W271 HW8

Subhashini R., Lei Yang, Ron Cordell

March 29, 2016

Build an univariate linear time series model (i.e AR, MA, and ARMA models) using the series in hw08_series.csv.

- Use all the techniques that have been taught so far to build the model, including date examination, data visualization, etc.
- All the steps to support your final model need to be shown clearly.
- Show that the assumptions underlying the model are valid.
- Which model seems most reasonable in terms of satisfying the model's underling assumption?
- Evaluate the model performance (both in- and out-of-sample)
- Pick your "best" models and conduct a 12-step ahead forecast. Discuss your results. Discuss the choice of your metrics to measure "best".

```
# Load the libraries and tools
library(astsa)
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(forecast)
```

```
## Loading required package: timeDate
```

```
## This is forecast 6.2
```

```
##
## Attaching package: 'forecast'
```

```
## The following object is masked from 'package:astsa':
##
##     gas
```

```
library(stargazer)
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2015). stargazer: Well-Formatted Regression and Summary Statistics Tables.
```

```
##  R package version 5.2. http://CRAN.R-project.org/package=stargazer
```

```
# load the CSV file
df <- read.csv('hw08_series.csv')
str(df)
```

```
## 'data.frame':    372 obs. of  2 variables:
##  $ X: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ x: num  40.6 41.1 40.5 40.1 40.4 41.2 39.3 41.6 42.3 43.2 ...
```

The CSV file for the HW8 time series consists of two variables: an X variable that is the time interval and an x value corresponding to the time period. There is no information about the time interval or units of the values.

A time series object is created from the dataframe for further analysis.

```
ts1 <- ts(df$x)
str(ts1)
```

```
##  Time-Series [1:372] from 1 to 372: 40.6 41.1 40.5 40.1 40.4 41.2 39.3 41.6 42.3 43.2 ...
```

```
summary(ts1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   36.00   57.38   76.45   84.83  111.50  152.60
```

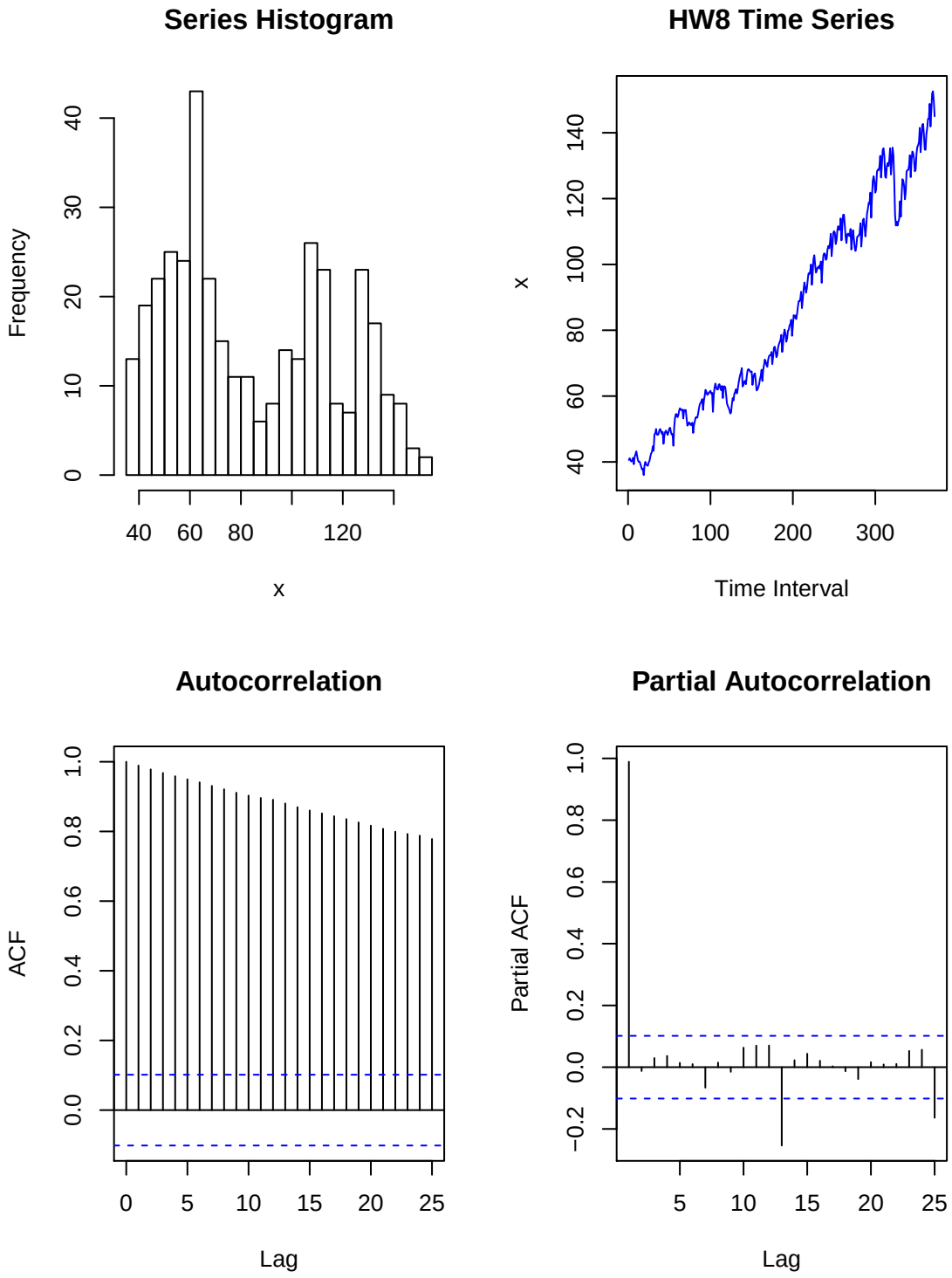
```
head(ts1)
```

```
## [1] 40.6 41.1 40.5 40.1 40.4 41.2
```

```
tail(ts1)
```

```
## [1] 141.9 146.9 152.0 152.6 149.7 145.0
```

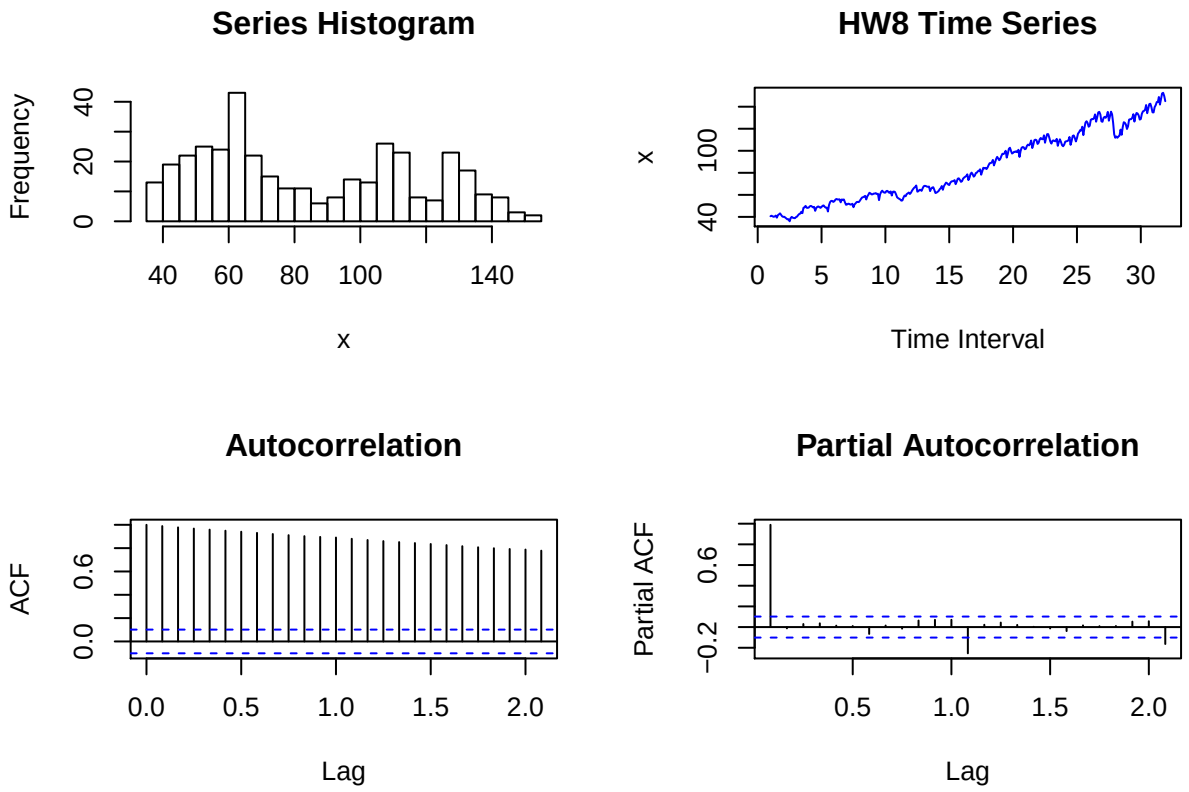
```r
par(mfrow=c(2,2))
hist(ts1, main='Series Histogram', xlab='x', breaks=20)
plot.ts(ts1, col='blue',
        xlab='Time Interval',
        ylab='x',
        main='HW8 Time Series')
acf(ts1, main='Autocorrelation')
pacf(ts1, main='Partial Autocorrelation')
```



The time series plot reveals that the HW8 time series is not stationary with a persistently upward trend. The time series also seems to exhibit some seasonality component. The series appears very much like a random walk with drift. The autocorrelation shows a very long decay over more than 25 lags that corresponds to the trend while the partial autocorrelation shows statistically significant results at lags 13 and 25.

We will re-import the time series now that we know there is a 12 period cycle and indicate this in the time series conversion. Then we replot the time series.

```
ts1 <- ts(df$x, frequency = 12)
par(mfrow=c(2,2))
hist(ts1, main='Series Histogram', xlab='x', breaks=20)
plot.ts(ts1, col='blue',
        xlab='Time Interval',
        ylab='x',
        main='HW8 Time Series')
acf(ts1, main='Autocorrelation')
pacf(ts1, main='Partial Autocorrelation')
```

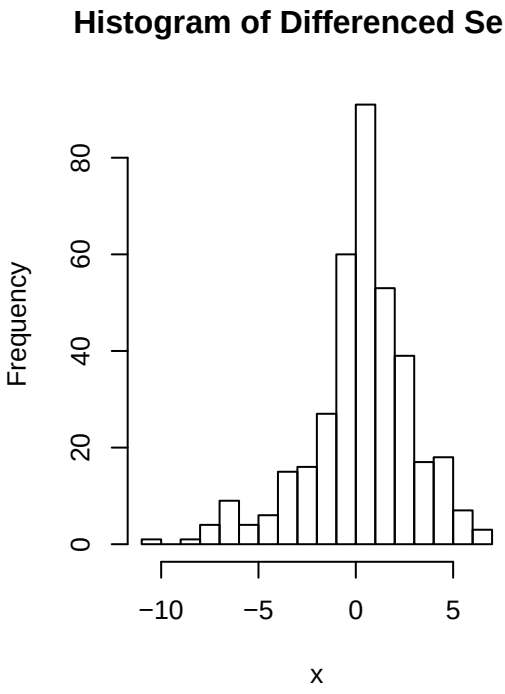


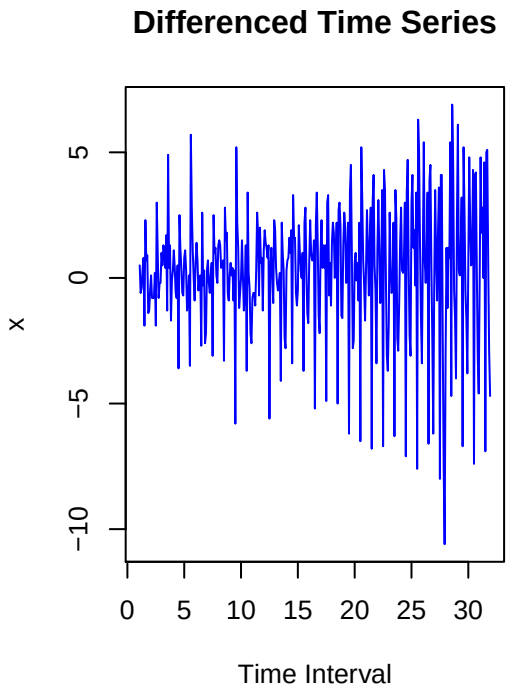
We will attempt to remove the trend by taking the first difference and then examine the resulting series.

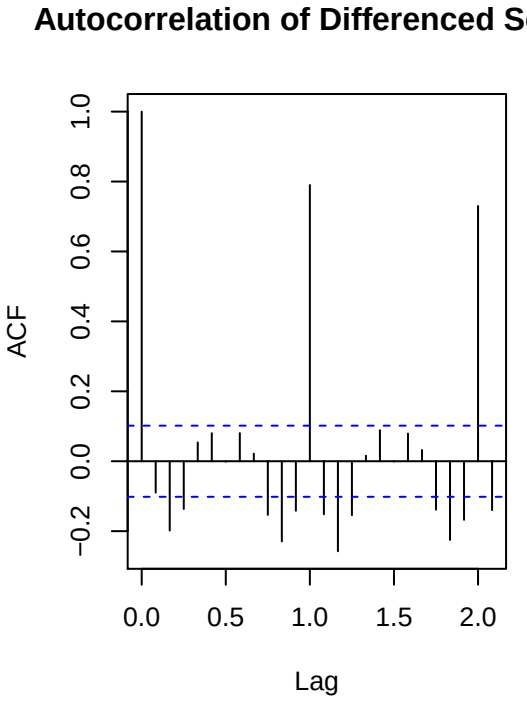
```
ts1.diff <- diff(ts1)
summary(ts1.diff)
```

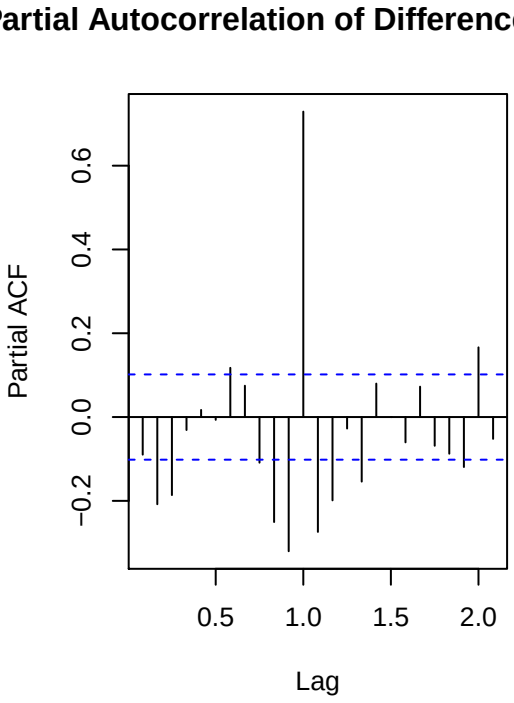
```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -10.6000  -0.8000   0.5000   0.2814   1.8000   6.9000
```

```
par(mfrow=c(2,2))
hist(ts1.diff, main='Histogram of Differenced Series', xlab='x', breaks=20)
plot.ts(ts1.diff, col='blue',
        xlab='Time Interval',
        ylab='x',
        main='Differenced Time Series')
acf(ts1.diff, main='Autocorrelation of Differenced Series')
pacf(ts1.diff, main='Partial Autocorrelation of Differenced Series')
```









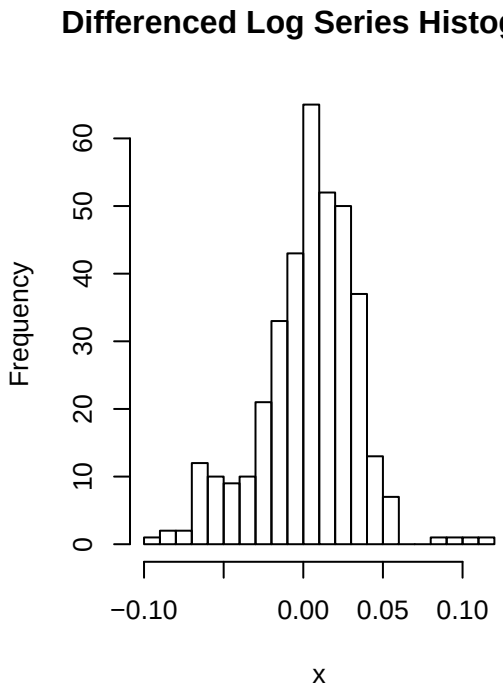
The resulting difference series shows an increasing seasonality and the autocorrelation and partial autocorrelation also show the seasonality. The seasonality appears to be a cycle of 12 periods with very strong peaks at periods 12 and 24. This may indicate a sales trend where months 12 and 24 are the winter holiday sales - this is just speculation, however.

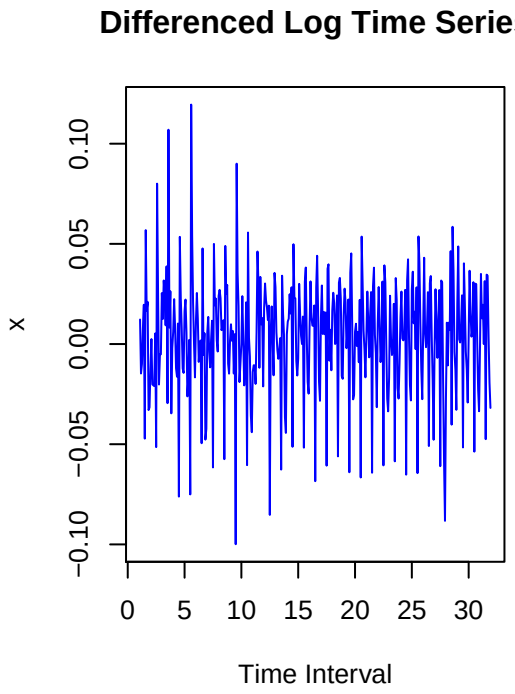
Let's take a look at the difference in log next.

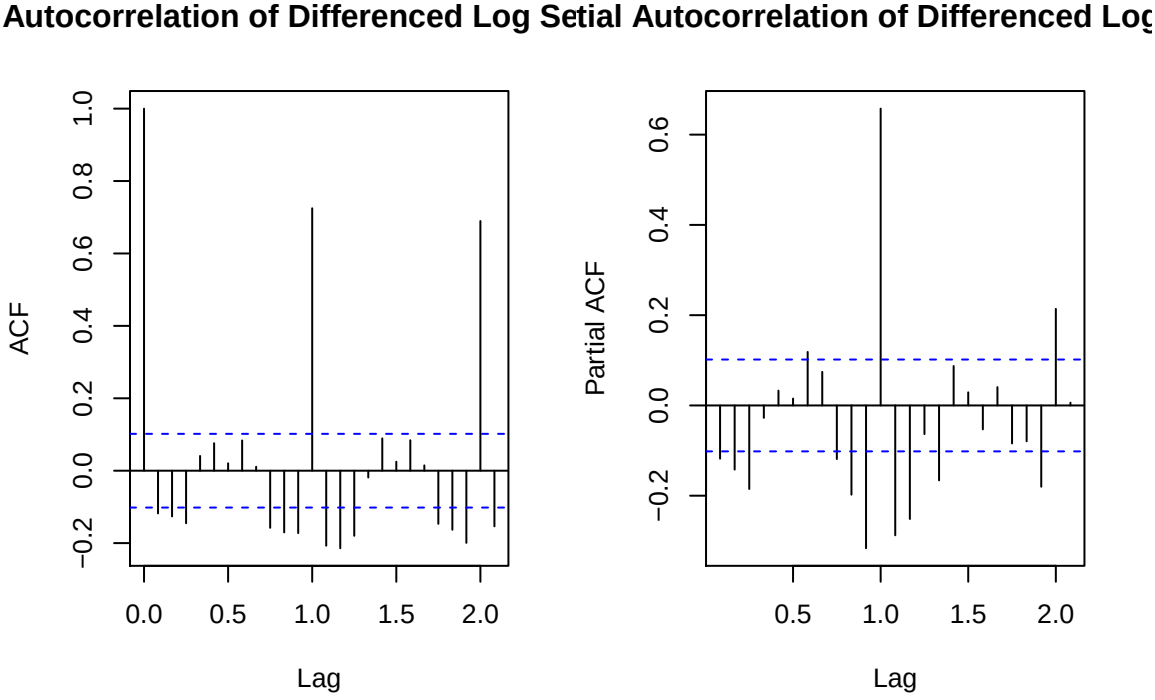
```r
ts1.diff_log <- diff(log(ts1))
summary(ts1.diff_log)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.099910 -0.012310  0.006413  0.003431  0.023520  0.119500
```

```r
par(mfrow=c(2,2))
hist(ts1.diff_log, main='Differenced Log Series Histogram', xlab='x', breaks=20)
plot.ts(ts1.diff_log, col='blue',
        xlab='Time Interval',
        ylab='x',
        main='Differenced Log Time Series')
acf(ts1.diff_log, main='Autocorrelation of Differenced Log Series')
pacf(ts1.diff_log, main='Partial Autocorrelation of Differenced Log Series')
```







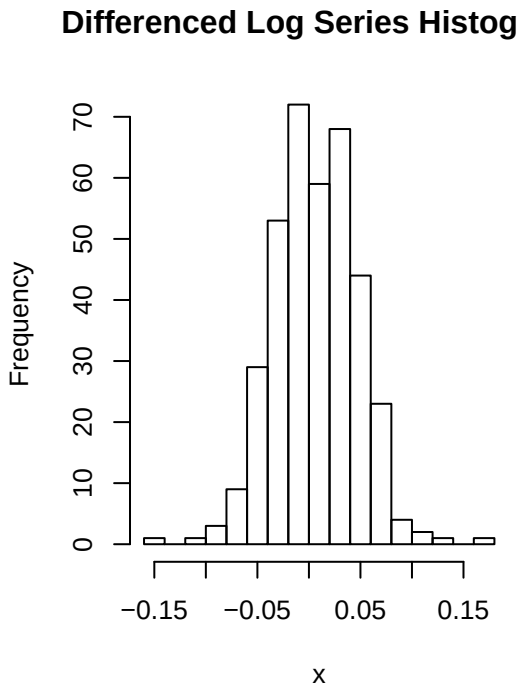
With the differenced log-transformed series the volatility of the seasonality is much more uniform and the seasonality persists.

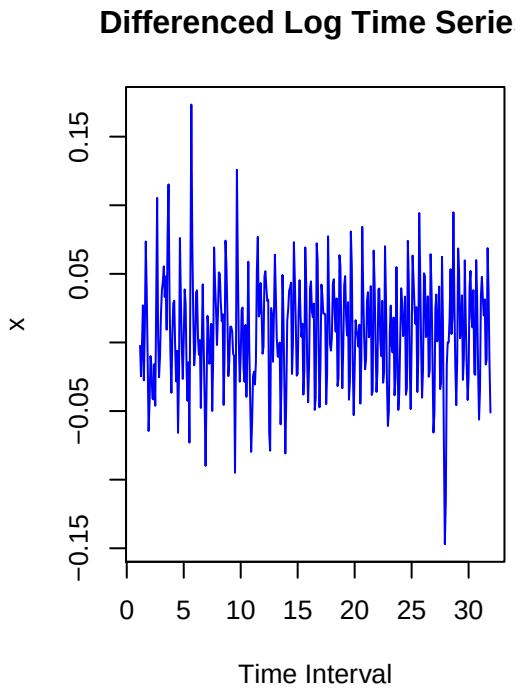
We take a second difference of the log-transformed series to see if the volatility is reduced in any way.

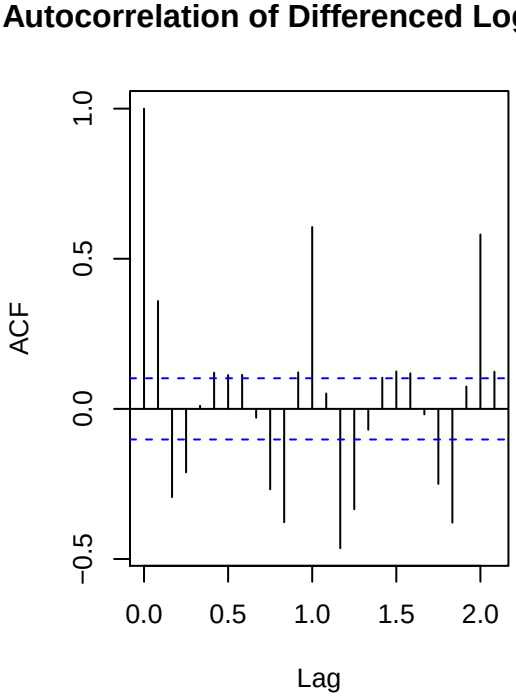
```
ts1.diff2_log <- diff(log(ts1), lag=2)
summary(ts1.diff2_log)
```

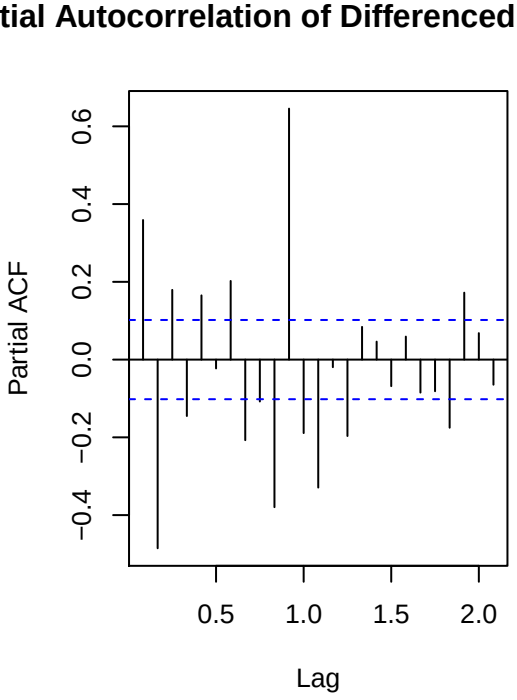
```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.147000 -0.020560  0.005120  0.006934  0.034540  0.173400
```

```r
par(mfrow=c(2,2))
hist(ts1.diff2_log, main='Differenced Log Series Histogram', xlab='x', breaks=20)
plot.ts(ts1.diff2_log, col='blue',
        xlab='Time Interval',
        ylab='x',
        main='Differenced Log Time Series')
acf(ts1.diff2_log, main='Autocorrelation of Differenced Log Series')
pacf(ts1.diff2_log, main='Partial Autocorrelation of Differenced Log Series')
```









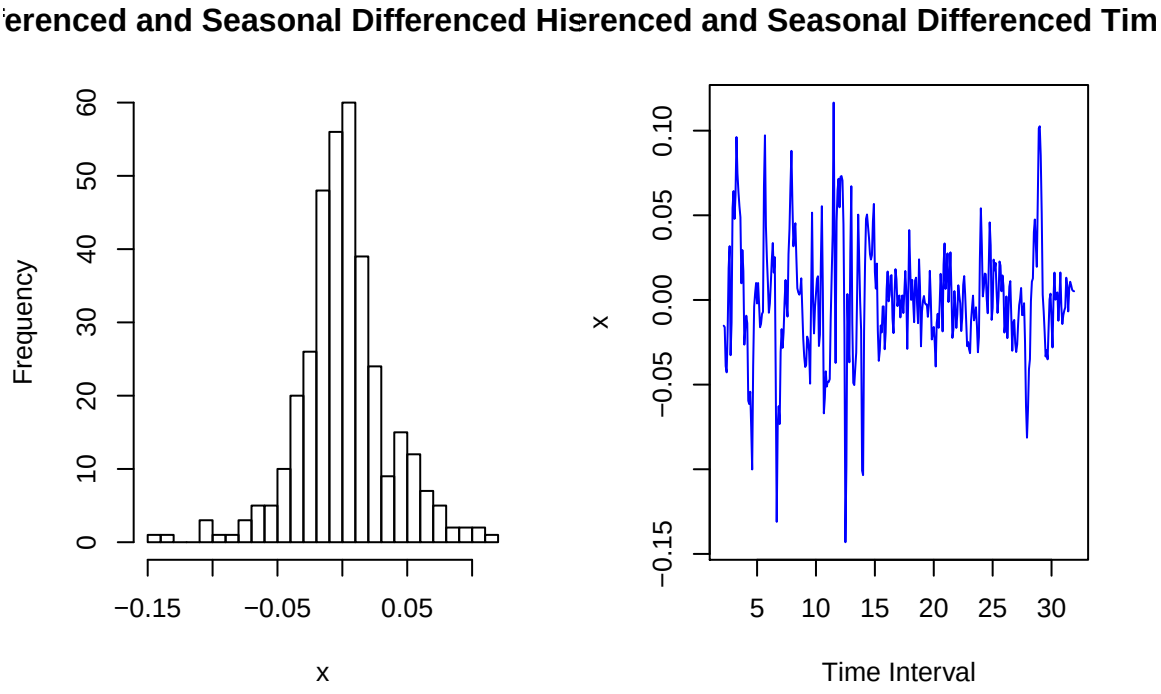
The series still shows strong seasonality, but the volatility is much more uniform.

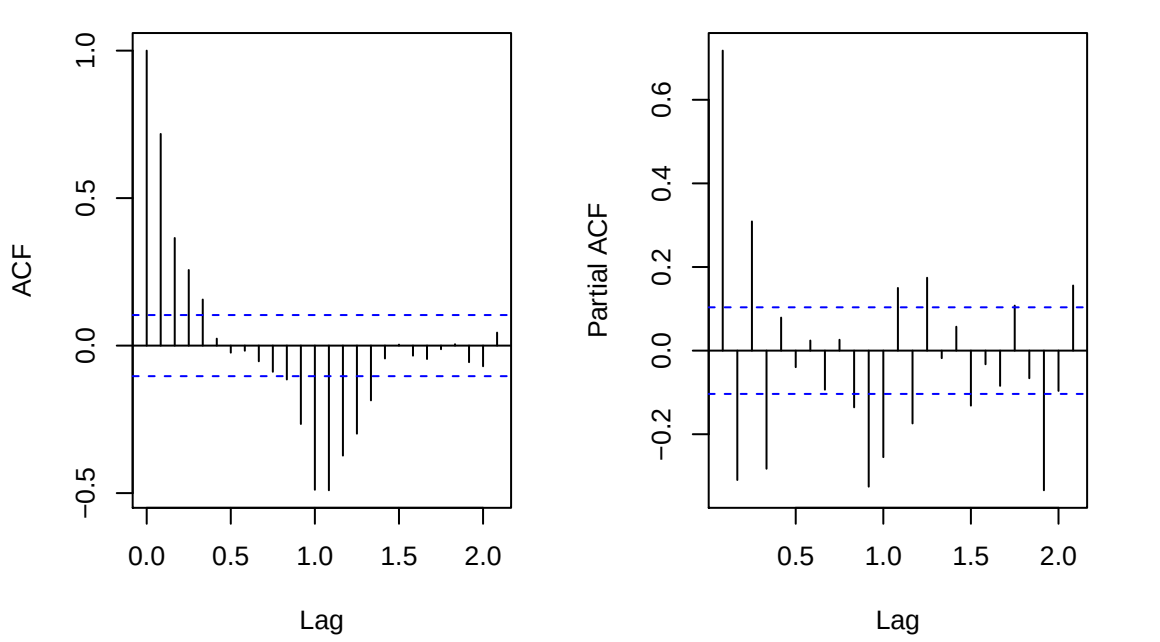
Now we will perform a seasonally differenced model on the log-transformed, differenced time series to remove the 12 period seasonality and examine the results.

```
ts1.diff2_s <- diff(ts1.diff2_log, lag = 12)
summary(ts1.diff2_s)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.1431000 -0.0178100 -0.0002484  0.0005092  0.0164900  0.1166000
```

```r
par(mfrow=c(2,2))
hist(ts1.diff2_s, main='Differenced and Seasonal Differenced Histogram', xlab='x', breaks=20)
plot.ts(ts1.diff2_s, col='blue',
        xlab='Time Interval',
        ylab='x',
        main='Differenced and Seasonal Differenced Time Series')
acf(ts1.diff2_s, main='Autocorrelation of Differenced and Seasonal Differenced Series')
pacf(ts1.diff2_s, main='Partial Autocorrelation of Differenced and Seasonal Differenced Series')
```





The seasonally differenced first difference series is beginning to appear stationary. The autocorrelation shows some periodicity remaining, as does the PACF. However the autocorrelation indicates the possibility of a ARMA(1,2) while the seasonally differenced seems to indicate an MA(1).

```
ts1.fit1 <- Arima(ts1, order=c(1,1,1), seasonal = c(0,1,1))
ts1.fit2 <- Arima(log(ts1), order=c(1,1,1), seasonal = c(0,1,1))
ts1.fit3 <- Arima(ts1, order=c(1,1,2), seasonal = c(0,1,1))
summary(ts1.fit1)
```

```
## Series: ts1
## ARIMA(1,1,1)(0,1,1)[12]
##
## Coefficients:
##          ar1      ma1     sma1
##       0.5827  -0.2857  -0.6831
## s.e.  0.1160   0.1357   0.0331
##
## sigma^2 estimated as 1.428:  log likelihood=-577.15
## AIC=1162.3   AICc=1162.41   BIC=1177.83
##
## Training set error measures:
##                      ME     RMSE       MAE        MPE     MAPE      MASE
## Training set 0.02995918 1.173821 0.8392451 0.03719054 1.068714 0.1534503
##                      ACF1
## Training set -0.001614412
```

```
summary(ts1.fit2)
```

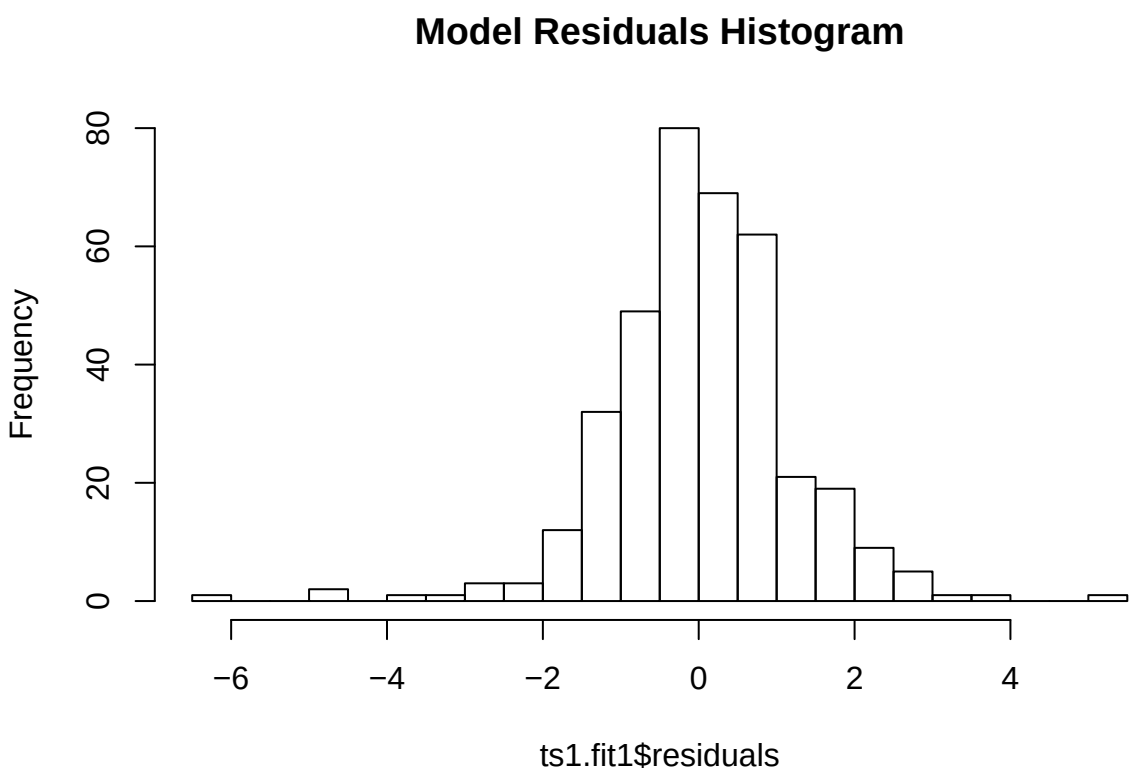
```
## Series: log(ts1)
## ARIMA(1,1,1)(0,1,1)[12]
##
## Coefficients:
##          ar1      ma1     sma1
##       0.5793  -0.3344  -0.8304
## s.e.  0.1215   0.1378   0.0309
##
## sigma^2 estimated as 0.0002317:  log likelihood=985.93
## AIC=-1963.85   AICc=-1963.74   BIC=-1948.32
##
## Training set error measures:
##                         ME       RMSE        MAE         MPE      MAPE
## Training set 0.0001734777 0.01497113 0.01009561 0.004426139 0.2373802
##                    MASE        ACF1
## Training set 0.1496079 -0.01061807
```

```
summary(ts1.fit3)
```

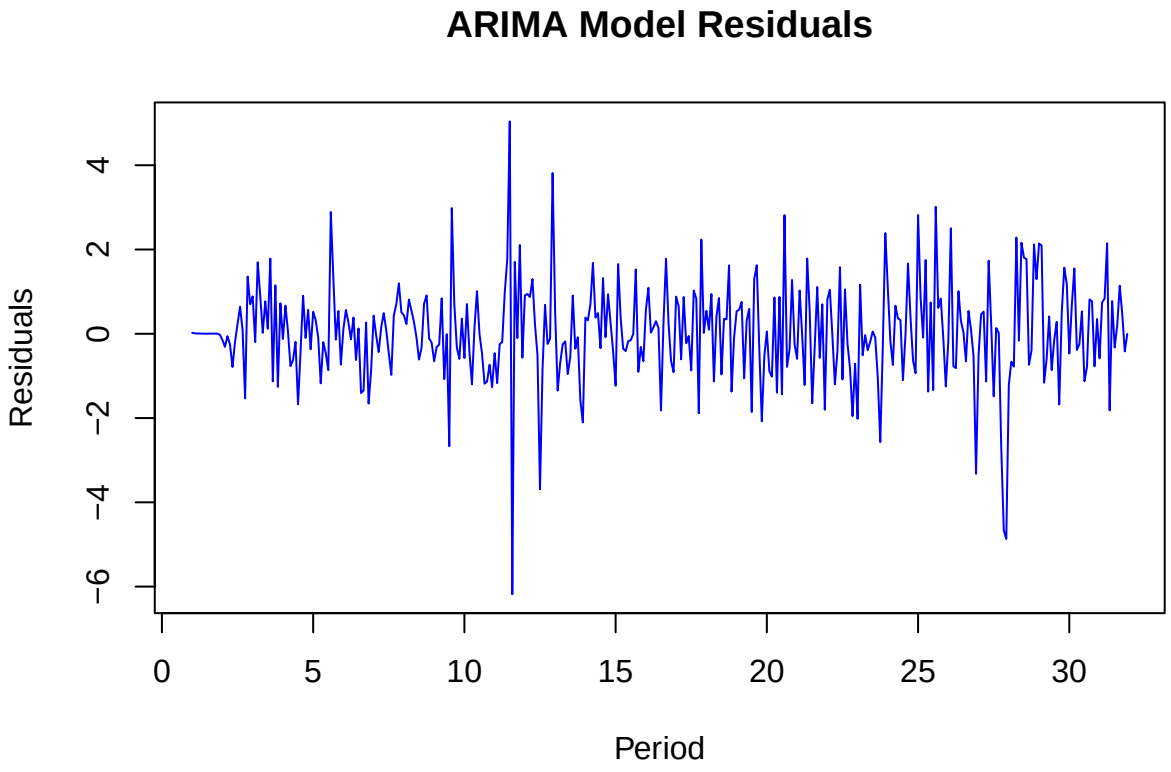
```
## Series: ts1
## ARIMA(1,1,2)(0,1,1)[12]
##
## Coefficients:
##          ar1      ma1     ma2     sma1
##       0.5619  -0.2665  0.0122  -0.6828
## s.e.  0.1782   0.1863  0.0741   0.0332
##
## sigma^2 estimated as 1.428:  log likelihood=-577.14
## AIC=1164.27   AICc=1164.44   BIC=1183.69
##
## Training set error measures:
##                      ME    RMSE       MAE        MPE     MAPE      MASE
## Training set 0.03009136 1.17379 0.8400327 0.03734825 1.069447 0.1535943
##                      ACF1
## Training set 4.321208e-05
```

The model with the lowest AIC is an ARIMA(1,1,1) (0,1,1)[12] model. Let's do some diagnostics on this model by assessing the residuals.

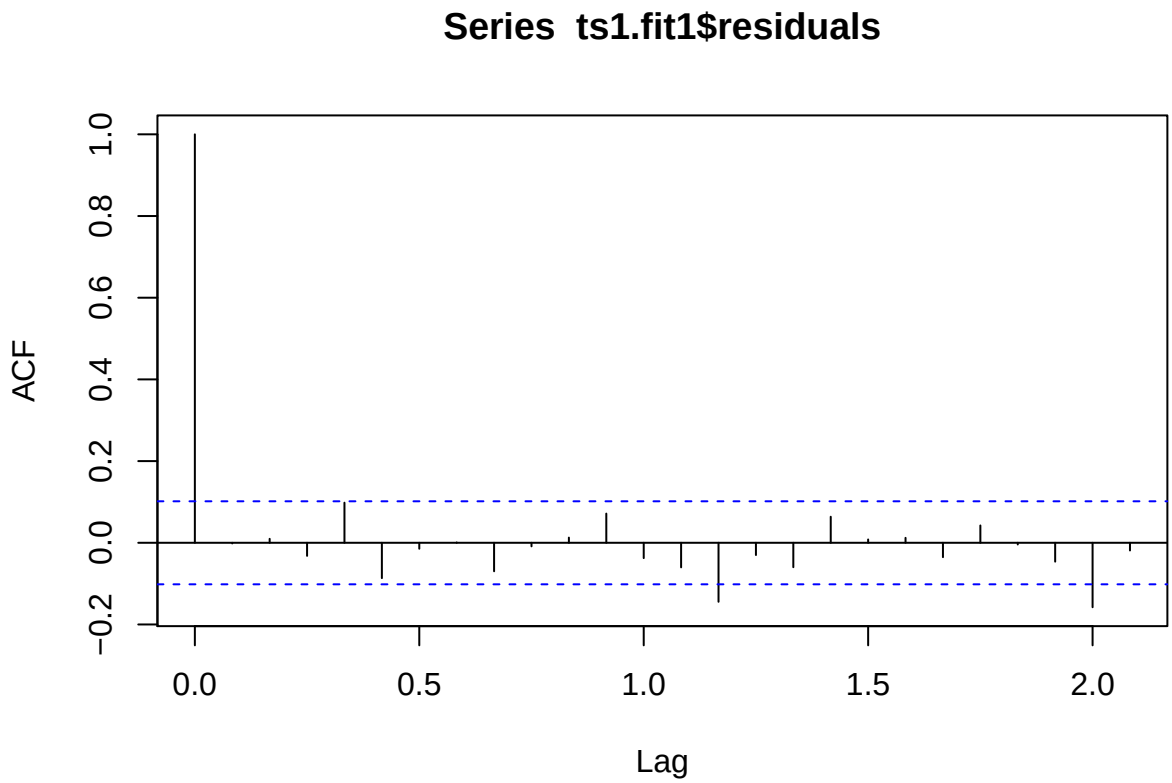
```
hist(ts1.fit1$residuals, main="Model Residuals Histogram", breaks=20)
```



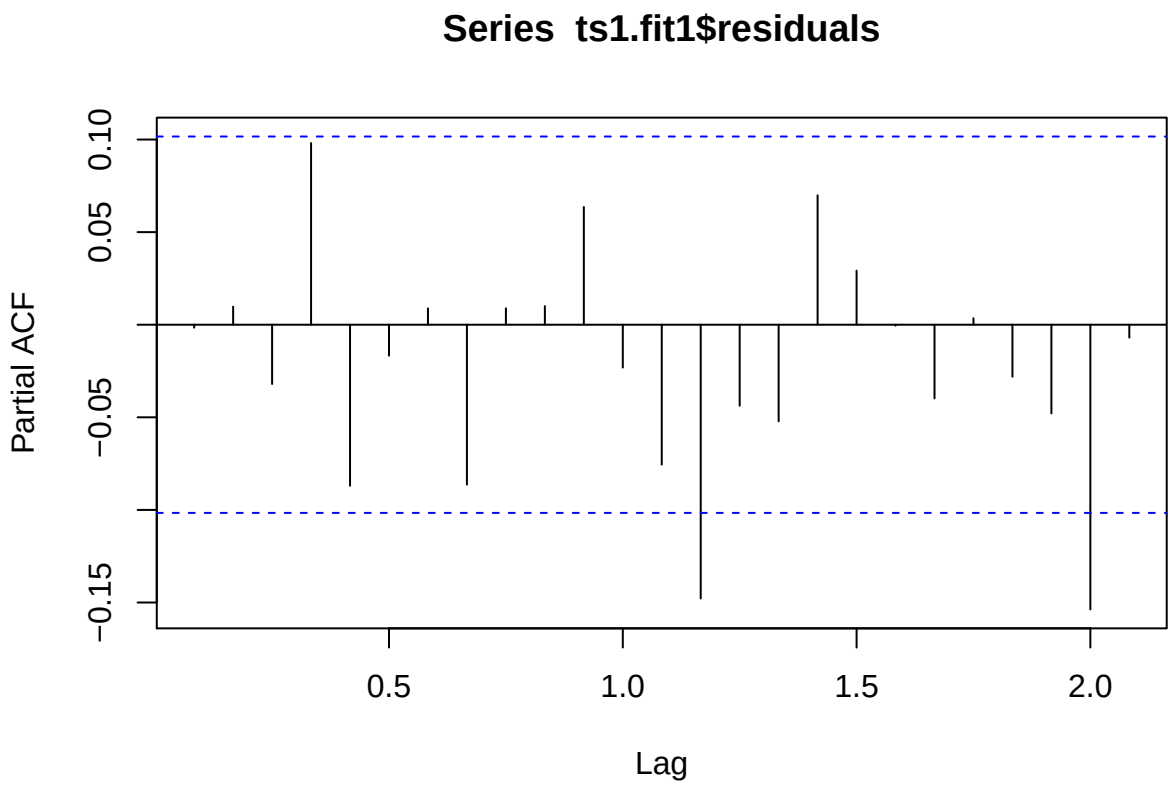

```
plot.ts(ts1.fit1$residuals, col='blue',
          xlab='Period',
          ylab='Residuals',
          main='ARIMA Model Residuals')
```




```
acf(ts1.fit1$residuals)
```



```
pacf(ts1.fit1$residuals)
```



Box-Ljung Test

```
Box.test(ts1.fit1$residuals, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  ts1.fit1$residuals
## X-squared = 0.00097739, df = 1, p-value = 0.9751
```

The null hypothesis of independence can not be rejected according to the Box-Ljung test.

The summary statistics of the model compared to the original series shows a very close correspondence in the mean, standard deviation and minimum/maximum values.

```
fit.df <- data.frame(cbind(ts1, fitted(ts1.fit1), ts1.fit1$residuals))
class(df)
```

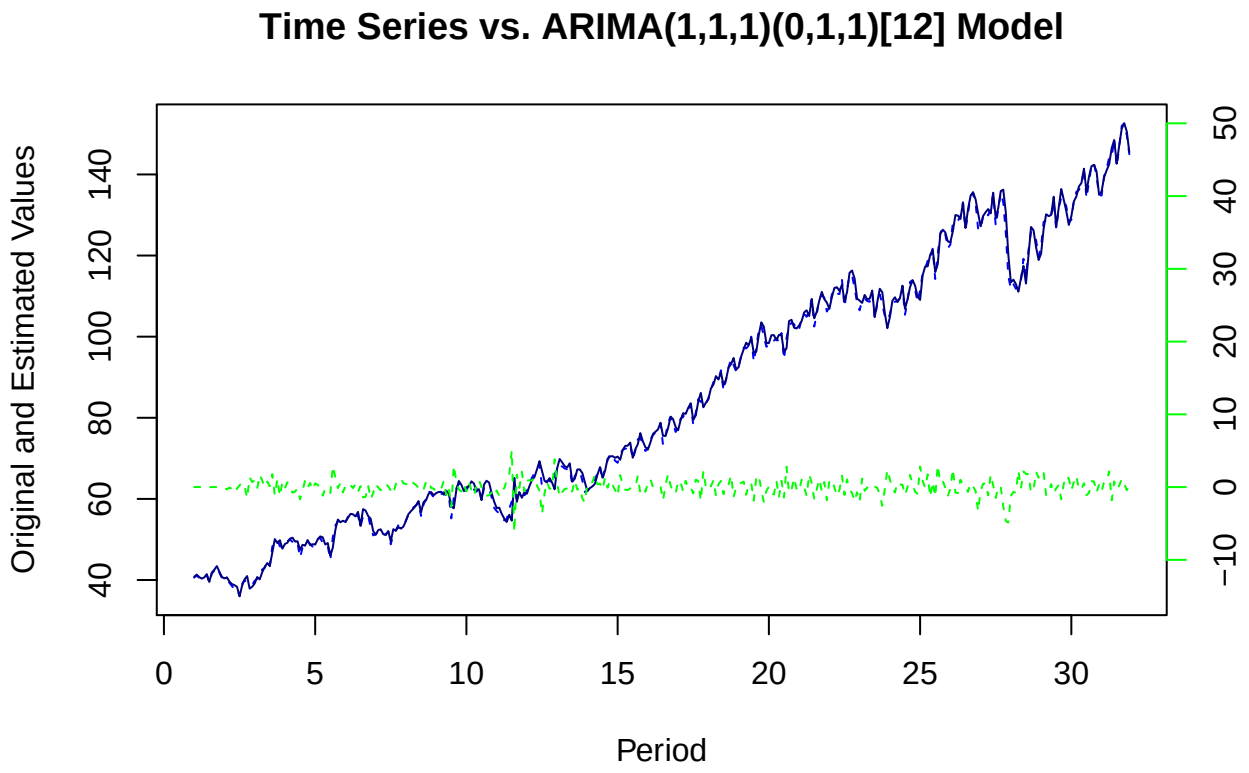
```
## [1] "data.frame"
```

```
stargazer(fit.df, type="text",title="Descriptive Statistics", digits=1)
```

```
##
## Descriptive Statistics
## ===============================================
```

```
## Statistic           N  Mean St. Dev. Min   Max
## ------------------------------------------------
## ts1                372 84.8   32.0   36.0 152.6
## fitted.ts1.fit1.   372 84.8   32.0   35.8 152.1
## ts1.fit1.residuals 372 0.03   1.2    -6.2  5.0
## ------------------------------------------------
```

```
par(mfrow=c(1,1))
plot.ts(ts1, col='blue',
        main='Time Series vs. ARIMA(1,1,1)(0,1,1)[12] Model',
        ylab='Original and Estimated Values', xlab='Period',
        pch=1, lty=2)
par(new=T)
plot.ts(fitted(ts1.fit1), col='navy', axes=F,
        xlab='',ylab='', lty=1)
leg.txt <-
par(new=T)
plot.ts(ts1.fit1$residuals, axes=F, xlab='', ylab='', col='green',
        lty=2, pch=1, col.axis='green', ylim=c(-15,50))
axis(side=4, col='green')
```



Forecast Model

```
ts1.fcast <- forecast.Arima(ts1.fit1, h=24)
ts1.fcast
```

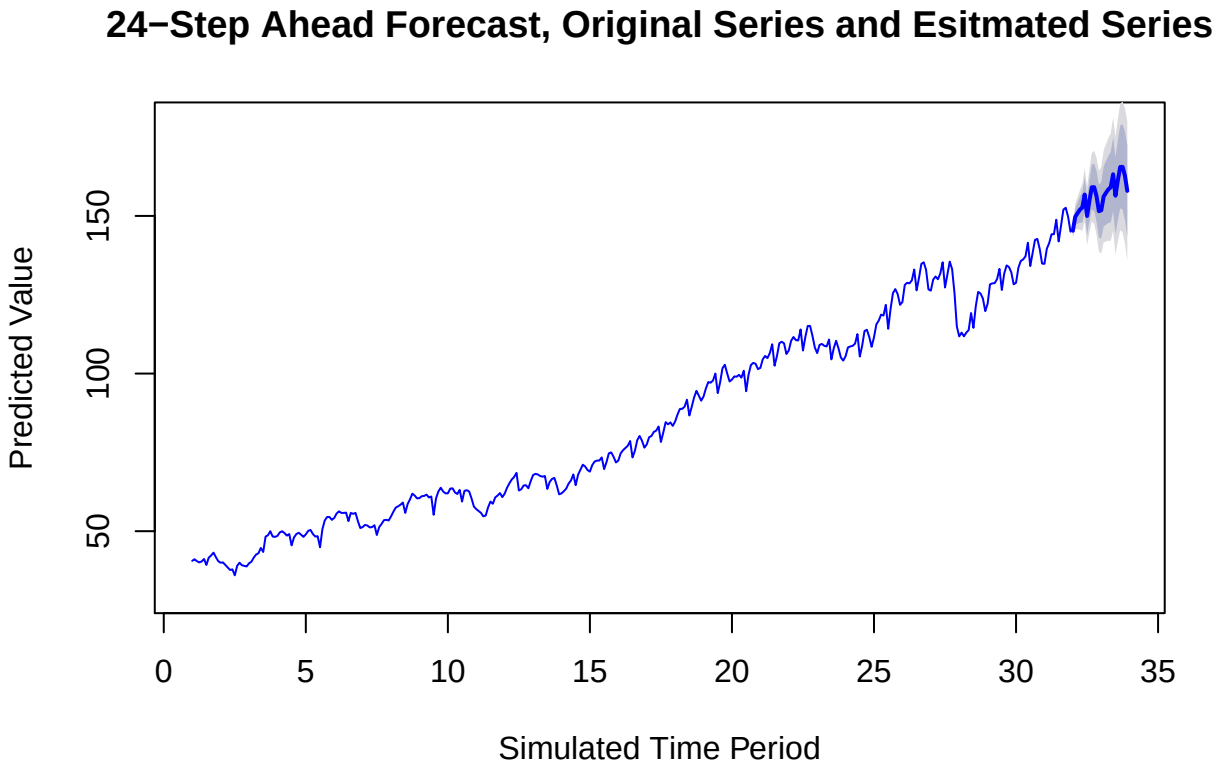
```
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 32       145.3095 143.7783 146.8408 142.9677 147.6514
## Feb 32       149.7276 147.2198 152.2354 145.8922 153.5630
```

```
## Mar 32       150.8772 147.5073 154.2471 145.7234 156.0311
## Apr 32       151.9939 147.8536 156.1343 145.6618 158.3260
## May 32       152.6748 147.8406 157.5090 145.2815 160.0680
## Jun 32       156.7868 151.3223 162.2513 148.4296 165.1441
## Jul 32       149.9763 143.9340 156.0185 140.7354 159.2171
## Aug 32       154.8314 148.2550 161.4078 144.7736 164.8891
## Sep 32       159.1054 152.0312 166.1795 148.2864 169.9243
## Oct 32       159.1258 151.5846 166.6669 147.5926 170.6590
## Nov 32       156.2655 148.2835 164.2476 144.0580 168.4730
## Dec 32       151.5096 143.1091 159.9101 138.6622 164.3570
## Jan 33       151.7858 142.8300 160.7416 138.0891 165.4825
## Feb 33       156.1845 146.6574 165.7115 141.6141 170.7548
## Mar 33       157.3228 147.2293 167.4163 141.8861 172.7594
## Apr 33       158.4329 147.7876 169.0782 142.1523 174.7135
## May 33       159.1099 147.9313 170.2885 142.0137 176.2061
## Jun 33       163.2197 151.5272 174.9122 145.3376 181.1018
## Jul 33       156.4078 144.2204 168.5953 137.7688 175.0469
## Aug 33       161.2622 148.5976 173.9268 141.8933 180.6310
## Sep 33       165.5357 152.4104 178.6610 145.4623 185.6092
## Oct 33       165.5559 151.9850 179.1267 144.8011 186.3107
## Nov 33       162.6955 148.6930 176.6980 141.2805 184.1105
## Dec 33       157.9395 143.5180 172.3609 135.8838 179.9951
```

```
summary(ts1.fcast$mean)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   145.3   151.9   156.6   156.4   159.1   165.6
```

```
par(mfrow=c(1,1))
plot(ts1.fcast,
     main='24-Step Ahead Forecast, Original Series and Esitmated Series',
     xlab='Simulated Time Period',
     ylab='Predicted Value',
     ylim=c(30,180), lty=1, col='blue')
```


24−Step Ahead Forecast, Original Series and Esitmated Series
Predicted Value
50
100
150
0
5
10
15
20
25
30
35
Simulated Time Period