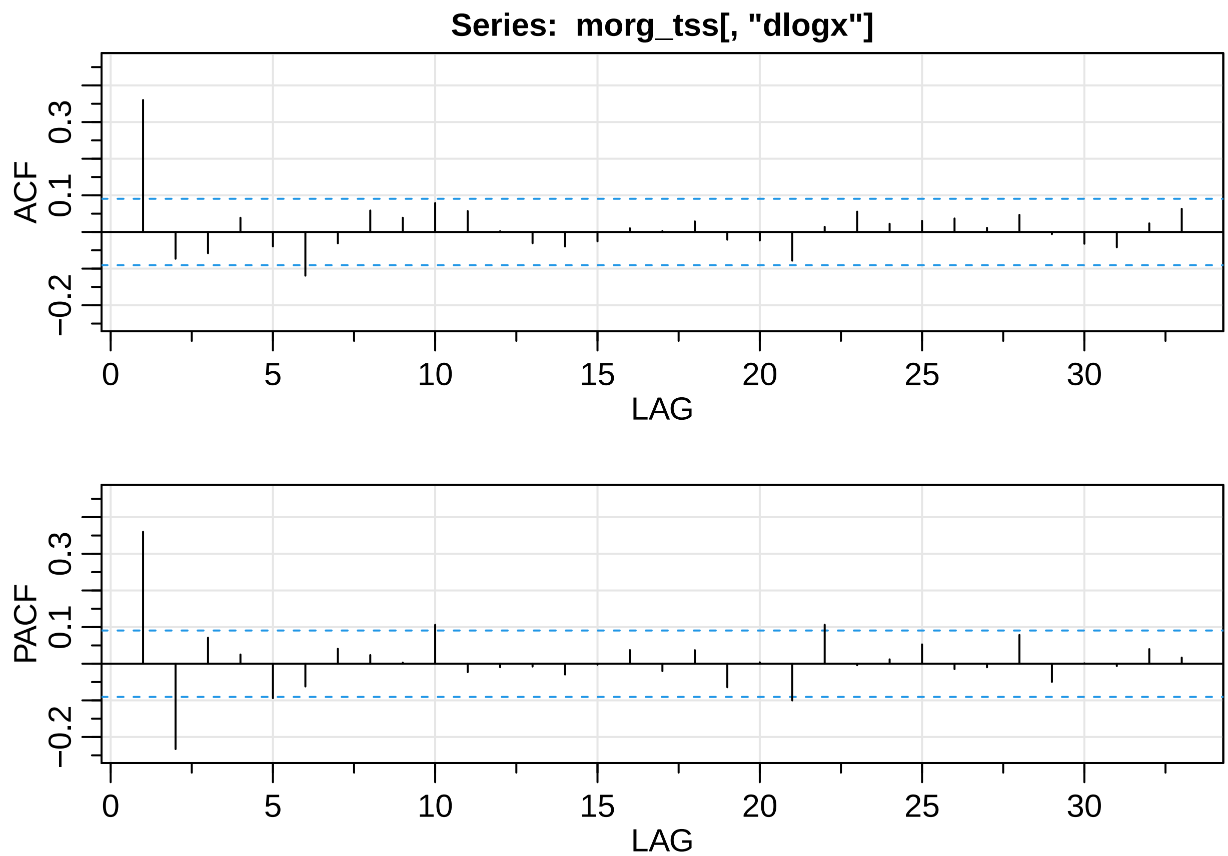

```
acf(morg_tss[, "dlogx"])
```




```
pacf(morg_tss[, "dlogx"])
```




```
#OR
acf2(morg_tss[, "dlogx"]) # <- this looks like AR(2), MA(1) or ARMA(2,1)
```

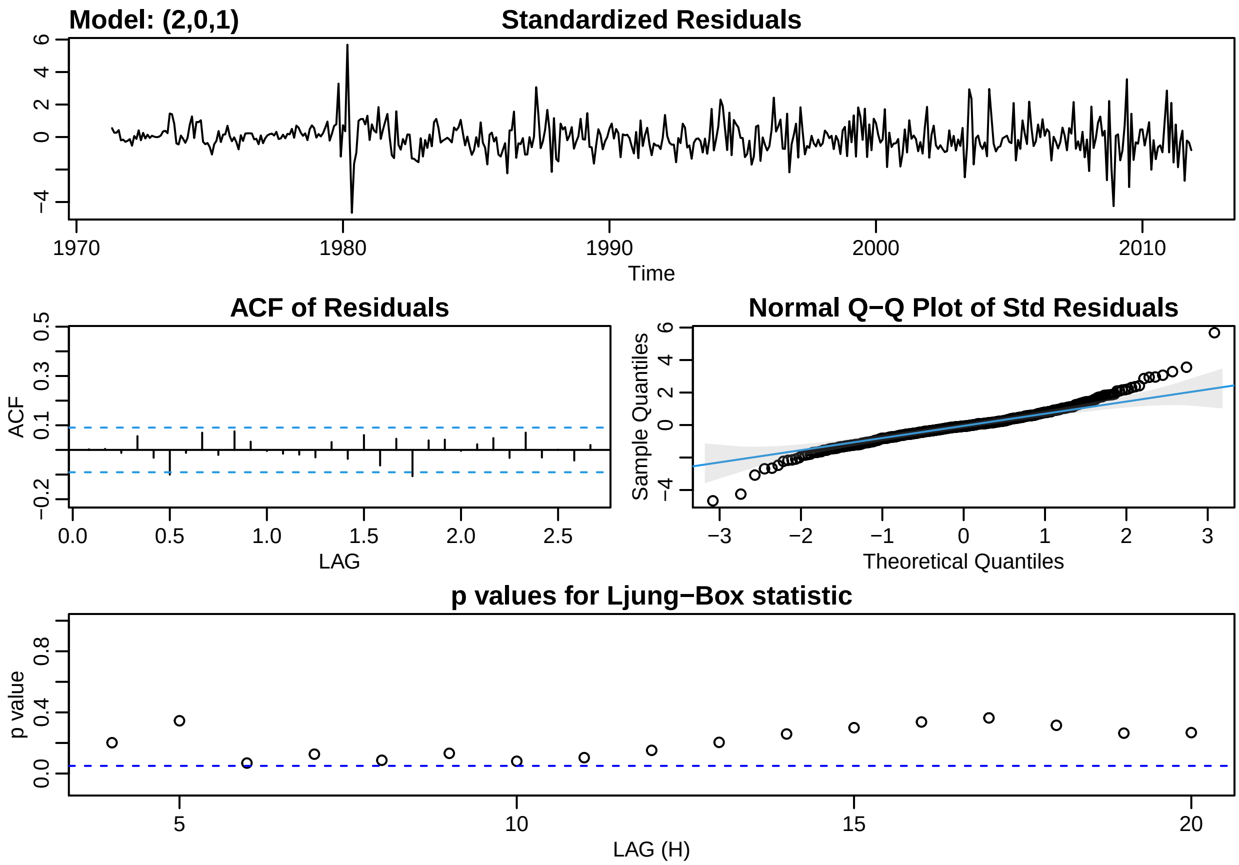



```
##       [,1]  [,2]  [,3] [,4]  [,5]  [,6]  [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## ACF  0.36 -0.07 -0.06 0.04 -0.04 -0.12 -0.03 0.06 0.04  0.08  0.06  0.00 -0.03
## PACF 0.36 -0.23  0.07 0.03 -0.09 -0.06  0.04 0.02 0.00  0.11 -0.02 -0.01 -0.01
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF  -0.04 -0.03  0.01  0.00  0.03 -0.02 -0.02 -0.08  0.01  0.06  0.02  0.03
## PACF -0.03  0.00  0.04 -0.02  0.04 -0.06  0.00 -0.10  0.11  0.00  0.01  0.05
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33]
## ACF   0.04  0.01  0.05 -0.01 -0.03 -0.04  0.02  0.06
## PACF -0.01 -0.01  0.08 -0.05  0.00 -0.01  0.04  0.02
```

```
sarima(diff(log(morg_ts)), p=2, d=0, q=1, no.constant = T)# -4.37 for AIC, -4.33 for BIC
```

```
## initial  value -3.512473
## iter   2 value -3.576250
## iter   3 value -3.610210
## iter   4 value -3.611795
## iter   5 value -3.611868
## iter   6 value -3.611875
## iter   7 value -3.611907
## iter   8 value -3.611927
## iter   9 value -3.611928
## iter  10 value -3.611933
## iter  11 value -3.611933
## iter  12 value -3.611935
## iter  13 value -3.611935
## iter  13 value -3.611935
## iter  13 value -3.611935
```

```
## final  value -3.611935
## converged
## initial  value -3.613365
## iter   1 value -3.613365
## final  value -3.613365
## converged
```

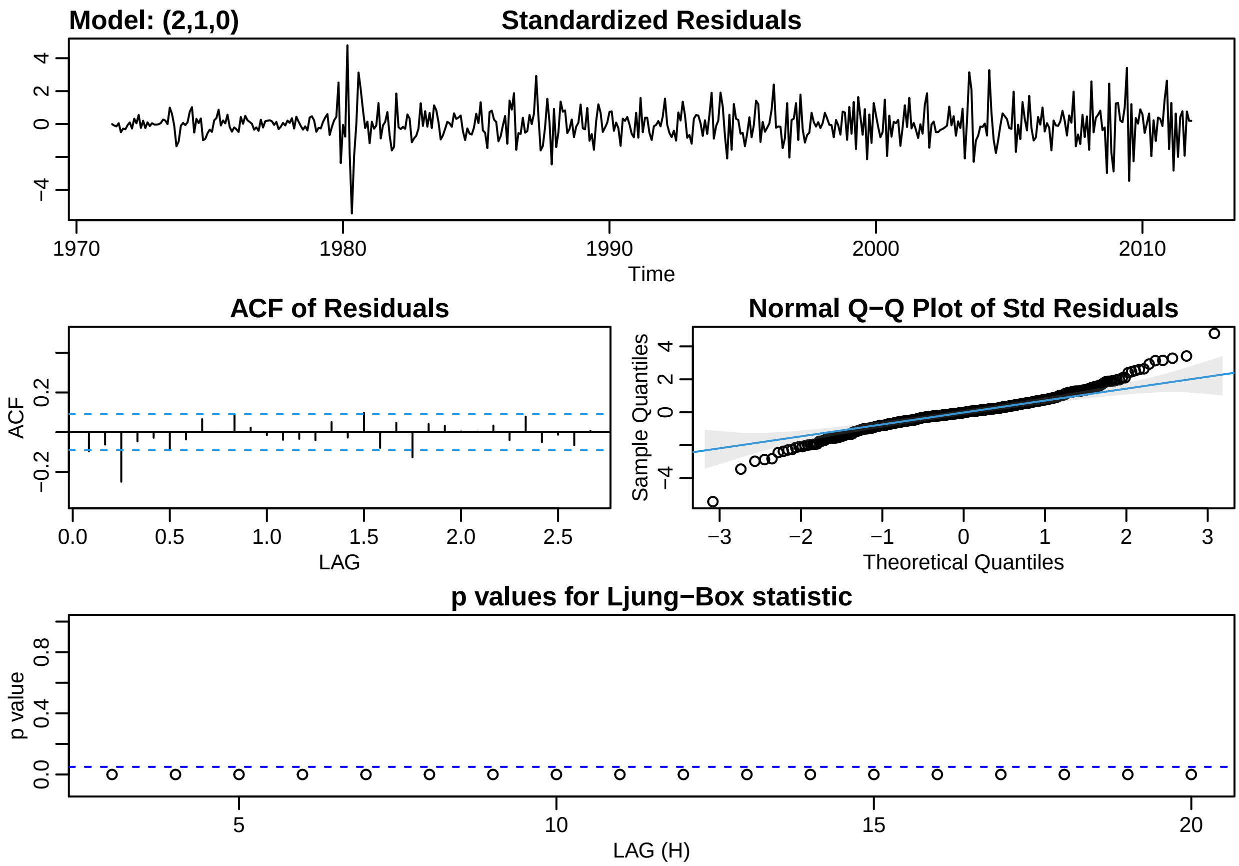



```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), xreg = xmean, include.mean = FALSE, transform.pars = trans,
##     fixed = fixed, optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ar2     ma1
##       0.2523  -0.1642  0.2046
## s.e.  0.1457   0.0716  0.1446
##
## sigma^2 estimated as 0.0007265:  log likelihood = 1068.69,  aic = -2129.37
##
## $degrees_of_freedom
## [1] 484
##
## $ttable
##     Estimate     SE t.value p.value
```

```
## ar1   0.2523 0.1457  1.7319  0.0839
## ar2  -0.1642 0.0716 -2.2934  0.0223
## ma1   0.2046 0.1446  1.4151  0.1577
##
## $AIC
## [1] -4.372426
##
## $AICc
## [1] -4.372324
##
## $BIC
## [1] -4.338026
```

```
sarima(diff(log(morg_ts)), p=2, d=1, q=0, no.constant = T)# -4.12 for AIC, -4.09 for BIC
```

```
## initial  value -3.389149
## iter   2 value -3.481149
## iter   3 value -3.483588
## iter   4 value -3.483858
## iter   5 value -3.483858
## iter   5 value -3.483858
## iter   5 value -3.483858
## final  value -3.483858
## converged
## initial  value -3.485530
## iter   2 value -3.485531
## iter   2 value -3.485531
## iter   2 value -3.485531
## final  value -3.485531
## converged
```

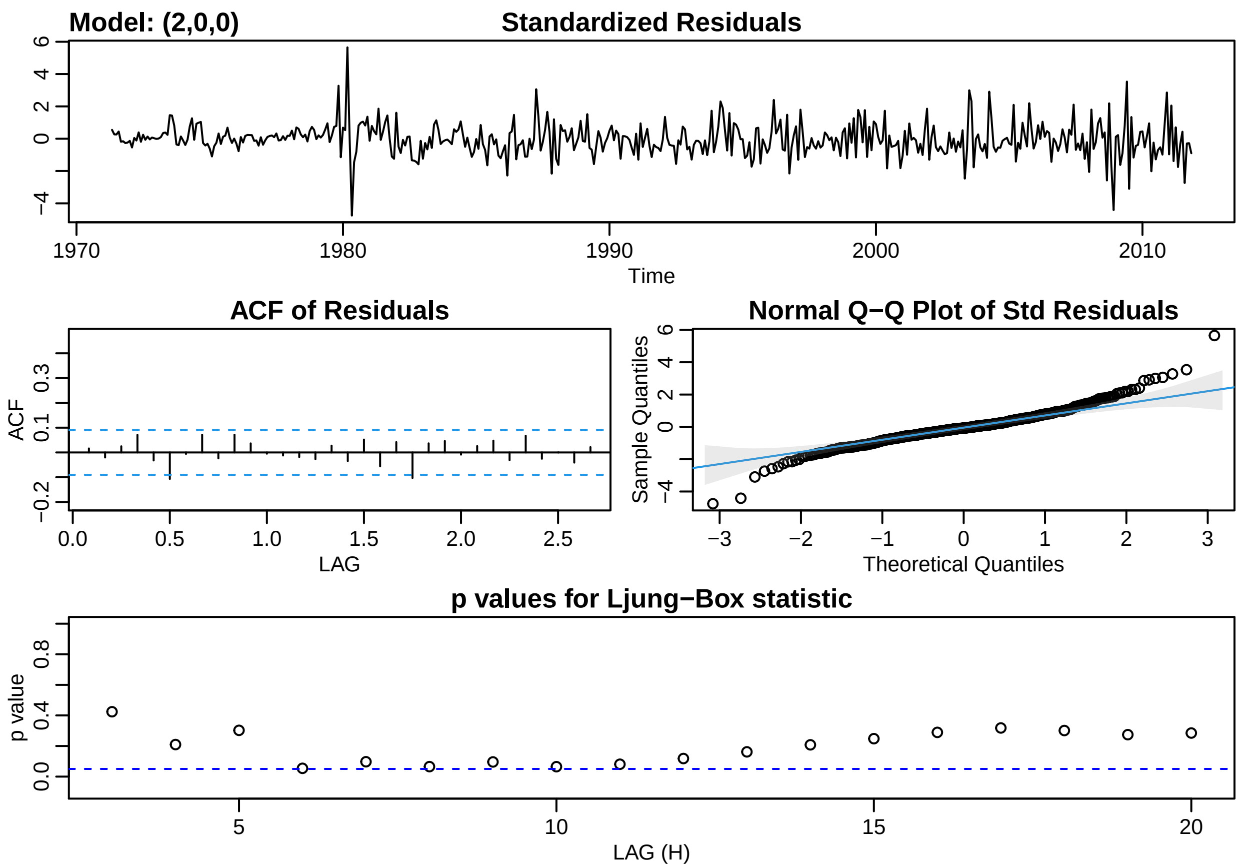



```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2
##       -0.2235  -0.3865
## s.e.   0.0418   0.0417
##
## sigma^2 estimated as 0.000938:  log likelihood = 1004.36,  aic = -2002.73
##
## $degrees_of_freedom
## [1] 484
##
## $ttable
##     Estimate     SE t.value p.value
## ar1  -0.2235 0.0418 -5.3467       0
## ar2  -0.3865 0.0417 -9.2598       0
##
## $AIC
## [1] -4.12084
##
## $AICc
```

```
## [1] -4.120789
##
## $BIC
## [1] -4.094999
```

```
sarima(diff(log(morg_ts)), p=2, d=0, q=0, no.constant = T)# -4.37 for AIC, -4.34 for BIC
```

```
## initial  value -3.512473
## iter   2 value -3.596192
## iter   3 value -3.609053
## iter   4 value -3.610076
## iter   5 value -3.610078
## iter   5 value -3.610078
## iter   5 value -3.610078
## final  value -3.610078
## converged
## initial  value -3.611496
## iter   2 value -3.611497
## iter   2 value -3.611497
## iter   2 value -3.611497
## final  value -3.611497
## converged
```

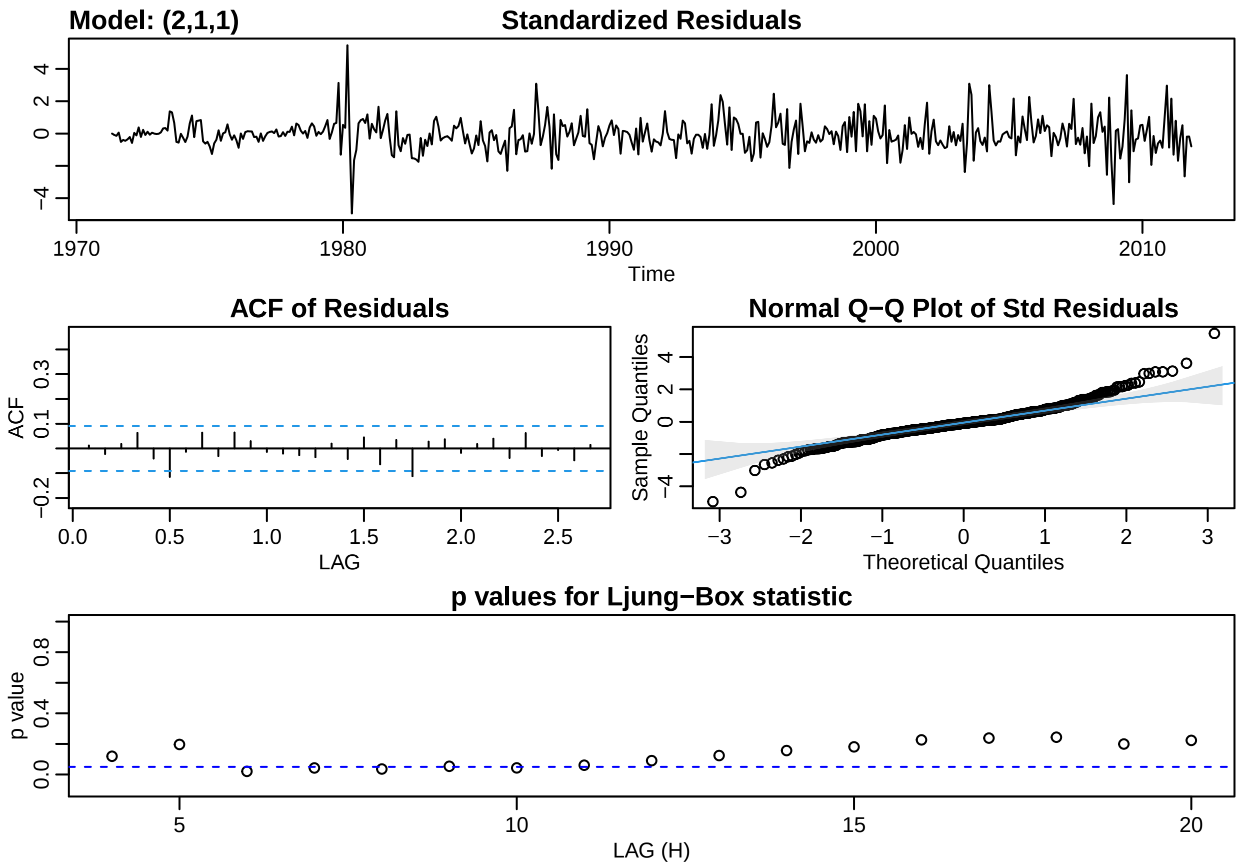



```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
```

```
##     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
##       0.4448  -0.2313
## s.e.  0.0441   0.0440
##
## sigma^2 estimated as 0.0007292:  log likelihood = 1067.78,  aic = -2129.55
##
## $degrees_of_freedom
## [1] 485
##
## $ttable
##     Estimate     SE t.value p.value
## ar1   0.4448 0.0441 10.0974       0
## ar2  -0.2313 0.0440 -5.2535       0
##
## $AIC
## [1] -4.372796
##
## $AICc
## [1] -4.372745
##
## $BIC
## [1] -4.346995
```

```
sarima(diff(log(morg_ts)), p=2, d=1, q=1, no.constant = T)# -4.35 for AIC, -4.32 for BIC
```

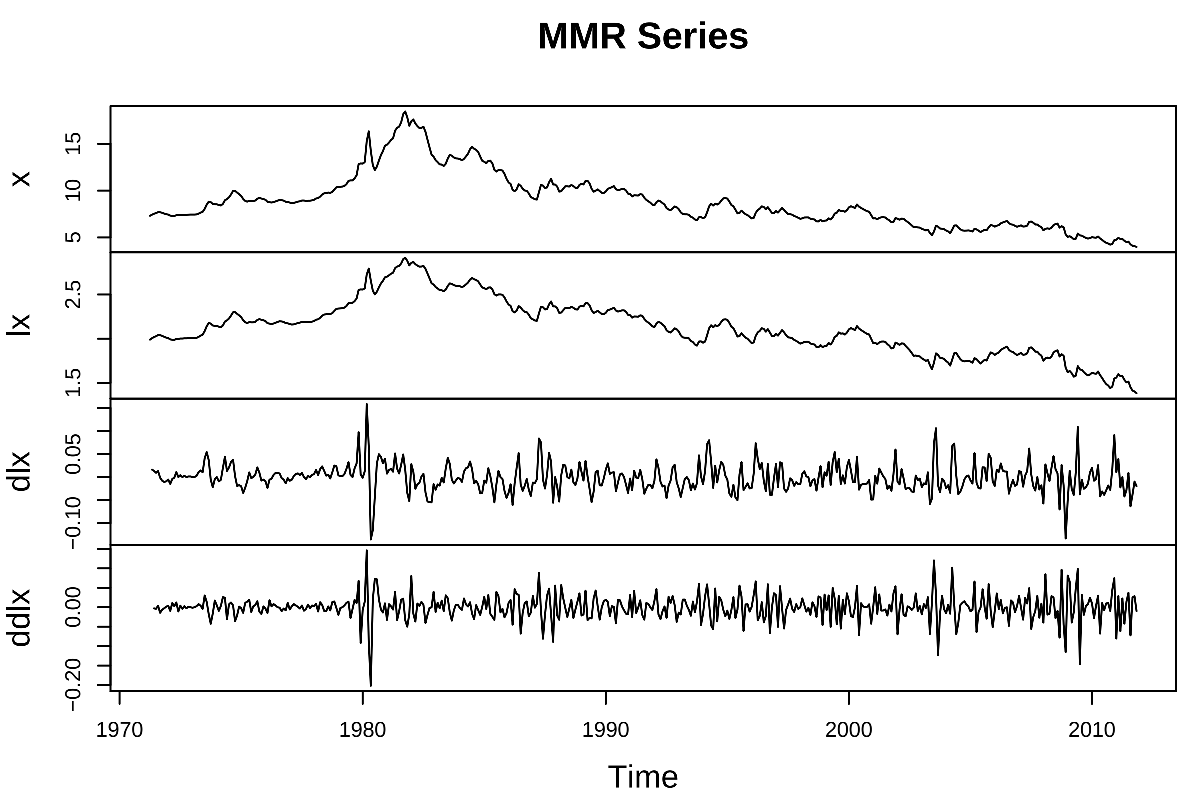
```
## initial  value -3.389149
## iter   2 value -3.495218
## iter   3 value -3.501875
## iter   4 value -3.510771
## iter   5 value -3.561889
## iter   6 value -3.596697
## iter   7 value -3.604364
## iter   8 value -3.605781
## iter   9 value -3.606410
## iter  10 value -3.606597
## iter  11 value -3.606603
## iter  12 value -3.606605
## iter  13 value -3.606605
## iter  13 value -3.606605
## iter  13 value -3.606605
## final  value -3.606605
## converged
## initial  value -3.606272
## iter   2 value -3.606429
## iter   3 value -3.606475
## iter   4 value -3.606475
## iter   4 value -3.606475
## iter   4 value -3.606475
## final  value -3.606475
## converged
```



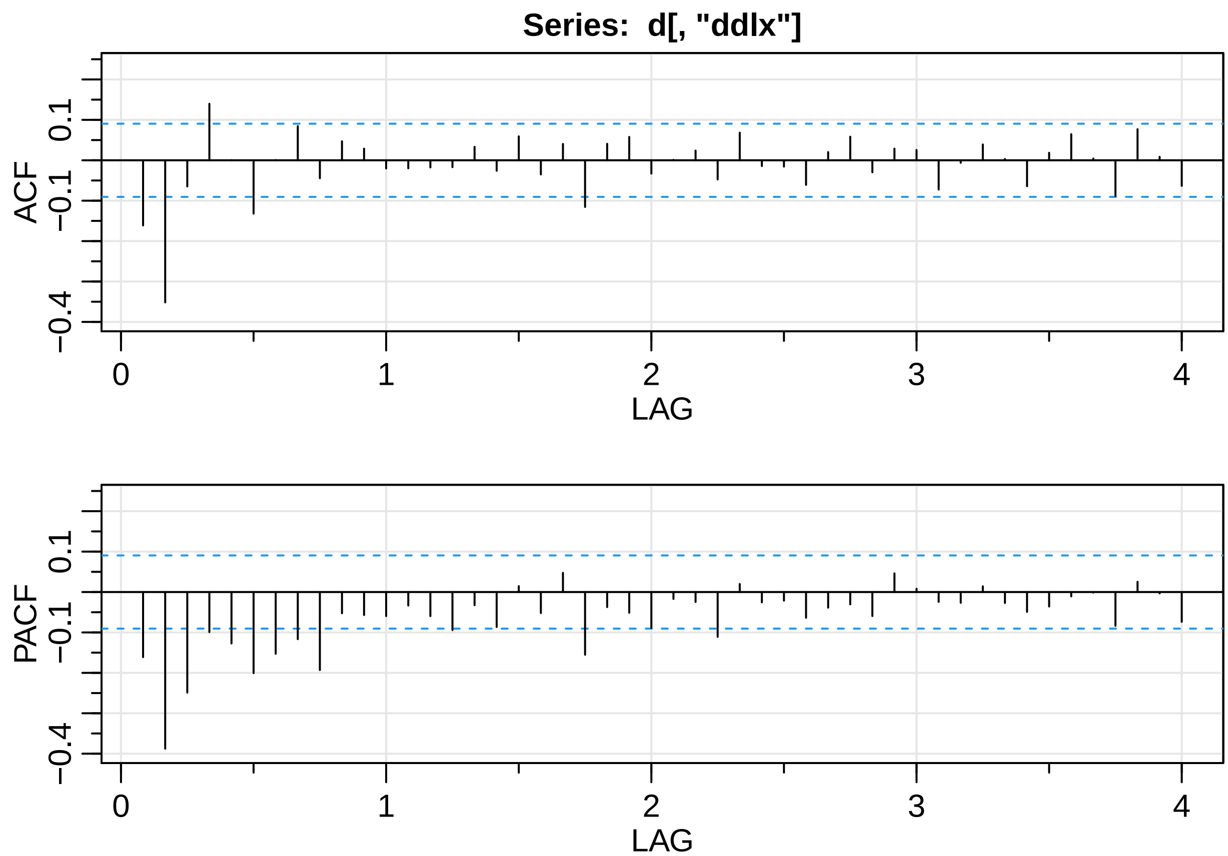

```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ar2      ma1
##       0.4402  -0.2357  -0.9912
## s.e.  0.0443   0.0442   0.0074
##
## sigma^2 estimated as 0.0007312:  log likelihood = 1063.14,  aic = -2118.29
##
## $degrees_of_freedom
## [1] 483
##
## $ttable
##     Estimate     SE   t.value p.value
## ar1   0.4402 0.0443    9.9433       0
## ar2  -0.2357 0.0442   -5.3318       0
## ma1  -0.9912 0.0074 -133.4495       0
##
## $AIC
## [1] -4.358612
##
```

```
## $AICc
## [1] -4.35851
##
## $BIC
## [1] -4.324158
```

```r
#A lower AIC or BIC indicates a better fit, so we stick with AR(2).
d = cbind(
  x = morg_ts,
  lx = log(morg_ts),
  dlx = diff(log(morg_ts)),
  ddlx = diff(diff(log(morg_ts))))
plot(d, main = "MMR Series")
```

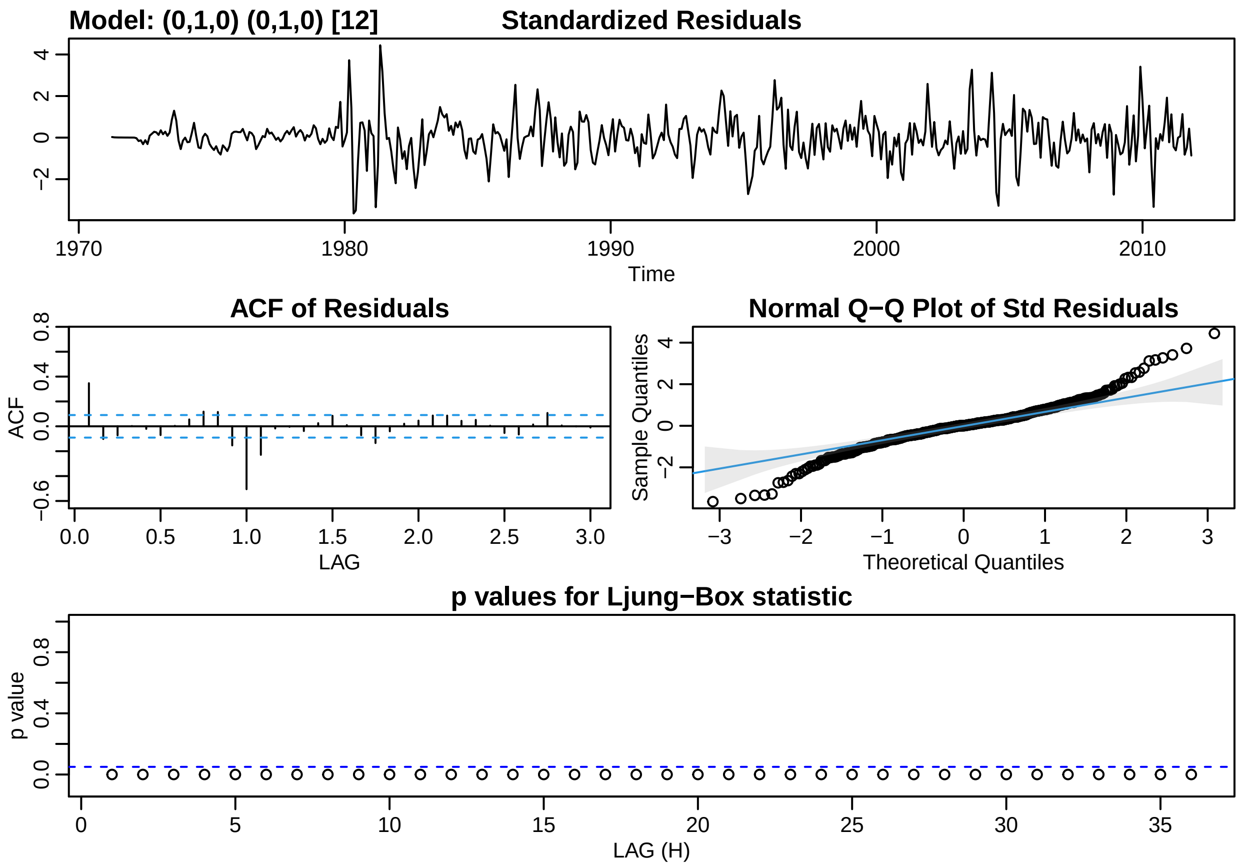



```r
acf2(d[,"ddlx"])
```




```
##       [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10] [,11] [,12]
## ACF  -0.16 -0.35 -0.06  0.14  0.00 -0.13  0.00  0.08 -0.04  0.05  0.03 -0.02
## PACF -0.16 -0.39 -0.25 -0.10 -0.13 -0.20 -0.15 -0.12 -0.19 -0.05 -0.06 -0.06
##      [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## ACF  -0.02 -0.02 -0.02  0.03 -0.03  0.06 -0.04  0.04 -0.12  0.04  0.06 -0.03
## PACF -0.03 -0.06 -0.09 -0.03 -0.09  0.01 -0.05  0.05 -0.16 -0.04 -0.05 -0.09
##      [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36]
## ACF   0.00  0.02 -0.05  0.07 -0.01 -0.02 -0.06  0.02  0.06 -0.03  0.03  0.03
## PACF -0.02 -0.02 -0.11  0.02 -0.03 -0.02 -0.06 -0.04 -0.03 -0.06  0.05  0.01
##      [,37] [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF  -0.07 -0.01  0.04  0.00 -0.06  0.02  0.06     0 -0.09  0.08  0.01 -0.06
## PACF -0.02 -0.03  0.01 -0.03 -0.05 -0.04 -0.01     0 -0.08  0.03  0.00 -0.07
```

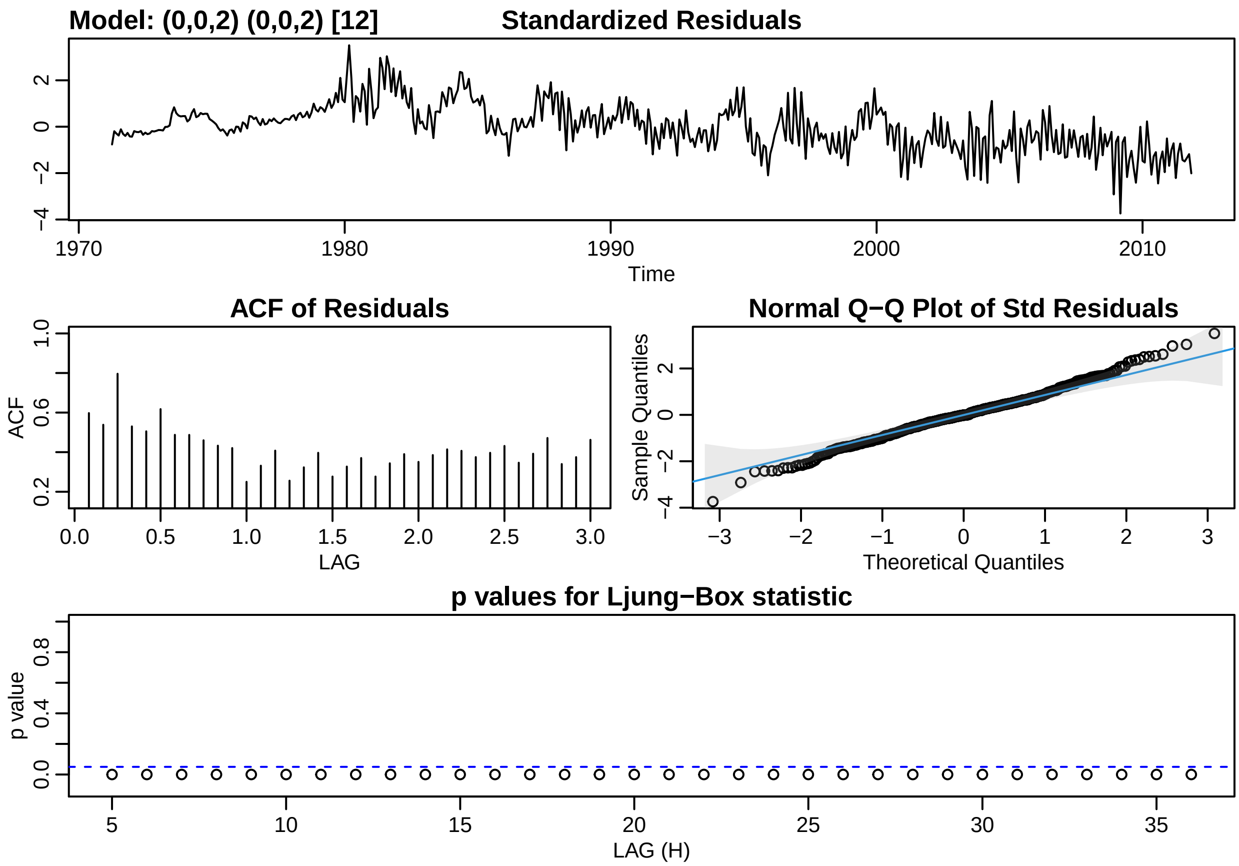
```
sarima(d[,"lx"], 0, 1, 0, 0, 1, 0, 12) # -3.41
```




```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
##
## sigma^2 estimated as 0.001769:  log likelihood = 831.17,  aic = -1660.34
##
## $degrees_of_freedom
## [1] 475
##
## $ttable
##      Estimate p.value
##
## $AIC
## [1] -3.416345
##
## $AICc
## [1] -3.416345
##
## $BIC
## [1] -3.407779
```

```r
sarima(d[,"lx"], 0, 0, 2, 0, 0, 2, 12) # -2.55
```

```
## initial  value -1.160846
## iter   2 value -2.632469
## iter   3 value -2.653743
## iter   4 value -2.696147
## iter   5 value -2.700156
## iter   6 value -2.706210
## iter   7 value -2.711295
## iter   8 value -2.712658
## iter   9 value -2.713460
## iter  10 value -2.714038
## iter  11 value -2.714263
## iter  12 value -2.714351
## iter  13 value -2.714358
## iter  14 value -2.714360
## iter  15 value -2.714361
## iter  16 value -2.714362
## iter  17 value -2.714362
## iter  18 value -2.714363
## iter  19 value -2.714363
## iter  19 value -2.714363
## iter  19 value -2.714363
## final  value -2.714363
## converged
## initial  value -2.706441
## iter   2 value -2.706625
## iter   3 value -2.706712
## iter   4 value -2.707635
## iter   5 value -2.707817
## iter   6 value -2.708432
## iter   7 value -2.708725
## iter   8 value -2.708758
## iter   9 value -2.708758
## iter   9 value -2.708758
## iter   9 value -2.708758
## final  value -2.708758
## converged
```

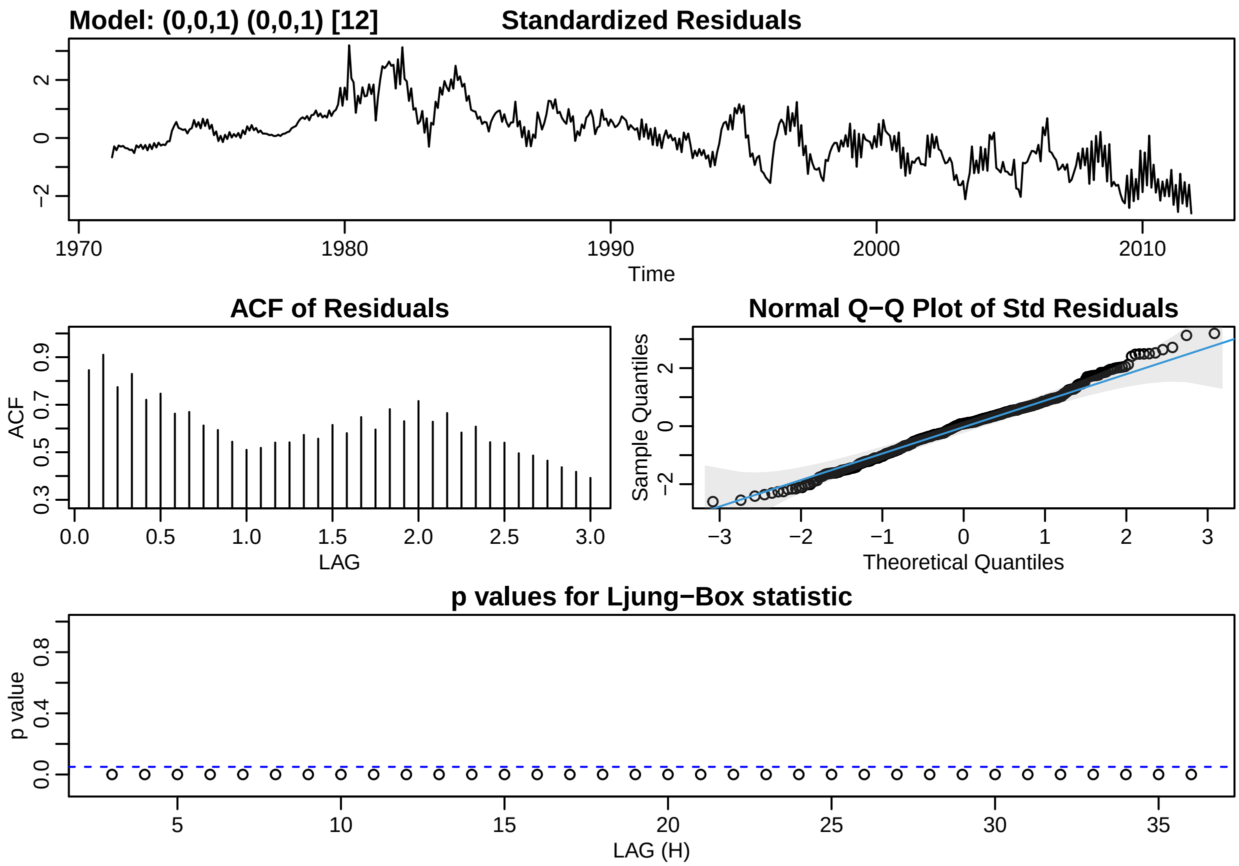



```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), xreg = xmean, include.mean = FALSE, transform.pars = trans,
##     fixed = fixed, optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##          ma1     ma2    sma1    sma2   xmean
##       1.0295  0.9444  0.8437  0.5123  2.1104
## s.e.  0.0151  0.0170  0.0412  0.0380  0.0204
##
## sigma^2 estimated as 0.00431:  log likelihood = 629.43,  aic = -1246.86
##
## $degrees_of_freedom
## [1] 483
##
## $ttable
##       Estimate     SE  t.value p.value
## ma1     1.0295 0.0151  68.2605       0
## ma2     0.9444 0.0170  55.5611       0
## sma1    0.8437 0.0412  20.4648       0
## sma2    0.5123 0.0380  13.4650       0
## xmean   2.1104 0.0204 103.5053       0
##
## $AIC
```

```
## [1] -2.555049
##
## $AICc
## [1] -2.554794
##
## $BIC
## [1] -2.503529
```

```
model1 <- sarima(d[,"lx"], 0, 0, 1, 0, 0, 1, 12) # -1.54
```

```
## initial  value -1.160846
## iter   2 value -2.134130
## iter   3 value -2.148147
## iter   4 value -2.170410
## iter   5 value -2.183279
## iter   6 value -2.205437
## iter   7 value -2.206102
## iter   8 value -2.207423
## iter   9 value -2.207445
## iter  10 value -2.207467
## iter  11 value -2.207514
## iter  12 value -2.207570
## iter  13 value -2.207578
## iter  14 value -2.207578
## iter  14 value -2.207578
## iter  14 value -2.207578
## final  value -2.207578
## converged
## initial  value -2.199863
## iter   2 value -2.199928
## iter   3 value -2.200020
## iter   4 value -2.200130
## iter   5 value -2.200280
## iter   6 value -2.200322
## iter   7 value -2.200326
## iter   8 value -2.200326
## iter   8 value -2.200326
## final  value -2.200326
## converged
```




```
model1$fit
```

```
##
## Call:
## stats::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:
##          ma1    sma1    xmean
##        0.957   0.738   2.1153
## s.e.   0.010   0.024   0.0167
##
## sigma^2 estimated as 0.01199:  log likelihood = 381.32,  aic = -754.63
```

Answer: Based on AIC selection criterion I decided that ARIMA(0,0,1) is the model.