FE 570 Homework 2

Arjun Koshal

10/21/2022

Loading Data Set

```
library(xts)
```

```
## Loading required package: zoo
```

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

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

```
library(highfrequency)
options(digits.secs=3)

load("sampleTQdata.RData")
Sys.setenv(TZ='GMT') # added to remove warnings about time zone mismatch

head(tqdata)
```

```
##                     SYMBOL EX  BID       BIDSIZ OFR       OFRSIZ MODE PRICE
## 2008-01-04 09:30:27 "XXX"  "N" "193.340" "4.5"  "193.890" "11.5" "12" "193.710"
## 2008-01-04 09:30:28 "XXX"  "N" "193.340" "4.5"  "193.890" "11.5" "12" "193.590"
## 2008-01-04 09:30:29 "XXX"  "N" "193.250" "12.5" "193.810" "8.5"  "12" "193.445"
## 2008-01-04 09:30:30 "XXX"  "N" "193.470" "0.5"  "193.630" "0.5"  "12" "193.380"
## 2008-01-04 09:30:31 "XXX"  "N" "193.470" "0.5"  "193.630" "0.5"  "12" "193.340"
## 2008-01-04 09:30:33 "XXX"  "N" "193.300" "2.5"  "193.640" "0.5"  "12" "193.520"
##                     SIZE
## 2008-01-04 09:30:27 "9100"
## 2008-01-04 09:30:28 "200"
## 2008-01-04 09:30:29 "200"
## 2008-01-04 09:30:30 "250"
## 2008-01-04 09:30:31 "300"
## 2008-01-04 09:30:33 "400"
```

```
tail(tqdata)
```

```
##                      SYMBOL EX  BID       BIDSIZ OFR       OFRSIZ MODE PRICE
## 2008-01-04 15:59:52 "XXX"  "N" "191.600" "60.5" "191.670" "3.5"  "12" "191.695"
## 2008-01-04 15:59:55 "XXX"  "N" "191.620" "0.5"  "191.790" "1.5"  "12" "191.620"
## 2008-01-04 15:59:57 "XXX"  "N" "191.600" "180"  "191.690" "27.5" "12" "191.690"
## 2008-01-04 15:59:58 "XXX"  "N" "191.600" "180"  "191.690" "27.5" "12" "191.650"
## 2008-01-04 15:59:59 "XXX"  "N" "191.600" "180"  "191.690" "27.5" "12" "191.620"
## 2008-01-04 16:00:00 "XXX"  "N" "191.600" "180"  "191.690" "27.5" "12" "191.670"
##                      SIZE
## 2008-01-04 15:59:52 "550"
## 2008-01-04 15:59:55 "1600"
## 2008-01-04 15:59:57 "350"
## 2008-01-04 15:59:58 "150"
## 2008-01-04 15:59:59 "50"
## 2008-01-04 16:00:00 "50"
```

Problem 2.1

i) How many trades are in the dataset?

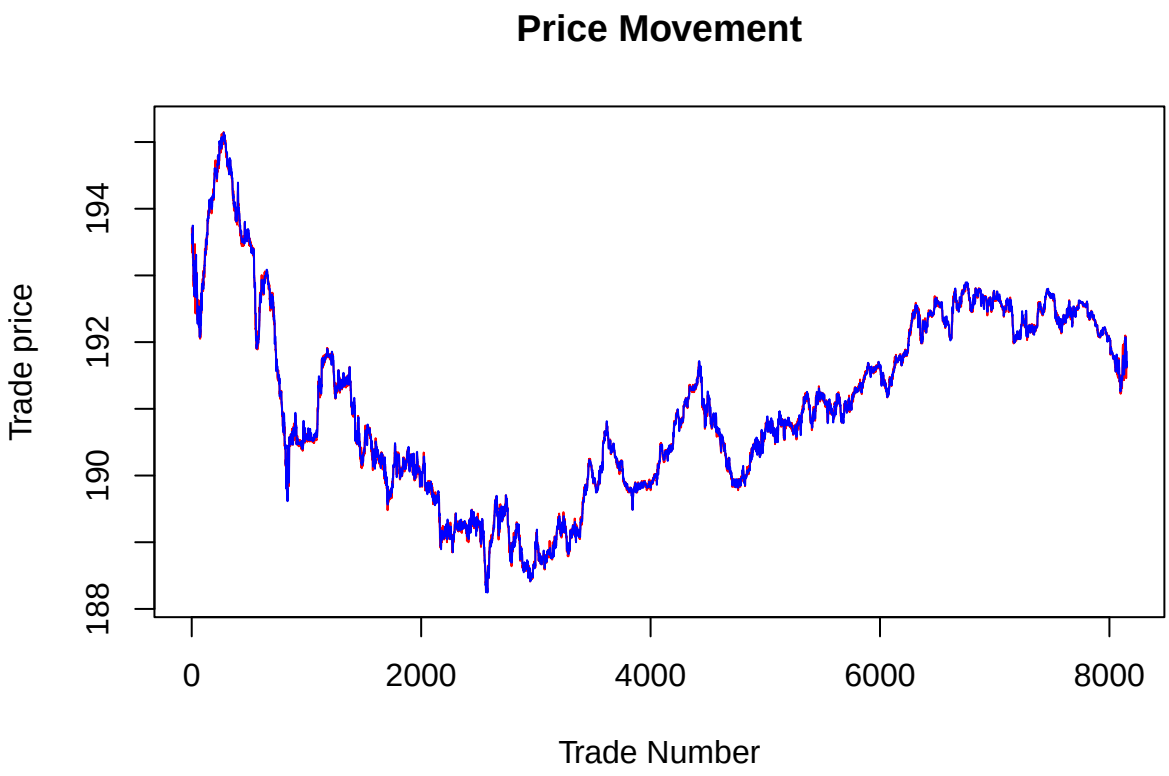
```
n.trades <- length(tqdata$SIZE)
n.trades
```

```
## [1] 8153
```

There are 8153 trades in the dataset.

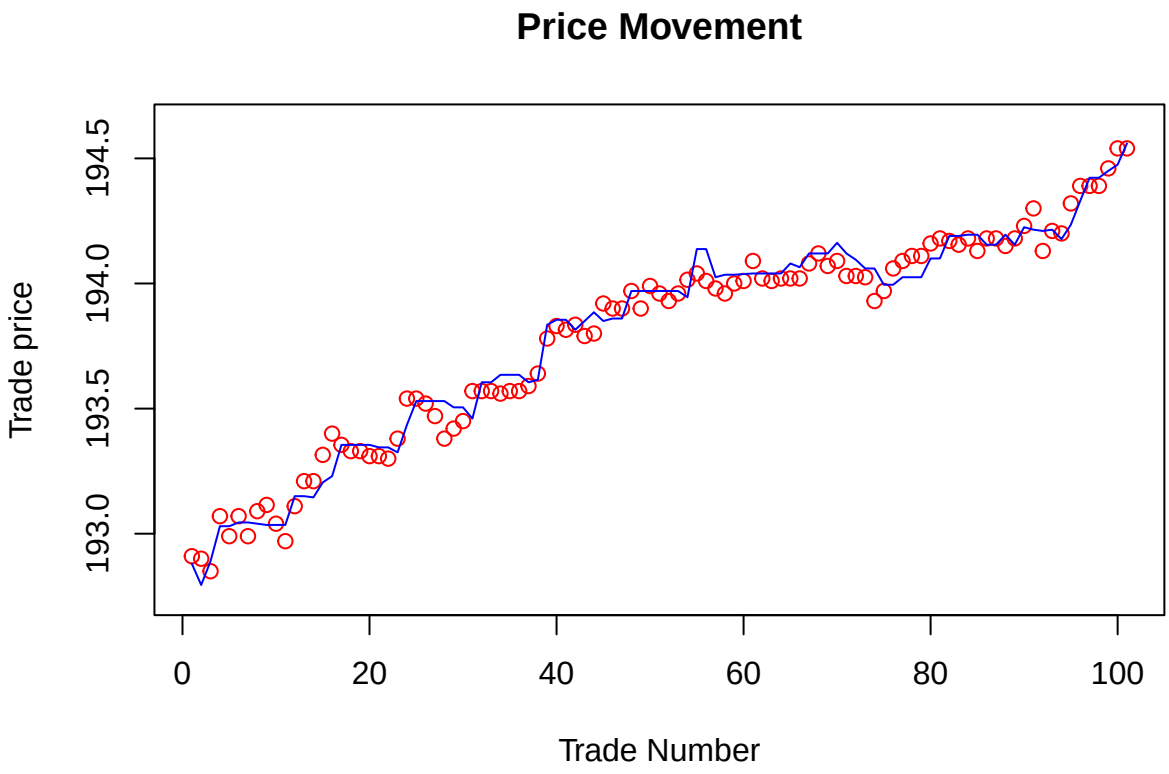
ii) plot the trade prices p_t and the best-bid b_t and best-ask prices a_t for the entire dataset.

```
TQ <- data.frame(Date=index(tqdata), tqdata)
asks <- as.numeric(tqdata$OFR)
bids <- as.numeric(tqdata$BID)
mids <- 0.5*bids + 0.5*asks

pmin = min(as.numeric(tqdata$PRICE))
pmax = max(as.numeric(tqdata$PRICE))
plot(as.numeric(tqdata$PRICE),col="red", type="l", ylab="Trade price",
     xlab="Trade Number", main="Price Movement", ylim=c(pmin-0.1,pmax+0.1))
lines(mids, type="l", col="blue")
```



iii) same as in ii) but only for trades with counts 100:200

```
# Slice the tqdata from 100 to 200.
tqdata1 <- tqdata[100:200]
asks1 <- as.numeric(tqdata1$OFR)
bids1 <- as.numeric(tqdata1$BID)
mids1 <- 0.5*bids1 + 0.5*asks1

TQ <- data.frame(Date=index(tqdata1), tqdata1)
pmin = min(as.numeric(tqdata1$PRICE))
pmax = max(as.numeric(tqdata1$PRICE))
plot(as.numeric(tqdata1$PRICE),col="red", type="p", ylab="Trade price",
     xlab="Trade Number", main="Price Movement", ylim=c(pmin-0.1,pmax+0.1))
lines(mids1, type="l", col="blue")
```



Problem 2.2

Count how many trades take place within the spread ($p_t \in (b_t, a_t)$), and how many at the touch ($p_t = b_t$ or $p_t = a_t$). Give separately the three numbers, and test if their sum reproduces the total trade count from 1.i).

```
bid <- sapply(tqdata$BID, FUN = as.numeric)
ask <- sapply(tqdata$OFR, FUN = as.numeric)
price <- sapply(tqdata$PRICE, FUN = as.numeric)
within <- length(which((price > bid) & (price < ask)))
outside <- length(which(((price < bid) | (price > ask))))
at_bid <- length(which(price == bid))
at_offer <- length(which(price == ask))

within
```

```
## [1] 2832
```

```
at_bid
```

```
## [1] 1709
```

```
at_offer
```

```
## [1] 1370
```

```
outside
```

```
## [1] 2242
```

```
within + at_bid + at_offer + outside
```

```
## [1] 8153
```

```
n.trades
```

```
## [1] 8153
```

Problem 2.3

Determine the "trade direction" d_t of each trade, which shows if it is a buy ($d_t = +1$) or a sell ($d_t = -1$). There are two ways to do this, implement each of them in R.

i) Tick test. This test uses only the trade prices p_t, but not the quotes a_t, b_t. Under this test the trade is classified as 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 $p_t = (20.0, 20.0, 20.0, 19.9)$, then the trade signs are $(+, +, +, ?)$.

```
tick_rule <- 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)
}
```

ii) Lee-Ready rule. This test uses both trade prices p_t and quotes a_t, 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.

```
lee_ready <- function(price) {
  tick <- tick_rule(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])/2
```

```
    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)
}
```

```
Tick_Rule_Function <- tick_rule(price)
Lee_Ready_Function <- lee_ready(price)
Lee_Ready_Actual <- getTradeDirection(tqdata)

length(which(Lee_Ready_Actual == Lee_Ready_Function))/length(Lee_Ready_Function)
```

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

```
length(which(Tick_Rule_Function ==
               Lee_Ready_Function))/length(Tick_Rule_Function)
```

```
## [1] 0.7944315
```

About 79% the trades are classified the same way by the two methods, Tick Rule and Lee-Ready.

Exactly all the trades are classified the same way by the built-in method, getTradeDirection, and the Lee Ready function that was created in Problem 3.2.