Analysis

PART I: Implementation in R

```
Data = read.csv(file="/Users/ruohanwu/Desktop/IBM.1min.TradesAndQuotes.20160128.csv",
                sep=",")
data = data.frame("Time" = Data$TimeBarStart)
data$BidPrice = Data$OpenBidPrice
data$AskPrice = Data$OpenAskPrice
data$Avg = (data$BidPrice + data$AskPrice) / 2
length = length(data$Avg)
data$MovingAvg_5[6:length] = rollmean(data$Avg, 6)
data$MovingAvg_20[21:length] = rollmean(data$Avg, 21)
```

```
# Signal of 1 means go long and Signal of 0 means go short
signal = array(dim = c(length,1))
signal[,1] = 0
for (i in 21:length){
  ind1 = (data$MovingAvg_5[i] > data$MovingAvg_20[i]
          & data$MovingAvg_5[i-1] <= data$MovingAvg_20[i-1])
  ind2 = (data$MovingAvg_5[i] < data$MovingAvg_20[i]
          & data$MovingAvg_5[i-1] >= data$MovingAvg_20[i-1])
  if (!is.na(ind1) & ind1){
    signal[i] = 1
  }
  if (!is.na(ind2) & ind2){
    signal[i] = -1
  }
}
data$signal = signal
```

```
#Time to go long
print(data$Time[which(data$signal == 1)])
```

```
##  [1] 06:01 07:00 08:09 08:51 09:54 10:36 11:03 12:06 13:11 13:35 14:11
## [12] 14:14 15:10 15:38 16:06 16:44 18:20 19:03 19:34 20:00
## 960 Levels: 04:01 04:02 04:03 04:04 04:05 04:06 04:07 04:08 04:09 ... 20:00
```

```
#Time to go short
print(data$Time[which(data$signal == -1)])
```

```
##  [1] 06:38 07:12 08:31 09:51 10:03 10:42 11:27 12:54 13:21 13:53 14:13
## [12] 14:40 15:25 15:46 16:18 16:59 18:40 19:33 19:38
## 960 Levels: 04:01 04:02 04:03 04:04 04:05 04:06 04:07 04:08 04:09 ... 20:00
```

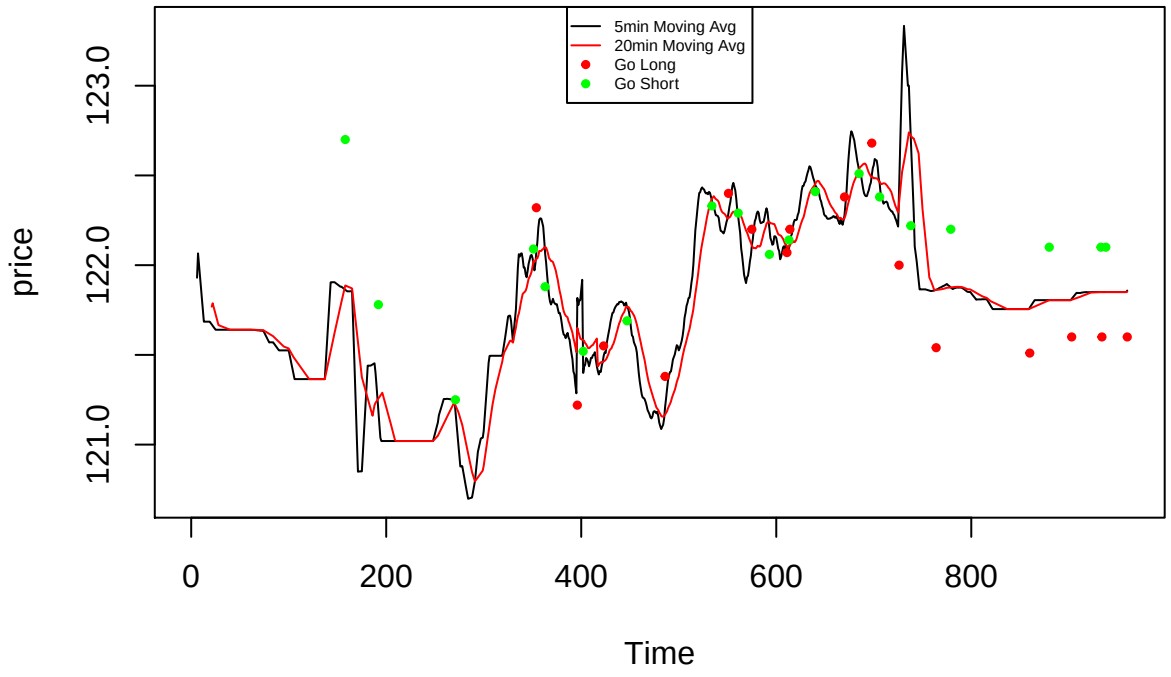
```
#Time to pass an order (go long or go short)
print(data$Time[which(data$signal == 1 | data$signal == -1)])
```

```
##  [1] 06:01 06:38 07:00 07:12 08:09 08:31 08:51 09:51 09:54 10:03 10:36
## [12] 10:42 11:03 11:27 12:06 12:54 13:11 13:21 13:35 13:53 14:11 14:13
## [23] 14:14 14:40 15:10 15:25 15:38 15:46 16:06 16:18 16:44 16:59 18:20
## [34] 18:40 19:03 19:33 19:34 19:38 20:00
## 960 Levels: 04:01 04:02 04:03 04:04 04:05 04:06 04:07 04:08 04:09 ... 20:00
```

Plot of the price change and the orders

The black curve is the 5-minutes moving average and the red curve is the 20-minutes moving average. The red dots denote the orders corresponding to our go-long signal. And the green dots are the orders to go short. All of their values are the prices.

```
plot(data$MovingAvg_5, type = 'l',xlab = "Time", ylab = "price")
lines(data$MovingAvg_20, col = 'red', type = 'l')
points(which(data$signal == 1),data$BidPrice[which(data$signal == 1)],
       pch = 19, cex = 0.5, col = 'red')
points(which(data$signal == -1),data$AskPrice[which(data$signal == -1)],
       pch = 19, cex = 0.5, col = 'green')
legend("top",cex=.5,legend=c("5min Moving Avg", "20min Moving Avg", "Go Long", "Go Short"),
       col=c("black", "red","red","green"), lty=c(1,1,0,0), pch = c(NA,NA,19,19))
```



PART II: Comparision with C++ Output

```
MyDataAvg <- read.csv(file="/Users/ruohanwu/Desktop/ieorhomework7/cmake-build-debug/averagePrice.csv",
                      header=FALSE, sep=",")
nRow = nrow(MyDataAvg)
MyDataOrder <- read.csv(file="/Users/ruohanwu/Desktop/ieorhomework7/cmake-build-debug/output.csv",
                        header=FALSE, sep=",")
BuyRow = (MyDataOrder[,4] == 1)
SellRow = (MyDataOrder[,4] == 0)
BuyOrder = MyDataOrder[BuyRow,]
SellOrder = MyDataOrder[SellRow,]
#Time to go long
print(BuyOrder$V1)
```

```
##  [1] 20160128 04:07:00 20160128 06:18:00 20160128 07:00:00
```

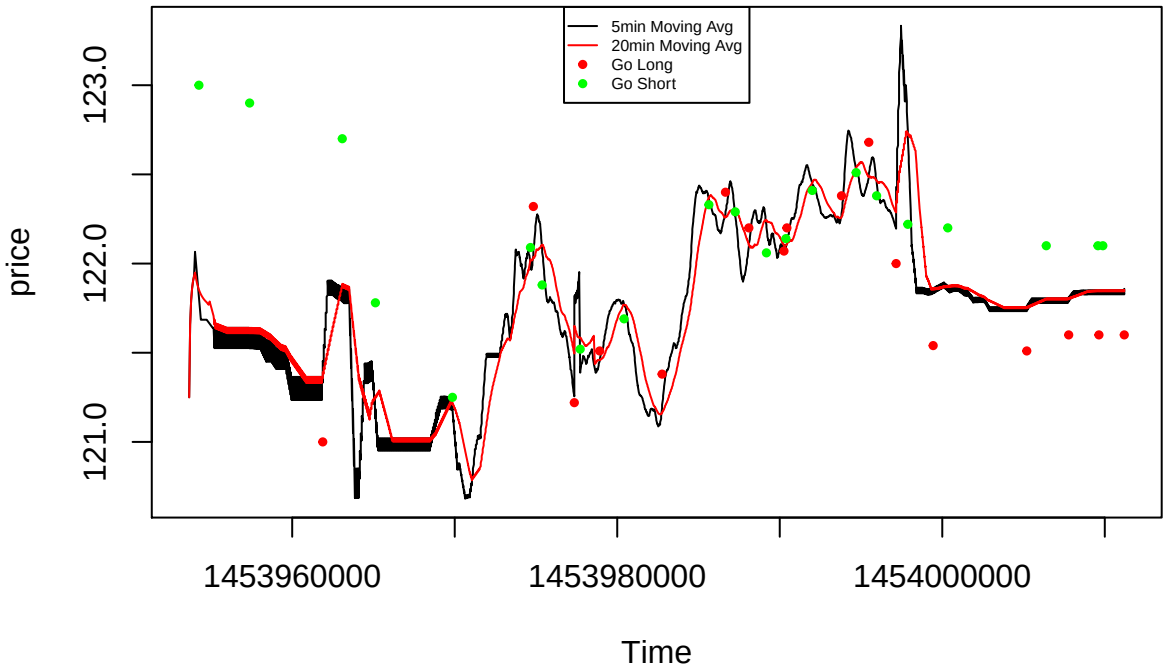
```
##  [4] 20160128 08:09:00 20160128 08:51:00 20160128 09:54:00
##  [7] 20160128 10:36:00 20160128 11:02:00 20160128 12:06:00
## [10] 20160128 13:11:00 20160128 13:35:00 20160128 14:11:00
## [13] 20160128 14:14:00 20160128 15:10:00 20160128 15:38:00
## [16] 20160128 16:06:00 20160128 16:44:00 20160128 18:20:00
## [19] 20160128 19:03:00 20160128 19:34:00 20160128 20:00:00
## 42 Levels: 20160128 04:07:00 20160128 04:11:00 ... 20160128 20:00:00
```

```
#Time to go short
print(SellOrder$V1)
```

```
##  [1] 20160128 04:11:00 20160128 05:03:00 20160128 06:38:00
##  [4] 20160128 07:12:00 20160128 08:31:00 20160128 09:51:00
##  [7] 20160128 10:03:00 20160128 10:42:00 20160128 11:27:00
## [10] 20160128 12:54:00 20160128 13:21:00 20160128 13:53:00
## [13] 20160128 14:13:00 20160128 14:40:00 20160128 15:25:00
## [16] 20160128 15:46:00 20160128 16:18:00 20160128 16:59:00
## [19] 20160128 18:40:00 20160128 19:33:00 20160128 19:38:00
## 42 Levels: 20160128 04:07:00 20160128 04:11:00 ... 20160128 20:00:00
```

When the simulator fill all the data we pass to the simulator, the plot will look like the following, where the red dots are orders to go long and the green dots are orders to go short.

```
plot(MyDataAvg[,1],MyDataAvg[,2],type = 'l',xlab = "Time", ylab = "price")
lines(MyDataAvg[,1],MyDataAvg[,3],type = 'l', col = 'red')
points(BuyOrder[,2],BuyOrder[,3], col = 'red',pch = 19, cex = 0.5)
points(SellOrder[,2],SellOrder[,3], col = 'green',pch = 19, cex = 0.5)
legend("top",cex=.5,legend=c("5min Moving Avg", "20min Moving Avg", "Go Long", "Go Short"),
       col=c("black", "red","red","green"), lty=c(1,1,0,0), pch = c(NA,NA,19,19))
```



We can see from the plot that the time given by the two different implementations to pass an order are the same.

If we want to make the simulator more realistic, which means the simulator will not always fill an order

whenever we pass it, we will introduce a probability p. So I introduce a binary random variable, which can give us the random probability p = 0.5 to fill an order that was passed to the simulator and 0.5 probability not to fill an order. We will plot such result in the following figure.

```
BuyRow_fill = (MyDataOrder[,4] == 1 & MyDataOrder[,5] == "FILLED")
BuyRow_rjct = (MyDataOrder[,4] == 1 & MyDataOrder[,5] == "REJECTED")
SellRow_fill = (MyDataOrder[,4] == 0 & MyDataOrder[,5] == "FILLED")
SellRow_rjct = (MyDataOrder[,4] == 0 & MyDataOrder[,5] == "REJECTED")
BuyOrder_fill = MyDataOrder[BuyRow_fill,]
BuyOrder_rjct = MyDataOrder[BuyRow_rjct,]
SellOrder_fill = MyDataOrder[SellRow_fill,]
SellOrder_rjct = MyDataOrder[SellRow_rjct,]
plot(MyDataAvg[,1],MyDataAvg[,2],type = 'l',xlab = "Time", ylab = "price")
lines(MyDataAvg[,1],MyDataAvg[,3],type = 'l', col = 'red')
points(BuyOrder_fill[,2],BuyOrder_fill[,3], col = 'red',pch = 19, cex = 0.5)
points(BuyOrder_rjct[,2],BuyOrder_rjct[,3], col = 'red',pch = 4, cex = 0.5)
points(SellOrder_fill[,2],SellOrder_fill[,3], col = 'green',pch = 19, cex = 0.5)
points(SellOrder_rjct[,2],SellOrder_rjct[,3], col = 'red',pch = 4, cex = 0.5)
legend("top",cex=.5,legend=c("5min Moving Avg", "20min Moving Avg",
                             "Filled Go Long","Rejected Go Long",
                             "Filled Go Short", "Rejected Go Short"),
       col=c("black", "red","red","red","green","green"),
       lty=c(1,1,0,0,0,0), pch = c(NA,NA,19,4,19,4))
```