Project Report IITM

Fruits and Vegetables Supply Chain

Submitted by

Siddharth Thakur (Roll Number: EE14B054) and

Abhishek Avhad (Roll Number: EE14B015)

in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY
in
ELECTRICAL ENGINEERING

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS

MAY 2018

REPORT CERTIFICATE

This is to certify that the report titled **Fruits and Vegetables Supply Chain**, submitted by **Siddharth Thakur** along with **Abhishek Avhad** to the Indian Institute of Technology Madras, for the award of the degree of **Bachelor of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Mrs. Usha Mohan,	**Dr. Puduru Viswanadha Reddy**,
Project Guide,	Project Co-Guide,
Assistant Professor,	Assistant Professor,
Department of Management Studies,	Department of Electrical Engineering,
IIT Madras - 600036	IIT Madras - 600036

Place: Chennai,
Date: 13th May, 2018

Acknowledgement

I would like to thank our project guide Mrs. Usha Mohan for providing us with an amazing opportunity to work on this project. She consistently guided and supported us on our venture throughout and also allowed us to work on our own timelines at our own pace. I would like to thank our project co-guide Dr. Puduru Viswanadha Reddy for essential support.

A huge vote of thanks to Waycool Enterprises to conduct a project along with them. I would especially like to thank Mr.Karthik Jayaraman, Mr Vignesh Kumar Manogaran, Mr. Vignesh Kumar Manogaran and Mr. Subu Akkulan from Waycool from providing us with essential information, cooperating and showing s the ropes on how to start and approach the project.

Finally, I would also take this opportunity to thank all my friends who have been a great source of motivation and encouragement for this project.

Place: Chennai,
Date: 13th May, 2018

Introduction:

India is mainly an agrarian economy. Agriculture contributes 16 percent of India's GDP. The supply chain originating from the farmer to the final consumer is extremely fragmented with huge inefficiencies in between. Agricultural products have traditionally moved from farmer to consumer through a series of intermediaries, who manage information, physical and financial movement through it, in both directions. While this supply chain has been seen effective in moving produce, it also suffers from high wastage and inefficiencies. Over 40% of fresh produce is believed to be wasted or suffers value diminution before it reaches the consumer. Fruits and vegetables cost four to ten times more at the retail outlet, compared to the farmgate. Food is moved and stored in less-than-hygienic conditions with quality degrading over time.

One another disadvantage is that often only a small percentage of profits of the produce going directly to the farmer. The farmers lose out a huge part of their revenue as the other constituents of this supply chain who are closer to the delivery and payment end of the supply chain often are the ones to take hue chunks of their profits. Also, in case of accidents and wastage, the farmers aren't repaid in full as the other constituents of supply chain also try to cover their loss. The inefficiencies lead to a higher price, decrease in quality which leads to decreasing popularity of Indian domestic produce in favor of industrial complements, which again harm the Indian farmers.

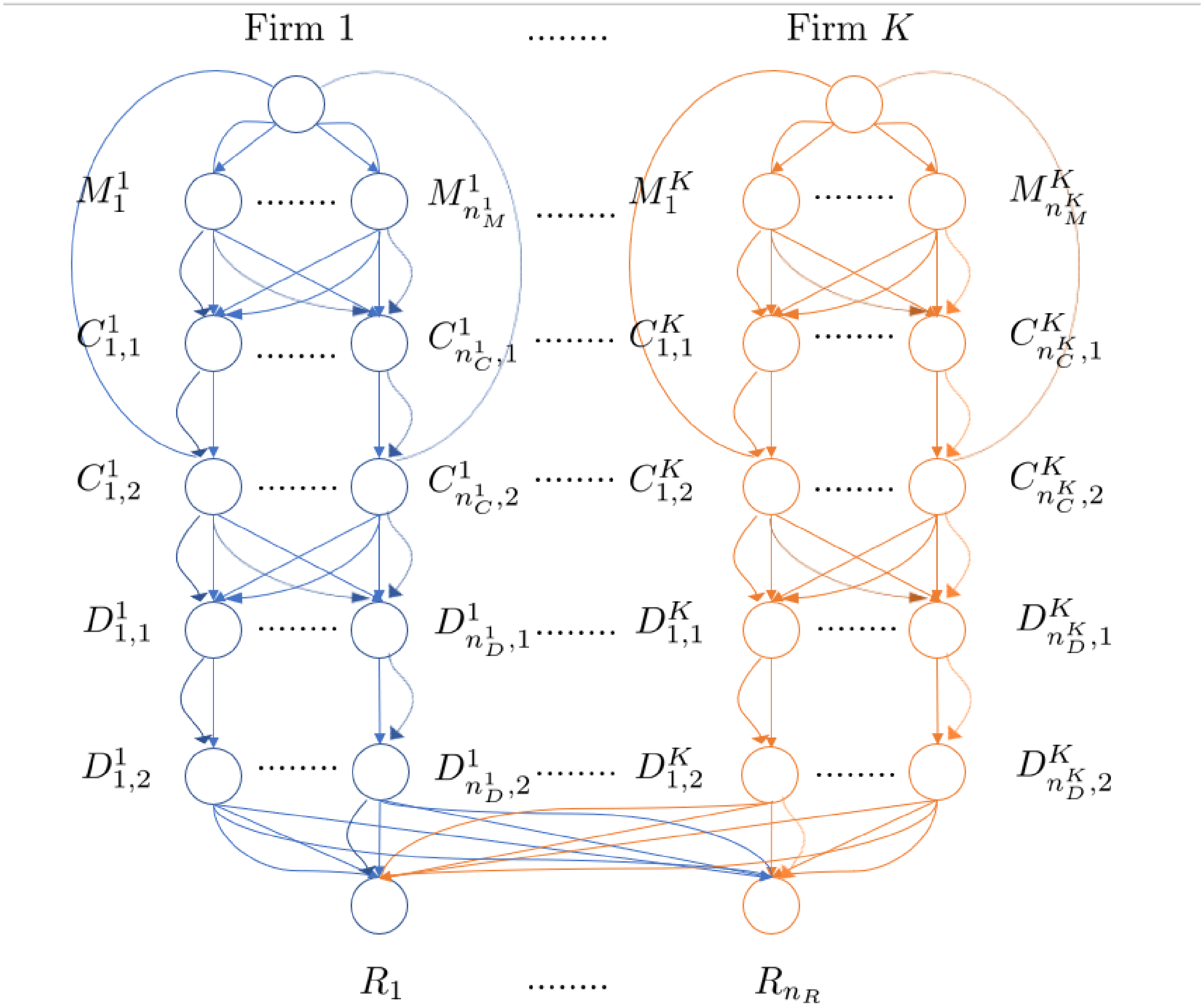
Waycool is a company that is attempting to solve this problem by controlling the entire supply chain and removing all such wastage using state of the art technology, like IT, Analytics and IOT. Our project consisted of looking at their model, understanding and trying to optimise it. We looked at their retail business chain, **Sunny Bee** and tried to analyze and provide inputs to their process. Also since the dataset is huge, spanning over a huge number of products and multiple stores, some optimizations had to be made to read and analyze the data meaningfully.

Theory Behind Model Optimization:

The fresh produce supply chain network oligopoly model is distinct from other product supply chains in the following aspects:

- It captures the deterioration of fresh food along the entire supply chain from a network perspective.
- It handles the exponential time decay through the introduction of arc multipliers.
- It captures oligopolistic competition with product differentiation.
- It includes the disposal of the spoiled food products, along with the associated costs.
- It allows for the assessment of alternative technologies for each supply chain activity.

Let us now consider a model as depicted in the following graph:



In this section, we consider a food supply chain network with a finite number of *K* food firms. The food supply chain network activities include **production, processing, storage, distribution**, and the **disposal** of the food. The firms will typically be vertically integrated, which, as a strategy, has become increasingly important as food systems become more consumer-driven. The firms are said to behave non- cooperatively, each looking out for its own best interests. We also take into consideration the product differentiation in the food market – i.e. same product may differ in quality, freshness (based on the transportation and stock parameters), the safety concerns and specific parameters required by different forms of business operating in the food market

The graph can be understood by supply chain networks of their economic activities in the following manner:

- There are total K number of firms ranging from 1 to K. Here only 2 firms are represented.
- M denotes the *production facilities*, C denotes the *processing facilities*, D denotes the *distribution centers* and R denotes the *demand markets*.

- Each firm i, has n_M^i production facilities, n_C^i processing facilities and n_D^i distribution centers. There is a total of n_R demand markets, where the j'th market is denoted by R_j.

This was the graph distribution for K firms. Now we loo at each firm and understand their supply chain as follows:

- The first set of links connecting the top two tiers of nodes corresponds to the food production at each of the production units of firm *I*. Production may involve seasonal operations such as soil agitation, sowing, pest control, nutrient and water management, and harvesting. The *multiple possible links* connecting each top tier node *I* with its production facilities, $M_1^I, M_2^I \dots \dots M_{n_M^{\mathrm{I}}}^{\mathrm{I}}$ represent alternative production technologies that may be associated with a given facility.

- The second set of links from the production facility nodes is connected to the processors of each firm *I* which are denoted by $C_{1,1}^I, \dots \dots, C_{n_C^{\mathrm{I}},1}^I$. These links correspond to the shipment links between the production units and the processors. The multiple shipment links denote different possible modes of transportation, characterized by varying time durations and environmental conditions.

- The third set of links connecting nodes $C_{1,1}^I, \dots \dots, C_{n_C^{\mathrm{I}},1}^I$ to $C_{1,2}^I, \dots \dots, C_{n_C^{\mathrm{I}},2}^I$ denotes the processing of fresh produce. The major food processing activities include cleaning, sorting, labeling, and simple packaging. Different processing activities and technologies may result in various levels of quality degradation.

- The next set of nodes represents the distribution centers. Thus, the fourth set of links connecting the processor nodes to the distribution centers is the set of shipment links. Such distribution nodes associated with firm *I* are denoted by $\mathrm{D}_{1,1}^I, \dots \dots, \mathrm{D}_{n_{\mathrm{D}}^{\mathrm{I}},1}^I$ .There are also multiple shipment links to capture different modes of transportation.

- The fifth set of links, which connects nodes $\mathrm{D}_{1,1}^I, \dots \dots, \mathrm{D}_{n_{\mathrm{D}}^{\mathrm{I}},1}^I$ to $\mathrm{D}_{1,2}^I, \dots \dots, \mathrm{D}_{n_{\mathrm{D}}^{\mathrm{I}},2}^I$, are the storage links. Since fresh produce items may require alternative storage conditions, we represent these alternative conditions through multiple links at this tier.

- The last set of links connecting the two bottom tiers of the supply chain network corresponds to the distribution links over which the fresh produce items are shipped from the distribution centers to the demand markets. Here we also allow for multiple modes of transportation.

- In addition, the curved links in the above figure joining the top-tiered nodes *I* with the processors, which are denoted by $C_{1,2}^I, \dots \dots, C_{n_C^{\mathrm{I}},2}^I$ capture the possibility of on-site production and processing.

- We denote N as the set of all nodes and L as the set of all links for the entire graph.

We will now solve for this model. The perishable nature of foods is captured as an exponential time decay to show either a decrease in quality, quantity or both. If we assume degradation in quality such that all the products deteriorate at the same rate simultaneously, it will not be completely correct as fresh produce all vary at different rates. This will be a better approximation for meat, dairy and bakery products. We thus assume that the quantity deteriorates of fresh produce. All post production activities will deal with exponential time decay. Food products deteriorate over time even under optimal conditions. We assume that the temperature and other environmental conditions associated with each postproduction activity/link are known and fixed. Given N_0 as the initial quantity of a good at time 0, the expected quantity surviving at time t is

$$N(t) = N_0 e^{-\alpha t}$$

The arc multiplier α_a for any postproduction link a is given as

$$\alpha_a = e^{-\lambda_a t_a}$$

Where λ_a and t_a are the decay constant and the time duration fixed associated with a given link a. They represent decay happening during and the time that is taken for the good to be processed at link a. Both values are fixed. We now define f_a as the initial quantity of product *flowing through a link a*. Let f_a' be the final quantity that is obtained at the link, the quantity that reaches the successor node of a. It's given by:

$$f_a' = \alpha_a f_a \qquad \forall a \in \mathrm{L}$$

Hence the number of units that got wasted on a link a is given as

$$f_a - f_a' = (1 - \alpha_a) f_a \qquad \forall a \in L$$

The wastage for a link a is a function of f_a. Thus, the total discarding cost function $\hat{z}_a$ for a path a is given as

$$\hat{z}_a = \hat{z}_a(f_a) \qquad \forall a \in L$$

Let P_k^i be the set of all paths that from origin node of firm i to destination node R_k. Let x_p denote the path flow of product on path p. Then we have,

$$x_p \geq 0, \forall p \in P_k^i\,;\ i = 1,2, \dots \dots, K; k = 1,2, \dots . . n_R$$

Since the path flow cannot be negative. Now we can establish the relation between the link flow f_a and path flows which can be expressed as,

$$f_a = \sum_{i=1}^{K} \sum_{\mathrm{k}=1}^{n_R} \sum_{p \in P_k^i} x_p\, \alpha_{ap} \qquad \forall a \in L$$

Let μ_p denotes the multiplier for path p, which can contain any number of links. It is defined as the product of all link multipliers of that path. Hence,

$$\mu_p = \prod_{a \in p} \alpha_a \qquad \forall p \in P_k^i; i = 1,2, \dots \dots, K; k = 1,2, \dots . ., n_R$$

The demand for the i'th firm for the demand market R_k is denoted by d_{ik}. The expression for $d_{\mathrm{i}k}$

Can be written as:

$$d_{ik} = \sum_{p \in P_k^i} x_p \, \mu_p, \qquad i = 1,2, \dots . K; k = 1,2, \dots . . n_R$$

We denote the demand price associated with food firm I's product at demand market R_k by ρ_{ik}. We denote $\hat{c}_a$ as the operational cost on a link a. We assume that it is a function of product flows through that link (which it generally is). That is,

$$\hat{c}_a = \hat{c}_a(f) \qquad \forall a \in L$$

The profit function of a food firm is defined as the difference between its revenue and its total costs (operational and discarding). Each firm *i* seeks to maximize its profit. The statement of the maximization of profits for firm *i*, in link flows, is

Maximize

$$\sum_{k=1}^{n_R} \rho_{ik}(d) d_{ik} - \sum_{a \in L^i} \left(\hat{c}_a(f) + \hat{z}_a(f_a) \right)$$

Which can be extended to write

Maximize

$$\sum_{k=1}^{n_R} \rho_{ik}(d) \sum_{p \in P_k^i} x_p \, \mu_p - \sum_{a \in L^i} \left(\hat{\mathrm{C}}_{\mathrm{p}}(\mathrm{x}) + \hat{\mathrm{Z}}_{\mathrm{p}}(\mathrm{x}) \right)$$

1.Quantity Prediction-

Objective-

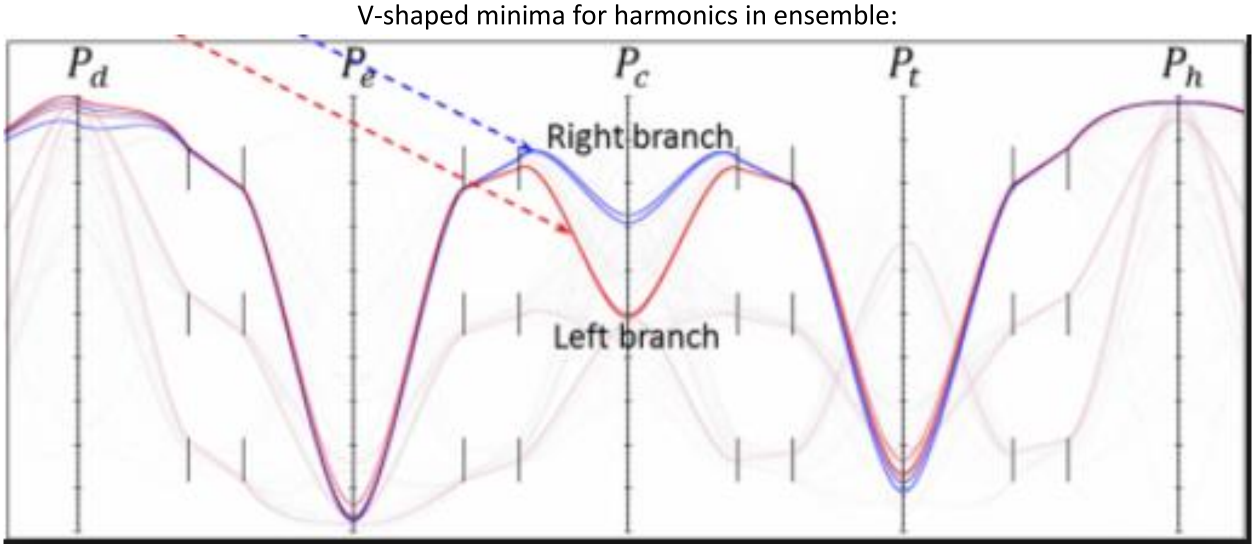
For any analysis of a food supply chain we need data for the entire supply chain so that the entire dead weight cost of the product can be accounted for and no product is left out of the calculation. Food & Vegetables supply chain is mostly composed of individual stakeholders and this may pose a challenge when finding out the competitors for the selected complete chain company. Game Theory analysis is restricted due to the sheer number of competitors and unknown price differences.

For a food supply chain analysis the foremost step lies in demand forecasting at the lowest levels of the supply chain. Next objective would be to assign weights to each path and optimize the flow by finding the minimum.

Theory-

Demand Forecasting is approached by division of ensemble into groups based on buying frequency. We know that majority of the consumers follow similar buying patterns which leads to our first assumption that for each product, there exists a dominant major buying frequency and the base of consumers within one buying time period will repeat. What this means is that if we have an ensemble of four, then most of the consumer base will repeat only once for every four days. There may be trend and seasonalities which we may have to discount for to obtain this buying frequency. After fitting a pattern and iterating over all such frequencies for seeing the best pattern, we find that the standard deviation of such model would be lowest and we would get similar lows in standard deviation for the harmonics of the buying frequency. Our goal is to capture the first lowest frequency dip and check for harmonics. If the standard deviation for one day lower, current and one day higher frequency is a V shaped minimum and this minima repeats on multiples then this drop in standard deviation is our buying frequency.

V-shaped minima for harmonics in ensemble:



Code instance:

We will now show a few instances of code to demonstrate how this is achieved:

```
#Total_Volume<-aggregate(data$Qty ~ data$Item.Name,ndf, sum)

DDay<-as.numeric(substr(data$Bill.Date,1,2))
DMonth<-as.numeric(substr(data$Bill.Date,4,5))
DYear<-as.numeric(substr(data$Bill.Date,9,10))
V<-DDay+DMonth*30+(DYear-17)*12*30
```

In here we have the basic variable initialization where we extract the time where each item was sold across different stores. We form a variable *V* which will help us form the timeline. Simply it gives us the number of days after which an item was sold.

```
#Since 1 Jan 2009
V<-max(V,na.rm=TRUE)-V
data<-cbind(data,V)
UI<- cbind(unique(data$Item.Name),14,0,0)
print(UI)
vpicker <- function(i)
{
return(subset(data[which(data$Item.Name==UI[i]),],select=c("Qty","Item.Rate","V")))
}
```

This age is considering Jan 2009 as the 'zero' of our timeline(the first time an item is sold) and we invert it to get our parameter of recency. Next, we add this to our main data set called 'data' and filter this to create a set of unique items which will be printed in the log output. *vpicker* function is used to row search all the items belonging to the input(name) from the history of bills so that an individual analysis is possible.

```
for(mi in 1:nrow(UI))
{
  UI[mi,3]<-median(vpicker(mi)[,2])
  UI[mi,4]<-median(vpicker(mi)[,1])
}
```

mi will pick some item and we find the corresponding median entries of price and recency for default entries.

```
sdcomp <- function(a,n)
  {
  #a=subset(data[which(data$Item.Name==UI[2]),],select=c("Qty","Item.Rate","V"))
  #n=5
  #print(n)
  a[,3]<-a[,3]/n
  glm.fit1 <- glm(Qty ~ Item.Rate +as.integer(a[,3]),data=a)
  a[,3]<-as.integer(a[,3]*n)
  #coef(glm.fit1)
  #plot(density(resid(glm.fit1)))
  #qqnorm(resid(glm.fit1))
  #summary(glm.fit1)
  sd(resid(glm.fit1))
  return(sd(resid(glm.fit1)))
  }
```

sdcomp function is used to compute the standard deviation of each items quantity when grouped with a frequency of 'n'. Refer to theory for the ensemble adjustments. We are iterating for a frequency from 1 to 30(assuming 30 as the average days in a month) for finding the ideal grouping frequency

```
fpicker<-function(g)
{
dsn=1
  for(sn in 1:30)
    {
      if(sdcomp(g,sn+1)<sdcomp(g,sn)&&sdcomp(g,sn+2)>sdcomp(g,sn+1)&&(nrow(g)>30))
        {
        dsn=sn+1
        break
        }
    }
#print(dsn)
  return(dsn)
}
```

fpicker function lets us pick this ideal frequency by finding the first ideal frequency, that is to say that there will be harmonics of this frequency which will also give the same minima. Here we find the first drop and rise in standard deviation. We are also setting a minimum quantity constraint of once in a month so that we can ignore outdated products

```
for(h in 1:nrow(UI))
{
  print(h)
  UI[h,2]<-fpicker(vpicker(h))
  print(UI[h,2])
}
```

```
write.csv(UI[which(UI[,2]!=1),],file="Unique.csv")
FUI <- read.table("Unique.csv",header=TRUE, sep = ",",stringsAsFactors = F)
```

We append this corrected frequency and store it in a csv file as this table gives us a saving point. In a sense that any manual changes can be incorporated here into this table and carried forward into the next part which is just regression.

```
finder<-function(inn)
{
  #inn=1
  sname<-FUI[inn,2]
  sdata=subset(data[which(data$Item.Name==sname),],select=c("Qty","Item.Rate","V"))
  sdata <-  subset(sdata,complete.cases(sdata))
  sdata[,3]<-as.numeric(sdata[,3])/as.numeric(FUI[inn,3])
  glm.fit2 <- glm(Qty ~ Item.Rate +as.integer(sdata[,3]),data=sdata)
  #sdata[,3]<-as.integer(sdata[,3]*n)
  #coef(glm.fit2)
  FQty<-0
  for(inni in 1:FUI[inn,3])
  {
    FQty<-FQty+coef(glm.fit2)[1]+coef(glm.fit2)[2]*FUI[inn,4]+coef(glm.fit2)[3]*(-inni)
  }
  return(FQty)
}
for(fi in 1:nrow(FUI))
{
  FUI[fi,5]<-finder(fi)
}
write.csv(FUI[which(FUI[,5]>0),],file="Final.csv")
```

We regress with the corrected frequency and manual entries of current price incorporated in the previously exported csv file to find the pattern. It is easy to see that the market is extremely price sensitive. Any more factors for regression can be incorporated here and they need not require the previous code. The previous part was to take care of the delay effect. Finally, we output our results to a csv file which will give us the quantity forecast for the given timeframe for the current price.

Conclusion:

Once we obtain the buying frequency for each item we can predict the quantities sold within one time period and also prescribe a harmonic of this as a stocking frequency. With this given the price of a good and a time period duration, one can predict the quantity that will be sold in that time period. Now we can predict in how many number of days does an average person buy a particular product and the total quantity of that product sold over that many number of days based on price.

2. Quantity correlation of goods and Economic Indicator-

Objective -

We would want to know what goods are sold together more on a relative scale and over what time frame: which part of year results in goods being complement to each other. We can come up then with a selling strategy based on time frame and get general views of buyer's psychology during different times of year. It is also important as we can source them differently and price matching need not be done for customer acquisition.

We also find the economic index of various places where the retail stores are located based on prices weighted over each good based on the amount of quantity for which they were sold. Based on stores location, we can get approximate economic indicators for a city.

Theory-

We find the correlation of quantity of all goods vs each other by using correlation matrix scaled over time periods. We also give each store an index, which is found by the sum of weighted price compared to other stores for each good, with each good being weighted by the quantity of that good sold.

Code instance for Quantity Coorelation:

```
#Sorting data by date
datasbd <- data[order(as.Date(data$Bill.Date,format = "%d-%m-%Y"),decreasing=FALSE),]

#Getting the dates and ordering them from increasing to decreasing
Dday<-as.Date(data$Bill.Date,format = "%d-%m-%Y")
Dday <- sort(Dday,FALSE)
Dday <- unique(Dday)

#Getting the unique values
itemnames <- unique(data$Item.Name)
outletnames <- unique(data$Outlet.Name)
customernames <- unique(data$User.Name)
```

Here we read the data from the table and extract information about dates, outlets and customer names.

```
#Defining an array of appropraite size
arr2 = array(0.0, c(NROW(Dday),NROW(itemnames)))

#Setting the initial and final date
tinitial = as.Date("28-07-2017",format = "%d-%m-%Y")
tfinal = as.Date("28-08-2017",format = "%d-%m-%Y")
```

We define an initial and final date from where we want our dataset to be obtained from. We set the range in which we want the correlation to be seen.

```
#matching where all places the date are same
dindinit = match(tinitial,as.Date(datasbd$Bill.Date,format = "%d-%m-%Y"))
dfinadinit = match(tfinal,as.Date(datasbd$Bill.Date,format = "%d-%m-%Y"))
datasbd = datasbd[dindinit:dfinadinit,]
dateindex =match(Dday,as.Date(datasbd$Bill.Date,format = "%d-%m-%Y"))
```

Since the amount of data is huge, here I have sorted the table by date and will get the quantity in sections of dates. This reduces the search time by a lot as now the program need not search the entire table and only a small subset of table.

```
#Getting all quantitities
for (i in 1:NROW(dateindex){
  datasbod = datasbd[dateindex[i]:(dateindex[i+1]-1),]
  for(j in 1:NROW(itemnames)){
    p = subset(datasbod[which(datasbod$Item.Name==itemnames[j]),],select=c("Qty"))
    p = mean(p[,1])*NROW(p)
    arr2[i,j] <- p
  }
}
```

Here we go in the given date range and sum up the quantities of each product.

```
#Finding coorelation, giving 0 to NA and writing to csv
arr2[is.na(arr2)] <- 0

corqt = cor(arr2)

corqt[is.na(corqt)] <- 0

rownames(arr2) <- as.character((Dday))
colnames(arr2) <- itemnames

colnames(corqt) <- itemnames
rownames(corqt) <- itemnames

write.csv(arr2,file="Quantityovertime.csv")
write.csv(corqt,file="Coorelation.csv")
```

Now we use the quantity matrix obtained to find the correlation of items over the given time period. We can do this over select stores also.

Code Instance for Economic Indicators:

```
#Getting the unique values
itemnames <- unique(dataw$Item.Name)
outletnames <- unique(dataw$Outlet.Name)
outletnames <- sort(outletnames,FALSE)

#Ordering data by ward names and finding the corresponding index in the table to make
computation easier
datasbw <- dataw[order(dataw$Outlet.Name,decreasing=FALSE),]
wardindex =match(outletnames,datasbw$Outlet.Name)
```

Getting the relevant data and ordering the date by ward to reduce search time to reduce computation.

```
#Defining an array of appropraite size
arrivw = array(0.0, c(NROW(itemnames),NROW(outletnames)))
qtyw = array(0.0, c(NROW(itemnames),NROW(outletnames)))
qty = 0
eci = array(0.0,c(1,NROW(outletnames)))
qmult = 0
pmult = 0
netq = 0

#Getting quantity vs wards
for (i in 1:NROW(outletnames)){
  datasbow = datasbw[wardindex[i]:(wardindex[i+1]-1),]
  for(j in 1:NROW(itemnames)){
    p
subset(datasbow[which(datasbow$Item.Name==itemnames[j]),],select=c("Qty","Item.Rate"))
    qty = mean(p[,1])*NROW(p)
    qtyw[j,i] <- qty
    wa = mean(p[,1]*p[,2])*NROW(p)
    arrivw[j,i] <- wa/qty
  }
}
```

First we have the variable declaration for later use. Then we get the quantity of each item for each ward. This is stored in the array *arrivw.*

```
#Economically classifying by quantity

qtyw[is.na(qtyw)] <- 0
arrivw[is.na(arrivw)] <- 0
netq = sum(colSums(qtyw))
eci = array(0.0,c(1,NROW(outletnames)
```

We define a few variables to be used later for calculation of economic indicators. *netq* is the net quantity of all products over all wards. *eci* is a numerical array for indexing different wards.

```
for(i in 1:NROW(itemnames)){
  qmult = sum(qtyw[i,])/netq
  for(j in 1:NROW(outletnames)){
    pmult = arrivw[i,j]/sum(arrivw[i,])
    z = pmult*qmult
    if(is.na(z)){
      break
    }
    eci[1,j] = eci[1,j] + z
  }
}

#Arranging by economic indicator
eci = order(eci)
eci = 12 - eci
```

Now we assign numerical identities to each store. We do this by taking each product, giving it a weight called *qmult.* Then for that product, we weight each store by the weighted price of that product, called *pmult* and then assign the product of both these quantities for that store. We do this over all products for each store and then sum these quantities to get a numerical indicator for each tore. We then order the store from ascending to descending by ordering these indicators.

Conclusion-

How goods are correlated across different time periods and how important are these patterns especially during festive season. Also, we are able to come up wit a rough economic indicator for each store. By correlating the prices of all products to economic indicators, we can partition the city into geographical zones (as per the city administrative board) and check if the economic status of that zone can be predicted. Also we can suggest new places where the nest store can be built for more revenue.

3. Optimizing the supply chain-

Theory-

Once we assign weights like losses and costs to each path, we find these quantities as a function of the amount of product flow. After mapping all practical ERP variables to theoretical variables we can write a function for four factors and take them into account as a total function. Now we just take derivatives to find the optimal flow amounts.

The total cost minimization objective faced by the organization includes the total cost of operating the various links, the total discarding cost of waste/loss over the links, and the expected total product supply shortage cost as well as the total discarding cost of outdated product at the demand points.

Theory-

We have quantity for each line and the costs associated with it. We know the entire supply chain losses and analysis of such costs becomes easier.

Challenges-

Priorities like no understocking distort the model to a simple version of demand forecasting and overstocking. This can be helped to a extent wherein the government buys the waste and sells it as manure to farmers.

Very few companies own the entire supply chain and hence it is difficult to get an exact idea as to how its competitors are sourcing. It is even more difficult to do a flow series analysis where the links are unknown that is beyond a intermediaries nearest sources.

Integration of many variables like predictor variables is yet to be done and categorization of these goods is yet to be implemented due to increasing operational costs.

References-

1) Springerbriefs in Optimization :Networks Against Time Supply Chain Analytics for Perishable Products
2) Quality in competitive fresh produce supply chains with application to farmers' markets by Deniz Besik , Anna Nagurney
3) Operational research models applied to the fresh fruit supply chain by Wladimir E. Soto-Silva, Esteve Nadal-Roig, Marcela C. González-Araya, Lluis M. Pla-Aragones.