CF963 Computational models in Economics and Finance

Name: Heera Kesavan Ezhuthachan

Registration Number: 2010488

Task 1:

Steps performed:

- Load data JET.L (JustEat stock 6M)
- Calculate 7 day and 14 day moving average creating For loop as function movmean() did not give appropriate results
- Generated BUY and SELL Signals when 7MA crossed 14 MA from below and when 7MA crossed 14MA from above respectively.
- Created an Auxiliary column for Number of Stocks in Budget using formula below:
- Stocks in Budget = round (InitialBudget/closeprice)
- Initial Budget =1000000 GBP
- Created a For loop with if- else conditions to calculate updating Budget and Number of Shares:
- if BUY_signal(i-1) == 1
- Budget(i) = Budget(i-1)-Stocks(i-1)*closeprice(i-1);
- elseif SELL_signal(i-1) == 1
- Budget (i) = Budget(i-1)+Shares(i-1)*closeprice(i-1);
- else
- Budget(i) = Budget(i-1);
- End
-
- Calculated and displayed results in table on which dates or when Algorithm buys or sells.
- Calculated and displayed results in table how much Algorithm buys or sells.
- As there is a profit , calculated total profit % which is 7.3% or 73094 GBP
- Results added after script.

```
%Load Data

TABLE=readtable("JETL.csv");
%Selecting Column with Closing prices
a_table=TABLE(:,5);
date=TABLE(:,1);
date(end,:) = [];
%Calculate Moving Average for 7 days and 14 days on Close price
closeprice = table2array(a_table);

%MA7 = movmean(a_array,7)
%MA14 = movmean(a_array,14)
g=1/7;
h=1/14;
[r,c] = size(closeprice);
```

```
%Movmean() gives wrong results hence finding 7 day and 14 day moving
average
%using the formula in a for loop
for k = 8:r
    MA7(k) = g*(closeprice(k-7)+closeprice(k-6)+closeprice(k-
5)+closeprice(k-4)+closeprice(k-3)+closeprice(k-2)+closeprice(k-1));
end
for k = 15:r
    MA14(k) = h*(closeprice(k-14)+closeprice(k-13)+closeprice(k-
12)+closeprice(k-11)+closeprice(k-10)+closeprice(k-9)+closeprice(k-
8)+closeprice(k-7)+closeprice(k-6)+closeprice(k-5)+closeprice(k-
4)+closeprice(k-3)+closeprice(k-2)+closeprice(k-1));
end
MovAvg7 = MA7.';
MovAvg14 = MA14.';
BUY_signal =[];
SELL_signal=[];
Budget = [];



%Generating BUY and SELL Signals when 7 day moving average crosses
14 day
%moving average from below and above respectively.

[numRows,numCols] = size(MovAvg7);
for i = 1:numRows-1
    disp("numRows")
    BUY_signal(i,1) = (MovAvg7(i,1) < MovAvg14(i,1)) &
(MovAvg7(i+1,1) > MovAvg14(i+1,1));
    SELL_signal(i,1) = (MovAvg7(i,1) > MovAvg14(i,1)) &
(MovAvg7(i+1,1) < MovAvg14(i+1,1));
end


InitialBudget = 1000000; %Available Budget
Shares(1) = 0; %Number of Shares we have initially
Stocks(1) = round(InitialBudget/closeprice(1));%Number of stocks in
budget initially
Budget(1)= InitialBudget;


for i = 2:r
    %Calculating Available Budget
    if BUY_signal(i-1) == 1
        Budget(i) = Budget(i-1)-Stocks(i-1)*closeprice(i-1);
    elseif SELL_signal(i-1) == 1
        Budget (i) = Budget(i-1)+Shares(i-1)*closeprice(i-1);
    else
        Budget(i) = Budget(i-1);
    end
    %Calculating Number of Shares
    if BUY_signal(i-1) == 1
        Shares(i) = Shares(i-1) + Stocks(i-1);
    elseif SELL_signal(i-1) == 1
        Shares(i) = 0;
```

```
    else
        Shares(i) =  Shares(i-1);
    end
    %Auxiliary values of Number of stocks
    Stocks(i) = round(Budget(i)/closeprice(i));
End


% Calculating when or on which dates this Algorithm Buys or Sells
BUY = array2table(BUY_signal);
SELL = array2table(SELL_signal);
DATA = [date BUY SELL];
DATA.BUY = BUY;
DATA.SELL = SELL;
Buydates = DATA(DATA.BUY_signal==1,:)
Selldates = DATA(DATA.SELL_signal==1,:)

%Calculating how much your algorithm buys or sells in each deal
Budget1 = Budget.';
Budget1(end,:) = [];
DATA1 = [Budget1 BUY_signal SELL_signal];
BuyAmount = DATA1(BUY_signal==1,:)
SellAmount = DATA1(SELL_signal==1,:)

%Calculate the Updated Budget which is equal to
Budget(126)+Shares(126)*closeprice(126) ,126 is last row
UpdatedBudget = Budget(126)+Shares(126)*closeprice(126)

%Profit is equal to UpdatedBudget - InitialBudget
Profit = UpdatedBudget - InitialBudget

% Calculate Percent profit
PercentProfit = (Profit/InitialBudget)* 100
```

Results/ Outputs for Task 1:

Date	BUY/SELL	Budget
26-08-2020	SELL	1000000
16-09-2020	BUY	1000000
28-10-2020	SELL	1992
17-12-2020	BUY	1066792
19-01-2021	SELL	-3178

Updated Budget = Budget(126)+ No. of Shares(126)*closeprice(126) = 1073094 GBP

The Total Profit of Algorithm = 73094 GBP

Percent Profit = 7.30%

Task 2:

Steps:

Created Random array with values from 1:200 with 40 elements

Sorted the Buyers valuation in ascending and Sellers cost in Descending

If Sellers(i)<=Buyers(i) , then p =((Sellers(i)+Buyers(i))/2) else No trade , ran simulation for 10 rounds

Code:

```
t0=0; %Beginning of time
dt = 1; %time step
T=10;%Number of Rounds

%Create a new random array of values between [1,40]
VC = randi([1 200],1,40)

%Initialize array for Buyers valuations and Sellers costs

Buyers =[];
Sellers =[];
probSwitch =0.5

%randomvalue = rand();
% for i=1:length(VC)
%     if randomvalue<probSwitch
%         isBuyer(i) = 0;
%         isSeller(i) =1;
%     else
%         isBuyer(i) = 1;
%         isSeller(i) = 0;
%     end
% end

Buyer = VC(1:20);
Seller = VC(21:end);

%Sorting the arrays (Buyers in ascending and Sellers in Descending)
Buyers = sort(Buyer)
Sellers = sort(Seller,'Descend')

%Running 10 rounds to check if there was a trade and to calculate price at
%which the items were purchased
for t=t0:dt:T
  for i=1:length(Sellers)
    if Sellers(i)<=Buyers(i)
```

```
        price =((Sellers(i)+Buyers(i))/2)
      else
        disp("No trade")
      end
    end
  end
```

Task 3:

(A) **Cournot duopoly model where the inverse demand function and the cost functions are as follows:**

$P = 120 – Q$

$c1 = 10 + 3q1$

$c2 = 12 + 6q2$

$Q = q1 + q2$

Calculating Profit Functions:

$\Pi1(q1, q2) = q1(120-(q1+q2)) – (10 + 3q1)$ ………………… (1)

$\Pi2(q1, q2) = q2(120-(q1+q2)) – (12 + 6q2)$ ………………… (2)

For Equation (1)

$\Pi1(q1, q2) = q1(120-(q1+q2)) – (10 + 3q1)$

$\Pi1(q1, q2) = 120q1 – (q1) ^2 – q1q2 -10 – 3q1$

Taking derivative w.r.t. q1

$120 – 2q1 -q2 -3 = 0$

$2q1 +q2 =117$ ………………. (3)

For Equation (2)

$\Pi2(q1, q2) = q2(120-(q1+q2)) – (12 + 6q2)$

$\Pi2(q1, q2) = 120q2 – q1q2 - (q2) ^2 -12 - 6q2$

Taking derivative w.r.t. q2

$120 – q1 – 2q2 -6 = 0$

$q 1 + 2q2 = 114$ ………………. (4)

To solve Equation (3) & (4)

Multiply Equation (3) with 2:

$4q1 + 2q2 = 234$ ………………… (5)

Subtracting Equation (5) by Equation (4)

$4q1 + 2q2 = 234$

$-q1 - 2q2 = -114$

$3q1 = 120$

$q1 = 40$

Substituting value of q1 in Equation (4)

$40 + 2q2 = 114$

$2q2 = 74$

$q2 = 37$

$Q = q1 + q2 = 40 + 37$

$Q = 77$

$P = 120 - Q$

$P = 120 - (q1 + q2)$

$P = 120 - (40 + 37)$

$P = 43$

Substituting values of q1 & q2 in Equation (1) & (2) Respectively

$\Pi 1 = (120*40) - (40)^2 - (40 * 37) - 10 - 3(40)$

$\Pi 1 = 1590$

$\Pi 2 = (120*37) - (40*37) - (37)^2 - 12 - 6(37)$

$\Pi 2 = 1357$

Consumer Surplus (CS)

$CS = [Q(120 - P)]/2$

$CS = [77(120 - 43)]/2$

$CS = 2964.5$

Total Surplus (TS)

$TS = CS + \pi$

$TS = 2964.5 + 1590 + 1357$

$TS = 5911.5$

Nash Equilibrium Quantities:

Quantity (Q)	Profit 1	Profit 2	Consumer Surplus (CS)	Total Surplus (TS)
77	**1590**	**1357**	**2964.5**	**5911.5**

(B) Leader-follower duopoly model with the inverse demand function and the cost functions and reaction function of firm 2 are as follows:

P = 120 – Q

c1 = 10 + 3q1

c2 = 12 + 6q2

r2(q1) = 57 – (q1/2)

Profit Functions:

Π1(q1, q2) = q1(120-(q1+q2)) – (10 + 3q1) ………………… (1)

Π2(q1, q2) = q2(120-(q1+q2)) – (12 + 6q2) ………………… (2)

Assuming Firm 1 moves and chooses q1 and after that Firm 2 moves and chooses q2.

Firm 1 chooses q1 to maximize:

Π1(q1, q2) = q1(120-(q1+q2)) – (10 + 3q1)

Π1(q1, q2) = q1(120 – q1 -q2) – 10 – 3q1

Reaction function of Firm 2

r2(q1) = 57 – (q1/2)

Replacing q2 with reaction function:

Π1(q1, q2) = q1(120 – q1 –(r2(q1)) – 10 – 3q1

Π1(q1, q2) = q1(120 – q1 – (57 – (q1/2))– 10 – 3q1

Π1(q1, q2) = 120q1 – (q1) ^2 – 57q1 – 0.5(q1) ^2 – 10 -3q1

Taking derivative w.r.t. q1

120 – 2q1 -57 -q1 -3 =0

3q1 = 60

$q_1 = 20$

Firm 2's response after Firm 1 moved:

$\Pi_2(q_1, q_2) = q_2(120-(q_1+q_2)) – (12 + 6q_2)$

$\Pi_2(q_1, q_2) = q_2(120-q_1-q_2) – 12 - 6q_2$

$\Pi_2(q_1, q_2) = 120q_2 - q_1q_2 – (q_2)^2 – 12 – 6q_2$

$120 - q_1 -2q_2 -6 =0$

$q_1 +2q_2 =114$ …………………… (3)

Substituting value of q1 in Equation (3)

$20 +2q_2 = 114$

$2q_2 = 94$

$q_2 = 47$

$Q = q_1 +q_2$

$Q = 67$

$P = 120 – Q$

$P = 120 – (q_1 + q_2)$

$P = 120 – (20+47)$

$P = 53$

Substituting values of q1 & q2 in Equation (1) & (2) Respectively

$\Pi_1 = (120*20) – (20)^2 – (20 * 47) – 10 – 3(20)$

$\Pi_1 = 2400 -400 -940 -10 -60$

$\Pi_1 = 990$

$\Pi_2 = (120*47) – (20*47) – (47)^2 -12 – 6(47)$

$\Pi_2 =5640 -940 -2209 -12 -282$

$\Pi_2 = 2197$

Consumer Surplus (CS)

$CS = [Q (120 – P)]/2$

$CS = [67(120 – 53)]/2$

$CS = 2244.5$

Total Surplus (TS)

TS = CS + π

TS = 2244.5 + 990 + 2197

TS = 5431.5

Nash Equilibrium Quantities:

Quantity (Q)	Profit 1	Profit 2	Consumer Surplus (CS)	Total Surplus (TS)
67	990	2197	2244.5	5431.5

Results:

	Quantity (Q)	Profit 1	Profit 2	Consumer Surplus (CS)	Total Surplus (TS)
Leader-Follower Duopoly model	67	990	2197	2244.5	5431.5
Cournot Duopoly model	77	1590	1357	2964.5	5911.5

Task 4: **FairLedger: A Fair Blockchain Protocol for Financial Institutions**

Summary:

Blockchains are regarded as trading technology of future for many industries. FairLedger is a permissioned blockchain BFT protocol for Financial Institutions, it ensures fairness and is designed to deal with rational behaviour that arises when financial institutions try to maximize their profit by maximizing rate of transactions as it is assumed they receive a fee for every transaction appended to ledger.[1] FairLedger is easy to understand and implement, it is safe under byzantine Failures and it detects and punishes deviating participants be it rational or byzantine. The FairLedger protocols includes detectable all-to-all (DA2A) that identifies deviating participants from communication protocol and can detect passive and active deviations. FairLedger's sequencing protocol works on Epochs and is key mechanism for fairness. This sequencing protocol is implemented in Hyperledger framework and is substituted for Iroha's Sumeragi in the Hyperledger framework. The Results show that FairLedger outperforms Iroha's byzantine fault-tolerant implementation in majority cases. FairLedger's sequencing protocol is also tested on PBFT Framework in a WAN Setting and it outperforms PBFT in failure-free normal mode and achieves slightly lower results in Alert mode.[1] Hyperledger protocols indicts correct participants, FairLedger never indicts correct participants It Identifies faulty components without accusing correct ones. FairLedger's protocol is novel and features the first byzantine fault-tolerant consensus engine to ensure fairness when all players are rational.

Critical Assessment:

The Assumptions made in this paper are as follows: Authors assume the model as Synchronous with coarse time bounds and use it to detect passive deviations and lack of progress. Authors assume that the financial entity receives fee or benefit for every transaction it appends to the shared ledger, and so entities are motivated to append as many transactions as possible and to reason such rational behaviour it is assumed that each entity can be either byzantine or rational. It is assumed byzantine entities can collude, but rational ones do not collude and that players do not deviate unless they benefit from doing so, also it is assumed byzantine failures are rare. It is assumed fixed ratio players prefer long ledgers and Players have a trusted Certification known to all players and unique pair of public and private keys in reliable communication channels. Authors also assume that there is a known upper bound on message latency as for implementation of DA2A Protocol as it is required.

Is the Methodology and Assumptions appropriate? The methodology of the FairLedger protocol is to define detectable-to-all protocol that identifies deviating participants. The FairLedger's sequencing protocol is substituted in Hyperledger framework's Iroha and FairLedger outperforms Iroha's BFT implementation in most cases. Also, FairLedger's sequencing protocol outperforms PBFT in normal mode. Since it is assumed rational players don't collude, it is impossible for a rational player to increase their ratio by passive deviation and since, it is assumed fixed ratio players prefer long ledgers sending protocol messages fast is an equilibrium. The Methodology does prevent Rational players from deviating and ensures fairness and outperforms the other frameworks in most cases and the Assumptions made by Authors supports the methodology. Hence, we can conclude Yes, the methodology and assumptions discussed in this paper are appropriate with respect to problem

How does Computational thinking help us analyse this situation? The Problem is decomposed to ensure fairness and safety to the participants of the permissioned blockchain system and pattern is recognized wherein participants tend to deviate to maximize profits. Hence, an Algorithm is developed with novel protocols for FairLedger to overcome this problem which is also easy to understand and implement.

Adaptation of Model: Financial Institutions and other major industries wherein permissioned blockchains are used and deployed over a secure WAN and have certified participants this model can be deployed to prevent the entities rational behaviour to maximize profit thus, ensuring Fairness.

References:

1. Lev-Ari, Kfir, et al. "Fairledger: A fair blockchain protocol for financial institutions." arXiv preprint arXiv:1906.03819 (2019).