Time series analysis on the stock price of Tesla Inc.

Wenhao Pan (3034946058), Ruojia Zhang, Mengzhu Sun, Xiangxi Wang, Mingmao Sun

November 13, 2021

Contents

1 Abstract

2 Introduction

Due to the increasing focus on carbon neutrality, the industry of replacing non-sustainable energy with sustainable energy has boomed in the past few years. Electricity, as a relatively environment-friendly energy, has been considered as replacement of some traditional energy, such as gasoline and diesel. Among all those enterprises pursing commercialized carbon neutrality, TSLA, as the largest electric car company, has been pioneering the fashion and aiming to transition the world to electric mobility. As the reflection of belief of the public, the stock price of TSLA has been sedentary for a period of time and has no evident increase until recent years. Therefore, we pick up the close price of TSLA stock of the recent 300 days to explore. In the following experiments, we utilize differencing, exponential smoothing, and fitting ARMA model, and combination of them to approximate the series.

3 Data Description

The TSLA stock price comes from Yahoo Finance (https://finance.yahoo.com). The stock price dataset consists of open price, close price, high price, and low price. Since they has roughly similar trend, we choose close price to experiment on. The whole volume of data, which contains 2791 data points, has variance 39768.49, max price 1208.59, min price 4.01, mean price 112.4271. The recent-300-day data has variance 22697.1, max price 1208.59, min price 330.21, mean price 654.1912.

```
##                        Data Variance     Max    Min     Mean Size
## 1   whole volume Tsla data 39768.49 1208.59   4.01 112.4271 2791
## 2 recent-300-day Tsla data 22697.10 1208.59 330.21 654.1912  300
```

4 Exploratory Data Analysis

To obtain a comprehensive understanding of the data, we conduct explanatory data analysis (EDA) first. Figure 1(a) is the time series plot of all the given time points. We observe that the stock prices of Tesla before 2020 are averagely and considerably lower than those after 2020. The significantly different scales of different parts of the time series make it hard to visually examine the trend and seasonality pattern of the time series. Moreover, since we are majorly interested in the recent activities of Tesla, we do not have to analyze all the available data. Therefore, for the sake of interest and convenience, we decide only to analyze the last 300 time points, which cover the period from `2020-08-26` to `2021-11-02` excluding weekends. Thus, whenever we use the word "data" in the following analysis, we implicitly mean the time series of the last three hundred time points.

Figure 1(b) is the time series plot of the close prices of Tesla in the last three hundred trading days before and including `2021-11-02`. We first observe that our data is roughly homoscedastic based on Figure 1(b). To verify our observation, we try the square root and natural log transformations and see whether they effectively stabilize the variance of the time series. Their plots are below in Figure 2.

We can see that both transformations unnecessarily increase the variance of the time series before mid-November in 2020 and do not change the variance of other time series data. Although both transformations shorten the vertical distance between the maximum and minimum of the time series after Oct. 2021, the spike after Oct. 2021 is more like an increasing trend than a considerable fluctuation. In short, both transformations are redundant, and we do not need to use any variance stabilizing transformation.

Back to Figure 1(b), intuitively, the data is not stationary because of a nonlinear and generally increasing trend. The trend first increases until around Feb. 2021 and then decreases until around Mid-May. 2021.

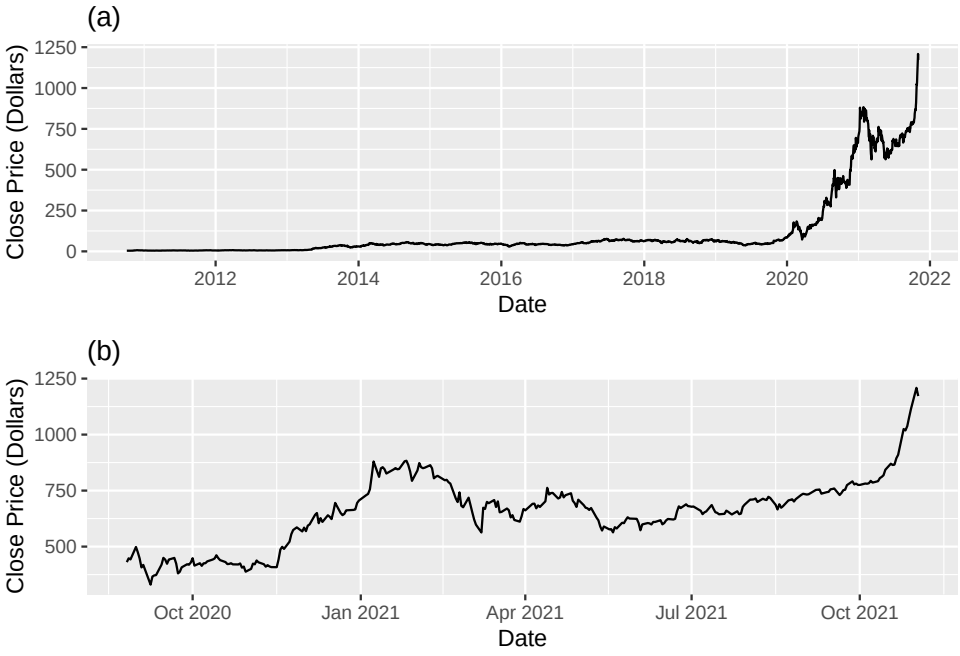


Figure 1: (a) Time series plot of all available trading days. (b) Time series plot of last 300 trading days

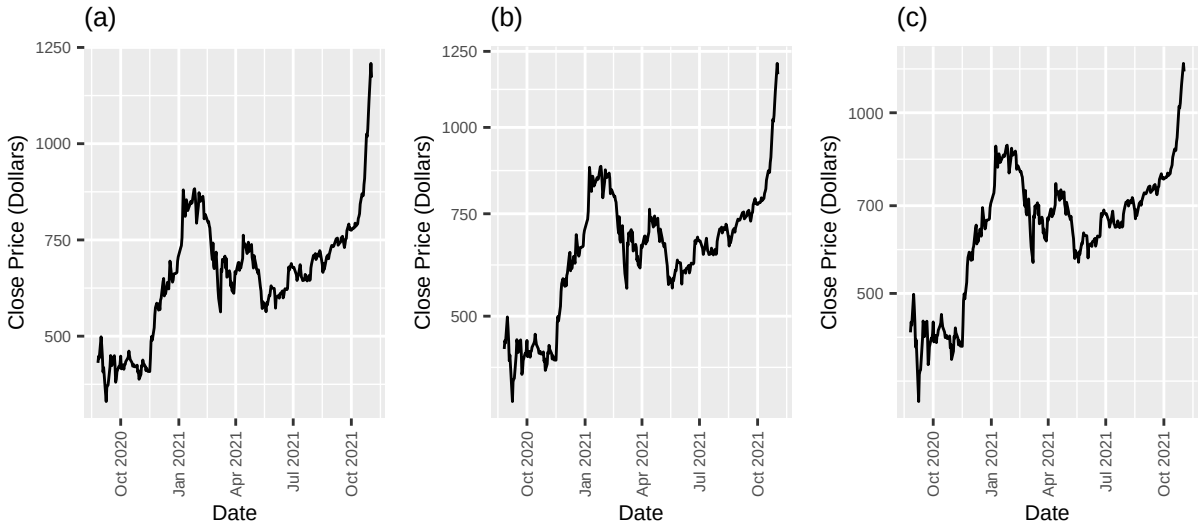


Figure 2: (a): Original time series. (b): Square root transfromed time seires. (c): Natural log transformed time series.

Finally, the trend increases again until the end of the time series. Nonetheless, we do not observe an obvious or significant seasonality pattern. It matches the intuition since the granularity of our data is day, and the structure of stock price data is too complicated to have a seasonality pattern.

In conclusion, based on all the previous discussions in EDA, we decide to construct possible models on the original time series data, including only the last three hundred time points.

5 Model Construction

With a comprehensive understanding of our data, we start to experiment and construct different time series model. We choose and build two non-parametric signal models of the trend and seasonality in our data. We aim to make the residuals approximately weekly stationary. We do not consider any parametric trend model because we think the trend of the stock price data is too complicated to be modeled by a parametric model, such as a high-order polynomial. Certainly, we could use a 15 or 20 order polynomial, but it may overfit the training data and produce imprecise predictions. We do not consider a parametric seasonality model either because we do not find a clear seasonality pattern in our data by the EDA. Finally, based on each signal model, we provide two ARMA models or its extension, such as SARMA or ARIMA, to whiten the residuals of the signal model. Thus, we have four candidate models, and we will explain how we select a final model among them in the next section.

5.1 Non-parametric Signal Model: exponential smoothing

In this signal model, we choose exponential smoothing with weight $\alpha = 0.8$ and lag $k = 10$ and a seasonal differencing with period $d = 5$.

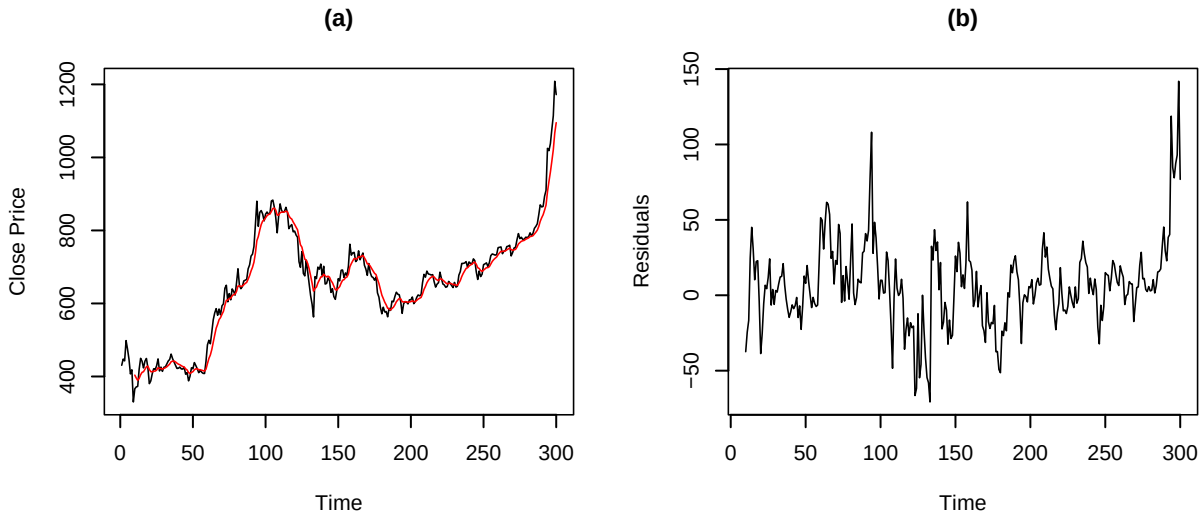


Figure 3: (a): Time series plot of the original data and fitted values. (b): The residual plot of exponential smoothing.

We experiment with different combinations of α and k with a careful consideration of overfitting issue. we choose $k = 10$ as the final value because we want to only use past two weeks, which are ten days in our data, to forecast. We choose $\alpha = 0.8$ as the final value because we think it best balances the smoothing effect and the capture of trend pattern among $(0, 1)$. Indeed, the smoothing line in Figure 4(a) fits the data in the way

that we want. Note that we lose the first nine time points due to the computation process of the exponential smoothing.

However, the residual plot Figure 4(b) is fairly non-stationary, as it has cycling fluctuation pattern and still slightly nonlinear trend. It might be due to that we intentionally let exponential smoothing not fit the data perfectly. Next, We use the seasonal differencing with period $d = 5$, which is one week in our data, to further make the residuals more stationary.

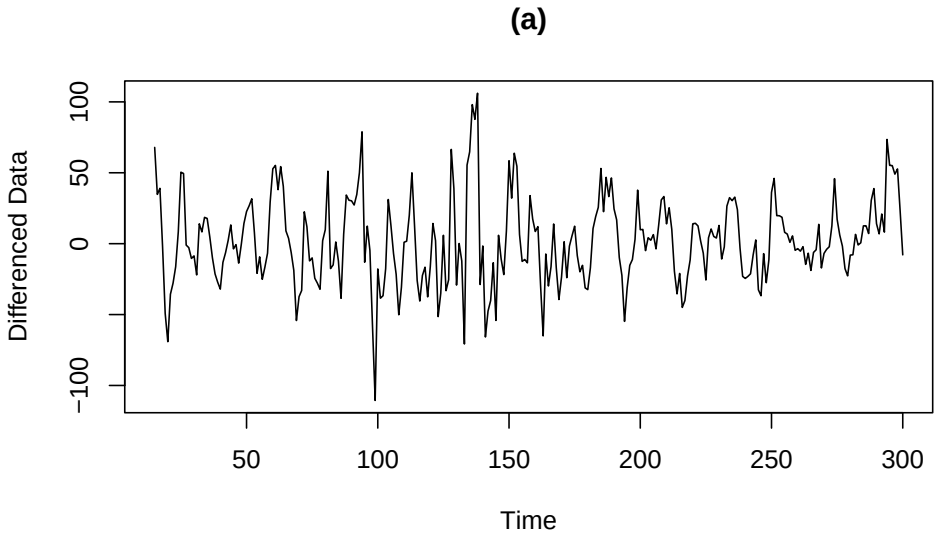


Figure 4: (a): Time series plot of the seasonal differenced (d = 5) residuals from the previous smoothing.

Indeed, now the differenced residuals become more stationary. There seems to be a contradiction that recalling in EDA, we claim that there is not a clear seasonality in our data. However, the effect of the seasonal differencing here implies a possible seasonality with period $d = 5$. We think it is because the seasonal differencing is actually removing the remaining trend left by the exponential smoothing instead of the seasonality. Nevertheless, We believe that the time series of the differenced residuals shown in Figure 5(a) is stationary enough for us to build ARMA models on it.

5.2 Non-parametric Signal Model: second-order differencing

In this model, we choose the second-order differencing to remove the trend. We observe that after the first-order differencing, there is still some trend pattern, such as the increasing one between 270 and 300, as shown by Figure TODO. This matches our previous analysis that the trend of our data is nonlinear in EDA. Thus, we take another differencing and acquire the second-order differencing data shown in Figure TODO.

The second-order differenced time series is more stationary than the first-order differenced time series. We can keep trying more higher-order differencings, but they may overfit our data. Therefore, we think the second-order differenced time series is already stationary enough for us to build ARMA model on it.

```
##       [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10] [,11] [,12]
## ACF  -0.52  0.00  0.03  0.02 -0.06  0.07 -0.06  0.00  0.07 -0.01 -0.07  0.05
## PACF -0.52 -0.37 -0.25 -0.14 -0.17 -0.07 -0.11 -0.15 -0.05  0.02 -0.05 -0.04
##      [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## ACF   0.01 -0.07  0.04  0.04 -0.07  0.10 -0.11 -0.04  0.14 -0.07 -0.11  0.18
## PACF  0.00 -0.08 -0.09  0.00 -0.04  0.09 -0.01 -0.15  0.00  0.01 -0.15  0.00
```

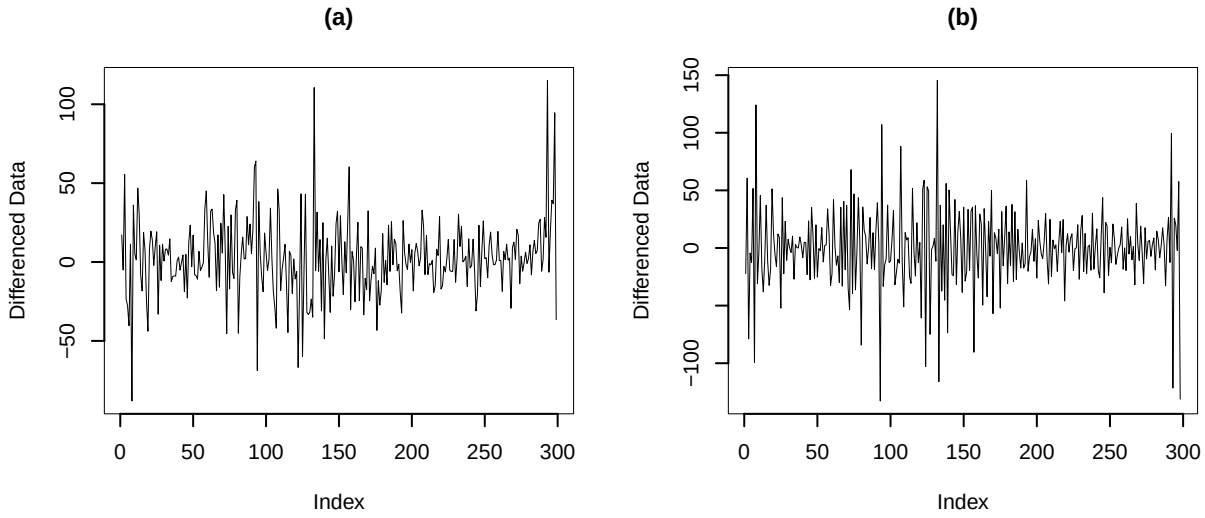


Figure 5: (a): The first-order differenced data. (b): The second-order differenced data.

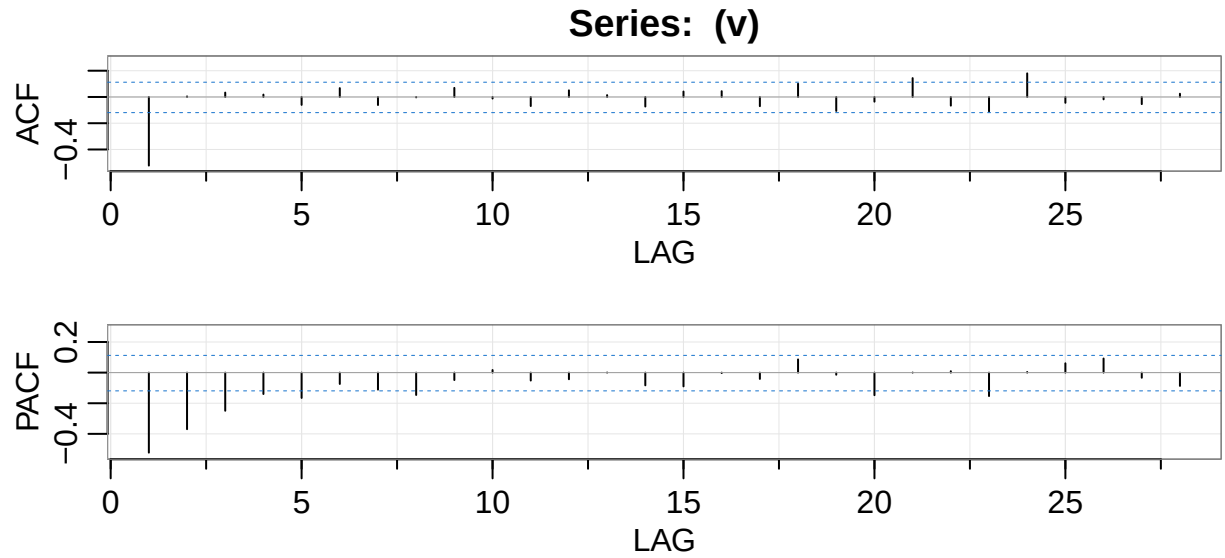


Figure 6: (a): The first-order differenced data. (b): The second-order differenced data.

```
##      [,25] [,26] [,27] [,28]
## ACF  -0.04 -0.02 -0.05  0.02
## PACF  0.06  0.09 -0.03 -0.09
```

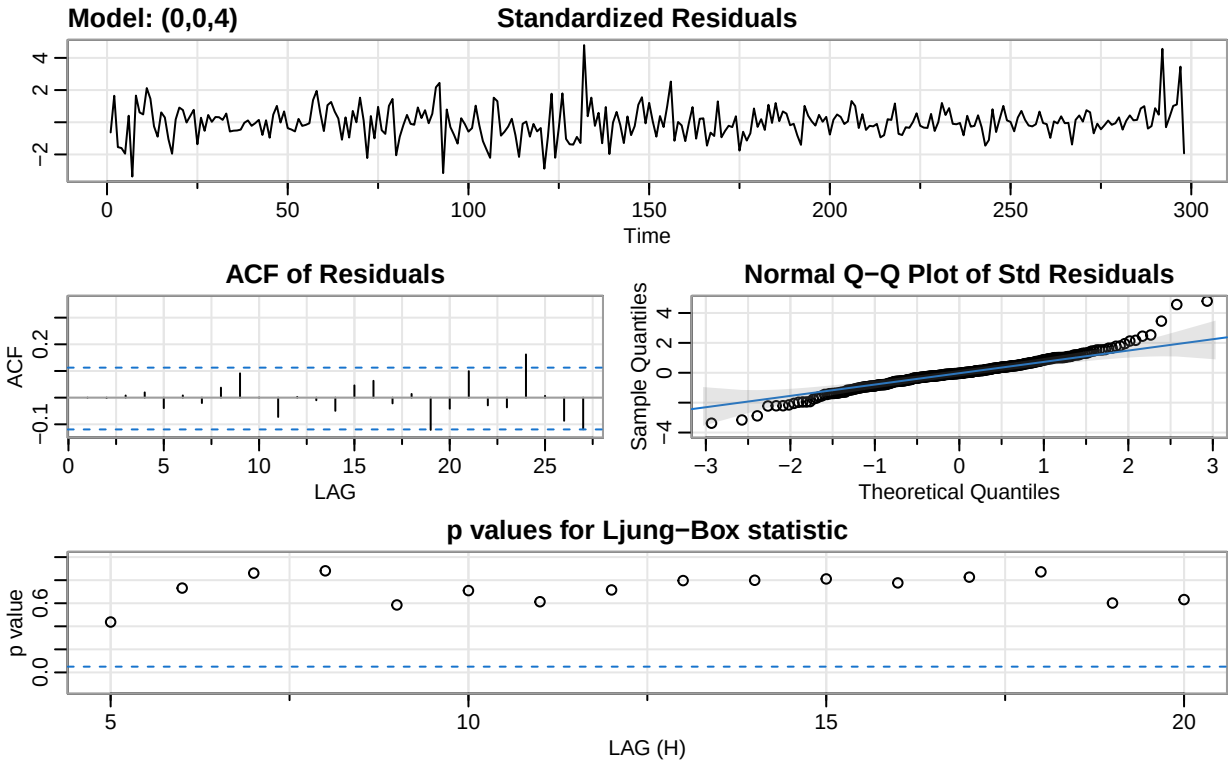
On the PACF graph of second order differencing series v, the absolute value of PACF is decreasing significantly from lag 1 to lag 5. On the ACF graph of v, the evident cutoff of absolute value of ACF occurs at lag 1. Therefore, it is reasonable to approximate the series v with MA model. However, the ACF and PACF plot are not strictly fitted on the theoretical $MA(q), q \in \{4, 5, 6\}$ model, so conduct parameter tuning experiment on q and possibility of $AR(p), p \in \{0, 1\}$.

With consideration of cross validation error, model AIC, model AICc, model BIC, the MA(4) model is the best one to approximate the second order differencing series v. For MA(4) model, the p values for Ljung-Box statistic are very large for all lag h, so MA(4) model passes the test. Besides MA(4) model, the MA(5) and ARMA(1,4) model serve as the second candidate group. In the perspective of cross validation error, ARMA(1,4) performs better than MA(5), while the thing is opposite for the aspect of AIC, AICc, and BIC statistics. Since discrepancy of the three model's performance statistics is not very large, all the three models can be explored in the further parameter tuning experiment.

```
##       Model Crossvalidation.error      AIC     AICc      BIC
## 1     MA(4)              22.55514 9.243639 9.244328 9.318077
## 2     MA(5)              22.59256 9.250292 9.251260 9.337136
## 3     MA(6)              23.18037 9.254172 9.255468 9.353423
## 4 ARMA(1,4)              22.57618 9.250360 9.251328 9.337204
## 5 ARMA(1,5)              23.09689 9.255031 9.256327 9.354282
## 6 ARMA(1,6)              23.43214 9.261673 9.263345 9.373330
```

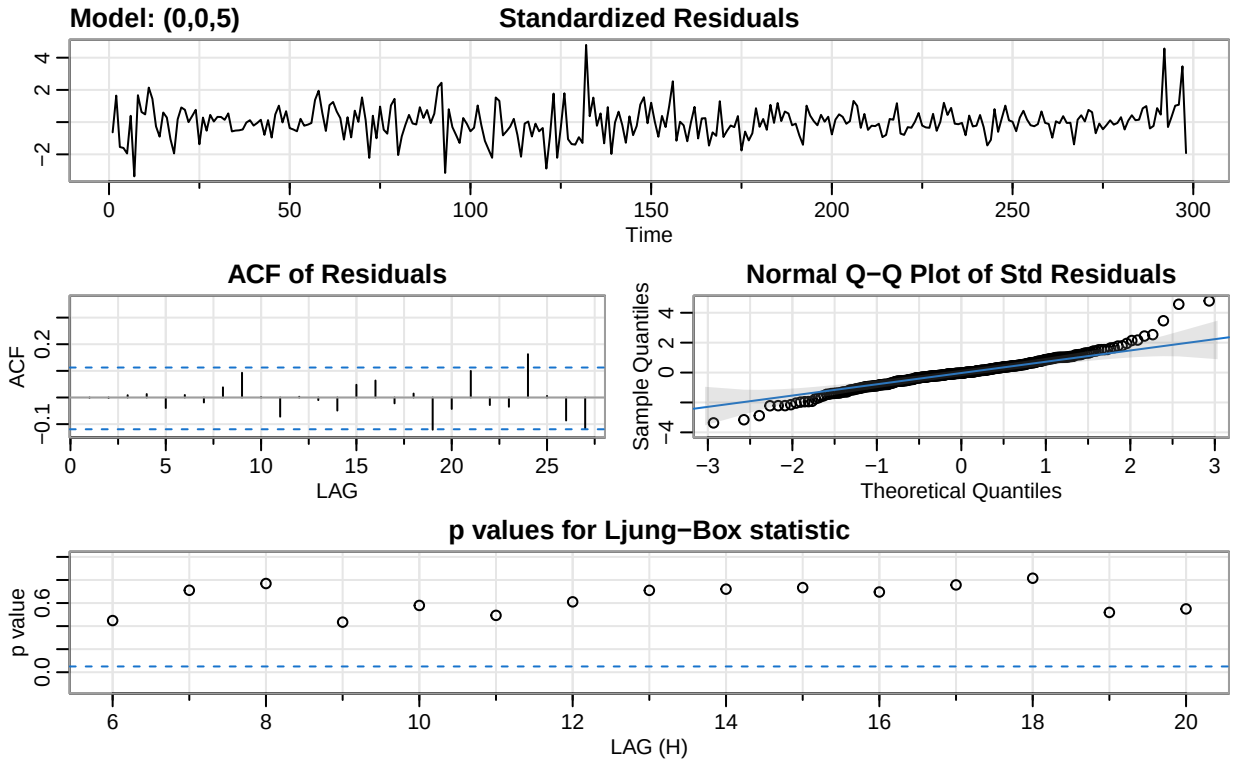
```
## initial  value 3.555728
## iter   2 value 3.327683
## iter   3 value 3.206791
## iter   4 value 3.206375
## iter   5 value 3.201768
## iter   6 value 3.194828
## iter   7 value 3.189862
## iter   8 value 3.188685
## iter   9 value 3.188558
## iter  10 value 3.188507
## iter  11 value 3.188492
## iter  12 value 3.188491
## iter  13 value 3.188491
## iter  14 value 3.188491
## iter  14 value 3.188491
## iter  14 value 3.188491
## final  value 3.188491
## converged
## initial  value 3.184013
## iter   2 value 3.183633
## iter   3 value 3.182927
## iter   4 value 3.182857
## iter   5 value 3.182753
## iter   6 value 3.182750
## iter   7 value 3.182748
## iter   8 value 3.182747
## iter   8 value 3.182747
## iter   8 value 3.182747
```

```
## final  value 3.182747
## converged
```



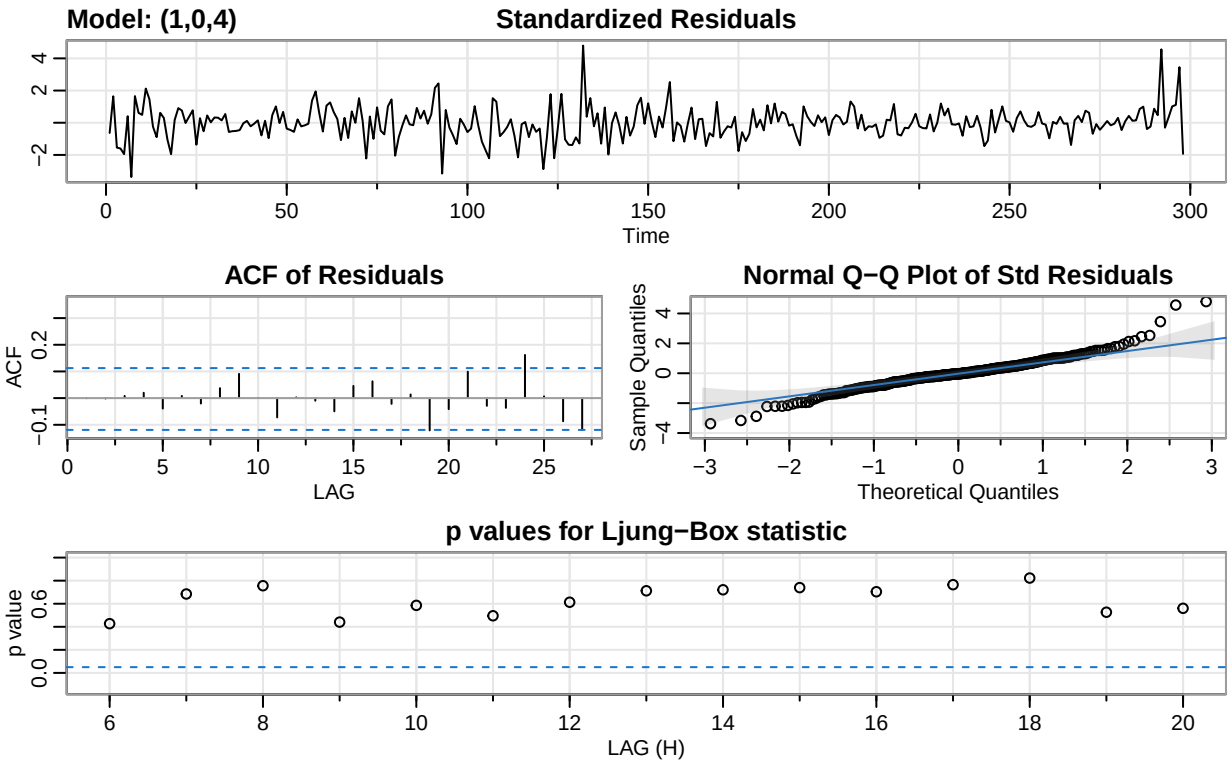

```
## initial  value 3.555728
## iter   2 value 3.324905
## iter   3 value 3.295634
## iter   4 value 3.243663
## iter   5 value 3.234430
## iter   6 value 3.225130
## iter   7 value 3.213137
## iter   8 value 3.198407
## iter   9 value 3.191747
## iter  10 value 3.189755
## iter  11 value 3.188779
## iter  12 value 3.188638
## iter  13 value 3.188468
## iter  14 value 3.188441
## iter  15 value 3.188440
## iter  16 value 3.188440
## iter  17 value 3.188437
## iter  18 value 3.188436
## iter  19 value 3.188435
## iter  20 value 3.188435
## iter  20 value 3.188435
## iter  20 value 3.188435
## final  value 3.188435
## converged
## initial  value 3.184028
```

```
## iter   2 value 3.183629
## iter   3 value 3.182938
## iter   4 value 3.182788
## iter   5 value 3.182723
## iter   6 value 3.182720
## iter   7 value 3.182718
## iter   8 value 3.182717
## iter   9 value 3.182717
## iter   9 value 3.182717
## iter   9 value 3.182717
## final  value 3.182717
## converged
```




```
## initial  value 3.556746
## iter   2 value 3.300946
## iter   3 value 3.255751
## iter   4 value 3.210799
## iter   5 value 3.208587
## iter   6 value 3.203416
## iter   7 value 3.196682
## iter   8 value 3.195570
## iter   9 value 3.181688
## iter  10 value 3.180859
## iter  11 value 3.180411
## iter  12 value 3.180041
## iter  13 value 3.179800
## iter  14 value 3.179658
## iter  15 value 3.179637
```

```
## iter  16 value 3.178929
## iter  17 value 3.178752
## iter  18 value 3.178719
## iter  19 value 3.178713
## iter  20 value 3.178712
## iter  21 value 3.178712
## iter  21 value 3.178712
## iter  21 value 3.178712
## final  value 3.178712
## converged
## initial  value 3.183113
## iter   2 value 3.183092
## iter   3 value 3.183012
## iter   4 value 3.182975
## iter   5 value 3.182971
## iter   6 value 3.182970
## iter   7 value 3.182970
## iter   8 value 3.182969
## iter   9 value 3.182967
## iter  10 value 3.182962
## iter  11 value 3.182950
## iter  12 value 3.182925
## iter  13 value 3.182885
## iter  14 value 3.182798
## iter  15 value 3.182796
## iter  16 value 3.182787
## iter  17 value 3.182766
## iter  18 value 3.182752
## iter  19 value 3.182751
## iter  19 value 3.182751
## iter  19 value 3.182751
## final  value 3.182751
## converged
```



6 Model Comparision and Selection

7 Final Model

7.1 Model interpretation

7.2 Prediction

8 Conclusion