STA137: Project 2

Selam Berekat

5/21/2021

Contents:

Section 1: Introduction:

-> Section(1) Introduction:

This is a time series data analysis project for a dataset of US monthly 30-year conventional mortgage rate for the period of April 1971 to November 2011. The data are collected from Federal Reserve Economic data. We can identify whether a dataset is a time series when the collected observation is a track of information based on yearly, monthly and daily.

Here, we will apply a time series data analysis to generate a model that represent the time series analyze a 30-year conventional mortgage. The data collects 488 observation of a monthly mortgage rate, and monthly federal rate for the time span of 1971 to 2011. The project information are organized as follows:

-> Section(2) Material and methods:

a. information about exploratory analysis technique to describe the data and
b. the appropriate method to determine the optimal model for the monthly mortgage rate series and monthly federal fund rate series. Thus, the analysis is divided into two problems:

Question of Interest_(1):

Finding the right model that predict the monthly mortgage rate time series

1. Check if the time series is stationary or not.
2. Apply transformation as needed.
3. Compute sample autocorrelation and partial autocorrelation
4. Select the optimal model based on the model selection criterion of Ljung-Box statistic with smallest AICc value.

Question of Interest_(2):

Check whether the monthly mortgage rate depends on the monthly Federal Fund rate time series

1. Check if the time series is stationary or not.
2. Apply transformation as needed.

3. Compute sample autocorrelation and partial autocorrelation
4. Select the optimal model based on the model selection criterion of Ljung-Box statistic with smallest AICc value.

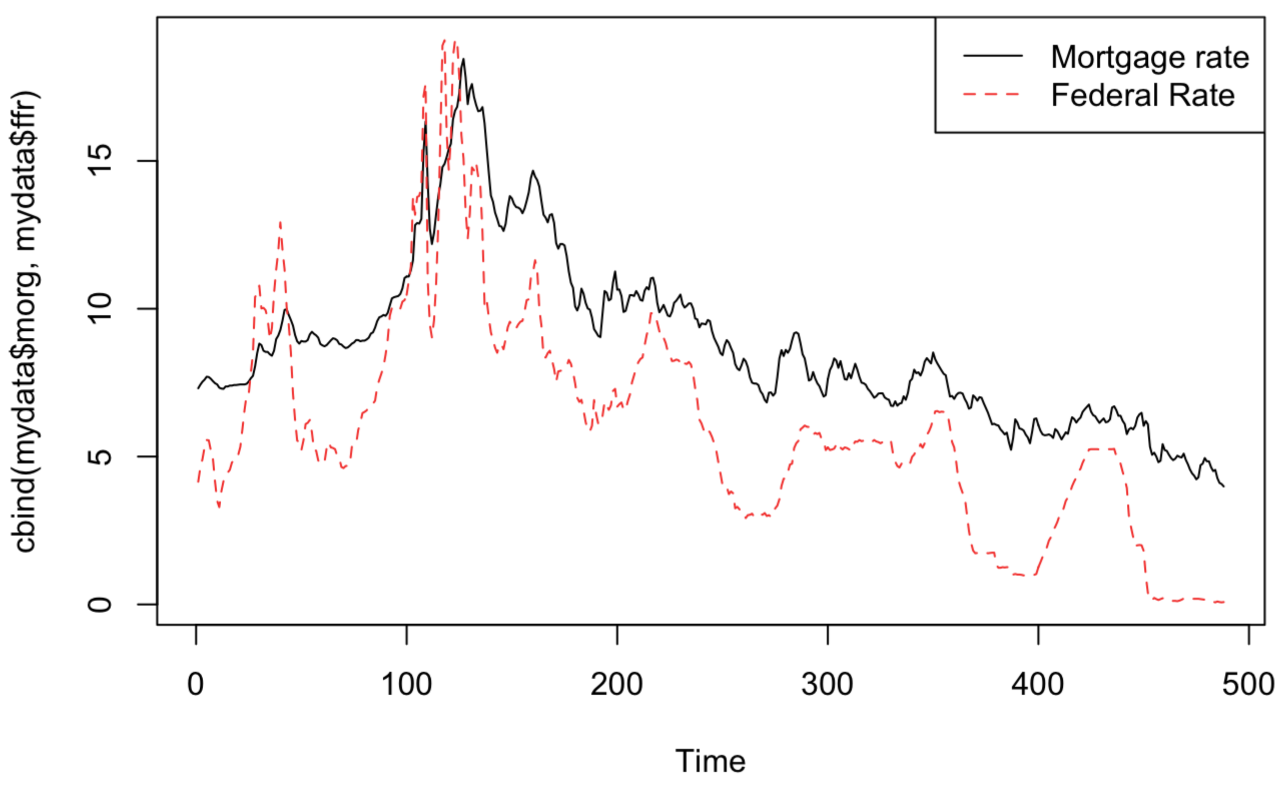
-> Section(3) Results: this section holds the final finding on the analysis.

-> Section(4) Appendix: this section holds the code used to apply and get the result of the analysis.

Section 2: Material and Methods:

(a) Description of data using exploratory analysis technique.

The initial step to build models for a data analysis is to plot the original time series data, and determine whether assumption of stationary holds. Stationary is a very important property that determines if the mean, variance, autocorrelation are constant over time. And, this can be helpful to apply right models to forecast future behaviors.

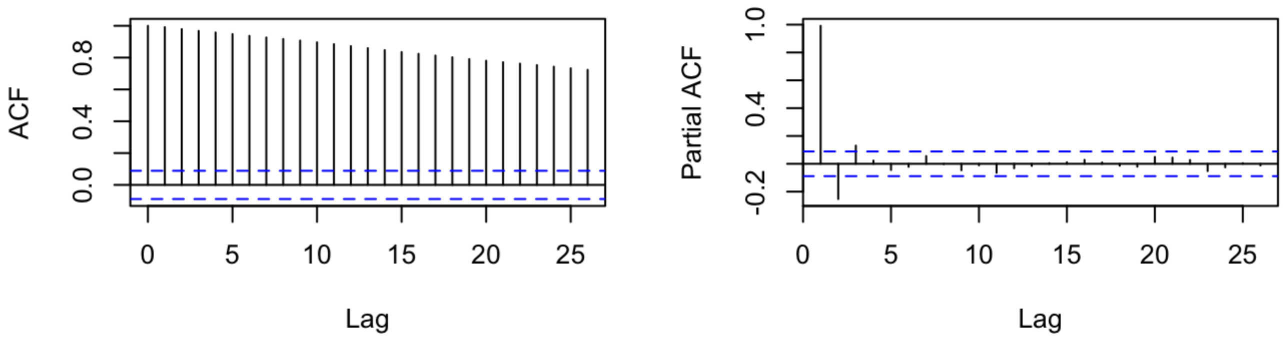


From the above plot, we observe our dataset exhibit a trend that decreases gradually and that do not repeat itself at any regular intervals and also that it is not a stationary. However, we do not observe an outlier or unusual observaton. Therefore, to proceed we will have to transform the dataset into an appropriate transformation method that is using differencing. We will apply the following analysis to answer our questions of interest.

b) Question of Interest_(1):

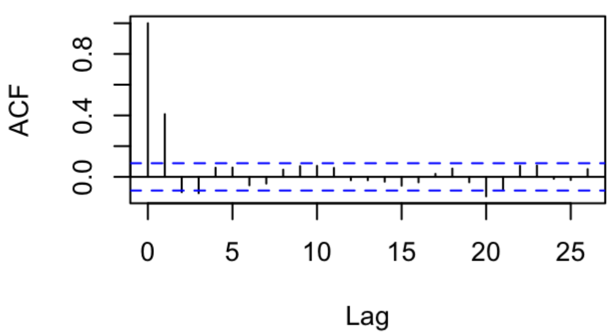
Finding the right model that predict the monthly mortgage rate time series

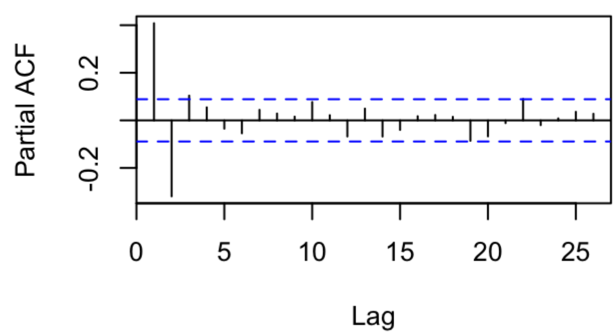
First, we know from the original plot stationary is violated and now we inspect if the property of sample autocorrelation and partial autocorrelation functions also holds by plotting it.



This plot suggests that the ACF is tailing off and the PACF is cutting off at lag 2. This might suggest an AR(2). Below we will apply differencing.

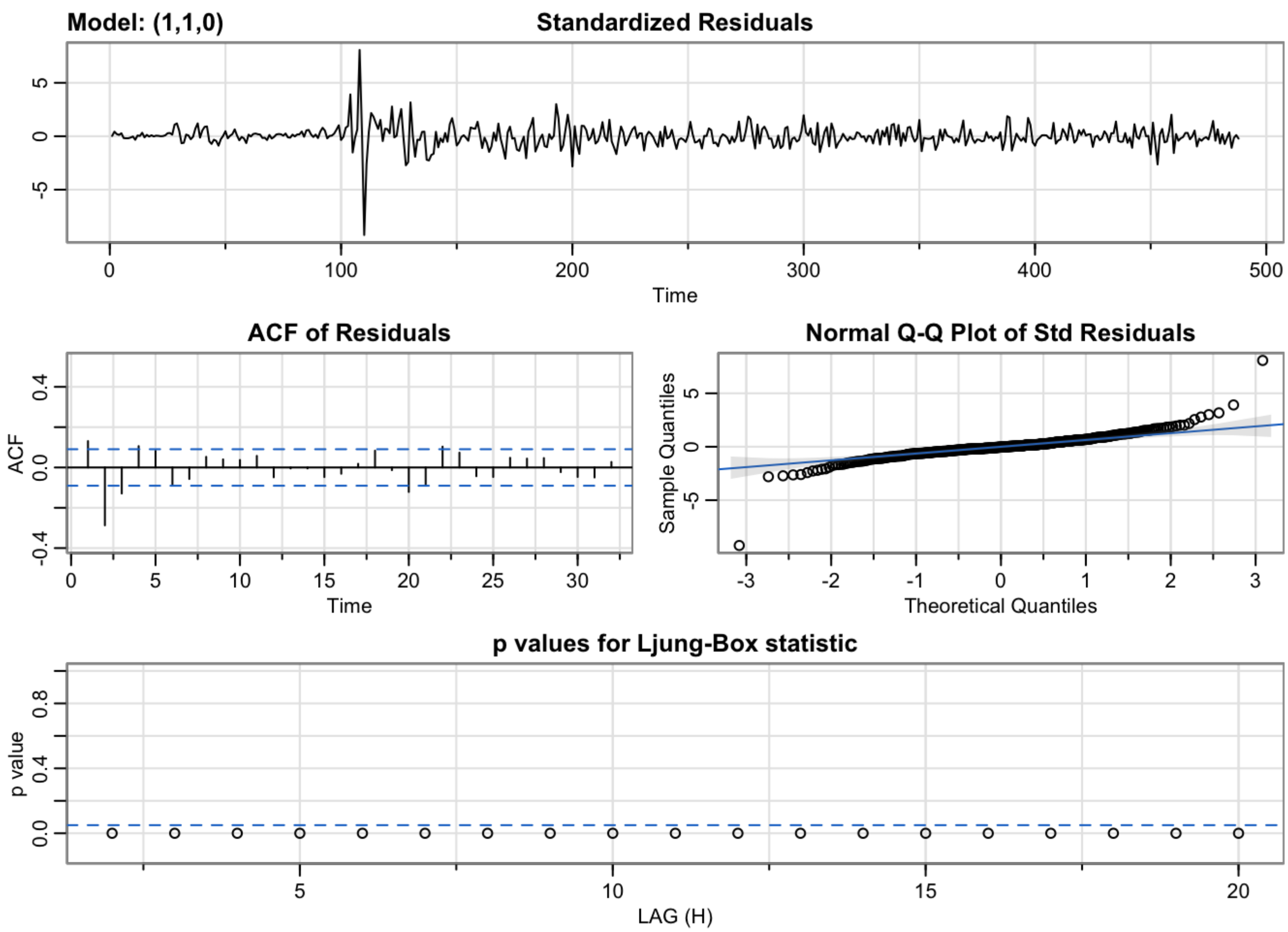

Transformed Mortgage Rate Time Series Plot
diff(mydata$morg)
2
1
0
-1
-2
0
100
200
300
400
500
Time


ACF
0.0
0.4
0.8
0
5
10
15
20
25
Lag

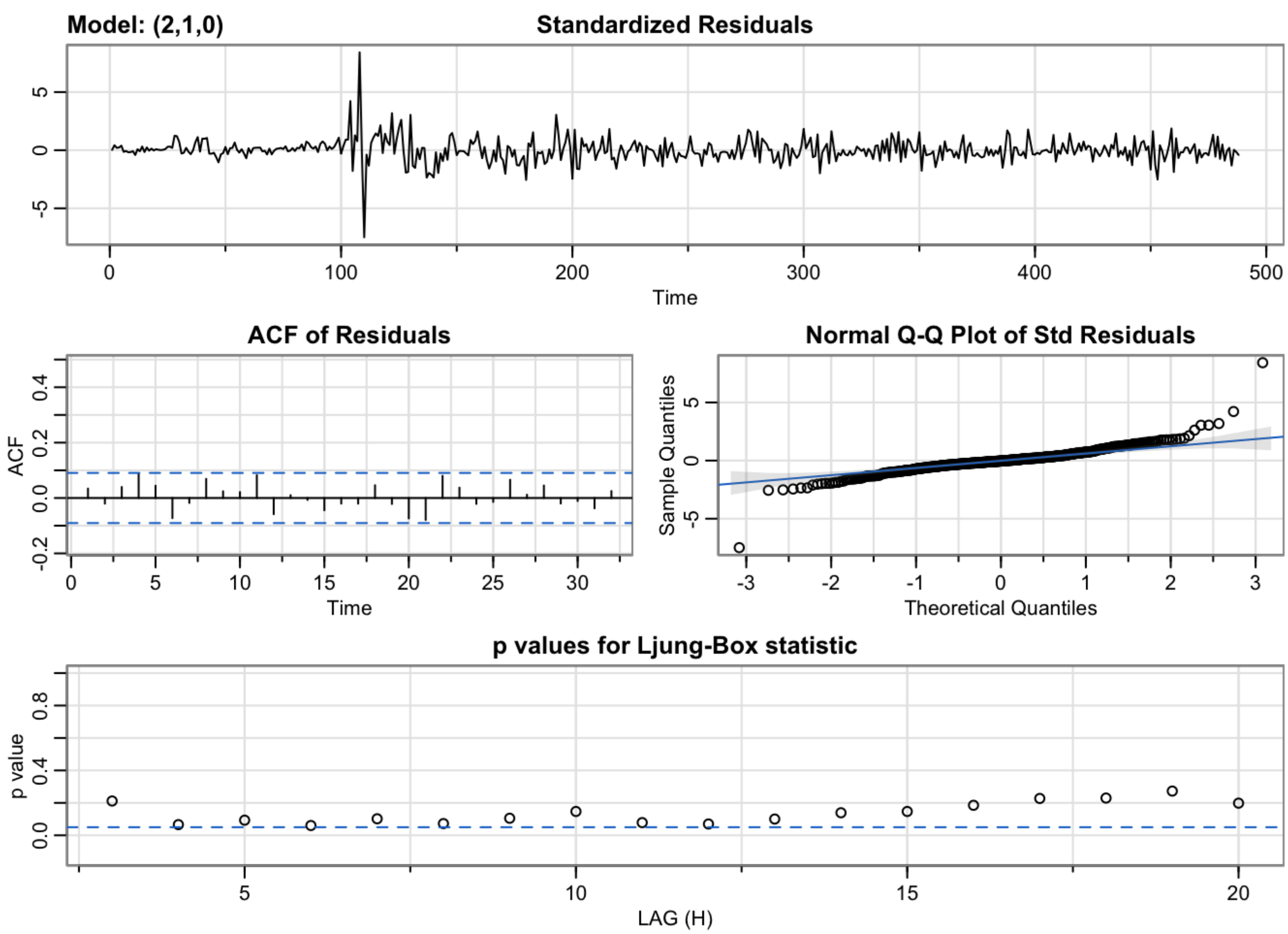

Partial ACF
-0.2
0.2
0
5
10
15
20
25
Lag

This result suggests transforming the dataset helped to correct the assumptions as the time values converge to mean. We follow it with model selecting.

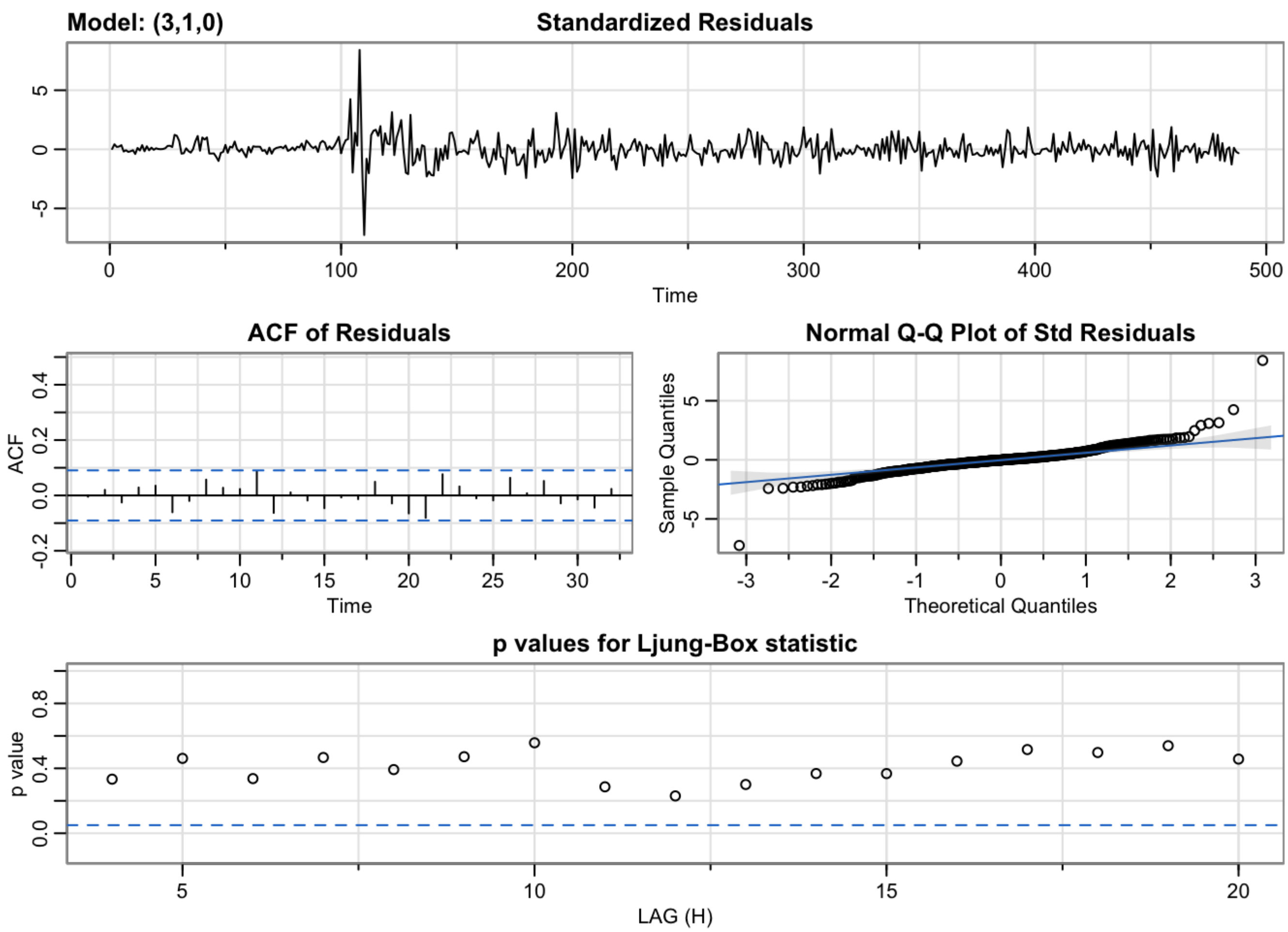
```
## initial  value -1.217709
## iter   2 value -1.308929
## iter   3 value -1.308929
## iter   4 value -1.308929
## iter   5 value -1.308929
## iter   5 value -1.308929
## iter   5 value -1.308929
## final  value -1.308929
## converged
## initial  value -1.309579
## iter   2 value -1.309580
## iter   3 value -1.309580
## iter   4 value -1.309580
## iter   5 value -1.309580
## iter   6 value -1.309580
## iter   6 value -1.309580
## final  value -1.309580
## converged
```



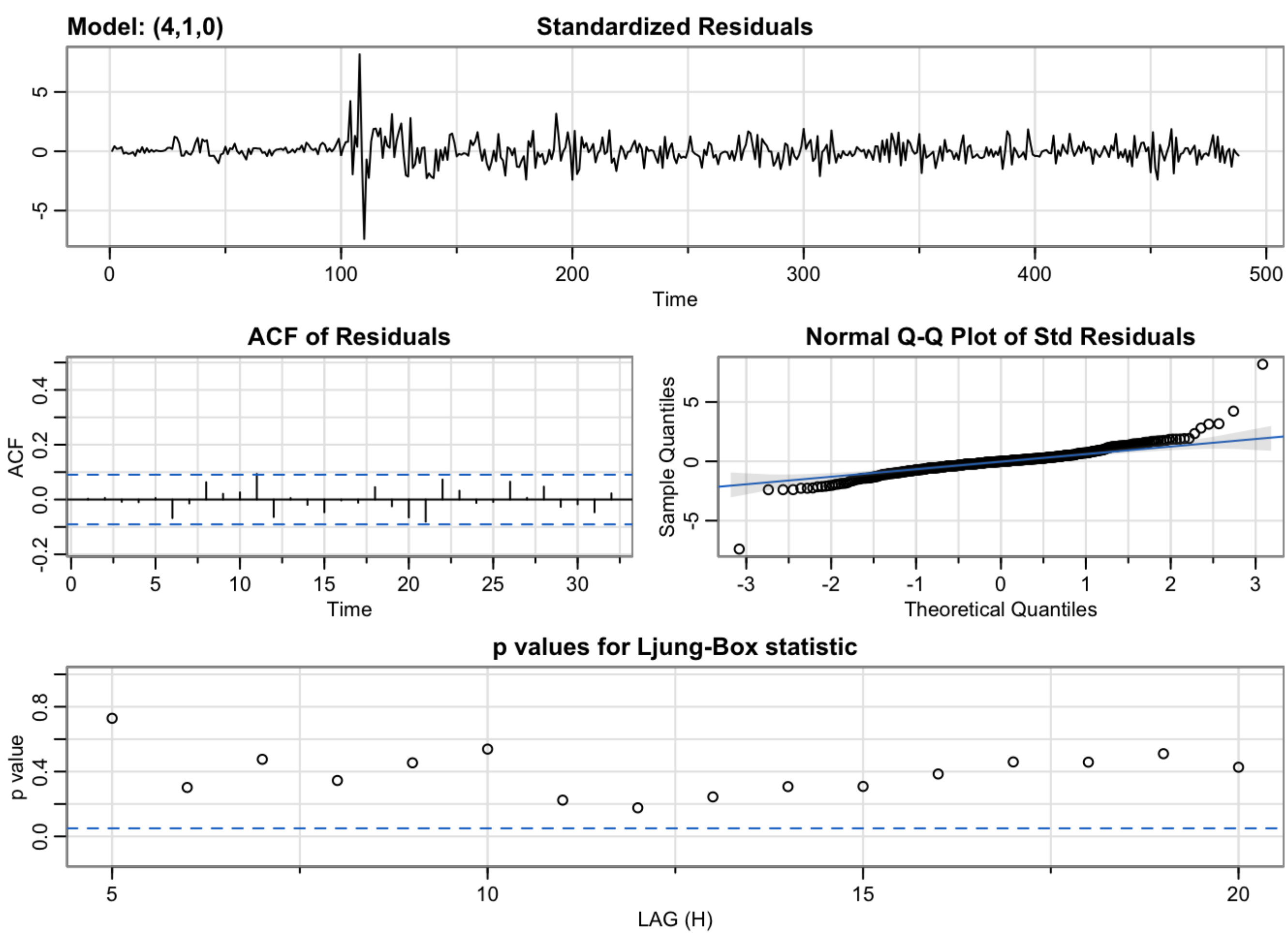
```
## initial  value -1.216814
## iter   2 value -1.334494
## iter   3 value -1.355518
## iter   4 value -1.361740
## iter   5 value -1.361758
## iter   6 value -1.361759
## iter   7 value -1.361759
## iter   8 value -1.361759
## iter   8 value -1.361759
## final  value -1.361759
## converged
## initial  value -1.363175
## iter   2 value -1.363176
## iter   3 value -1.363177
## iter   4 value -1.363177
## iter   5 value -1.363177
## iter   5 value -1.363177
## iter   5 value -1.363177
## final  value -1.363177
## converged
```



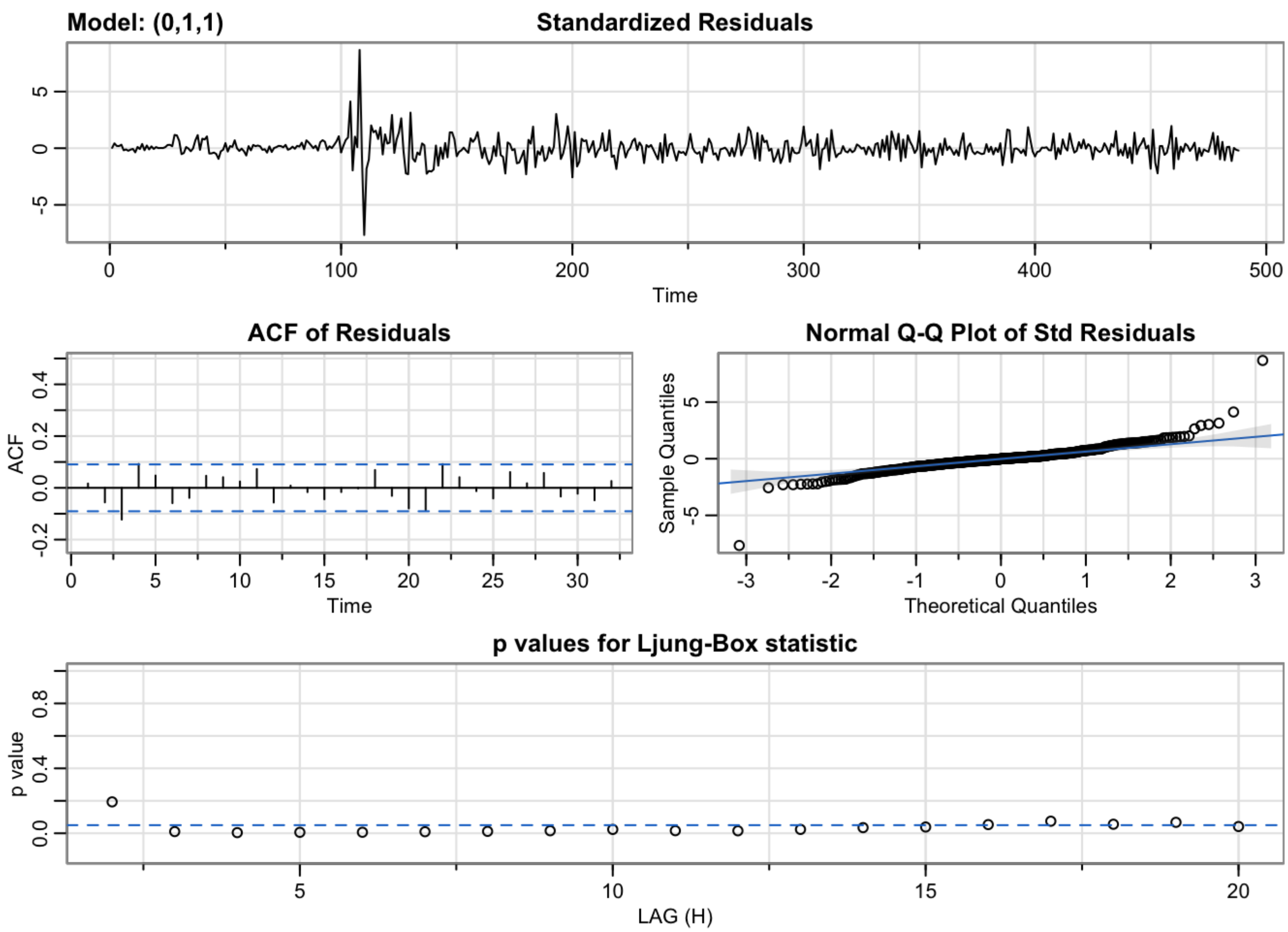
```
## initial  value -1.215851 
## iter   2 value -1.329738
## iter   3 value -1.358749
## iter   4 value -1.365203
## iter   5 value -1.366111
## iter   6 value -1.366241
## iter   7 value -1.366246
## iter   8 value -1.366246
## iter   9 value -1.366246
## iter  10 value -1.366246
## iter  10 value -1.366247
## final  value -1.366247 
## converged
## initial  value -1.368603 
## iter   2 value -1.368604
## iter   3 value -1.368604
## iter   4 value -1.368605
## iter   5 value -1.368605
## iter   5 value -1.368605
## iter   5 value -1.368605
## final  value -1.368605 
## converged
```



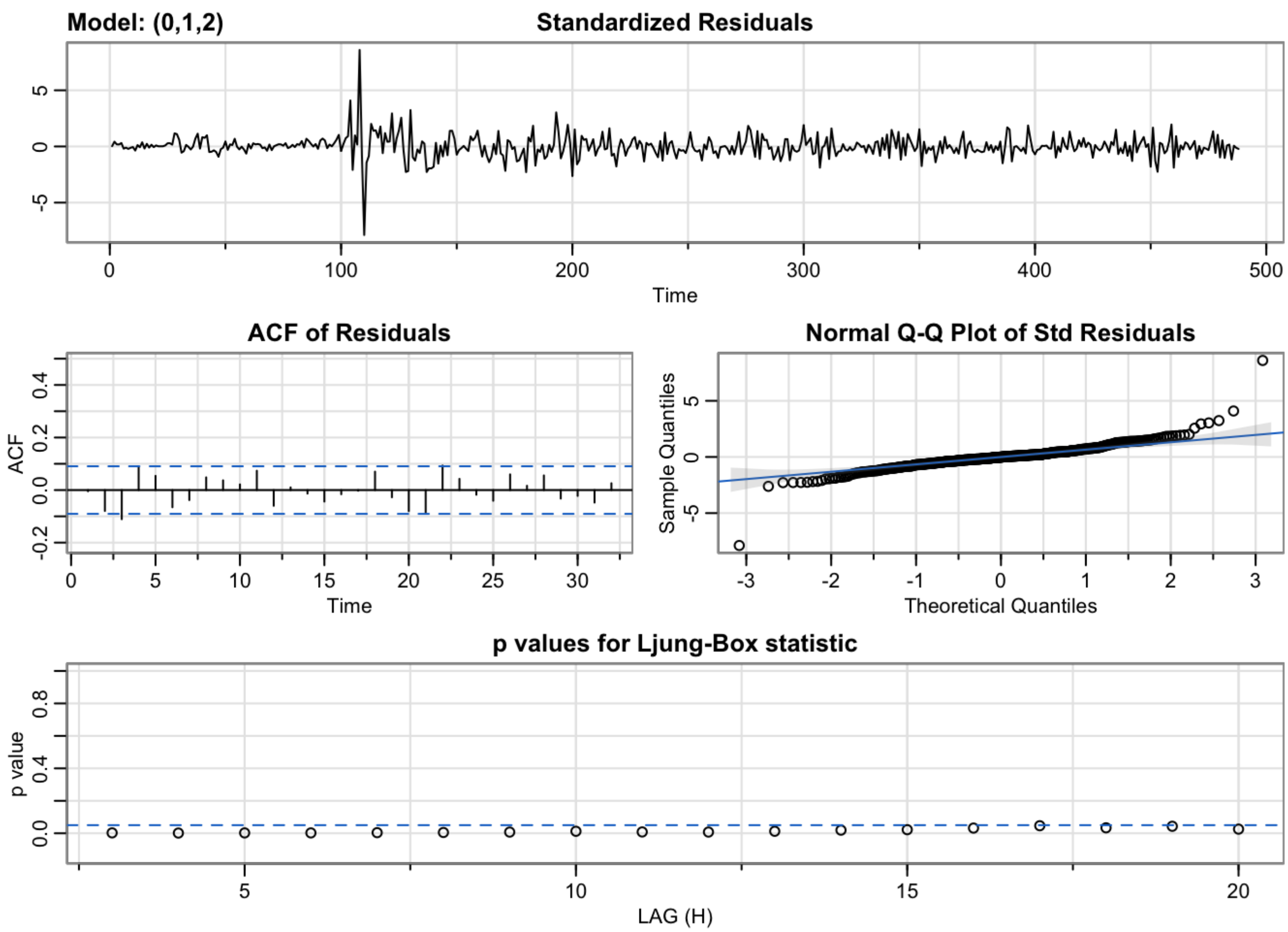
```
## initial  value -1.214951
## iter   2 value -1.336935
## iter   3 value -1.356037
## iter   4 value -1.364332
## iter   5 value -1.366744
## iter   6 value -1.366856
## iter   7 value -1.366858
## iter   8 value -1.366862
## iter   9 value -1.366862
## iter  10 value -1.366862
## iter  10 value -1.366862
## iter  10 value -1.366862
## final  value -1.366862
## converged
## initial  value -1.370107
## iter   2 value -1.370109
## iter   3 value -1.370110
## iter   4 value -1.370111
## iter   5 value -1.370112
## iter   6 value -1.370112
## iter   6 value -1.370112
## iter   6 value -1.370112
## final  value -1.370112
## converged
```



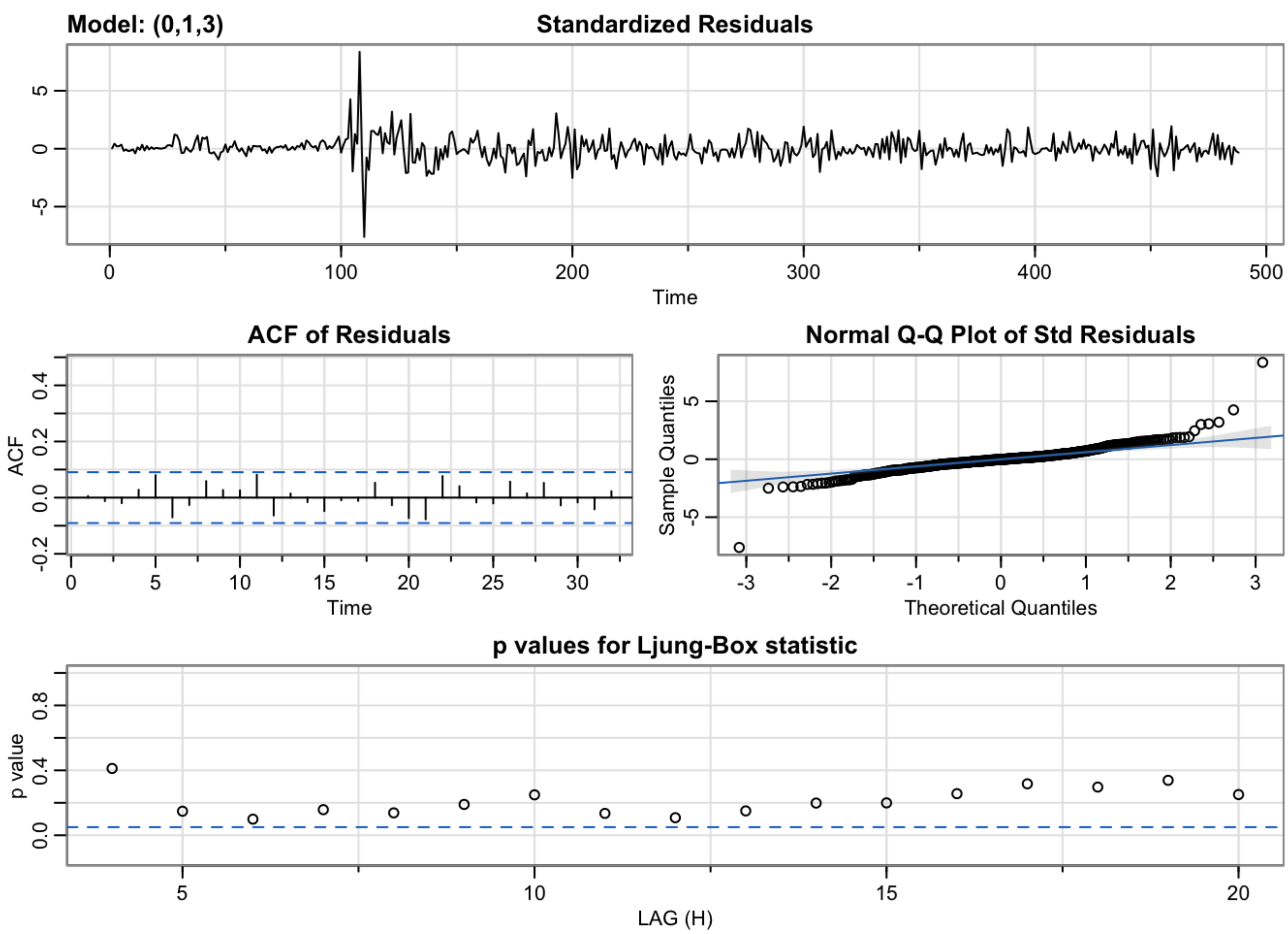
```
## initial  value -1.218548
## iter   2 value -1.343122
## iter   3 value -1.351713
## iter   4 value -1.355731
## iter   5 value -1.356360
## iter   6 value -1.356384
## iter   7 value -1.356384
## iter   8 value -1.356384
## iter   8 value -1.356384
## iter   8 value -1.356384
## final  value -1.356384
## converged
## initial  value -1.356055
## iter   2 value -1.356055
## iter   3 value -1.356056
## iter   3 value -1.356056
## iter   3 value -1.356056
## final  value -1.356056
## converged
```



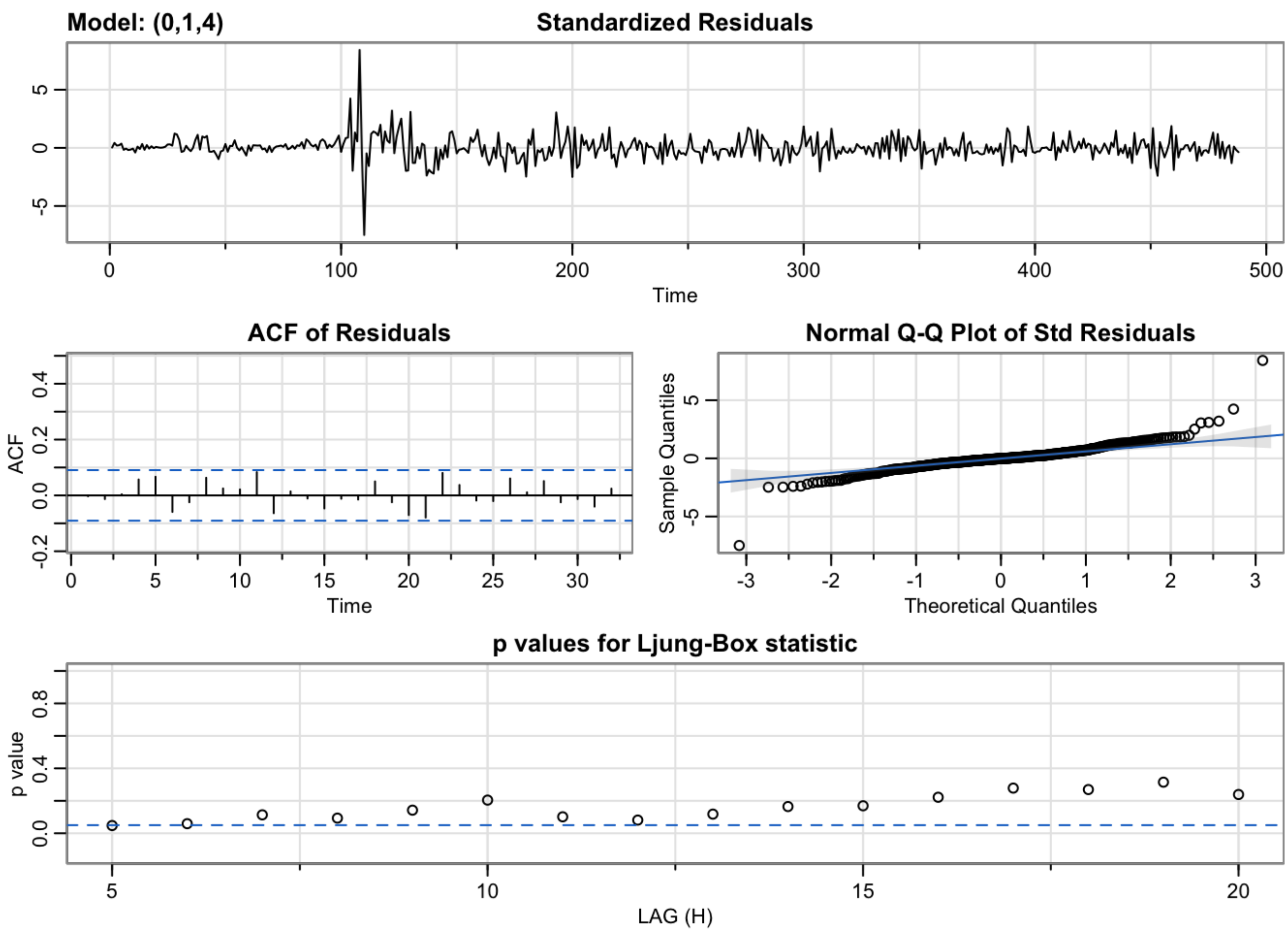
```
## initial  value -1.218548
## iter   2 value -1.345223
## iter   3 value -1.348834
## iter   4 value -1.356793
## iter   5 value -1.356955
## iter   6 value -1.356963
## iter   7 value -1.356963
## iter   7 value -1.356963
## iter   7 value -1.356963
## final  value -1.356963
## converged
## initial  value -1.356630
## iter   2 value -1.356630
## iter   3 value -1.356630
## iter   4 value -1.356630
## iter   4 value -1.356630
## iter   4 value -1.356630
## final  value -1.356630
## converged
```



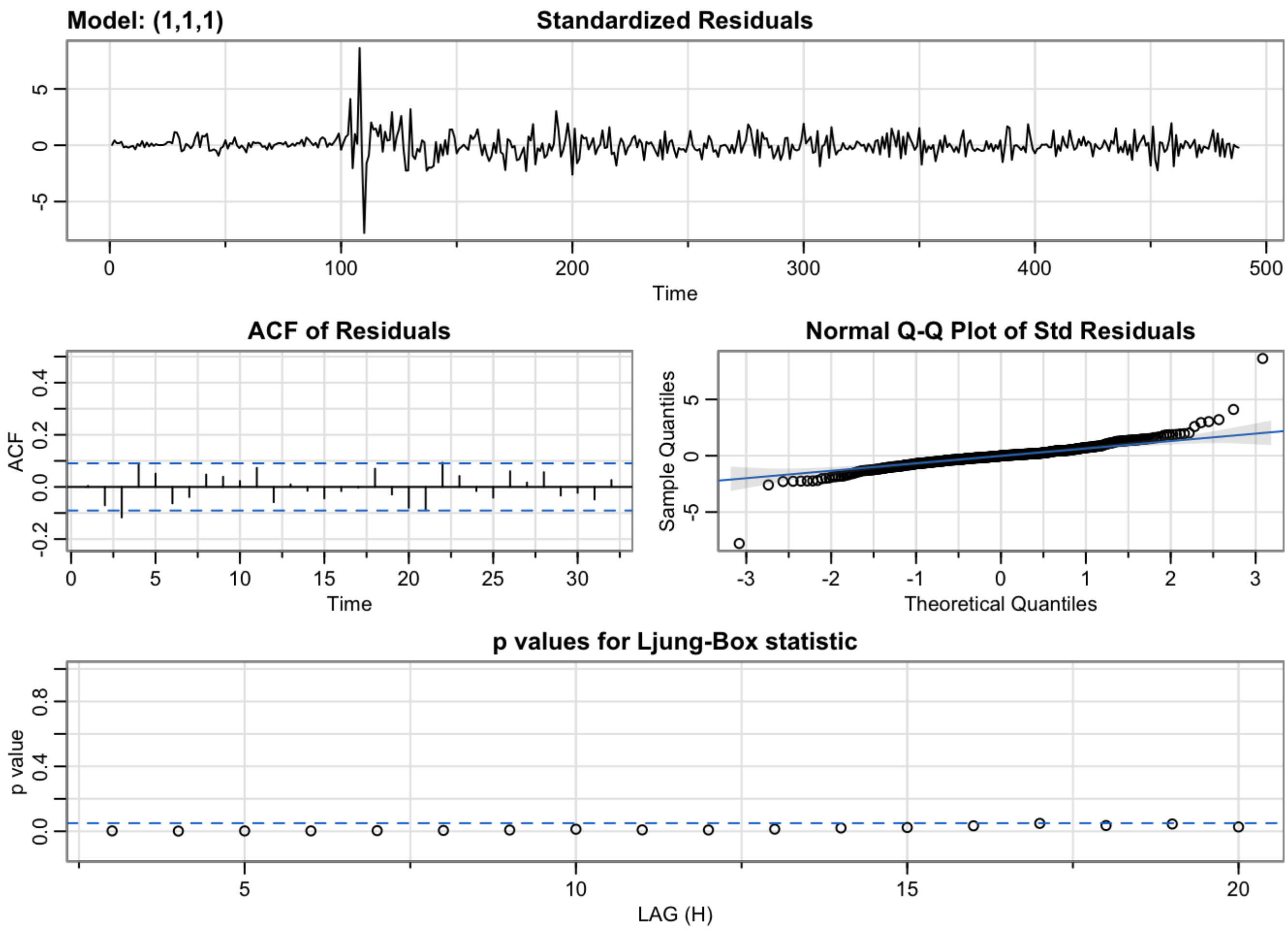
```
## initial  value -1.218548
## iter   2 value -1.353762
## iter   3 value -1.356280
## iter   4 value -1.363789
## iter   5 value -1.364861
## iter   6 value -1.365883
## iter   7 value -1.365909
## iter   8 value -1.365910
## iter   9 value -1.365910
## iter   9 value -1.365910
## iter   9 value -1.365910
## final  value -1.365910
## converged
## initial  value -1.365555
## iter   2 value -1.365555
## iter   3 value -1.365555
## iter   4 value -1.365555
## iter   4 value -1.365555
## iter   4 value -1.365556
## final  value -1.365556
## converged
```



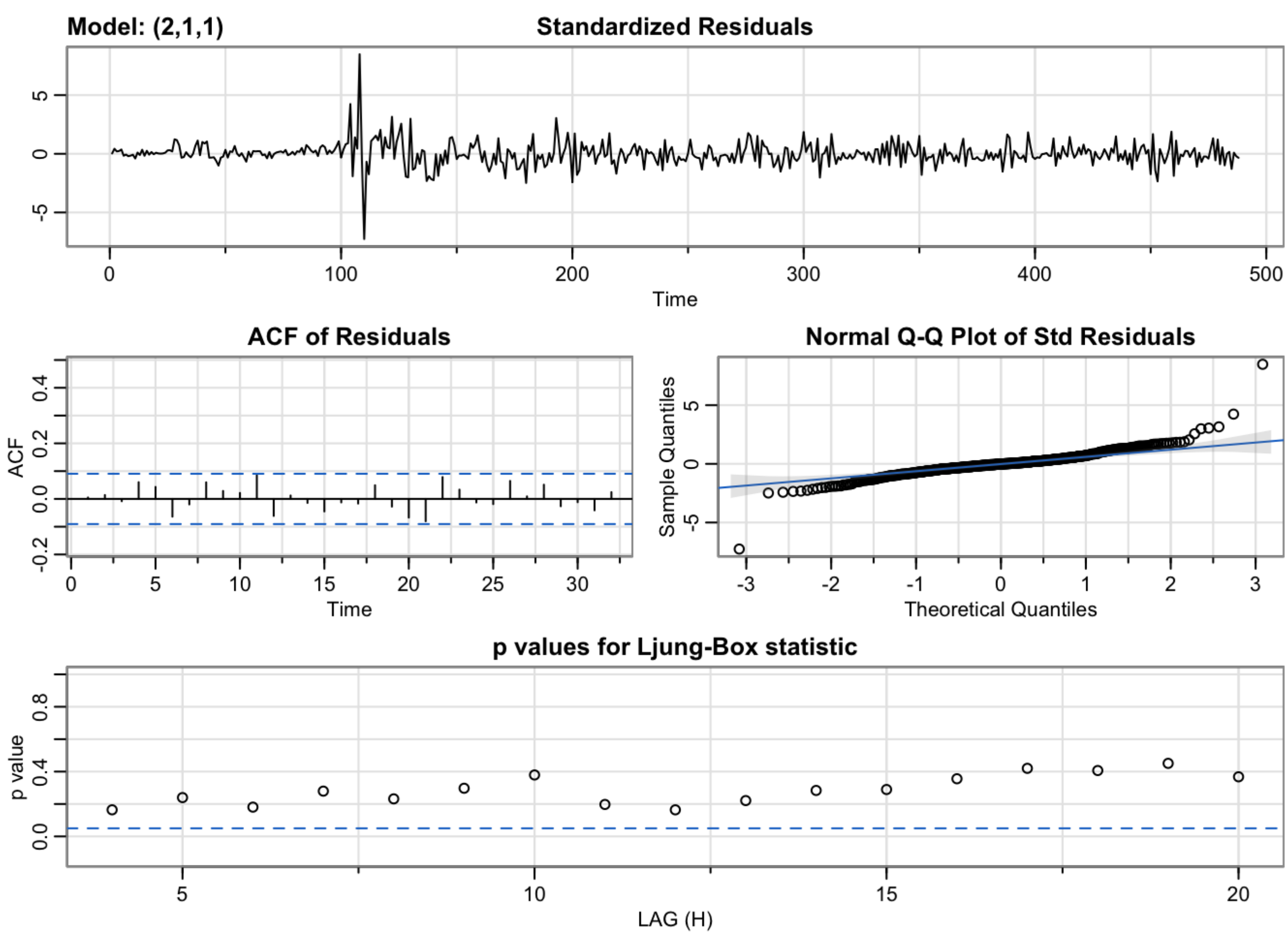
```
## initial  value -1.218548
## iter   2 value -1.352691
## iter   3 value -1.358394
## iter   4 value -1.365987
## iter   5 value -1.366782
## iter   6 value -1.366788
## iter   7 value -1.366798
## iter   8 value -1.366798
## iter   9 value -1.366799
## iter   9 value -1.366799
## iter   9 value -1.366799
## final  value -1.366799
## converged
## initial  value -1.366425
## iter   2 value -1.366426
## iter   3 value -1.366426
## iter   4 value -1.366426
## iter   5 value -1.366426
## iter   5 value -1.366426
## iter   5 value -1.366426
## final  value -1.366426
## converged
```



```
## initial  value -1.217709
## iter   2 value -1.317676
## iter   3 value -1.345417
## iter   4 value -1.350072
## iter   5 value -1.354591
## iter   6 value -1.355664
## iter   7 value -1.355779
## iter   8 value -1.355781
## iter   8 value -1.355781
## final  value -1.355781
## converged
## initial  value -1.356389
## iter   2 value -1.356389
## iter   3 value -1.356389
## iter   4 value -1.356389
## iter   5 value -1.356389
## iter   5 value -1.356389
## iter   5 value -1.356389
## final  value -1.356389
## converged
```

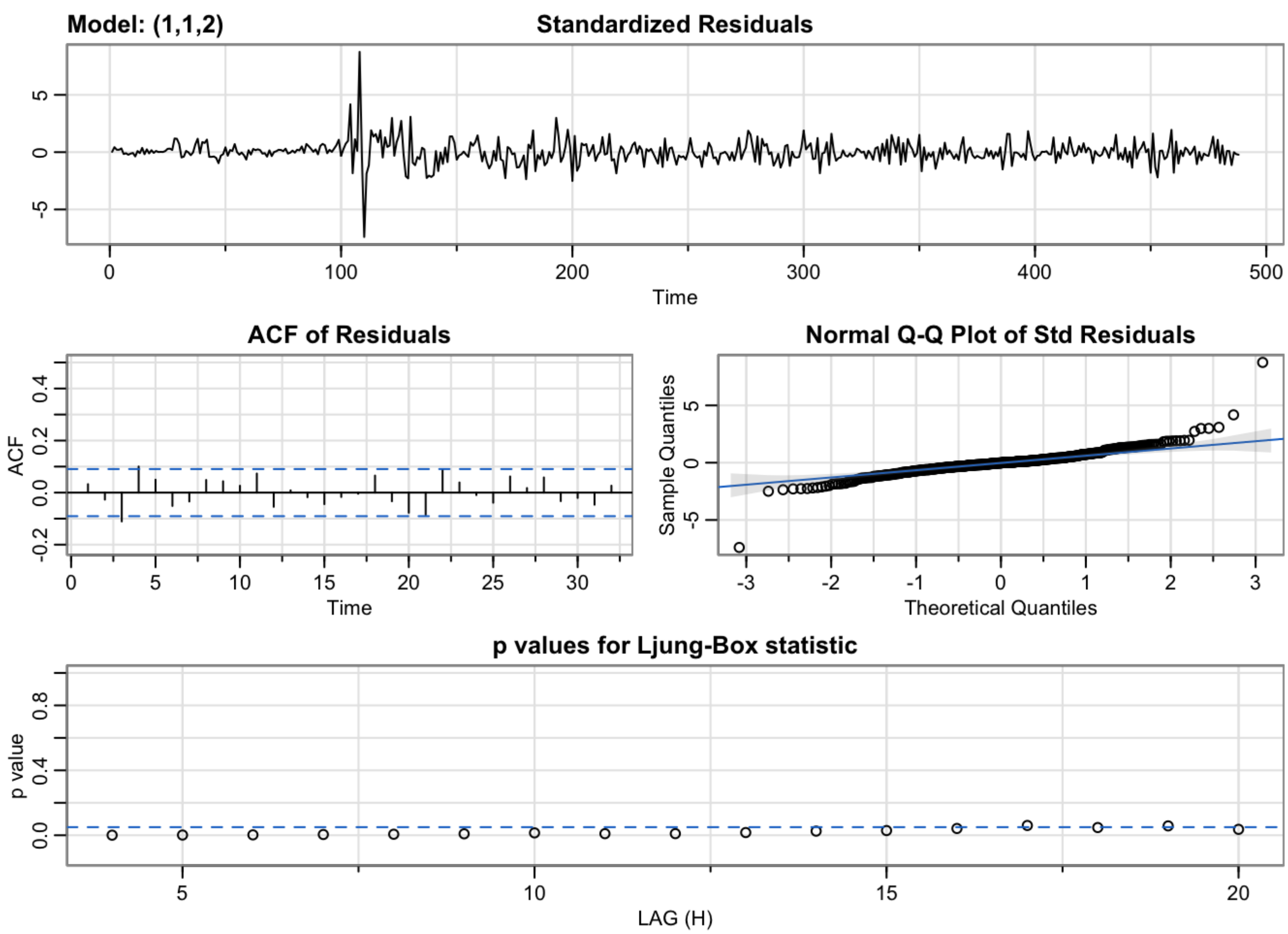


```
## initial  value -1.216814
## iter   2 value -1.329476
## iter   3 value -1.363149
## iter   4 value -1.364820
## iter   5 value -1.365086
## iter   6 value -1.365180
## iter   7 value -1.365507
## iter   8 value -1.365567
## iter   9 value -1.365572
## iter  10 value -1.365573
## iter  10 value -1.365573
## final  value -1.365573
## converged
## initial  value -1.366982
## iter   2 value -1.366982
## iter   3 value -1.366984
## iter   4 value -1.366984
## iter   5 value -1.366984
## iter   6 value -1.366984
## iter   6 value -1.366984
## iter   6 value -1.366984
## final  value -1.366984
## converged
```



```
## initial  value -1.217709
## iter   2 value -1.320773
## iter   3 value -1.349610
## iter   4 value -1.355276
## iter   5 value -1.355531
## iter   6 value -1.355558
## iter   7 value -1.355559
## iter   8 value -1.355566
## iter   9 value -1.355584
## iter  10 value -1.355625
## iter  11 value -1.355699
## iter  12 value -1.355718
## iter  13 value -1.355728
## iter  14 value -1.355751
## iter  15 value -1.355765
## iter  16 value -1.355776
## iter  17 value -1.355788
## iter  18 value -1.355820
## iter  19 value -1.355918
## iter  20 value -1.355974
## iter  21 value -1.356013
## iter  22 value -1.356029
## iter  23 value -1.356056
## iter  24 value -1.356084
## iter  25 value -1.356090
## iter  26 value -1.356116
## iter  27 value -1.356157
## iter  28 value -1.356229
## iter  29 value -1.356275
## iter  30 value -1.356292
## iter  31 value -1.356303
## iter  32 value -1.356312
## iter  33 value -1.356326
## iter  34 value -1.356327
## iter  35 value -1.356332
## iter  36 value -1.356339
## iter  37 value -1.356352
## iter  38 value -1.356356
## iter  39 value -1.356359
## iter  40 value -1.356360
## iter  41 value -1.356363
## iter  42 value -1.356363
## iter  43 value -1.356363
## iter  44 value -1.356364
## iter  45 value -1.356364
## iter  46 value -1.356365
## iter  47 value -1.356366
## iter  48 value -1.356366
## iter  49 value -1.356366
## iter  50 value -1.356367
## iter  51 value -1.356367
## iter  52 value -1.356367
```

```
## iter  53 value -1.356367
## iter  54 value -1.356367
## iter  55 value -1.356368
## iter  56 value -1.356368
## iter  57 value -1.356368
## iter  58 value -1.356368
## iter  59 value -1.356368
## iter  59 value -1.356368
## iter  59 value -1.356368
## final  value -1.356368
## converged
## initial  value -1.356776
## iter   2 value -1.356777
## iter   3 value -1.356778
## iter   4 value -1.356779
## iter   4 value -1.356779
## iter   4 value -1.356779
## final  value -1.356779
## converged
```



```
##                   AR(1)     AR(2)      AR(3)      AR(4)
## fit               List,14   List,14    List,14    List,14
## degrees_of_freedom 485      484        483        482
## ttable            Numeric,8 Numeric,12 Numeric,16 Numeric,20
## AIC               0.2310364 0.1279502  0.1212004  0.1222933
## AICc              0.2310873 0.1280522  0.1213708  0.1225495
## BIC               0.2568368 0.1623507  0.1642011  0.1738941
```

```
##                   MA(1)     MA(2)      MA(3)      MA(4)
## fit               List,14   List,14    List,14    List,14
## degrees_of_freedom 485      484        483        482
## ttable            Numeric,8 Numeric,12 Numeric,16 Numeric,20
## AIC               0.1380862 0.1410435  0.1272999  0.1296651
## AICc              0.1381371 0.1411455  0.1274704  0.1299212
## BIC               0.1638866 0.175444   0.1703006  0.1812659
```

```
##                   ARMA(1,1)  ARMA(2,1)  ARMA(1,2)
## fit               List,14    List,14    List,14
## degrees_of_freedom 484       483        483
## ttable            Numeric,12 Numeric,16 Numeric,16
## AIC               0.1415253  0.1244426  0.1448532
## AICc              0.1416273  0.124613   0.1450236
## BIC               0.1759258  0.1674433  0.1878538
```

From the above computation and residual plots of different model, we would select the optimal model as the model with the lowest AICc. This AICc is a model selection criteria that helps us chooses the optimal model to predict future behavior.

The selected optimal model is AR(3) with the following important statistics.

Optimal Fitted Model: X_t + mu = phi1(X_t-1 - mu) - phi2(X_t-2 - mu) + phi3(X_t-3 - mu)+w_t, where mu = -0.0068 and w_t ~ iid(0, sigma^2 = 0.06469) on 483 degrees of freedom

And below is the AIC value, coefficients of phi and summary statistic.

```
## [1] 0.1213708
```

```
##          ar1          ar2          ar3     constant
##  0.571045866 -0.374164628  0.103672726 -0.006847106
```

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control = list(trac
e = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##         ar1      ar2     ar3  constant
##       0.571  -0.3742  0.1037   -0.0068
## s.e.  0.045   0.0490  0.0450    0.0165
##
## sigma^2 estimated as 0.06469:  log likelihood = -24.51,  aic = 59.02
##
## $degrees_of_freedom
## [1] 483
##
## $ttable
##          Estimate     SE t.value p.value
## ar1        0.5710 0.0450 12.6783  0.0000
## ar2       -0.3742 0.0490 -7.6293  0.0000
## ar3        0.1037 0.0450  2.3061  0.0215
## constant  -0.0068 0.0165 -0.4157  0.6778
##
## $AIC
## [1] 0.1212004
##
## $AICc
## [1] 0.1213708
##
## $BIC
## [1] 0.1642011
```

The plot suggest ACF is tailing off and according to the normal Q-Q plot, the fitted values suggest linearity, normality and equal variance among the data. this means the data is normally distributed and also no obvious departure of residuals from whiteness. And according to the Ljung-Box plot where the pvalue is greater than 0.01, we fail to reject the null hypothesis and conclude that the model does not show lack of fit and there is autocorrelation as the p-value indicate significance. The statistic coefficients suggests all regression coefficients have a significance, including the average monthly mortgage rate which is -0.0068.

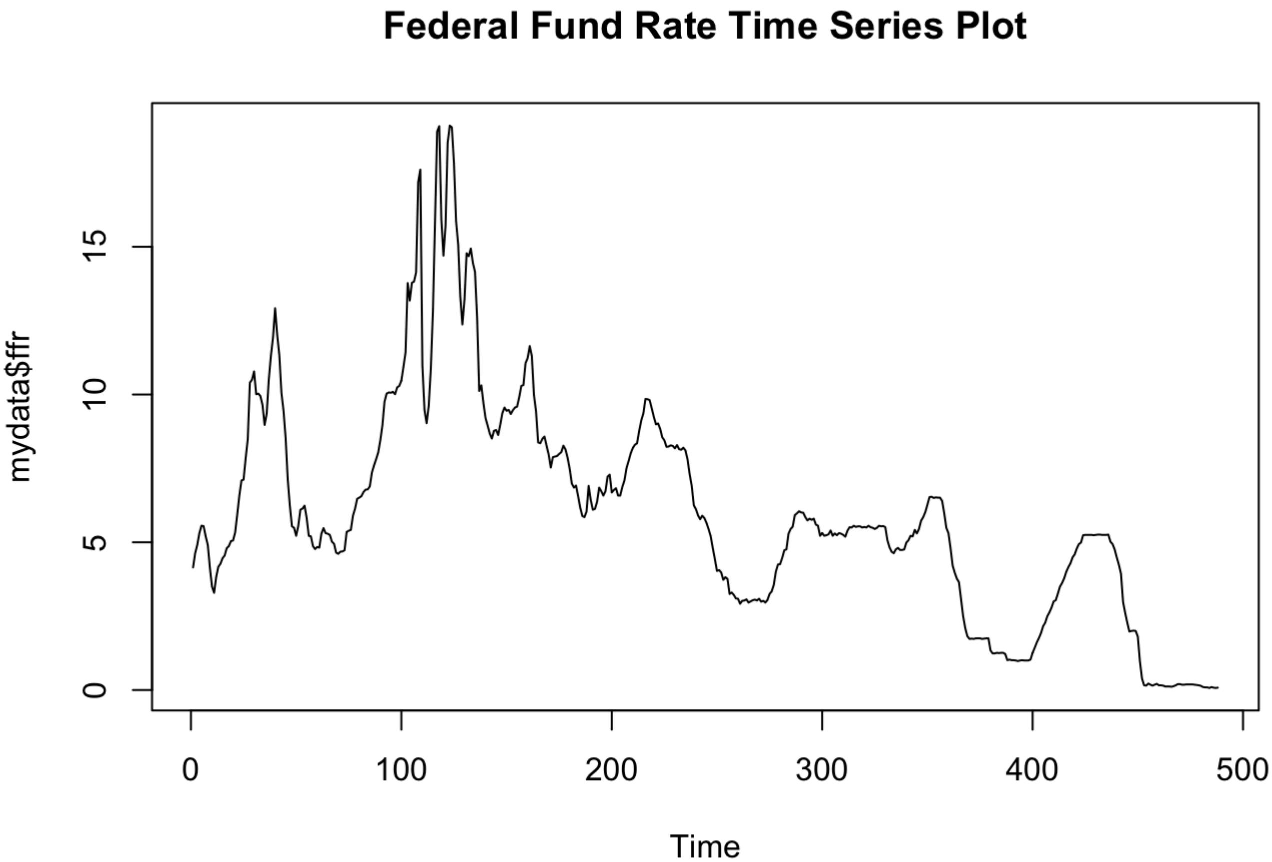
Question of Interest_(2):

Check whether the monthly mortgage rate depends on the monthly Federal Fund rate time series

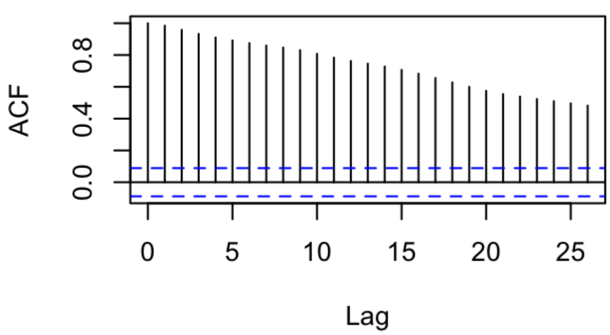
We would like to determine whether mortgage rate depends on the Federal Fund rate. Therefore, we will take the following steps.

1. Check if the time series is stationary or not.
2. Apply regression method and apply transformation as needed.

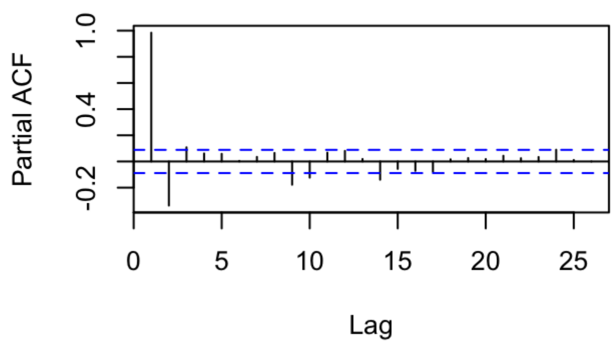
3. Compute sample autocorrelation and partial autocorrelation
4. Build a time series model for the mortgage rate using the lag-1 federal funds rate as an explanatory variable.
5. Perform model checking and select the optimal model based on the model selection criterion of Ljung-Box statistic with smallest AICc value.



Federal Fund Rate Time Series Plot
mydata$ffr
0
5
10
15
0
100
200
300
400
500
Time



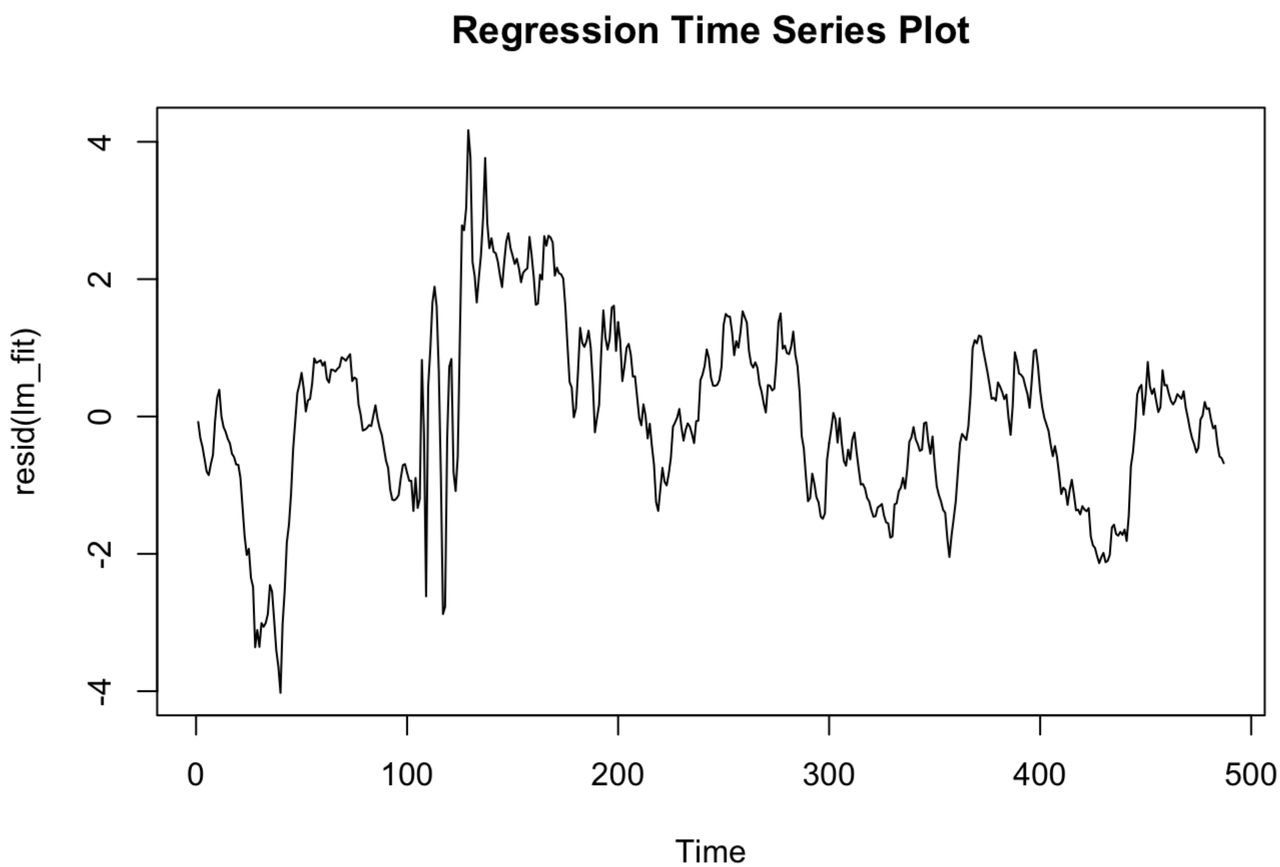
ACF
0.0
0.4
0.8
0
5
10
15
20
25
Lag

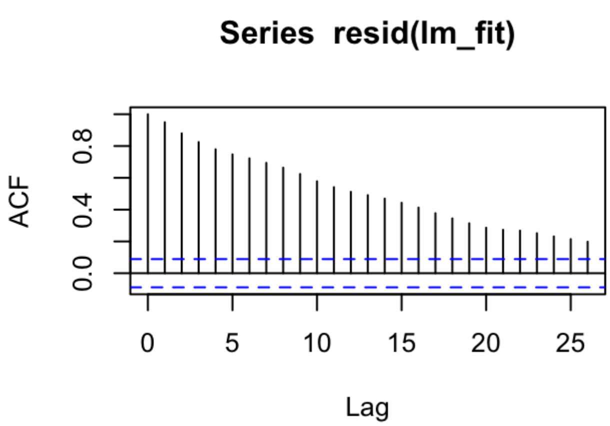


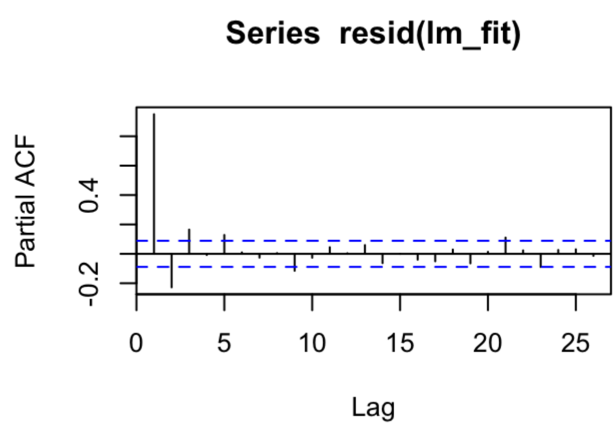
Partial ACF
-0.2
0.4
1.0
0
5
10
15
20
25
Lag

From the above plot, we observe again the dataset for Federal Fund Rate exhibits a trend that decreases gradually and that do not repeat itself at any regular intervals and also that it is not a stationary. However, we do not observe an outliers or unusual observation.

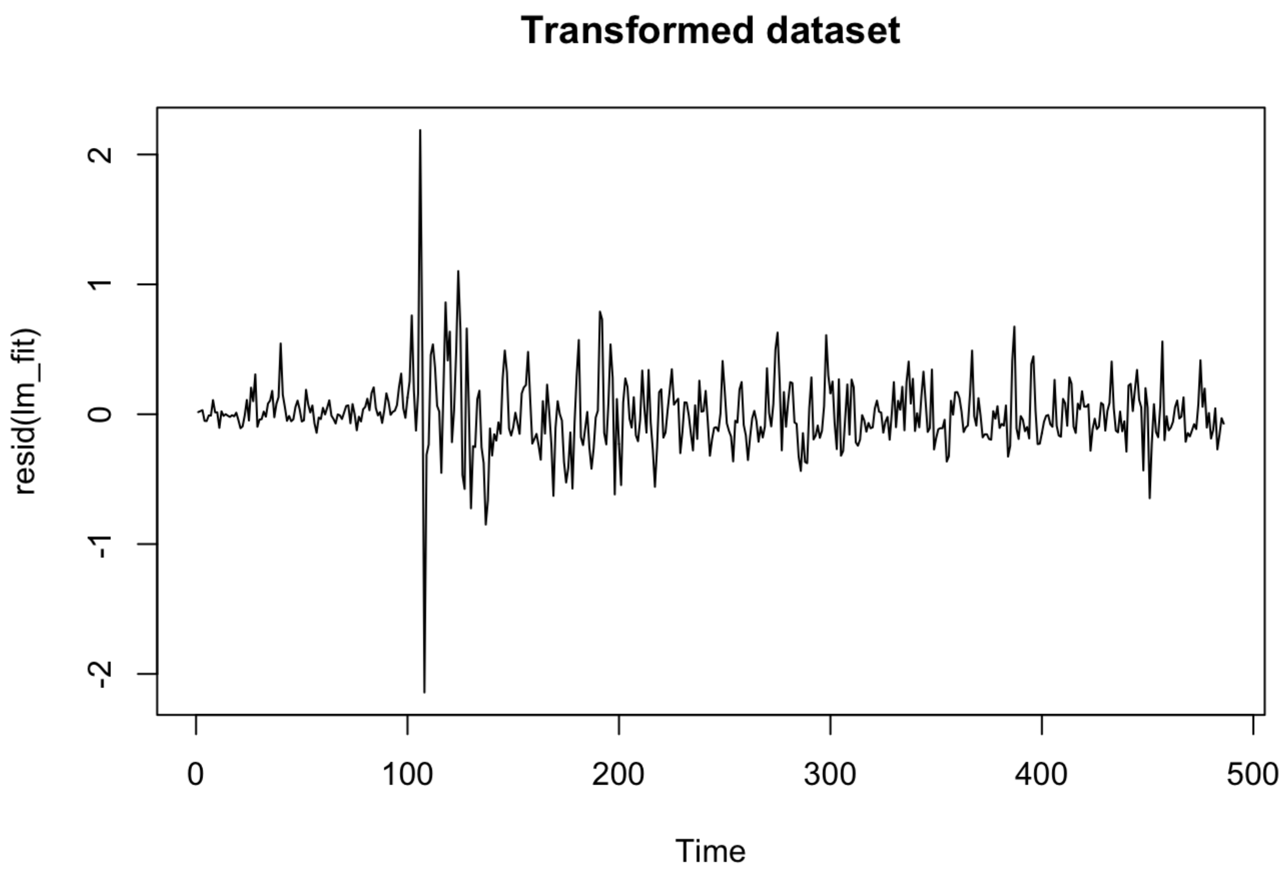
First, since we are interested to observe if there exist a relationship between mortgage rate and federal fund rate, we will apply a linear regression method. Also, plot both the sample autocorrelation and partial autocorrelation functions to check if the property holds.

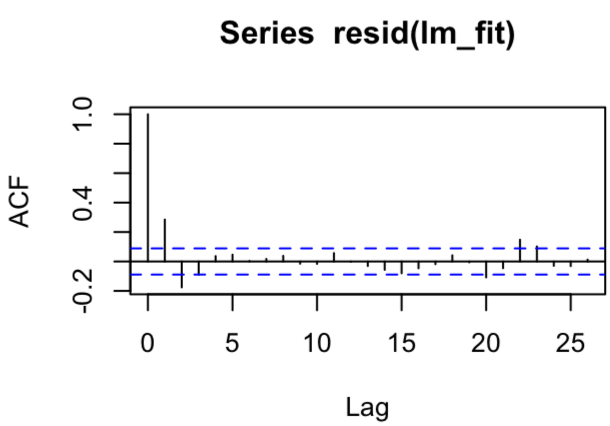

Regression Time Series Plot
resid(lm_fit)
4
2
0
-2
-4
0
100
200
300
400
500
Time

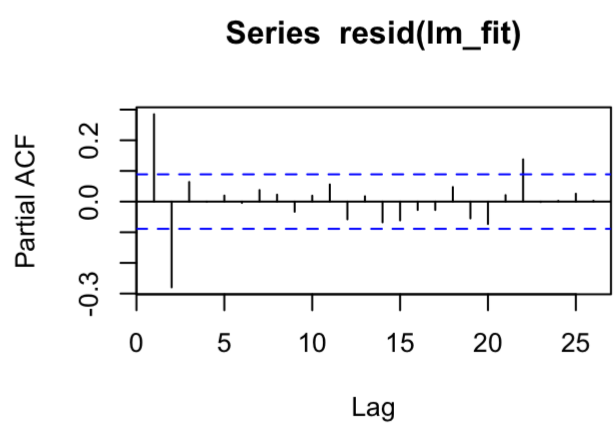

Series resid(lm_fit)
ACF
0.0
0.4
0.8
0
5
10
15
20
25
Lag


Series resid(lm_fit)
Partial ACF
-0.2
0.4
0
5
10
15
20
25
Lag

The plot suggest that the time series violates the assumption of linearity, normality and equal variables. Also, that it is not an identically independent distribution, and thus we would apply a transformation.

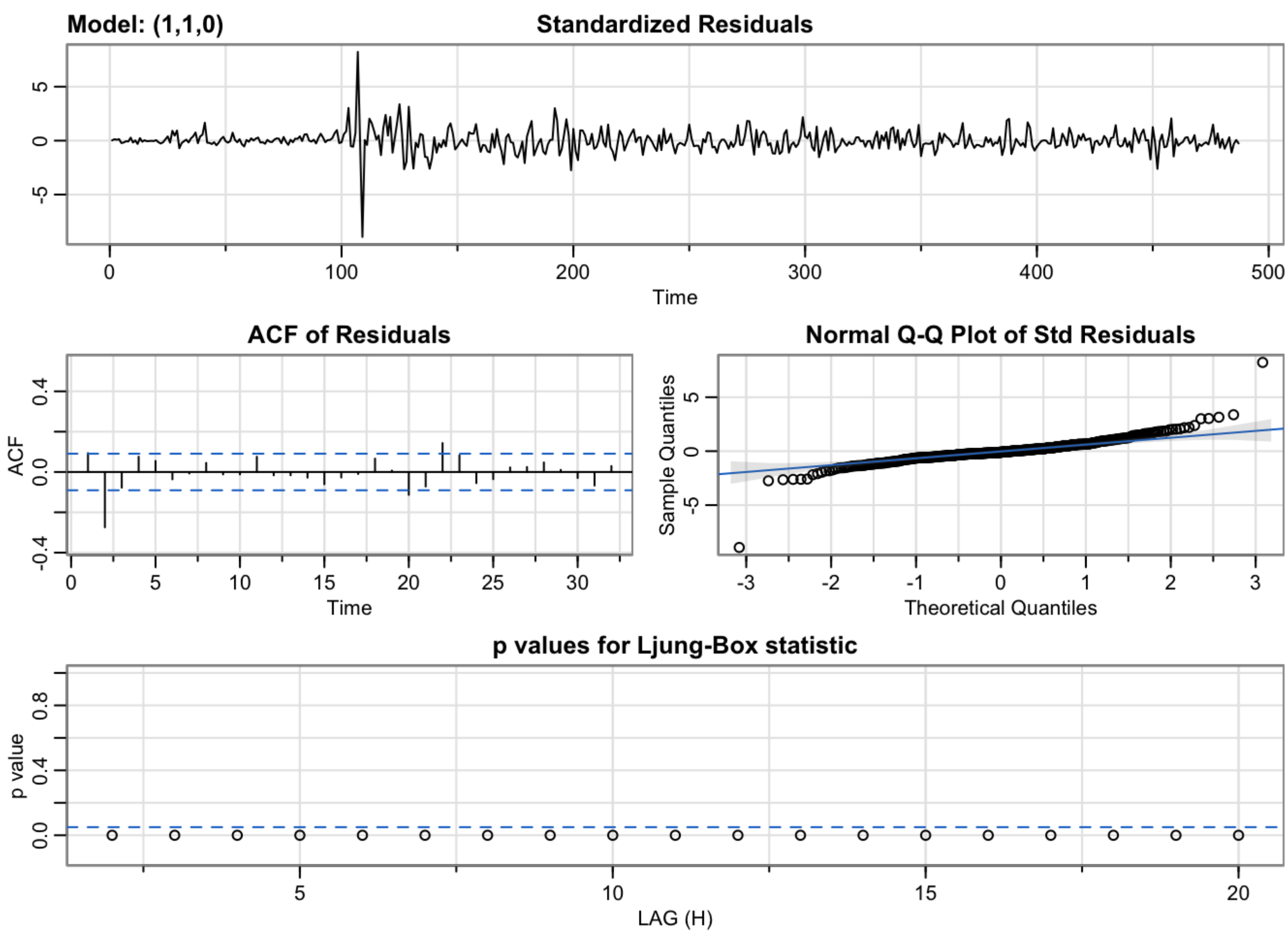

Transformed dataset
resid(lm_fit)
2
1
0
-1
-2
0
100
200
300
400
500
Time


Series resid(lm_fit)
ACF
1.0
0.4
-0.2
0
5
10
15
20
25
Lag

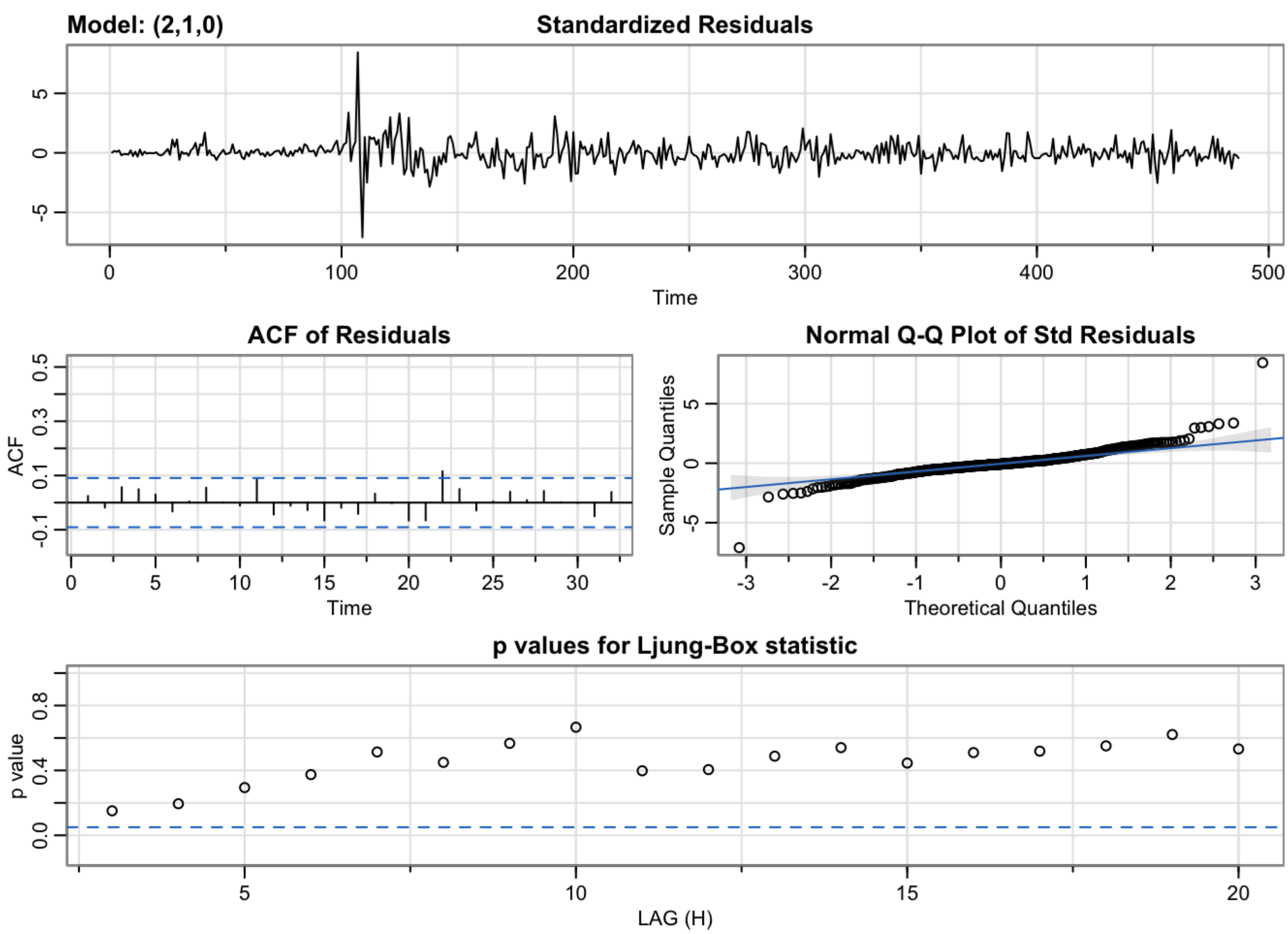

Series resid(lm_fit)
Partial ACF
0.2
0.0
-0.3
0
5
10
15
20
25
Lag

After transforming the data, we observe the time values converge to mean indicating stationatity assumption holds and the IID assumptions holding well.

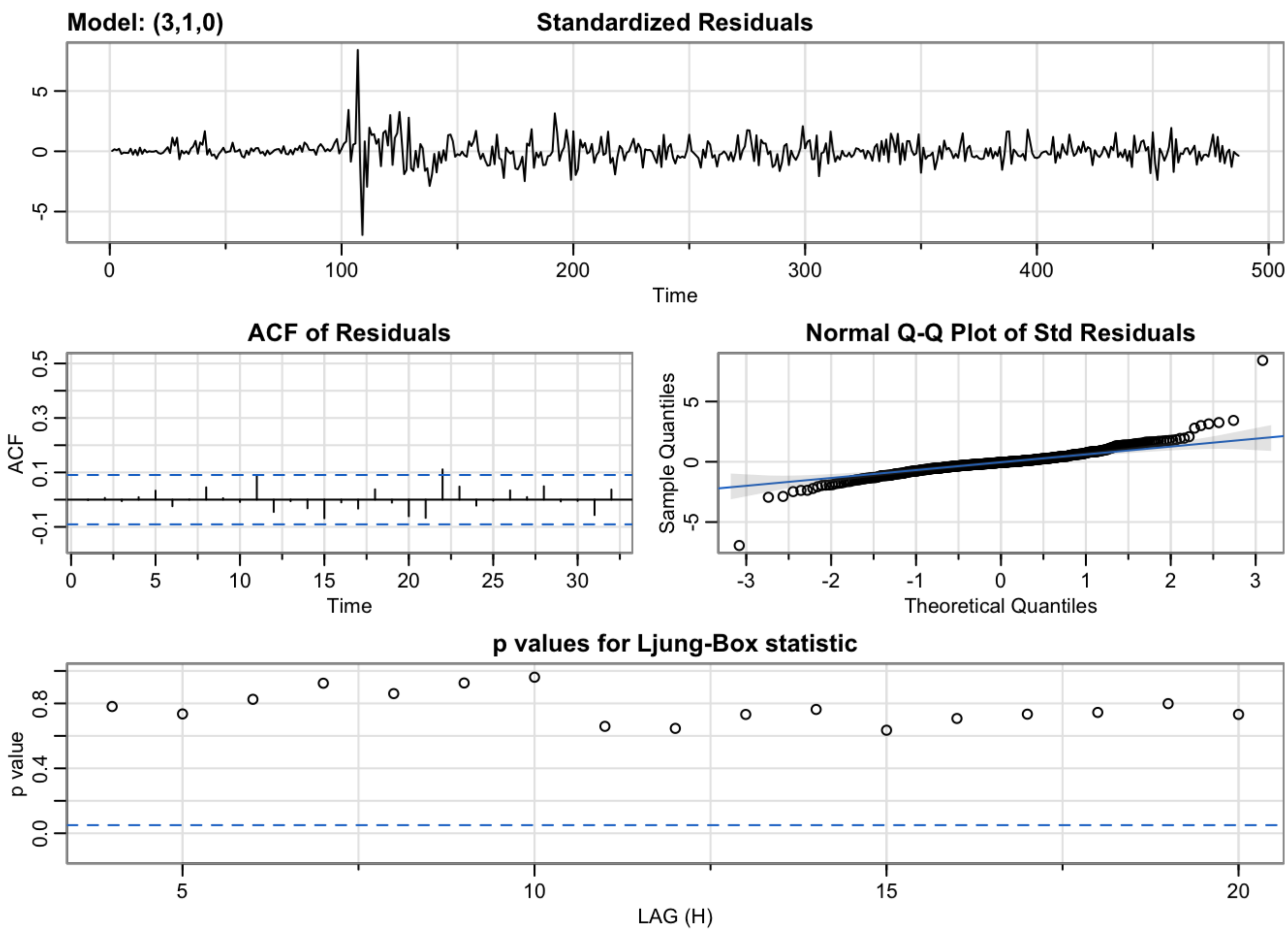
```
## initial  value -1.292874
## iter   2 value -1.335274
## iter   3 value -1.336976
## iter   4 value -1.340823
## iter   5 value -1.340824
## iter   5 value -1.340824
## iter   5 value -1.340824
## final  value -1.340824
## converged
## initial  value -1.341723
## iter   2 value -1.341723
## iter   3 value -1.341723
## iter   4 value -1.341723
## iter   4 value -1.341723
## iter   4 value -1.341723
## final  value -1.341723
## converged
```



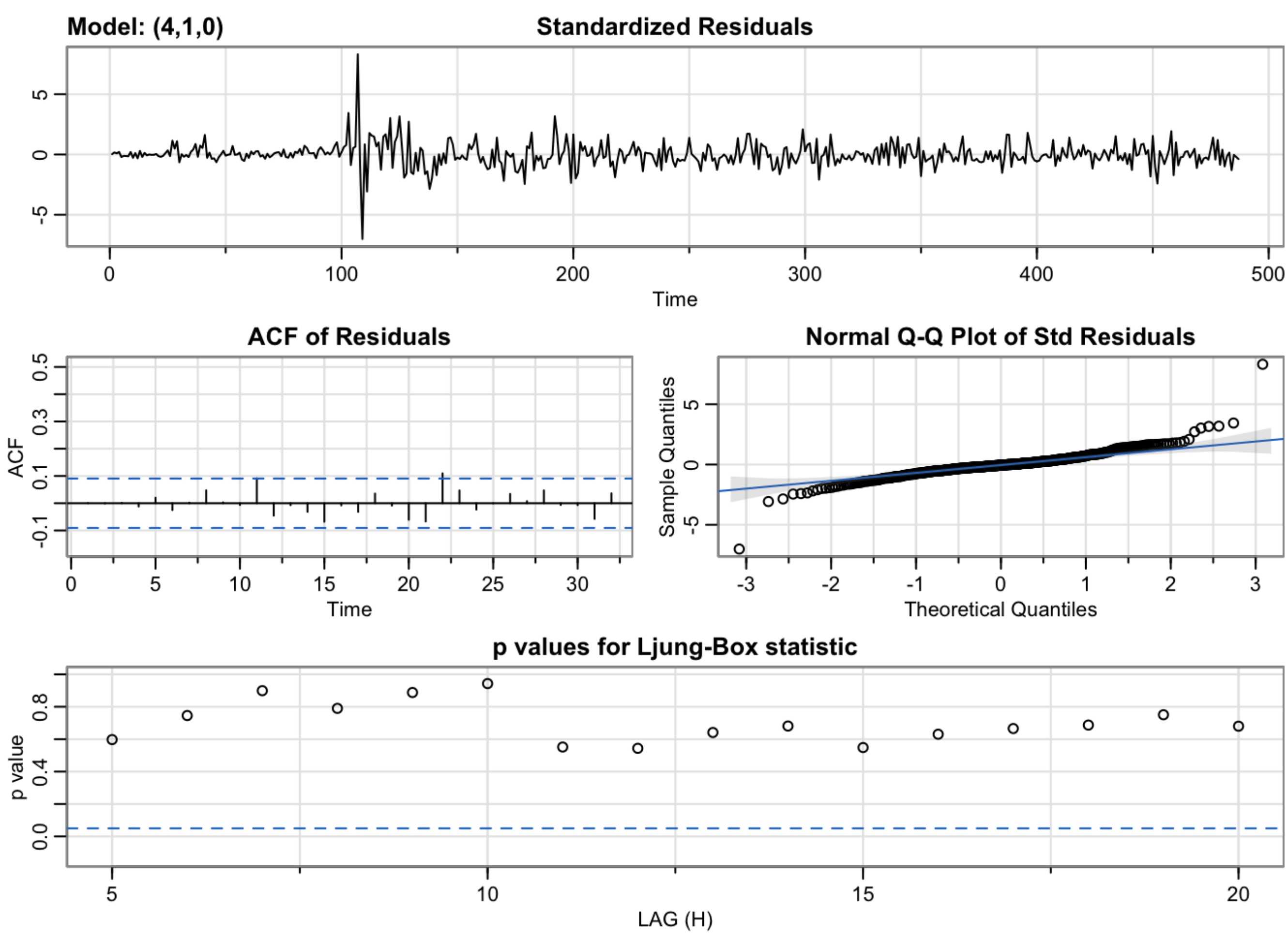
```
## initial  value -1.291846
## iter   2 value -1.367617
## iter   3 value -1.376856
## iter   4 value -1.379512
## iter   5 value -1.383373
## iter   6 value -1.384152
## iter   7 value -1.384231
## iter   8 value -1.384232
## iter   8 value -1.384232
## final  value -1.384232
## converged
## initial  value -1.385952
## iter   2 value -1.385953
## iter   3 value -1.385953
## iter   4 value -1.385953
## iter   5 value -1.385953
## iter   5 value -1.385953
## iter   5 value -1.385953
## final  value -1.385953
## converged
```



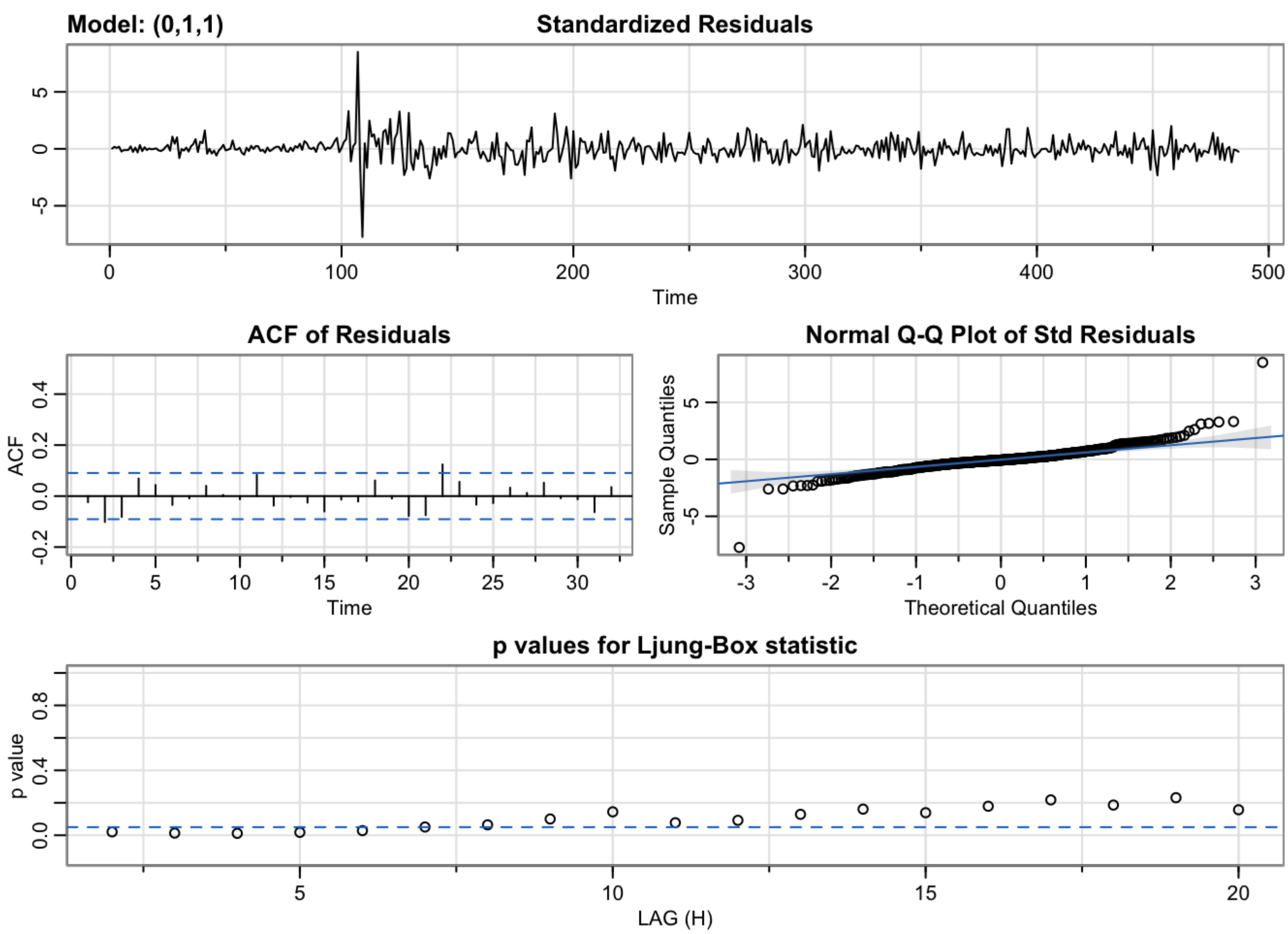
```
## initial  value -1.290821
## iter   2 value -1.360867
## iter   3 value -1.377336
## iter   4 value -1.381694
## iter   5 value -1.384226
## iter   6 value -1.386503
## iter   7 value -1.386738
## iter   8 value -1.386760
## iter   9 value -1.386763
## iter   9 value -1.386763
## iter   9 value -1.386763
## final  value -1.386763
## converged
## initial  value -1.389440
## iter   2 value -1.389441
## iter   3 value -1.389442
## iter   4 value -1.389442
## iter   5 value -1.389442
## iter   5 value -1.389442
## iter   5 value -1.389442
## final  value -1.389442
## converged
```



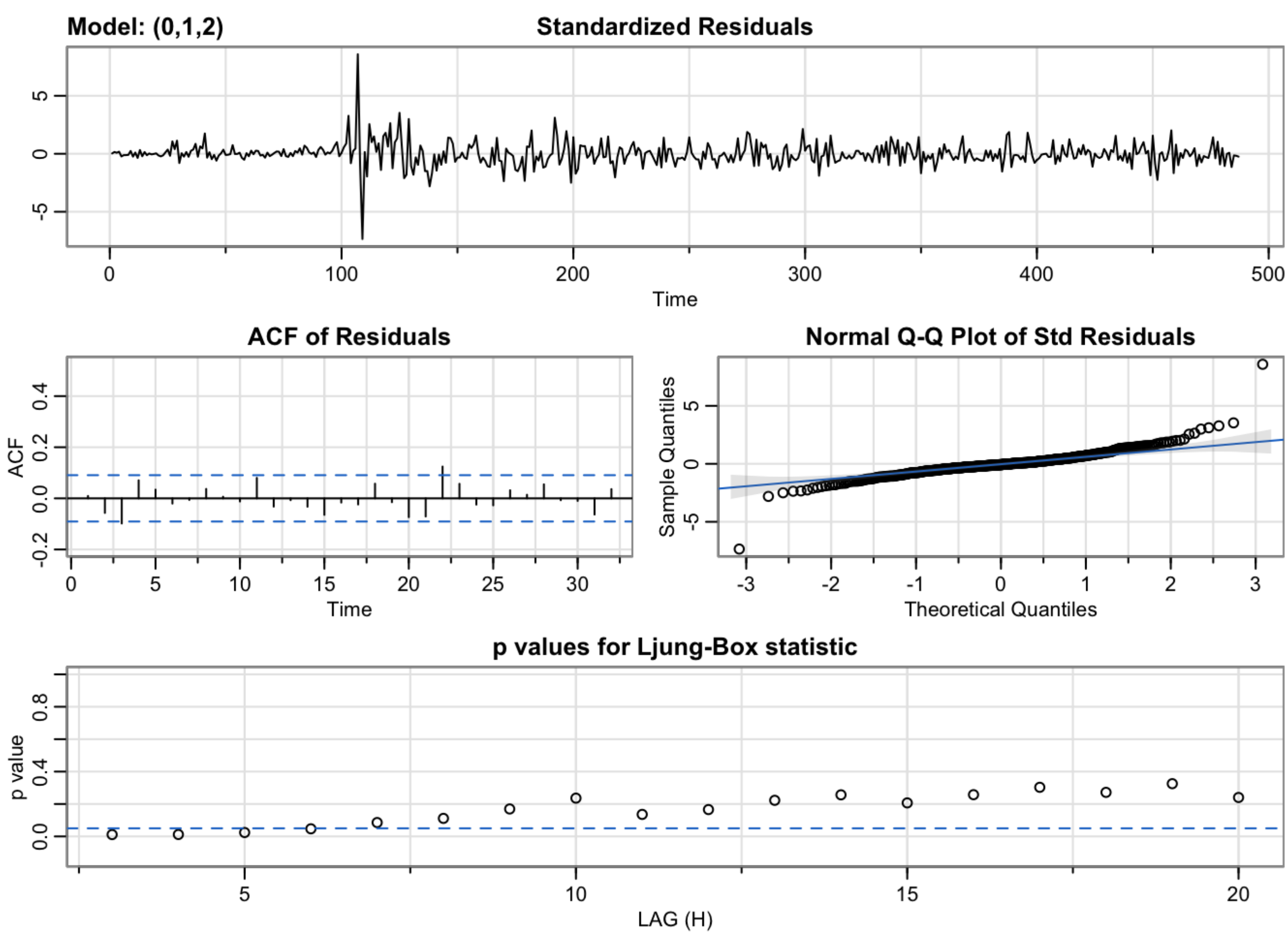
```
## initial  value -1.289828
## iter   2 value -1.361520
## iter   3 value -1.375182
## iter   4 value -1.381272
## iter   5 value -1.383806
## iter   6 value -1.385288
## iter   7 value -1.386042
## iter   8 value -1.386159
## iter   9 value -1.386169
## iter  10 value -1.386169
## iter  10 value -1.386169
## iter  10 value -1.386169
## final  value -1.386169
## converged
## initial  value -1.389825
## iter   2 value -1.389826
## iter   3 value -1.389826
## iter   4 value -1.389826
## iter   5 value -1.389826
## iter   5 value -1.389826
## iter   5 value -1.389826
## final  value -1.389826
## converged
```



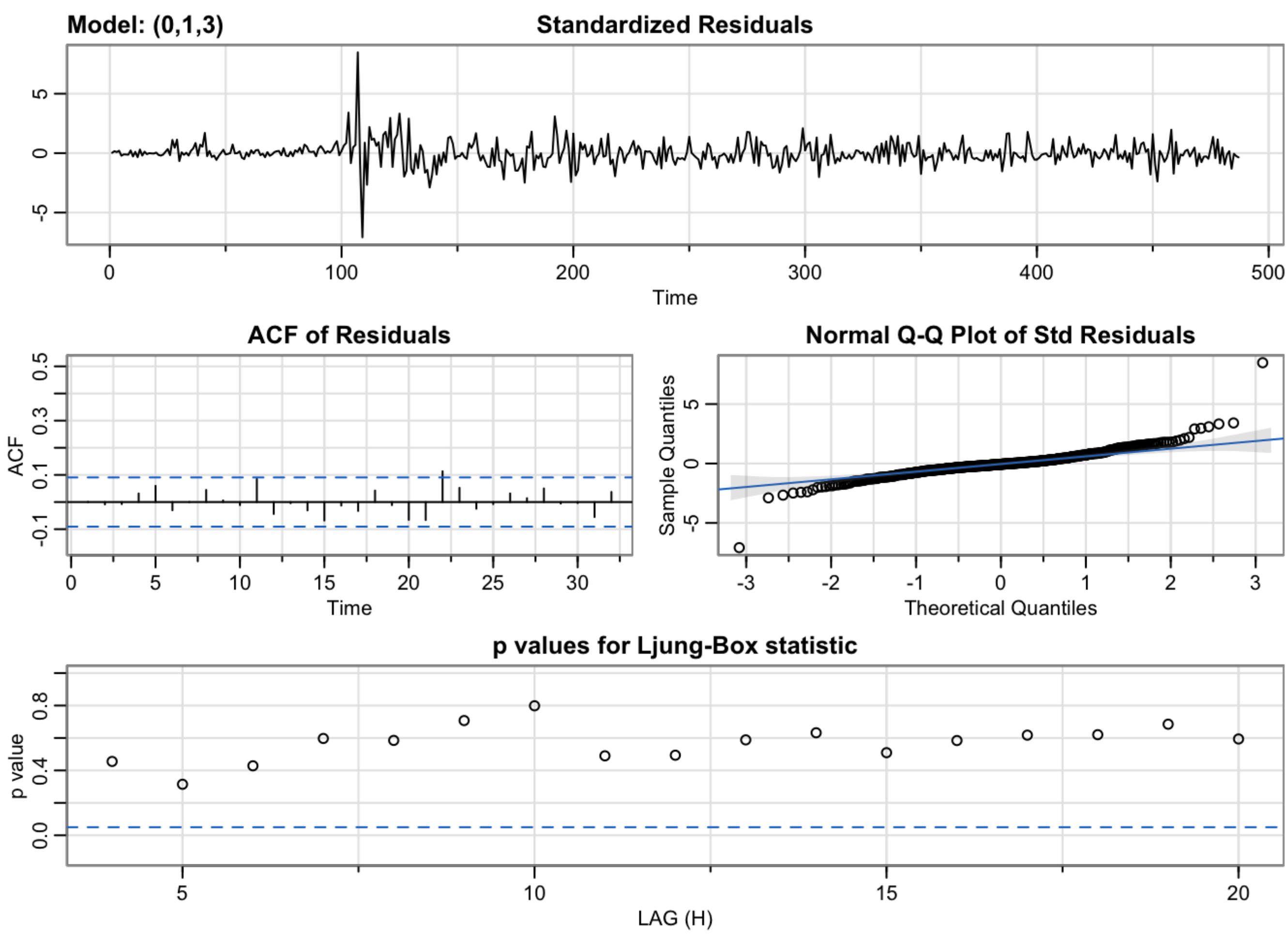
```
## initial  value -1.293902
## iter   2 value -1.354815
## iter   3 value -1.363632
## iter   4 value -1.371765
## iter   5 value -1.374593
## iter   6 value -1.378942
## iter   7 value -1.379023
## iter   8 value -1.379037
## iter   9 value -1.379037
## iter  10 value -1.379037
## iter  10 value -1.379037
## iter  10 value -1.379037
## final  value -1.379037
## converged
## initial  value -1.378758
## iter   2 value -1.378758
## iter   3 value -1.378758
## iter   3 value -1.378758
## iter   3 value -1.378758
## final  value -1.378758
## converged
```



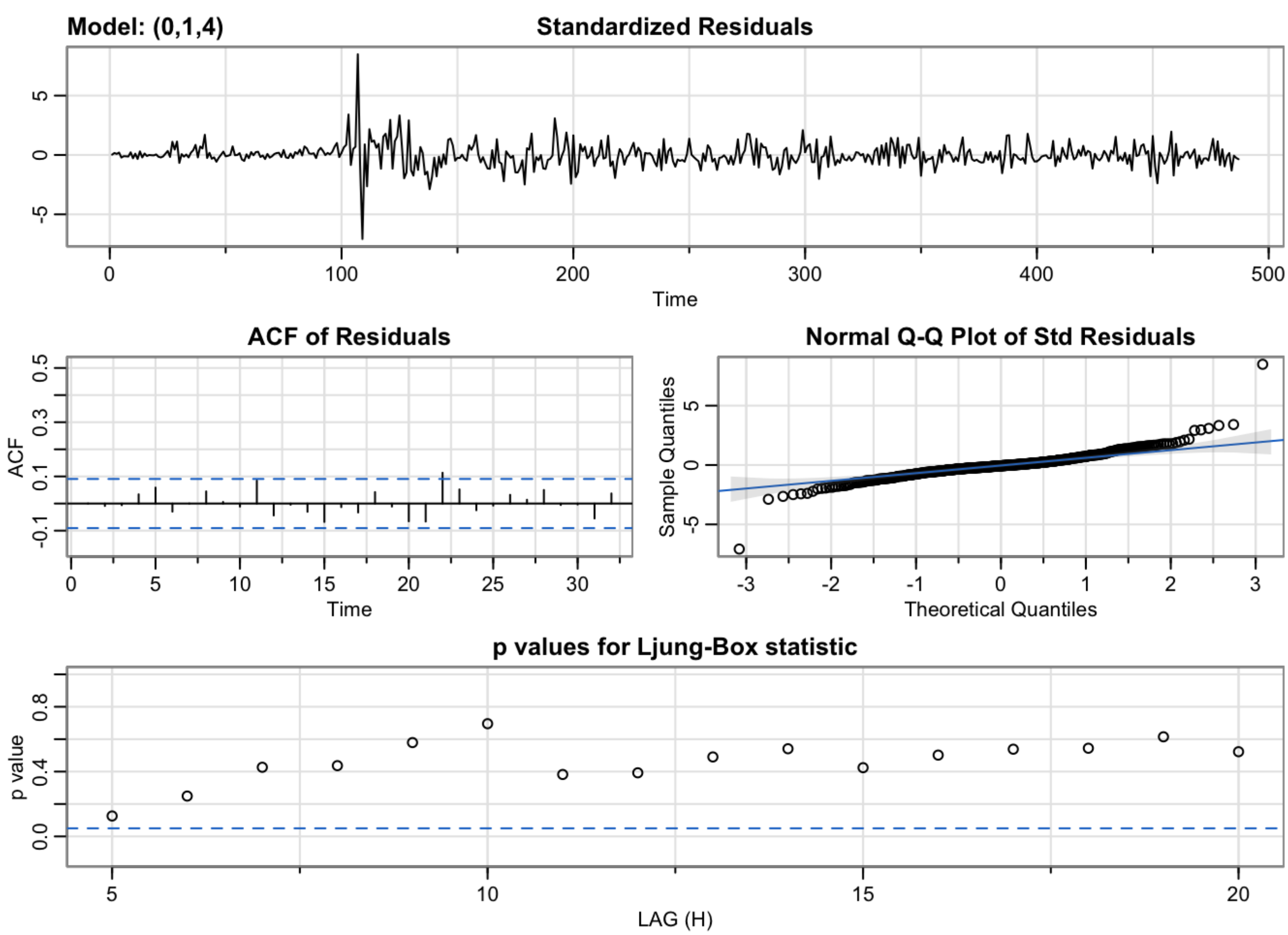
```
## initial  value -1.293902 
## iter   2 value -1.369518
## iter   3 value -1.372536
## iter   4 value -1.376692
## iter   5 value -1.379172
## iter   6 value -1.380725
## iter   7 value -1.380904
## iter   8 value -1.380917
## iter   9 value -1.380917
## iter  10 value -1.380917
## iter  11 value -1.380917
## iter  12 value -1.380917
## iter  12 value -1.380917
## iter  12 value -1.380917
## final  value -1.380917 
## converged
## initial  value -1.380649 
## iter   2 value -1.380650
## iter   3 value -1.380650
## iter   3 value -1.380650
## iter   3 value -1.380650
## final  value -1.380650 
## converged
```



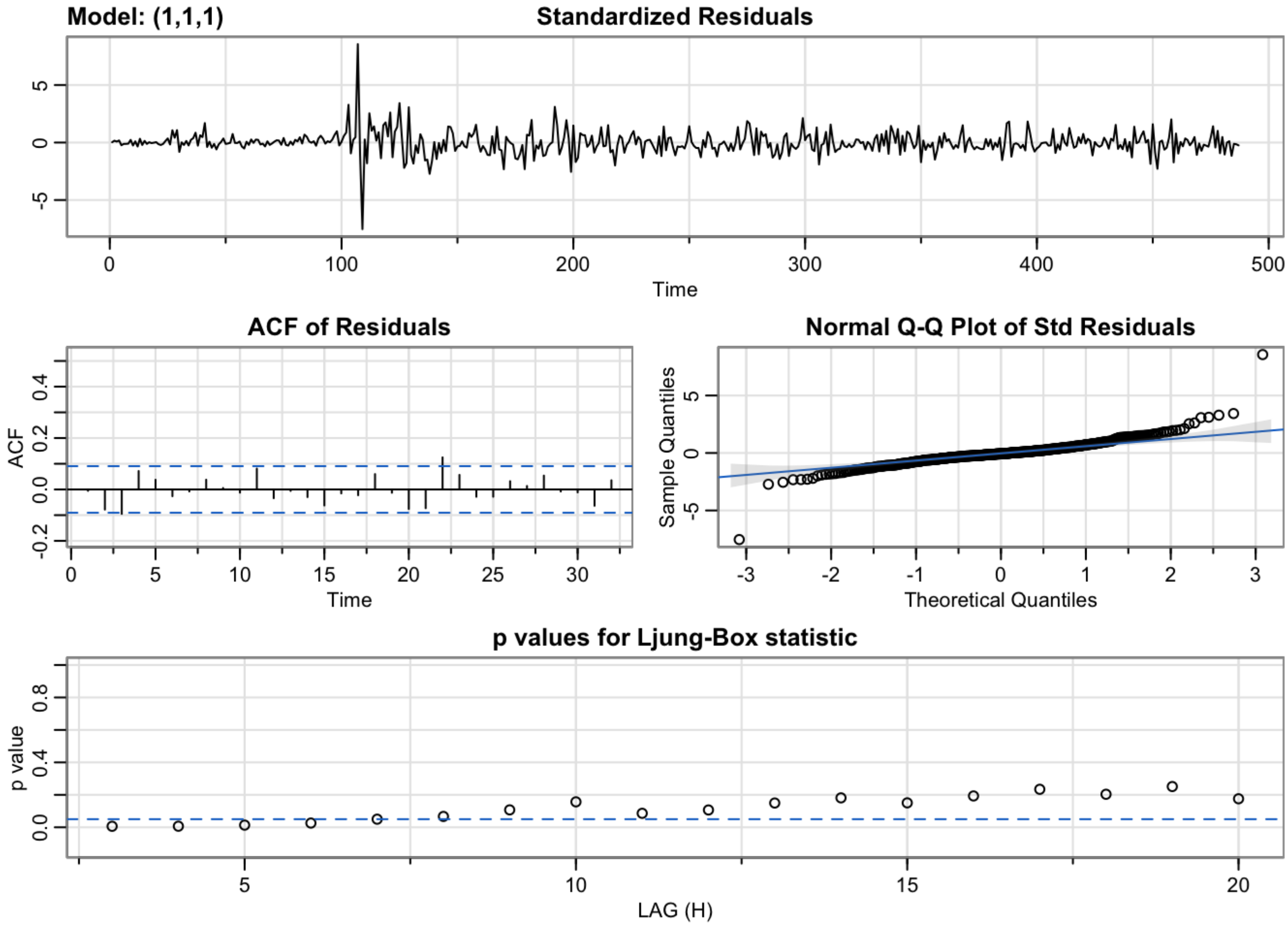
```
## initial  value -1.293902
## iter   2 value -1.373043
## iter   3 value -1.375037
## iter   4 value -1.381745
## iter   5 value -1.384038
## iter   6 value -1.387721
## iter   7 value -1.387725
## iter   8 value -1.387725
## iter   8 value -1.387725
## iter   8 value -1.387725
## final  value -1.387725
## converged
## initial  value -1.387419
## iter   2 value -1.387420
## iter   3 value -1.387420
## iter   4 value -1.387420
## iter   4 value -1.387420
## iter   4 value -1.387420
## final  value -1.387420
## converged
```



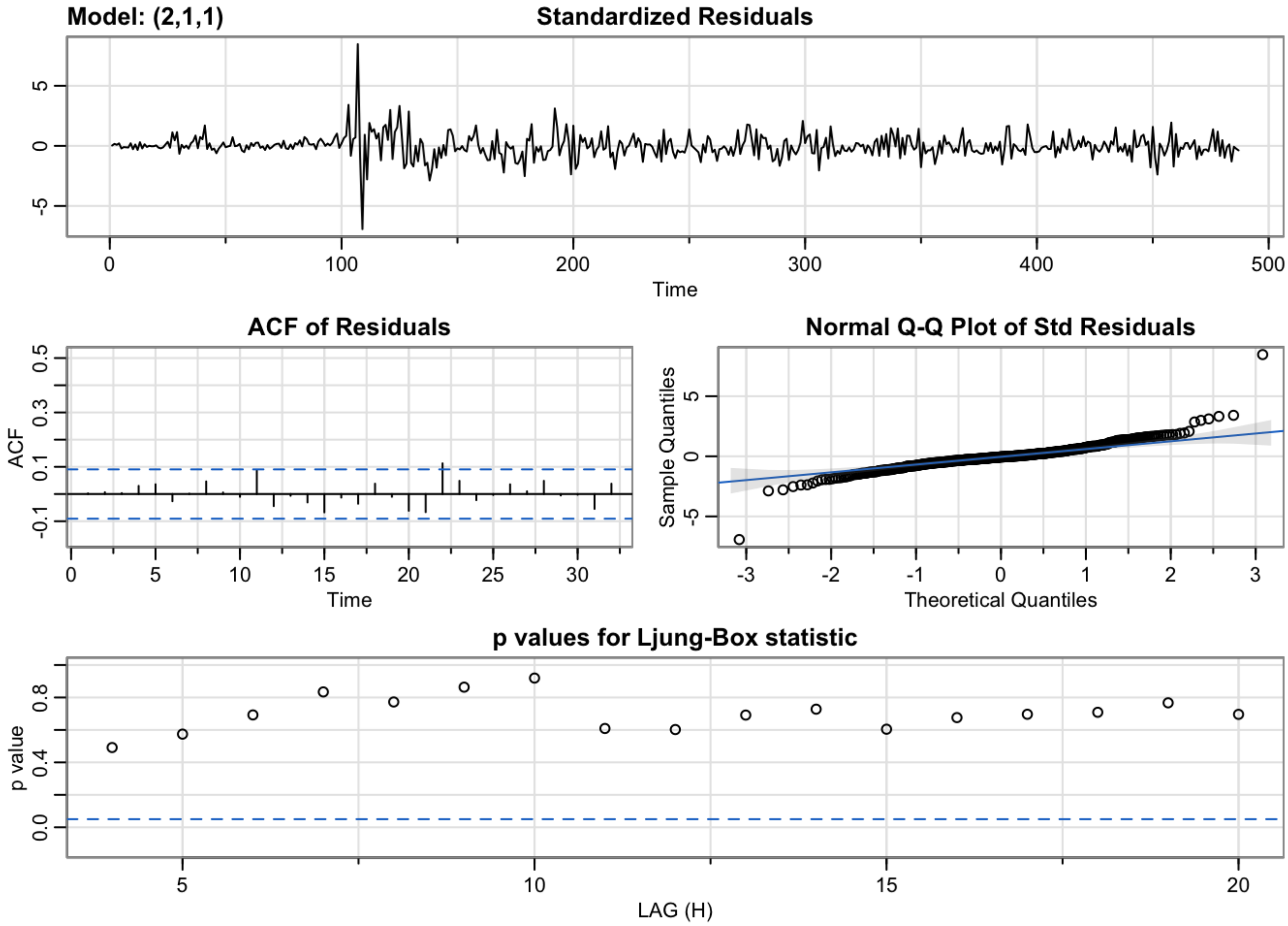
```
## initial  value -1.293902
## iter   2 value -1.373167
## iter   3 value -1.374120
## iter   4 value -1.381392
## iter   5 value -1.383804
## iter   6 value -1.387399
## iter   7 value -1.387709
## iter   8 value -1.387728
## iter   9 value -1.387730
## iter  10 value -1.387730
## iter  10 value -1.387730
## iter  10 value -1.387730
## final  value -1.387730
## converged
## initial  value -1.387424
## iter   2 value -1.387424
## iter   3 value -1.387425
## iter   4 value -1.387425
## iter   5 value -1.387425
## iter   5 value -1.387425
## iter   5 value -1.387425
## final  value -1.387425
## converged
```



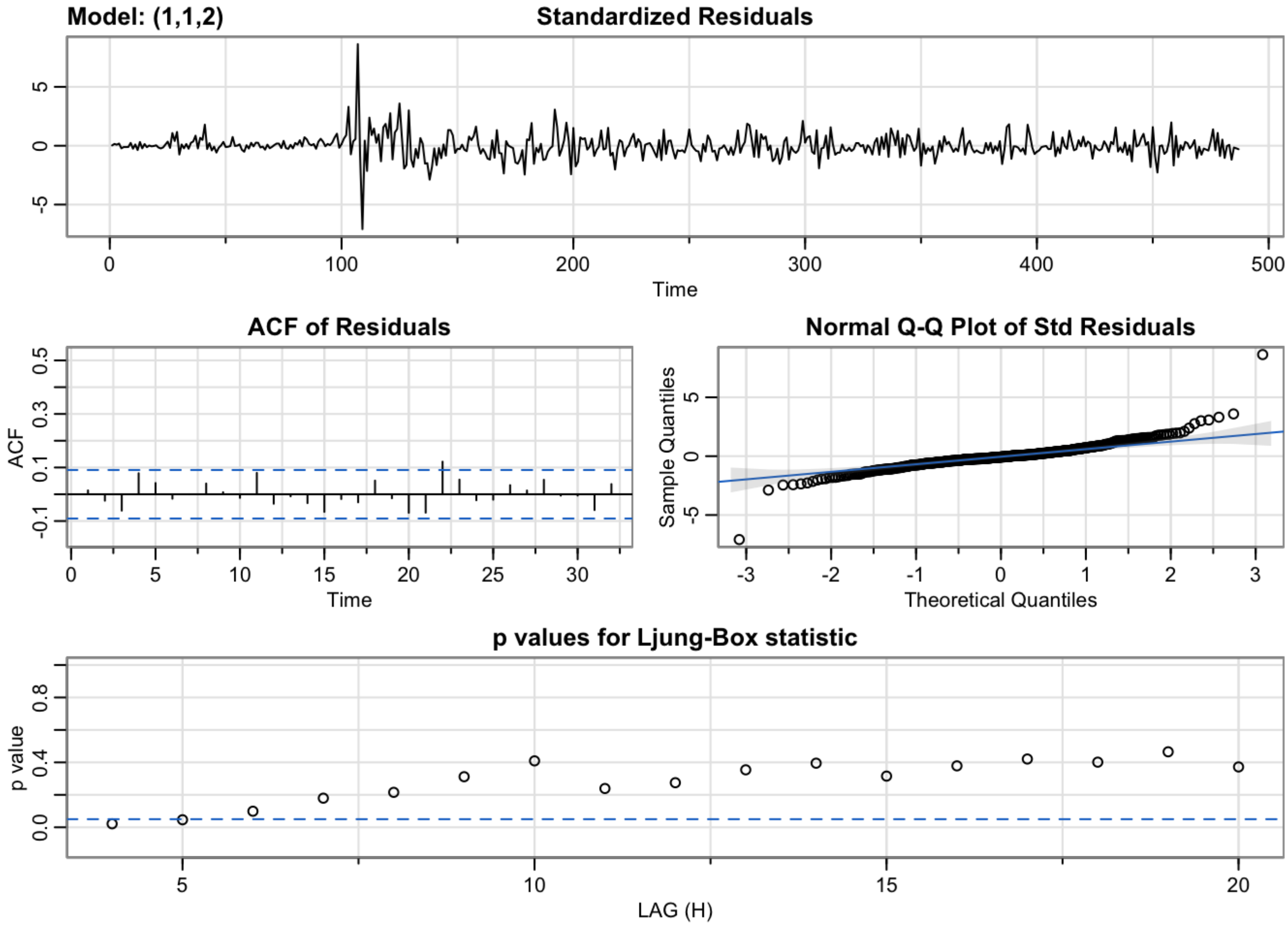
```
## initial  value -1.292874
## iter   2 value -1.337561
## iter   3 value -1.356532
## iter   4 value -1.364706
## iter   5 value -1.376416
## iter   6 value -1.378308
## iter   7 value -1.379026
## iter   8 value -1.379078
## iter   9 value -1.379086
## iter  10 value -1.379106
## iter  11 value -1.379111
## iter  12 value -1.379111
## iter  13 value -1.379111
## iter  13 value -1.379111
## iter  13 value -1.379111
## final  value -1.379111
## converged
## initial  value -1.379834
## iter   2 value -1.379834
## iter   3 value -1.379835
## iter   4 value -1.379835
## iter   5 value -1.379835
## iter   5 value -1.379835
## iter   5 value -1.379835
## final  value -1.379835
## converged
```


Model: (1,1,1)
Standardized Residuals
Time
ACF of Residuals
ACF
Time
Normal Q-Q Plot of Std Residuals
Sample Quantiles
Theoretical Quantiles
p values for Ljung-Box statistic
p value
LAG (H)

```
## initial  value -1.291846
## iter   2 value -1.358729
## iter   3 value -1.379782
## iter   4 value -1.382219
## iter   5 value -1.386253
## iter   6 value -1.387047
## iter   7 value -1.387088
## iter   8 value -1.387089
## iter   9 value -1.387091
## iter  10 value -1.387094
## iter  11 value -1.387095
## iter  12 value -1.387095
## iter  13 value -1.387096
## iter  14 value -1.387096
## iter  15 value -1.387097
## iter  16 value -1.387097
## iter  17 value -1.387097
## iter  18 value -1.387097
## iter  18 value -1.387097
## iter  18 value -1.387097
## final  value -1.387097
## converged
## initial  value -1.388806
## iter   2 value -1.388806
## iter   3 value -1.388807
## iter   4 value -1.388807
## iter   5 value -1.388807
## iter   5 value -1.388807
## iter   5 value -1.388807
## final  value -1.388807
## converged
```


Model: (2,1,1)
Standardized Residuals
Time
ACF of Residuals
ACF
Time
Normal Q-Q Plot of Std Residuals
Sample Quantiles
Theoretical Quantiles
p values for Ljung-Box statistic
p value
LAG (H)

```
## initial  value -1.292874
## iter   2 value -1.354655
## iter   3 value -1.374860
## iter   4 value -1.377976
## iter   5 value -1.379964
## iter   6 value -1.381788
## iter   7 value -1.382097
## iter   8 value -1.382134
## iter   9 value -1.382195
## iter  10 value -1.382328
## iter  11 value -1.382348
## iter  12 value -1.382433
## iter  13 value -1.382451
## iter  14 value -1.382612
## iter  15 value -1.382739
## iter  16 value -1.382802
## iter  17 value -1.382843
## iter  18 value -1.382870
## iter  19 value -1.382885
## iter  20 value -1.382888
## iter  21 value -1.382891
## iter  22 value -1.382892
## iter  23 value -1.382892
## iter  24 value -1.382893
## iter  25 value -1.382893
## iter  26 value -1.382893
## iter  27 value -1.382893
## iter  28 value -1.382893
## iter  29 value -1.382893
## iter  30 value -1.382893
## iter  30 value -1.382893
## iter  30 value -1.382893
## final  value -1.382893
## converged
## initial  value -1.383605
## iter   2 value -1.383605
## iter   3 value -1.383606
## iter   4 value -1.383606
## iter   5 value -1.383606
## iter   5 value -1.383606
## iter   5 value -1.383606
## final  value -1.383606
## converged
```




```
##                    AR(1)     AR(2)      AR(3)      AR(4)
## fit                List,14   List,14    List,14    List,14
## degrees_of_freedom 484       483        482        481
## ttable             Numeric,8 Numeric,12 Numeric,16 Numeric,20
## AIC                0.1667759 0.08243151 0.07956906 0.08291563
## AICc               0.166827  0.08253396 0.07974017 0.08317283
## BIC                0.1926167 0.1168859  0.1226371  0.1345972
```

```
##                    MA(1)      MA(2)      MA(3)      MA(4)
## fit                List,14    List,14    List,14    List,14
## degrees_of_freedom 484        483        482        481
## ttable             Numeric,8  Numeric,12 Numeric,16 Numeric,20
## AIC                0.0927065  0.09303814 0.08361276 0.08771915
## AICc               0.09275763 0.09314059 0.08378387 0.08797635
## BIC                0.1185473  0.1274925  0.1266808  0.1394007
```

```
##                    ARMA(1,1)  ARMA(2,1)  ARMA(1,2)
## fit                List,14    List,14    List,14
## degrees_of_freedom 483        482        482
## ttable             Numeric,12 Numeric,16 Numeric,16
## AIC                0.0946686  0.08083899 0.09124176
## AICc               0.09477105 0.0810101  0.09141287
## BIC                0.129123   0.123907   0.1343098
```

From the above computation and residual analysis plots of different model, we would select the optimal model as the model with the lowest AICc. This AICc is a model selection criteria that helps us chooses the optimal model to predict future behavior.
The selected optimal model is AR(3) with the following important statistics. *Optimal Fitted Model: X_t - mu = phi1(X_t-1 - mu) + phi2(X_t-2 - mu) + phi3(X_t-3 - mu)+w_t, where mu = 0.1098 and w_t ~ iid(0, sigma^2 = 0.06207) on 482 degrees of freedom* And below is the AIC value, coefficients of phi and statistic summary.

```
## [1] 0.07974017
```

```
##        ar1        ar2        ar3       xreg
##  0.4613887 -0.3301643  0.0838379  0.1097510
```

```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), xreg = xreg, transform.pars = trans, fixed = fixed, optim.contro
l = list(trace = trc,
##     REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ar2     ar3    xreg
##       0.4614  -0.3302  0.0838  0.1098
## s.e.  0.0517   0.0484  0.0454  0.0237
##
## sigma^2 estimated as 0.06207:  log likelihood = -14.34,  aic = 38.67
##
## $degrees_of_freedom
## [1] 482
##
## $ttable
##      Estimate     SE t.value p.value
## ar1    0.4614 0.0517  8.9281  0.0000
## ar2   -0.3302 0.0484 -6.8276  0.0000
## ar3    0.0838 0.0454  1.8453  0.0656
## xreg   0.1098 0.0237  4.6368  0.0000
##
## $AIC
## [1] 0.07956906
##
## $AICc
## [1] 0.07974017
##
## $BIC
## [1] 0.1226371
```

The plot suggest ACF is tailing off and according to the normal Q-Q plot, the fitted values suggest linearity, normality and equal variance among the data. this means the data is normally distributed and also no obvious departure of residuals from whiteness. And, according to the Ljung-Box plot, we fail to reject the null hypothesis and conclude that the model does not show lack of fit and the p-value indicate significance. The statistic coefficients suggests all regression coefficients have a significance, including the average monthly federal fund rate which is 0.1098.

Section 3: Results:

Upon applying the building model techniques and selection criteria, we believe the optimal model to determine the monthly mortgage rent and to confirm whether the monthly federal fund rate depends on mortgage rent is AR(3). From the above analysis, we can support our claim through the transformed dataset computation and charts of ACF and PACF.

Section 4: Appendix:

```r
library(astsa)
mydata <- read.csv("mortgage.txt", header = T, sep = " ")
plot.ts(cbind(mydata$morg, mydata$ffr), plot.type='single', col=1:2, lty=1:2)
legend( 'topright', col=1:2,lty=1:2, legend=c("Mortgage rate", "Federal Rate") )
par(mfrow = c(2,2))
#acf
acf(mydata$morg, main= "")

#pacf
pacf(mydata$morg, main= "")
#transform
plot.ts(diff(mydata$morg), main = "Transformed Mortgage Rate Time Series Plot")
par(mfrow = c(2,2))
#acf
acf(diff(mydata$morg), main= "")
#pacf
pacf(diff(mydata$morg), main= "")

#test models
AR1 <- sarima(mydata$morg, p=1, d=1, q=0)
AR2 <- sarima(mydata$morg, p=2, d=1, q=0)
AR3 <- sarima(mydata$morg, p=3, d=1, q=0)
AR4 <- sarima(mydata$morg, p=4, d=1, q=0)

MA1 <- sarima(mydata$morg, p=0, d=1, q=1)
MA2 <- sarima(mydata$morg, p=0, d=1, q=2)
MA3 <- sarima(mydata$morg, p=0, d=1, q=3)
MA4 <- sarima(mydata$morg, p=0, d=1, q=4)

ARMA1 <- sarima(mydata$morg, p=1, d=1, q=1)
ARMA2 <- sarima(mydata$morg, p=2, d=1, q=1)
ARMA3 <- sarima(mydata$morg, p=1, d=1, q=2)

#using Ljung-Box statistic to determine p-value
ar_aic <- cbind(AR1, AR2, AR3, AR4)
colnames(ar_aic) <- c("AR(1)", "AR(2)", "AR(3)", "AR(4)")
ar_aic

ma_aic <- cbind(MA1, MA2, MA3, MA4)
colnames(ma_aic) <- c("MA(1)", "MA(2)", "MA(3)", "MA(4)")
ma_aic

arma_aic <- cbind(ARMA1, ARMA2, ARMA3)
colnames(arma_aic) <- c("ARMA(1,1)", "ARMA(2,1)", "ARMA(1,2)")
arma_aic
# AR3 <- sarima(mydata$morg, p=3, d=1, q=0)
AR3$AICc
AR3$fit$coef
AR3
n <- length(mydata$ffr)
plot.ts(mydata$ffr, main = "Federal Fund Rate Time Series Plot")
```

```
par(mfrow = c(2,2))
#acf
acf(mydata$ffr, main= "")

#pacf
pacf(mydata$ffr, main= "")

lm_fit <- lm( mydata$morg[-1] ~ mydata$ffr[-n] )
plot.ts( resid(lm_fit) )
par(mfrow=c(2,2))
acf(resid(lm_fit))
pacf(resid(lm_fit))

lm_fit <- lm( diff(mydata$morg[-1]) ~ diff(mydata$ffr[-n]) )
plot.ts( resid(lm_fit), main = "Transformed dataset")

par(mfrow=c(2,2))
acf(resid(lm_fit))
pacf(resid(lm_fit))
#test models
AR1 <-  sarima(mydata$morg[-1], p=1,d=1,q=0, xreg=mydata$ffr[-n])
AR2 <-  sarima(mydata$morg[-1], p=2,d=1,q=0, xreg=mydata$ffr[-n])
AR3 <-  sarima(mydata$morg[-1], p=3,d=1,q=0, xreg=mydata$ffr[-n])
AR4 <-  sarima(mydata$morg[-1], p=4,d=1,q=0, xreg=mydata$ffr[-n])

MA1 <-  sarima(mydata$morg[-1], p=0,d=1,q=1, xreg=mydata$ffr[-n])
MA2 <-  sarima(mydata$morg[-1], p=0,d=1,q=2, xreg=mydata$ffr[-n])
MA3 <-  sarima(mydata$morg[-1], p=0,d=1,q=3, xreg=mydata$ffr[-n])
MA4 <-  sarima(mydata$morg[-1], p=0,d=1,q=4, xreg=mydata$ffr[-n])

ARMA1 <- sarima(mydata$morg[-1], p=1, d=1, q=1, xreg=mydata$ffr[-n])
ARMA2 <- sarima(mydata$morg[-1], p=2, d=1, q=1, xreg=mydata$ffr[-n])
ARMA3 <- sarima(mydata$morg[-1], p=1, d=1, q=2, xreg=mydata$ffr[-n])

#using Ljung-Box statistic to determine p-value
ar_aic <- cbind(AR1, AR2, AR3, AR4)
colnames(ar_aic) <- c("AR(1)", "AR(2)", "AR(3)", "AR(4)")
ar_aic

ma_aic <- cbind(MA1, MA2, MA3, MA4)
colnames(ma_aic) <- c("MA(1)", "MA(2)", "MA(3)", "MA(4)")
ma_aic

arma_aic <- cbind(ARMA1, ARMA2, ARMA3)
colnames(arma_aic) <- c("ARMA(1,1)", "ARMA(2,1)", "ARMA(1,2)")
arma_aic
AR3$AICc
AR3$fit$coef
AR3
```

Thank you!