174_finalproj

2023-05-17

(1)

Loading in Data.

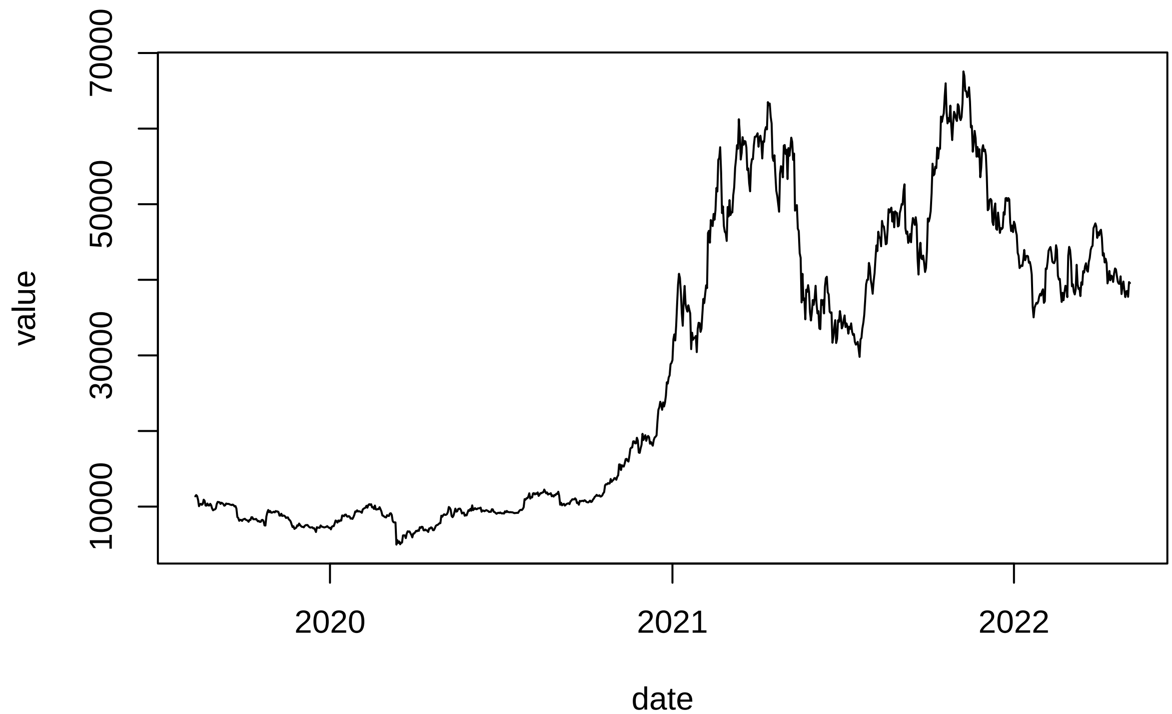
```
btc_data <- read.csv('data/BTC-USD.csv')
head(btc_data)
```

```
##         Date    Open    High     Low   Close Adj.Close   Volume
## 1 2014-09-17 465.864 468.174 452.422 457.334   457.334 21056800
## 2 2014-09-18 456.860 456.860 413.104 424.440   424.440 34483200
## 3 2014-09-19 424.103 427.835 384.532 394.796   394.796 37919700
## 4 2014-09-20 394.673 423.296 389.883 408.904   408.904 36863600
## 5 2014-09-21 408.085 412.426 393.181 398.821   398.821 26580100
## 6 2014-09-22 399.100 406.916 397.130 402.152   402.152 24127600
```

```
df1 = data.frame(date = btc_data$Date[1789:2788], adj_close = btc_data$Adj.Close[1789:2788])
head(df1)
```

```
##         date adj_close
## 1 2019-08-10  11354.02
## 2 2019-08-11  11523.58
## 3 2019-08-12  11382.62
## 4 2019-08-13  10895.83
## 5 2019-08-14  10051.70
## 6 2019-08-15  10311.55
```

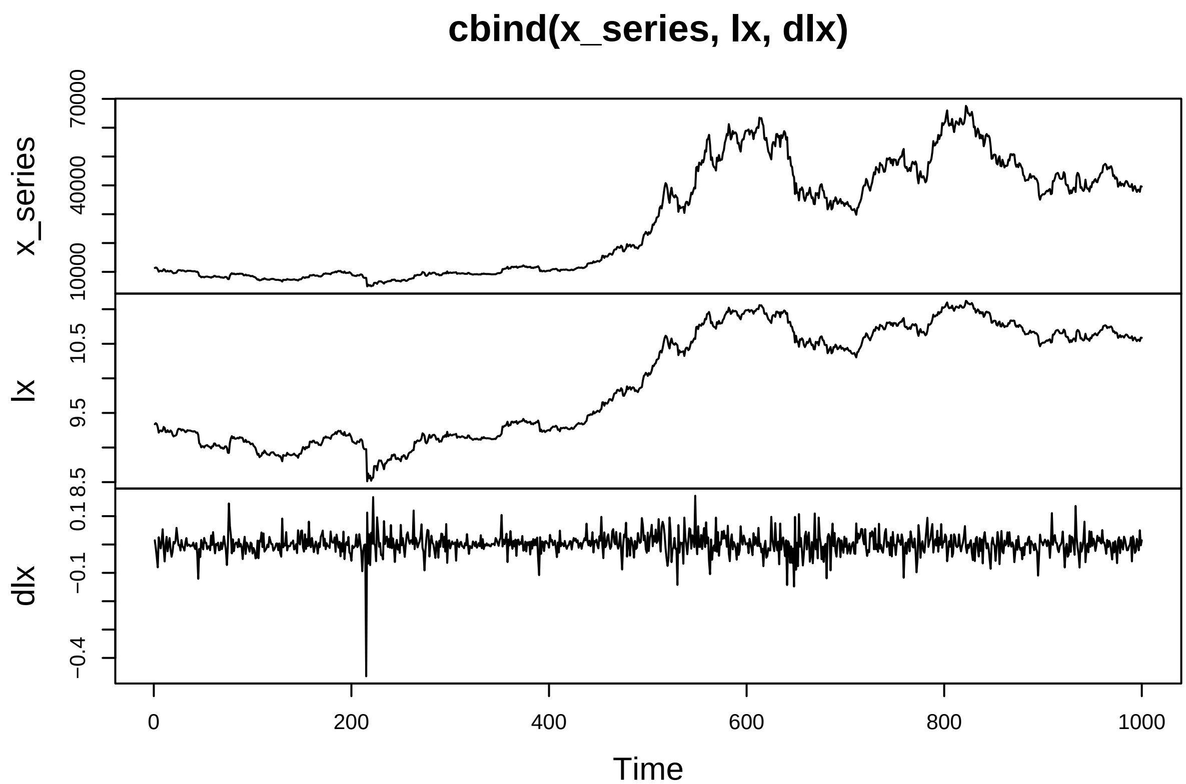
```
# Convert data frame to time series with appropriate frequency
ts_aclose_data <- ts(df1$adj_close, start = c(2019, 8), frequency = 365)  # Assuming daily observations

# Create time series data frame with ordered dates
ts_aclose_df <- data.frame(date = as.Date(df1$date), value = ts_aclose_data)
ts_aclose_df <- ts_aclose_df[order(ts_aclose_df$date), ]

# Plot the time series
plot(ts_aclose_df, type = "l")
```



(2) Transforming series to stabilize variance and ensure stationary.

```
x <- ts_aclose_df
x_series = x$value      # actual adjusted close price values
lx = log(x$value)       # logged values to stabilize variance
dlx = diff(lx)          # logged difference to make series stationary
ddlx = diff(lx, 30)
plot.ts(cbind(x_series, lx, dlx))
```

```
## Warning in cbind(x_series, lx, dlx): number of rows of result is not a multiple
## of vector length (arg 3)
```



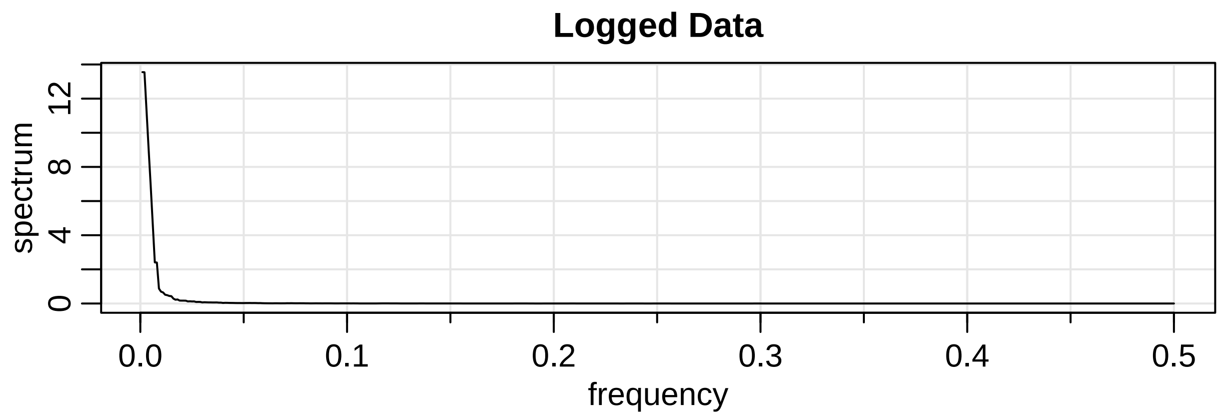
- Use auto.arima() on differenced log data (transformed) bc it's stationary with a zero mean –> get best (p,d,q) values based on AIC and BIC

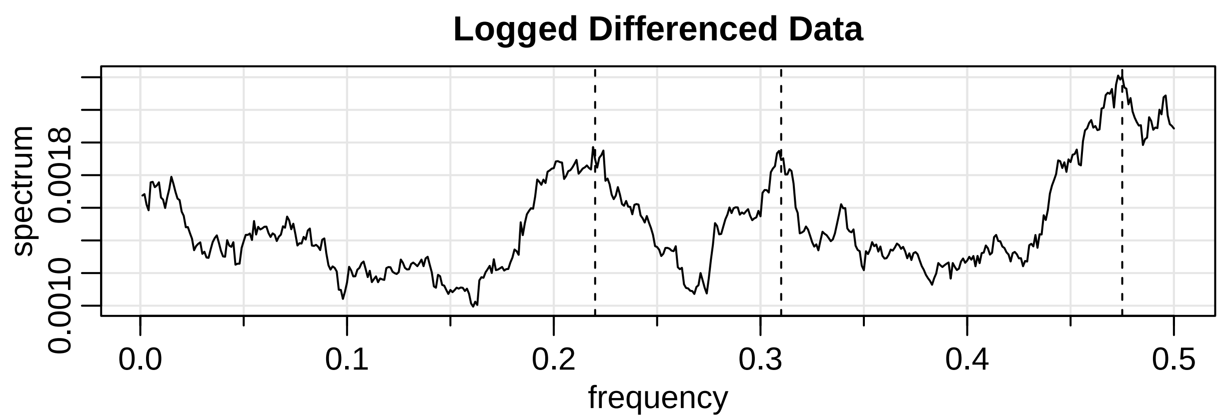
```
auto.arima(lx)
```

```
## Series: lx
## ARIMA(1,1,1)
##
## Coefficients:
##           ar1     ma1
##       -0.8058  0.7520
## s.e.   0.1005  0.1111
##
## sigma^2 = 0.001485:  log likelihood = 1836.49
## AIC=-3666.97   AICc=-3666.95   BIC=-3652.25
```

- Trying to identify seasonal components through periodograms of logged data and logged difference of data.
- Can't see any significant peaks in the logged data, doesn't indicate seasonal components although logged differenced data shows some peaks.

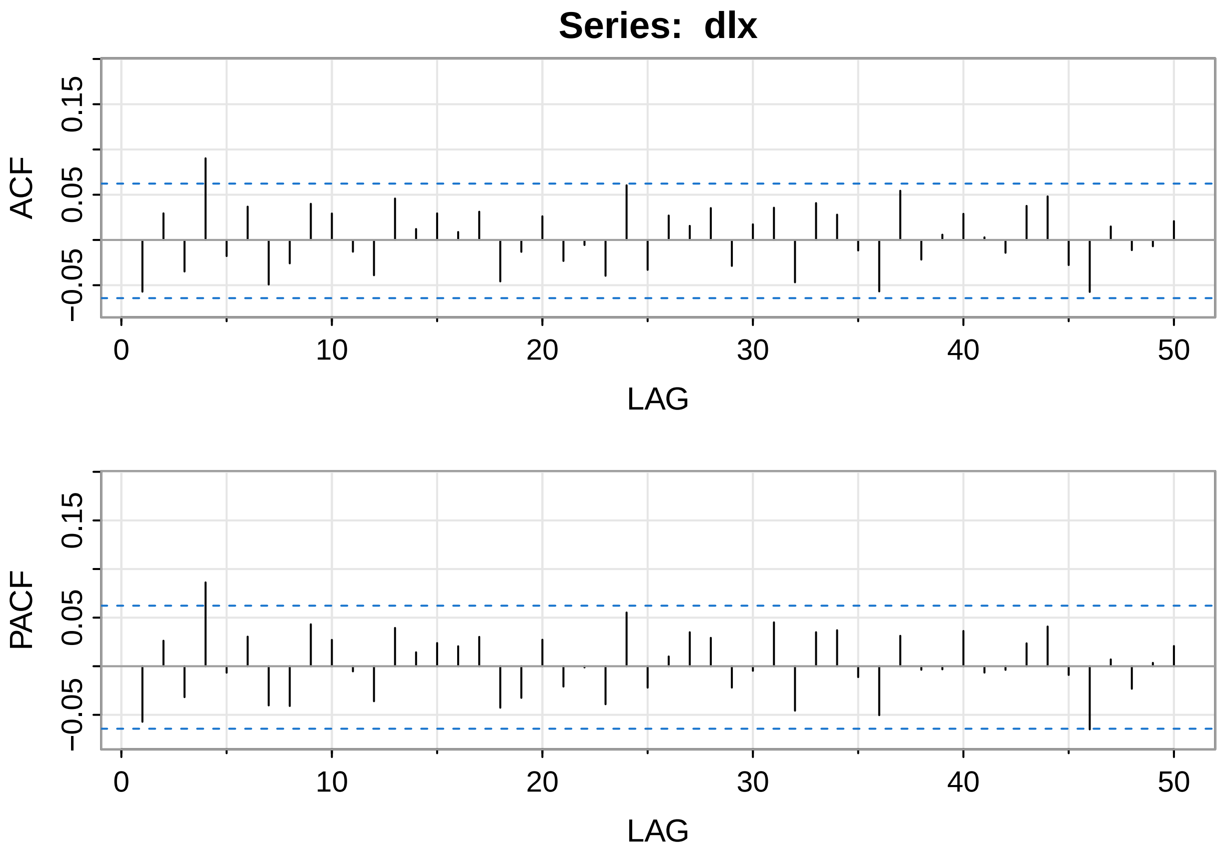
```
# why is differenced logged appear to have significant peaks, whereas logged doesn't --> (differenced l
par(mfrow = c(2,1))
mvspec(lx, kernel('daniell', 4), main = 'Logged Data')
mvspec(dlx, kernel('daniell', 20), main = 'Logged Differenced Data')
abline(v = c(0.22,0.31,0.475), lty = 2)
```



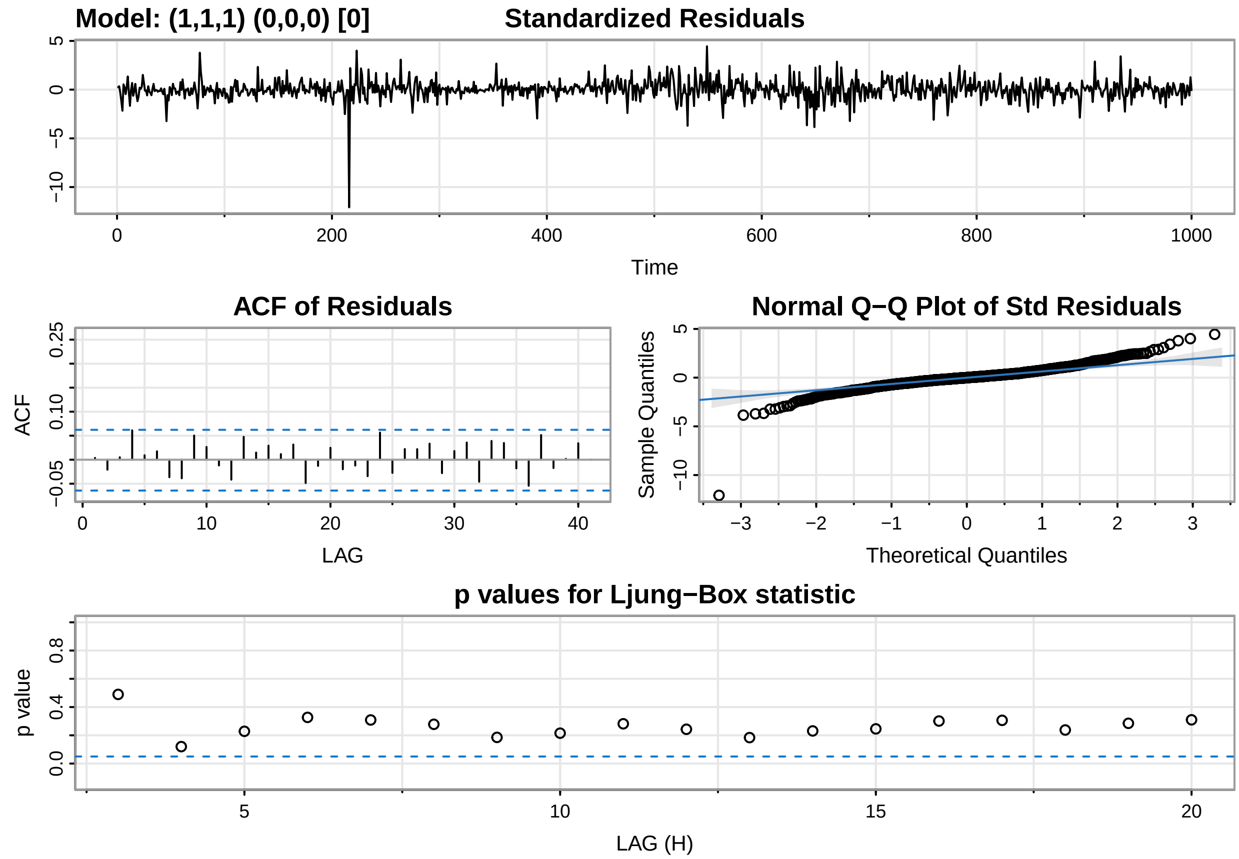


- Because the pacf of the logged difference data is significant at lag 4, an AR(4) model or ARIMA(4,1,0) [on logged data] may be good fits.
- There's no significant lags at any harmonics, nothing to indicate seasonality.

```
acf2(dlx, 50)
```

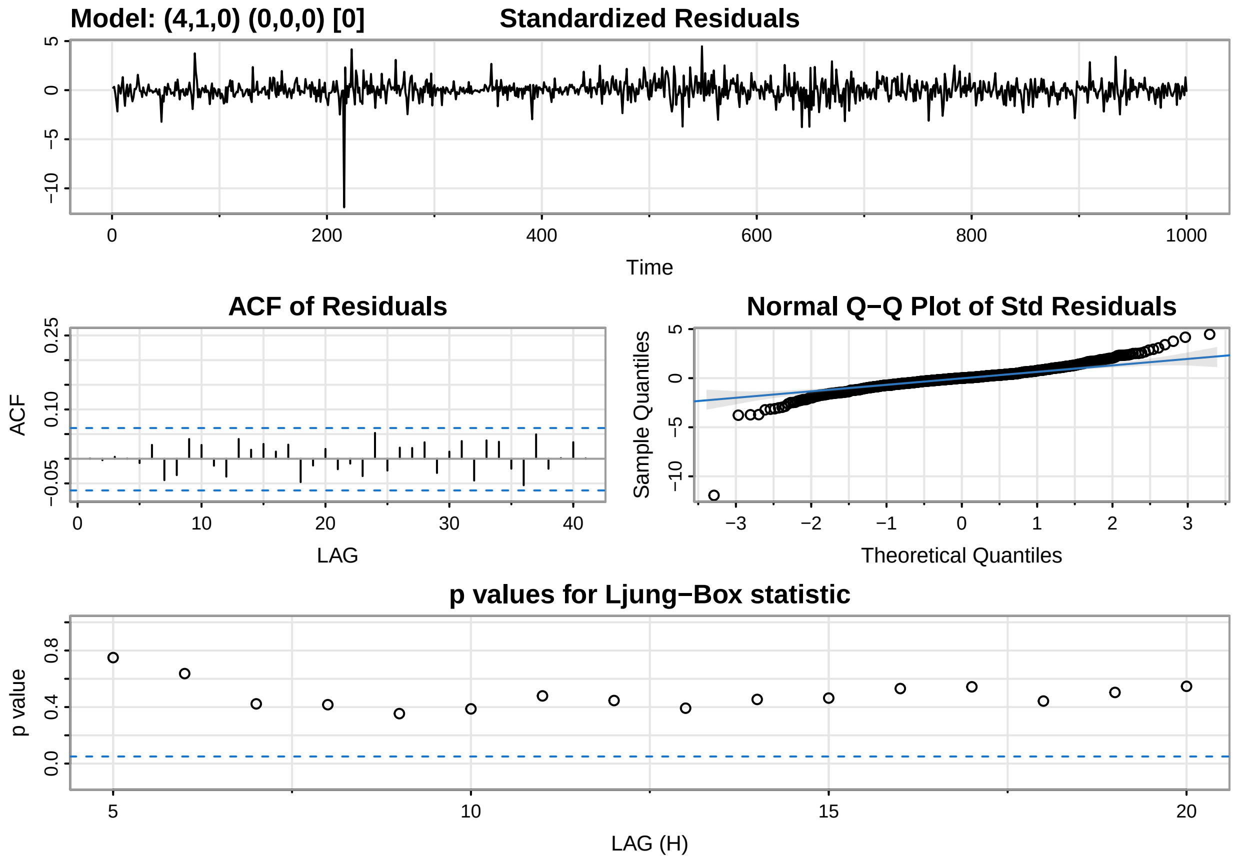



```
# fitting sarima with arima order specified by auto.arima() --> (ARIMA(1,0,1) on the logged difference
# no seasonal component bc while looking at the PACF, only lag 4 is significant, nothing else to indica
par(mfrow = c(2,1))
sarima(lx, 1, 1, 1, 0, 0, 0, 0)
```



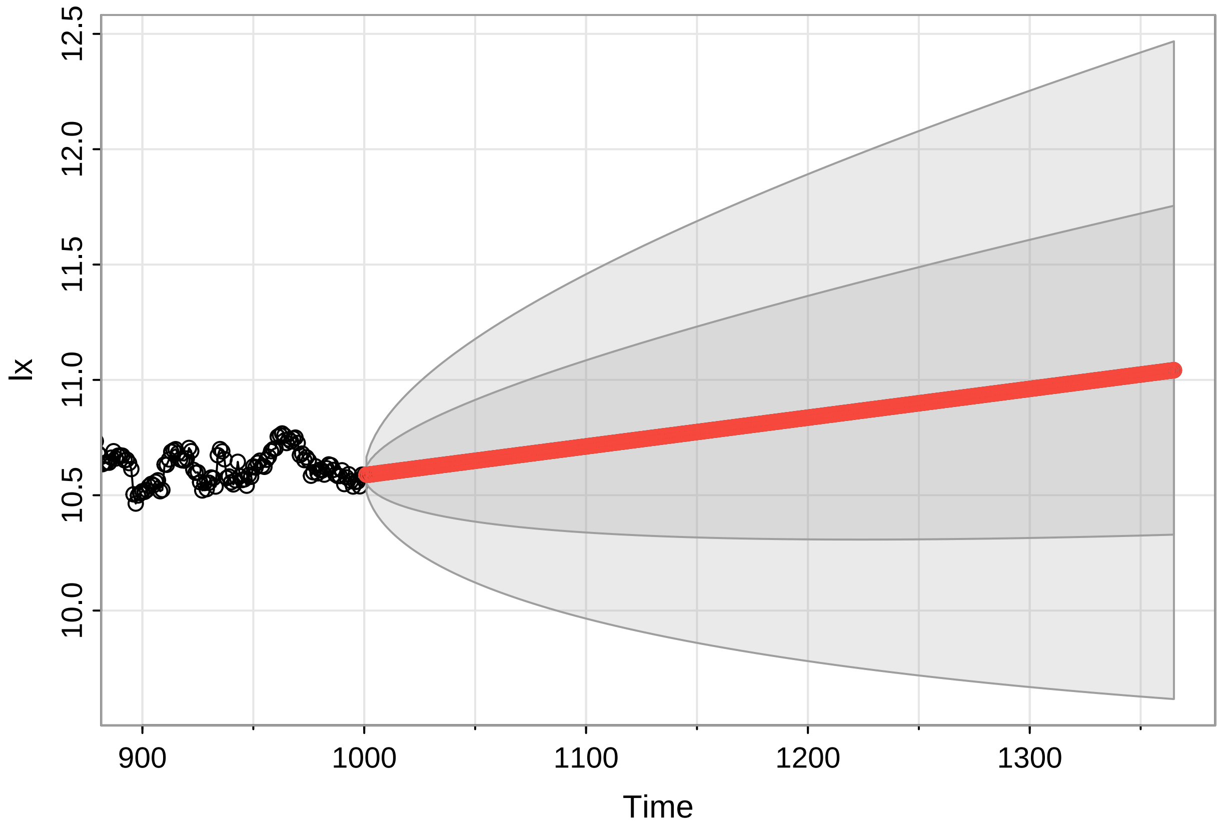
- Fitting sarima with arima order specified by auto.arima() –> (ARIMA(1,0,1) on the logged difference data) which is the same as (ARIMA(1,1,1) on the logged data)
- No seasonal component bc while looking at the PACF, only lag 4 is significant, nothing else to indicate seasonal component in time series data

```
# fitting sarima with arima order (4,0,0) bc pacf is only significant at lag = 4 and nowhere else --> i
# no seasonal component bc while looking at the PACF, only lag 4 is significant, nothing else to indica
sarima(lx, 4, 1, 0, 0, 0, 0)
```



- Both models ARIMA(1,1,1) and ARIMA(4,1,0) seem to perform the about the same looking at the error statistics. We forecast using ARIMA(1,1,1) on the logged data (no seasonal component).
- Because there's no seasonal component, forecast doesn't show much because it's just using an ARIMA(1,1,1) to get future values.

```
sarima.for(lx, 365, 1, 1, 1, 0, 0, 0, 0)
```



12.5
12.0
11.5
11.0
10.5
10.0
lx
900
1000
1100
1200
1300
Time