FE570 - Midterm Exam

I pledge my honor that I have abided by the Stevens Honor System.

Sid Bhatia

2023-10-23

Problem 11

The data for this problem is contained in the file *taqdata BTCUSD.RData*. This is a trade-and-quote file giving the trade price, size, and the quotes at the time of each trade for Bitcoin trades during 24 hours (19-Apr-2023).

```r
# Load necessary packages.
library(xts)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

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

##
## ################################### WARNING ###################################
## # We noticed you have dplyr installed. The dplyr lag() function breaks how    #
## # base R's lag() function is supposed to work, which breaks lag(my_xts).      #
## #                                                                             #
## # If you call library(dplyr) later in this session, then calls to lag(my_xts) #
## # that you enter or source() into this session won't work correctly.         #
## #                                                                             #
## # All package code is unaffected because it is protected by the R namespace   #
## # mechanism.                                                                  #
## #                                                                             #
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning.  #
## #                                                                             #
## # You can use stats::lag() to make sure you're not using dplyr::lag(), or you #
## # can add conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop   #
## # dplyr from breaking base R's lag() function.                                #
## ################################### WARNING ###################################
```

```r
library(highfrequency)

# Load in data set.
options(digits.secs=3)
absolute_path <- 'C:/Users/sbhatia2/My Drive/University/Academics/Semester V/FE570 - Market Microstructu
load(paste(absolute_path, "taqdata_BTCUSD.RData", sep = ""))

# Added to remove warnings about time zone mismatch.
```

```r
Sys.setenv(TZ='GMT')

head(tqdata, 10)
```

```
##                           DT SYMBOL   BID     OFR OFRSIZ BIDSIZ   PRICE
##  1: 2023-04-19 04:00:01.024 XBTUSD 30375 30375.5 189700  56200 30375.0
##  2: 2023-04-19 04:00:01.206 XBTUSD 30375 30375.5 189700  55600 30375.0
##  3: 2023-04-19 04:00:07.138 XBTUSD 30375 30375.5 224100  69300 30375.0
##  4: 2023-04-19 04:00:08.724 XBTUSD 30375 30375.5 227300  54800 30375.5
##  5: 2023-04-19 04:00:11.802 XBTUSD 30375 30375.5 226300  53100 30375.0
##  6: 2023-04-19 04:00:14.295 XBTUSD 30375 30375.5 227500  38600 30375.5
##  7: 2023-04-19 04:00:14.458 XBTUSD 30375 30375.5 227400  38600 30375.5
##  8: 2023-04-19 04:00:15.096 XBTUSD 30375 30375.5 222900  32700 30375.0
##  9: 2023-04-19 04:00:15.224 XBTUSD 30375 30375.5 222900  16200 30375.0
## 10: 2023-04-19 04:00:15.239 XBTUSD 30374 30375.0   7700   2500 30375.0
##     NUMTRADES  SIZE SIDE
##  1:         4 92900 Sell
##  2:         1   600 Sell
##  3:         1   900 Sell
##  4:         1   300  Buy
##  5:         1 16200 Sell
##  6:         1   100  Buy
##  7:         1   100  Buy
##  8:         1 20400 Sell
##  9:         3 21000 Sell
## 10:         2 17300 Sell
```

i. Report the number of trades in the dataset, and the minimum and maximum trade price during the time interval in the dataset.

```r
# Retrieve the number of trades in the dataset.
num_of_trades <- nrow(tqdata)
num_of_trades
```

```
## [1] 58793
```

```r
price <- as.numeric(tqdata$PRICE)

# Establish minimum and maximum prices quoted.
p_min <- min(price)
p_max <- max(price)

p_min
```

```
## [1] 28534.75
```

```r
p_max
```

```
## [1] 30407.5
```

The number of trades is **58793** with the minimum price at **28534.75** and maximum price at **30407.50**.

ii. For each transaction, compute the spread measures:

$$\text{Quoted Spread} : qs_t = \text{Ask}_t - \text{Bid}_t$$

$$\text{Effective Spread}: es_t = 2d_t(p_t - \text{Mid}_t)$$

.

```r
# Compute the bids and asks for each transaction.
ask <- as.numeric(tqdata$OFR)
bid <- as.numeric(tqdata$BID)

# Compute the quoted spread.
quoted_spread <- ask - bid

head(quoted_spread, 50)
```

```
##  [1] 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 1.0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
## [20] 1.5 1.5 5.0 0.5 0.5 0.5 1.0 1.5 0.5 0.5 1.0 1.5 3.5 3.5 0.5 0.5 0.5 0.5 3.0
## [39] 0.5 2.5 0.5 1.0 1.0 2.5 3.5 2.5 0.5 0.5 0.5 0.5
```

```r
tail(quoted_spread, 50)
```

```
##  [1] 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
## [20] 0.5 0.5 0.5 0.5 0.5 0.5 4.5 8.5 0.5 0.5 0.5 8.0 5.0 1.0 0.5 0.5 0.5 7.0 0.5
## [39] 0.5 0.5 0.5 0.5 0.5 0.5 6.0 3.0 0.5 3.5 4.5 5.0
```

```r
# Compute the mid prices (average of best bid and best ask prices).
mid <- (ask + bid) * 0.5

# Retrieve the trade sign for each transaction.
sign <- tqdata$SIDE

# Convert the trade sign for a "Buy" and "Sell" to 1 and -1, respectively.
sign_converted <- sign
sign_converted[sign_converted == "Buy"] <- 1
sign_converted[sign_converted == "Sell"] <- -1

sign_converted <- as.numeric(sign_converted)

head(sign, 10)
```

```
##  [1] "Sell" "Sell" "Sell" "Buy"  "Sell" "Buy"  "Buy"  "Sell" "Sell" "Sell"
```

```r
head(sign_converted, 10)
```

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

```r
# Calculate the effective spread.
effective_spread <- 2 * sign_converted * (price - mid)

head(effective_spread, 50)
```

```
##  [1]  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5 -1.0  0.5  0.5  0.5  0.5 -0.5
## [16] -0.5  0.5  0.5  0.5 -0.5 -0.5 -4.0  0.5  0.5  0.5  0.0 -0.5  0.5  0.5  0.0
## [31]  0.5  3.5  3.5  0.5  0.5  0.5  0.5 -2.0  0.5 -1.5  0.5  0.0  0.0 -1.5  1.5
## [46] -1.5  0.5  0.5  0.5  0.5
```

```r
tail(effective_spread, 50)
```

```
##  [1]  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5 -0.5
## [16]  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.0  6.0  0.5  0.5  0.5
```

```
## [31]  0.0  0.0  1.0  0.5  0.5  0.5  3.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  1.0
## [46]  3.0  0.5 -2.5 -3.5 -3.0
```

```r
mean(quoted_spread)
```

```
## [1] 4.816764
```

```r
mean(effective_spread)
```

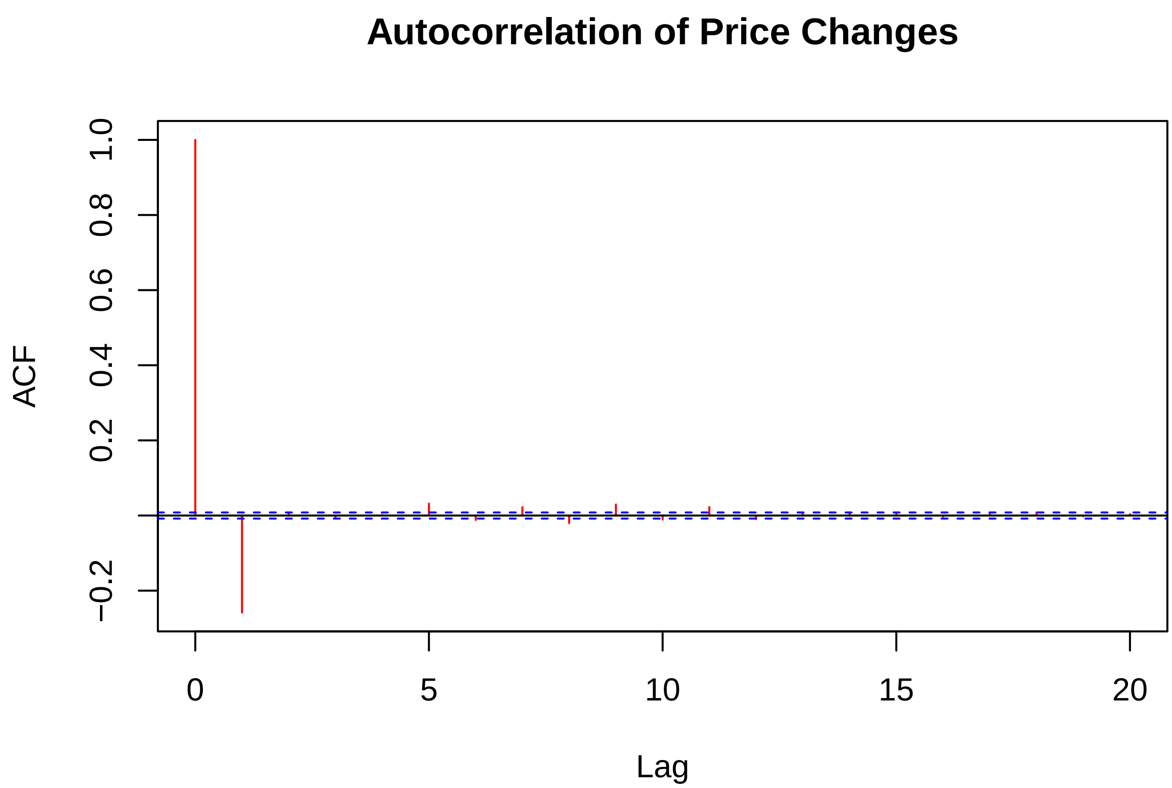
```
## [1] 1.956644
```

The average quoted spread is **4.817** and the average effective spread is **1.957**.

iii. Compute the Roll's estimate of the bid-ask spread.

```r
# Calculate the difference in price changes.
dprice <- diff(price)

# Compute and plot the autocorrelation of price changes.
ac_pr <- acf(dprice, lag.max=20, type="correlation", plot=FALSE)
plot(ac_pr, col="red", main="Autocorrelation of Price Changes")
```




```r
# Compute the covariances of the price changes.
covpr <- acf(dprice, lag.max=20, type="covariance", plot=FALSE)

# Retrieve gamma1 as the covariance at lag 0.
gamma0 <- covpr$acf[1]
gamma0
```

```
## [1] 21.90887
```

```r
# Retrieve gamma1 as the covariance at lag 1.
gamma1 <- covpr$acf[2]
gamma1
```

```
## [1] -5.655469
```

```r
# Compute the volatility of the efficient price.
sig2u <- gamma0 + 2 * gamma1
sigu <- sqrt(sig2u)
sigu
```

```
## [1] 3.255447
```

```r
# Compute the c paramater as the sqrt(-gamma1)
cparam <- sqrt(-gamma1)
cparam
```

```
## [1] 2.378123
```

```r
# Compute the spread as 2 * the c parameter.
roll_spread <- cparam * 2

roll_spread
```

```
## [1] 4.756246
```

As such, the Roll's model estimate of the bid-ask spread is **4.756** with $c = 2.378$ and $\sigma_u = 3.255$.

iv. Compare the trade sign in SIDE with the prediction of the Lee-Ready empirical rule.

What is the accuracy of the Lee-Ready rule?

This can be measured as the percentage of trade signs which are predicted correctly by the Lee-Ready rule.

Tick Test: Use only the trade prices p_t, but not the quotes a_t and b_t. Under the test, the trade is classified as a buy/sell according to: - $d_t = +1$ (buy) if $p_t > p_{t-1}$ (uptick) or if $p_t = p_{t-1} > p_{t-2}$ (zero-uptick) - $d_t = -1$ (sell) if $p_t < p_{t-1}$ (downtick) or if $p_t = p_{t-1} < p_{t-2}$ (zero-downtick)

Note that zero-uptick/downtick results apply also if there are multiple (more than 2) trades with the same price.

For example if the trade prices are $p_t = (19.9, 20.0, 20.0, 20.0)$ (increasing t order), then the trade signs are (?, +, + , +).

Lee-Ready Rule: Use both p_t and quotes a_t and b_t. The Lee-Ready Rule decides if a trade is a buy or sell by comparing the trade price p_t with the mid-price $m_t = \frac{1}{2}(a_t + b_t)$ (the half-point between best-bid b_t and best-ask a_t).

If the trade price is exactly equal to the mid-price, $p_t = m_t$, then use the tick rule in point (i) above.

```r
# Create a function that implements the Tick Test.
tick_test <- function(price)
{
    sign <- c(1)
    for(i in 2:(length(price)))
    {
        if(price[i] < price[i - 1])
        {
            sign <- c(sign, -1)
```

```
        }
        else if(price[i] > price[i - 1])
        {
            sign <- c(sign, 1)
        }
        else
        {
            sign <- c(sign, sign[i - 1])
        }
    }
    return(sign)
}

# Create a function that implements the Lee-Ready Rule.
lee_ready_rule <- function(price)
{
    tick <- tick_test(price)
    sign <- c(1)
    bid <- sapply(tqdata$BID, FUN = as.numeric)
    ask <- sapply(tqdata$OFR, FUN = as.numeric)

    for(i in 2:(length(price)))
    {
        mid <- (bid[i] + ask[i]) * 0.5

        if(price[i] > mid)
        {
            sign <- c(sign, 1)
        }
        else if(price[i] < mid)
        {
            sign <- c(sign, -1)
        }
        else
        {
            sign <- c(sign, tick[i])
        }
    }
    return(sign)
}

Lee_Ready_Rule_TQ <- lee_ready_rule(price)

Lee_Ready_Rule_Actual <- getTradeDirection(tqdata)

# Check to see if Lee-Ready implementation is the same.
length( which(Lee_Ready_Rule_Actual == Lee_Ready_Rule_TQ ) ) / length(Lee_Ready_Rule_TQ)
```

```
## [1] 0.999983
```

```
# Check to see accuracy of Lee-Ready Rule.
length( which(sign_converted ==  Lee_Ready_Rule_Actual) ) / length(sign_converted)
```

```
## [1] 0.7426564
```

```r
length( which(sign_converted ==  Lee_Ready_Rule_TQ) ) / length(sign_converted)
```

```
## [1] 0.7426394
```

As such, the Lee-Ready rule is 74.3% accurate in terms of the trade signs it correctly predicted.