Pricing Strategies for Iron Mountain Incorporated

by Xiaodan Ding, Hongyuan Zhu, Dahyun Kim, Ruxin Cheng, Huiwen Ge
02/02/2020

Methodology

Factors Related to Price Variance

- Market (market name)
- Country (US & CA)
- Product Type (service, storage & transportation)

Optimal Price of the Product

- Regression model to find relationship between revenue and price
- Child account size classification (small, medium & large)
- Calculating the optimal price of different products in each market and different child account sizes

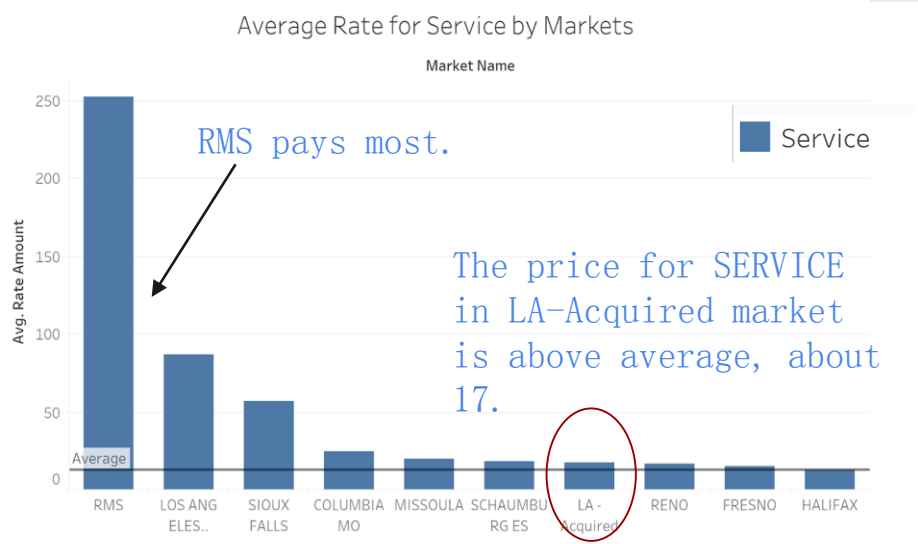
Recommendation

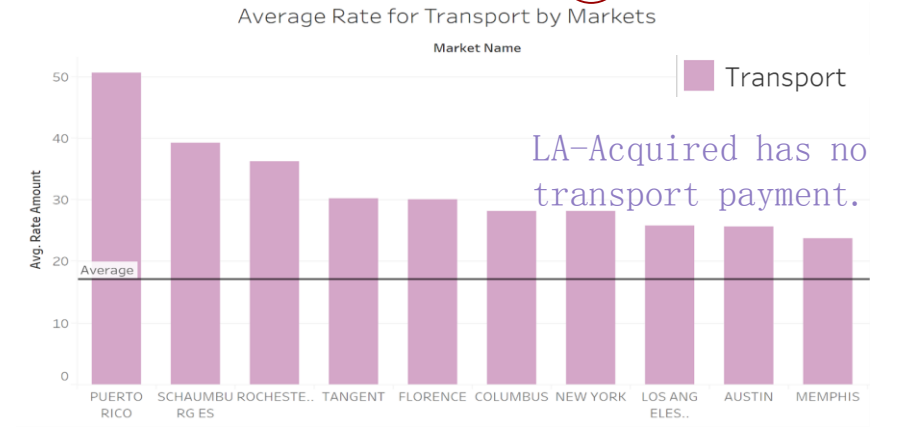
- Assessment of the current price
- Pricing strategy for future prices

Highlights

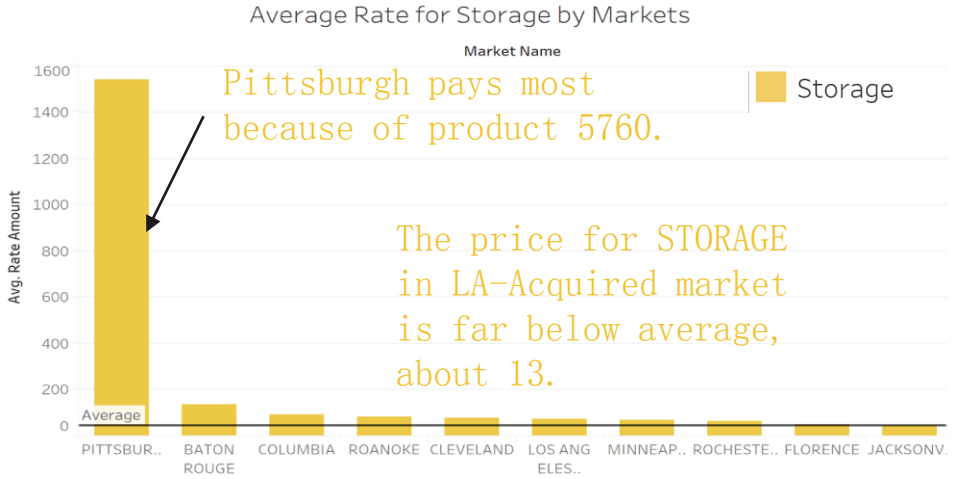
- Recommendation:We suggest that IRM should take different pricing strategy in different product types based on both geographical features and account size of customers, which can be divided into three categories (Small, Median and Large)
- Transport: There exists a common phenomenon that IRM is upselling their product in spite of customer account size.
- Storage: No need to change a lot for large account. But IRM should reset the pricing strategy for small and median account since half of markets are upselling and half of them are setting prices too low.
- Service: IRM should lower prices for most of the markets.

Price Variance by Markets and Bill Code Type

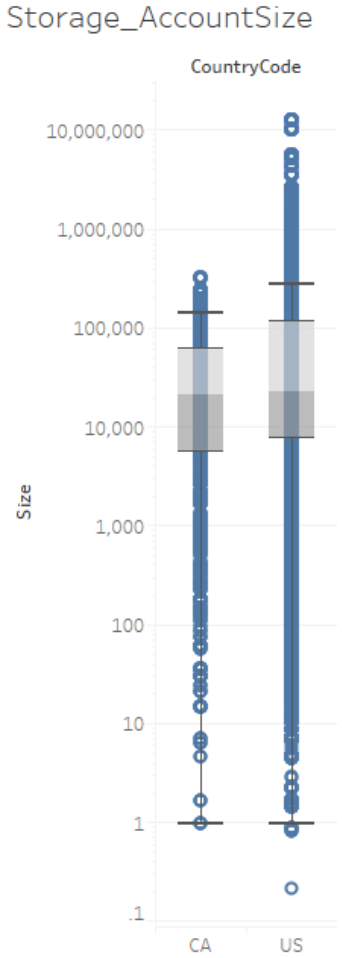


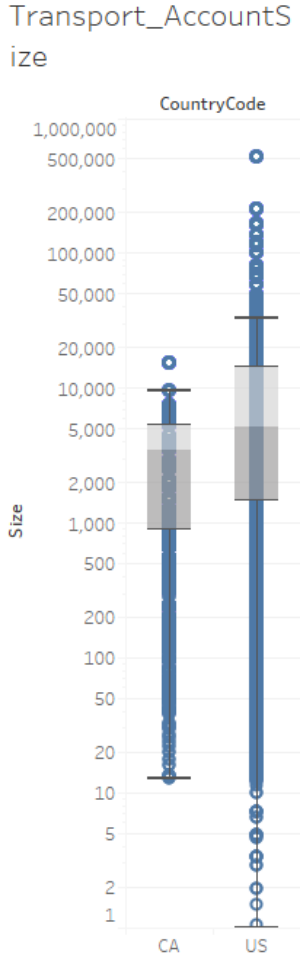


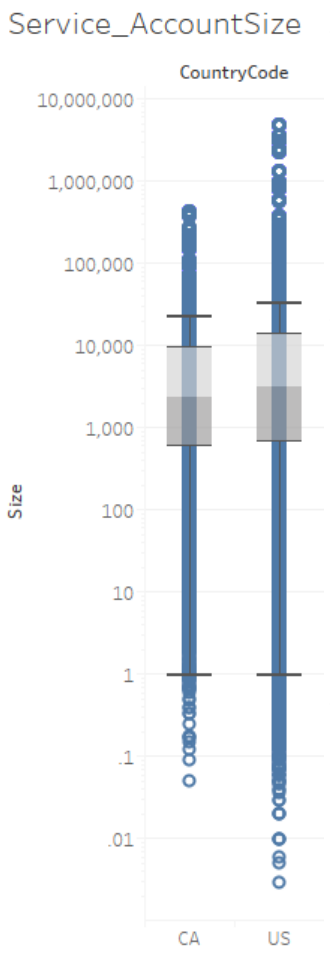
- The 3 bill code types are ***Storage, Service, and Transport.***
- The average amount rate differs by ***locations and bill code types.***
- Some markets have only one type of consumption.



Data Description: Account size







- Three Box and Whisker Plots:
  Each plot has two countries (US & CA);
  Size represents the sum of InvoiceRevenue of every single child account;
  There are three types of product: Storage, Transport and Service.
- Three level of size:
  US: Small: InvoiceRevenue <= 681
  Medium: 681 < InvoiceRevenue < 13928
  Large: InvoiceRevenue >= 13928
  CA: Small: InvoiceRevenue <= 594.6
  Medium: 594.6I<nvoiceRevenue<9464.6
  Large: InvoiceRevenue >= 9464.6
- The median size of CA is slightly less than the US in each plot but CA accounts for only 4.3% of the total InvoiceRevenue, which means that there may exist some problems in the current price

Calculating the Optimal Price of the Product

Assumption

1、Since we are not given cost information, we assume that there is little cost for the service.

2、The number of transaction is equal to the number of rows of the dataset, which means one row represents one transaction.

Only the revenue is taken into account when calculating the optimal price.

The optimal price (RateAmount) of each transaction can be obtained by maximizing the profit(revenue).

Calculation Process

Regression Model

- Get the relationship between InvoiceQuantity and RateAmount

Optimal Price of Each Transaction

- Use the regression and the equation Revenue = InvoiceQuantity * RateAmount
- Calculate the optimal price that maximized the revenue

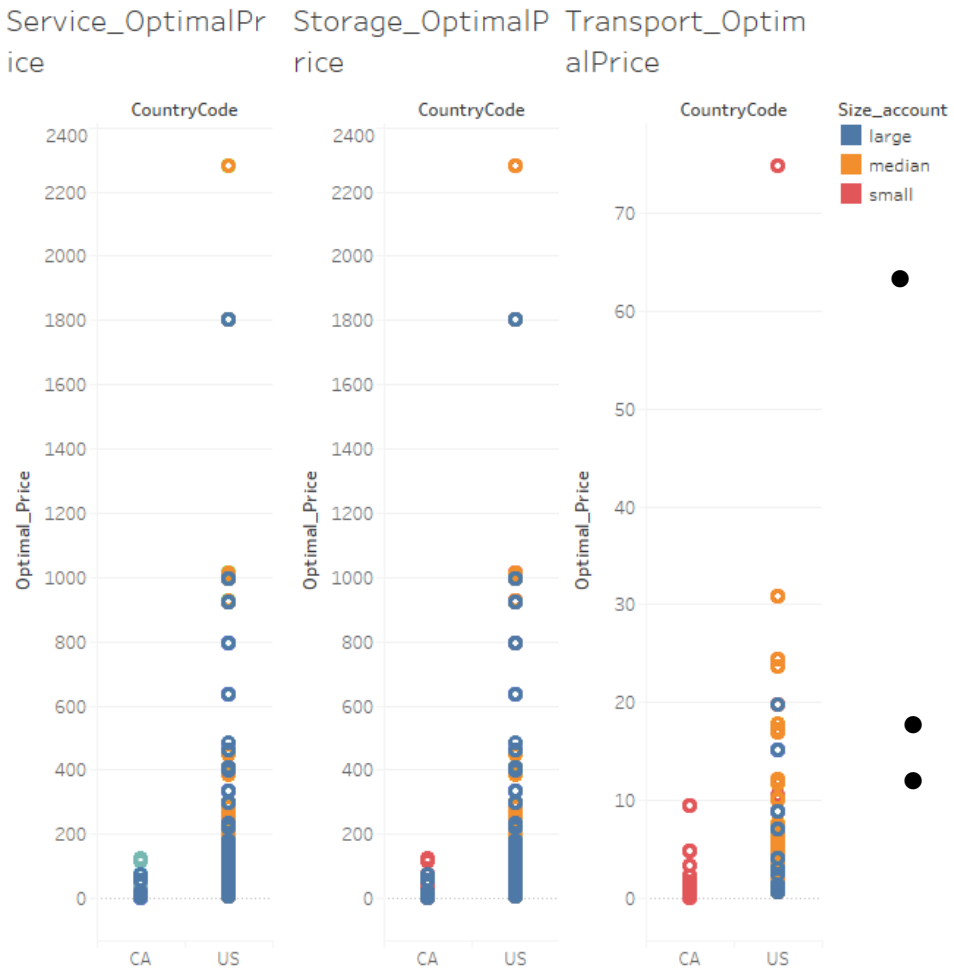
Account Size Classification

- Child accounts are classified into three levels according to summation of InvoiceRevenue of each child account: small, medium and large size

Optimal Price of Each Market in Each Level of Account Size

- **Under each level of account size, the optimal price of each market is obtained by the average of optimal prices in each market in the second step**

Optimal Price

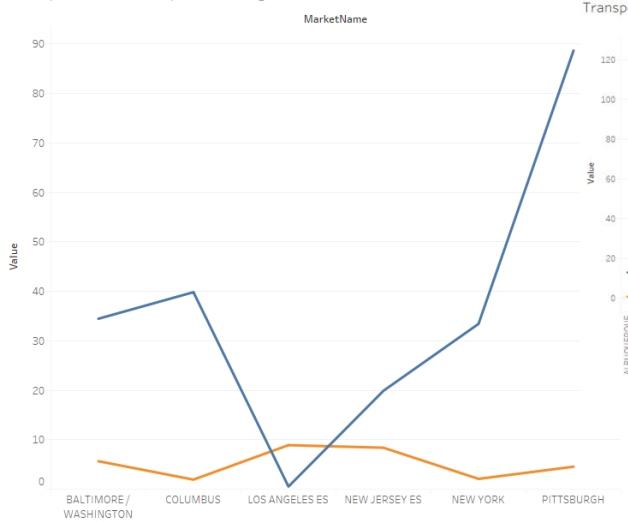


- Linear Regression: the reason we chose size and country code as control variables is that we think people who purchase quantity also depend on their transaction history and the country they belong to since the US and Canada may have different geographical features that might affect customer choice.

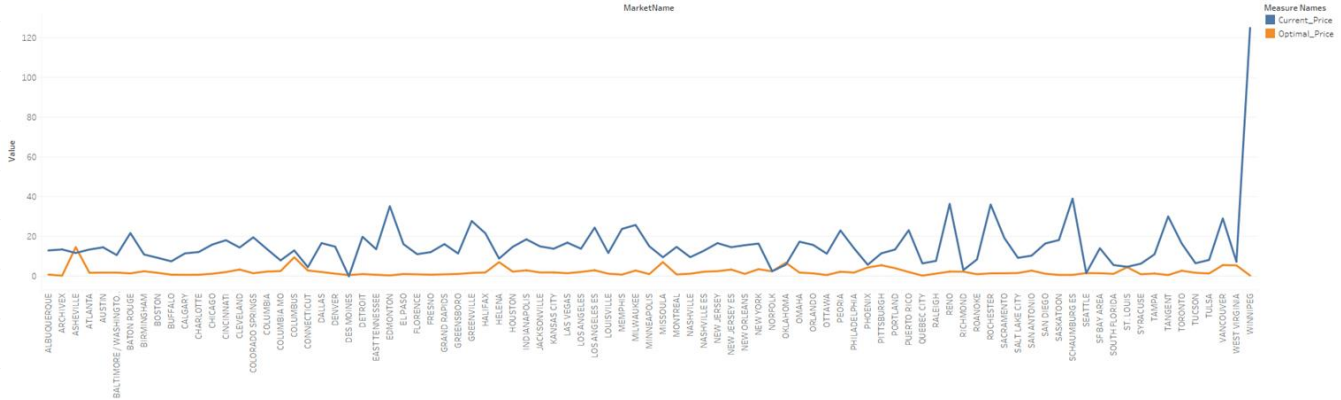
- Account size: larger account – lower unit price
- Difference in country: Company **should set more pricing strategy for the US** since Canada get a lower number of places and markets to sell.

Upsell vs Low prices? – Transport Related Product

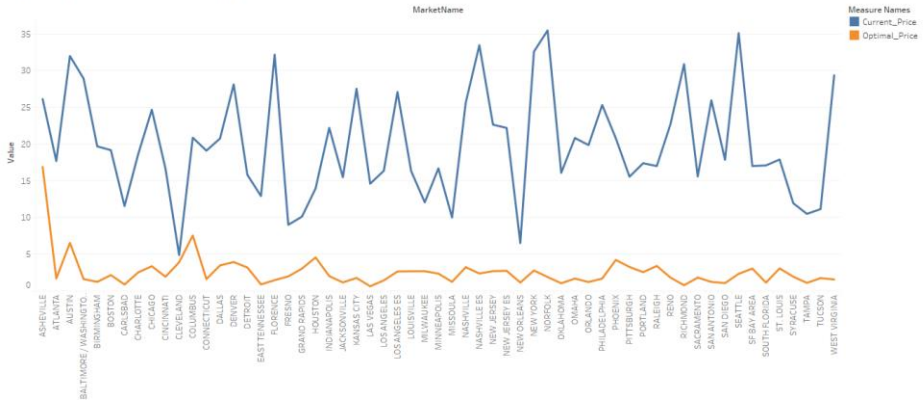
Transport_PriceComparisonLarge



Transport_PriceComparisonSmall

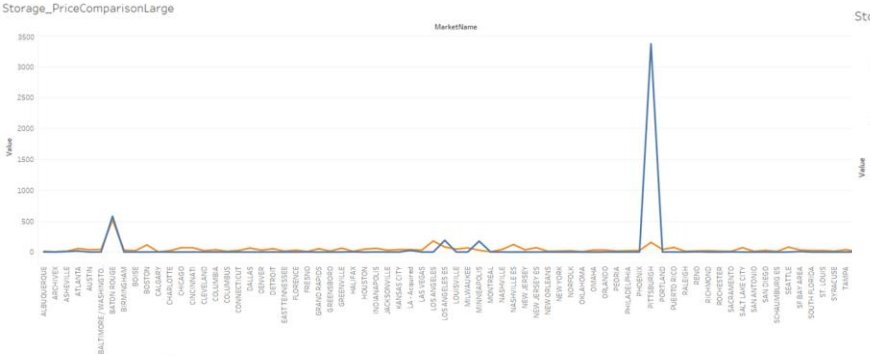


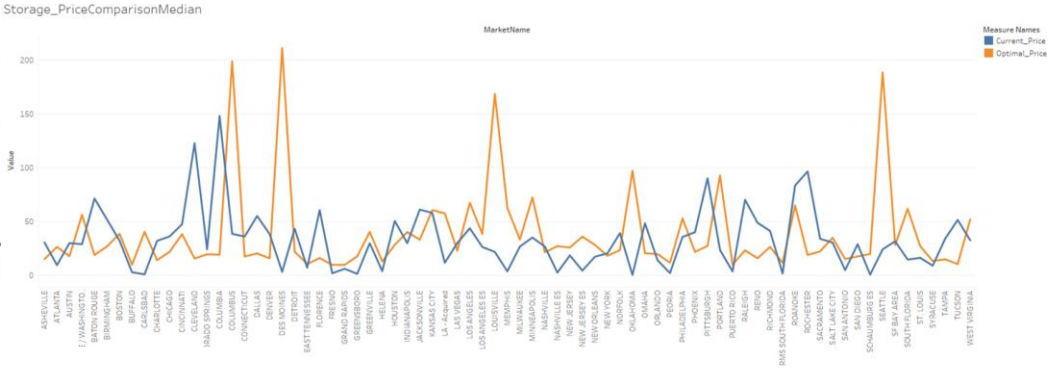
Transport_PriceComparisonMedian

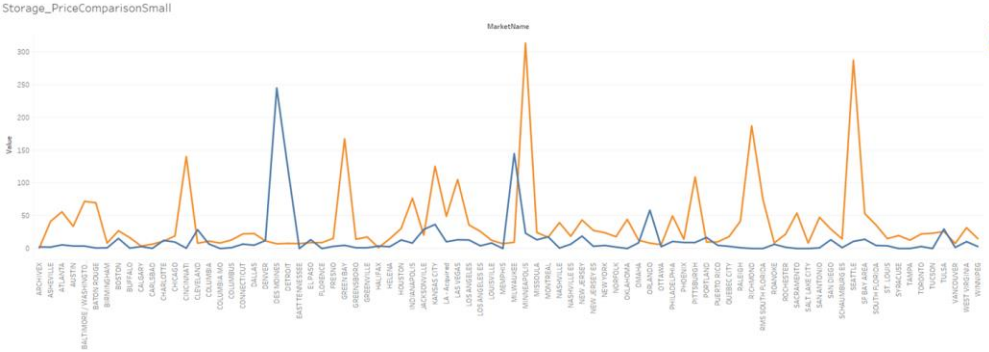


- It is clear that almost all the markets are upselling their products despite account size for customers.

Upsell vs low prices? – Storage related product

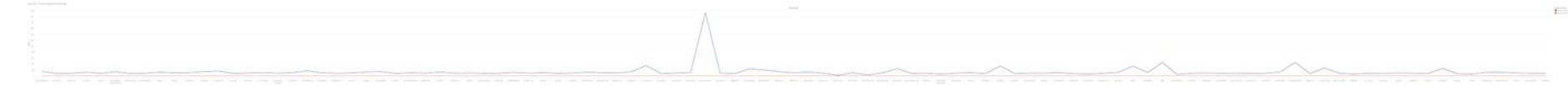


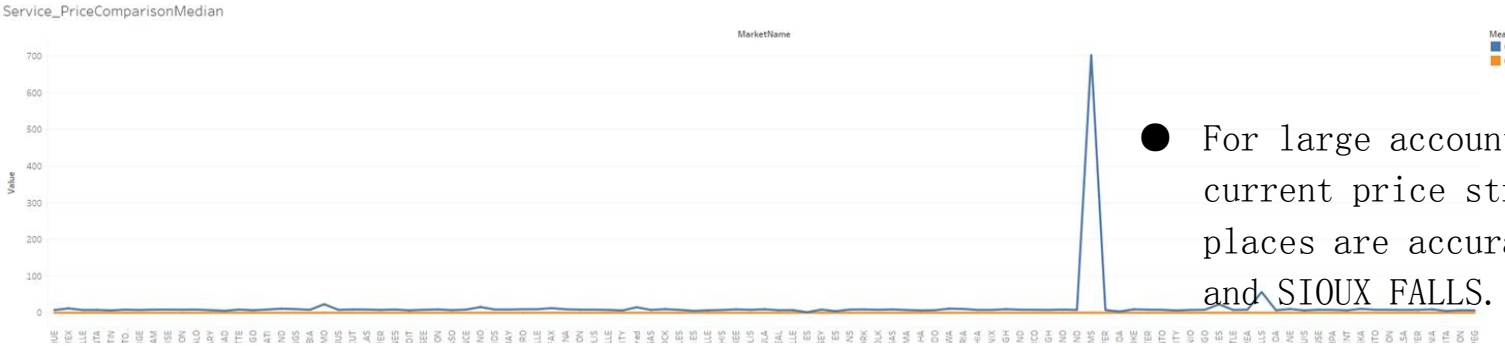




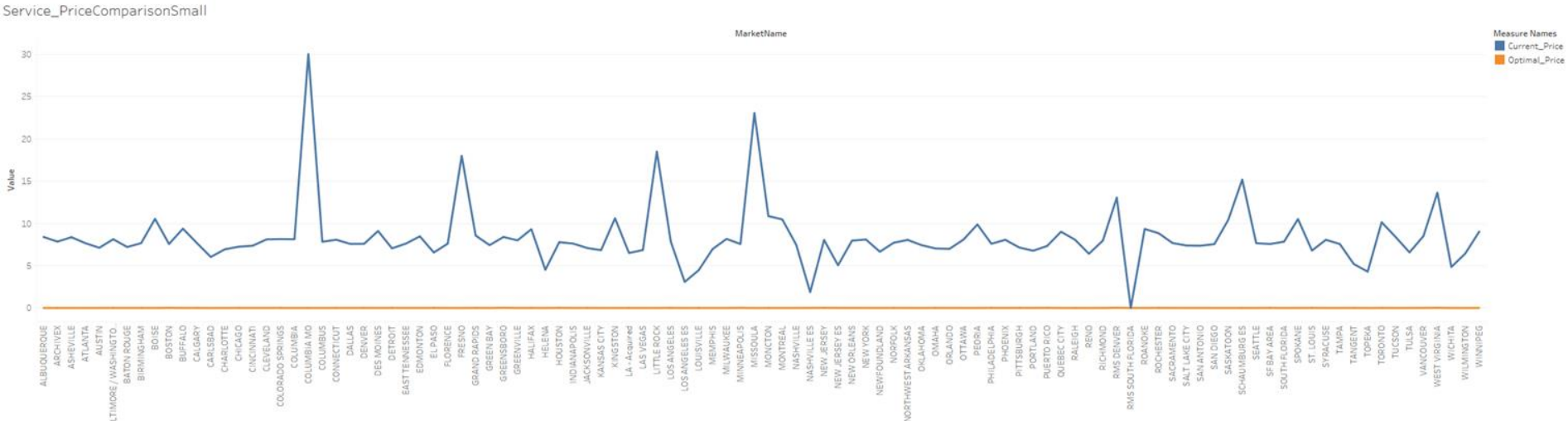
- Company should still keep same prices for large account customers except for Pittsburg, whose current Price is higher than optimal one.
- For small and median account, Company should change pricing strategy since most of the places are not in the optimal prices.

Upsell vs low prices? – Service related product





- For large account customers, the current price strategy for most places are accurate except for RMS and SIOUX FALLS.



Appendix – R Code

```
#Data classfied into three groups by bill code type
Data = read.csv("pricing competition.csv")
Data$price = Data$InvoiceRevenue/Data$InvoiceQuantity
summary(Data)
unique(Data$BillCodeType)
Data_service<-Data[Data$BillCodeType=='SERVICE - PERMANENT WITHDRAWAL'|Data$BillCodeType=='SERVICE - RECEIVING AND ENTRY'|Data$BillCodeType=='SERVICE - TRANSPORTATION
HANDLING'|Data$BillCodeType=='SERVICE - RETRIEVAL'|Data$BillCodeType=='SERVICE - REFILE'|Data$BillCodeType=='SERVICE - MINIMUM SERVICE CHARGE PER
ORDER'|Data$BillCodeType=='Service',]
Data_storage<-Data[Data$BillCodeType=='STORAGE - CARTON'|Data$BillCodeType=='STORAGE - LTO'|Data$BillCodeType=='STORAGE - TAPE'|Data$BillCodeType=='STORAGE -
HD'|Data$BillCodeType=='STORAGE - MINIMUM STRG CHARGE'|Data$BillCodeType=='STORAGE - PALLET'|Data$BillCodeType=='STORAGE - FLAT'|Data$BillCodeType=='STORAGE -
BULK'|Data$BillCodeType=='STORAGE - STRG,LOCKED AREA'|Data$BillCodeType=='Storage',]
Data_transport<-Data[Data$BillCodeType=='Transport',]
#Transport
library('Matrix')
library('lfe')
library('data.table')
library('tidyverse')
Data_transport$size = 0
Data_transport = Data_transport[,-21]
transport = aggregate(Data_transport$InvoiceRevenue, by = list(Data_transport$CustomerAccountName), FUN = sum)
transport = transport %>%
  rename(CustomerAccountName = Group.1,
         size = x)
Data_transport = merge(Data_transport, transport, by = 'CustomerAccountName')
reg_1 = felm(InvoiceQuantity~RateAmount+size.y+factor(CountryCode)|factor(GLPeriodQuarter), data = Data_transport)
summary(reg_1)
Data_transport$Optimal_Price = 0
Data_transport$Optimal_Price[Data_transport$CountryCode == 'US' ] = -(0.000702*Data_transport$size.y+2.244)/2*(-0.4849)
Data_transport$Optimal_Price[Data_transport$CountryCode == 'CA' ] = -(0.000702*Data_transport$size.y)/2*(-0.4849)
transport_res = Data_transport[,c(5,7,9,10,11,12,22,23)]
summary(transport_res$size.y[transport_res$CountryCode == 'US'])
summary(transport_res$size.y[transport_res$CountryCode == 'CA'])
transport_res$Size_account[transport_res$size.y<1456 & transport_res$CountryCode == 'US' ] = 'small'
transport_res$Size_account[transport_res$size.y<5662 & transport_res$CountryCode == 'CA' ] = 'small'
transport_res$Size_account[transport_res$size.y>118691 & transport_res$CountryCode == 'US'] = 'large'
transport_res$Size_account[transport_res$size.y>62167 & transport_res$CountryCode == 'CA'] = 'large'
transport_res$Size_account[transport_res$size.y>=7797 & transport_res$size.y<=118691 & transport_res$CountryCode == 'US'] = 'median'
transport_res$Size_account[transport_res$size.y>=5662 & transport_res$size.y<=62167 & transport_res$CountryCode == 'US'] = 'median'
t_market = aggregate(transport_res$Optimal_Price, by = c(list(transport_res$Size_account), list(transport_res$MarketName)) , FUN = mean)
t_res = t_market %>%
  rename(Account_Size = Group.1,
         market = Group.2,
         optimal_price = x)
```

Appendix – R Code

```
#Storage
Data_storage<-Data[Data$BillCodeType=='STORAGE - CARTON'|Data$BillCodeType=='STORAGE - LTO'|Data$BillCodeType=='STORAGE - TAPE'|Data$BillCodeType=='STORAGE - HD'|Data$BillCodeType=='STORAGE - MINIMUM STRG CHARGE'|Data$BillCodeType=='STORAGE - PALLET'|Data$BillCodeType=='STORAGE - FLAT'|Data$BillCodeType=='STORAGE - BULK'|Data$BillCodeType=='STORAGE - STRG,LOCKED AREA'|Data$BillCodeType=='Storage',]
storage = aggregate(Data_storage$InvoiceRevenue, by = list(Data_storage$CustomerAccountName), FUN = sum)
storage = storage %>%
  rename(CustomerAccountName = Group.1,
         Size = x)
Data_storage = merge(Data_storage, storage, by = 'CustomerAccountName')
reg_2 = felm(InvoiceQuantity~RateAmount+Size+factor(CountryCode)|factor(GLPeriodQuarter), data = Data_storage)
summary(reg_2)
Data_storage$Optimal_Price = 0
Data_storage$Optimal_Price[Data_storage$CountryCode == 'US' ] = -(0.001637*Data_storage$Size+64.82)/2*(-0.1965)
Data_storage$Optimal_Price[Data_storage$CountryCode == 'CA' ] = -(0.001637*Data_storage$Size)/2*(-0.1965)
storage_res = Data_storage[,c(5,7,9,10,11,12,21,22,23)]
summary(storage_res$Size[storage_res$CountryCode == 'US'])
summary(storage_res$Size[storage_res$CountryCode == 'CA'])
storage_res$Size_account[storage_res$Size<7797 & storage_res$CountryCode == 'US' ] = 'small'
storage_res$Size_account[storage_res$Size<5662 & storage_res$CountryCode == 'CA' ] = 'small'
storage_res$Size_account[storage_res$Size>118691 & storage_res$CountryCode == 'US'] = 'large'
storage_res$Size_account[storage_res$Size>62167 & storage_res$CountryCode == 'CA'] = 'large'
storage_res$Size_account[storage_res$Size>=7797 & storage_res$Size<=118691 & storage_res$CountryCode == 'US'] = 'median'
storage_res$Size_account[storage_res$Size>=5662 & storage_res$Size<=62167 & storage_res$CountryCode == 'US'] = 'median'
storage_market = aggregate(storage_res$Optimal_Price, by = c(list(storage_res$Size_account), list(storage_res$MarketName)) , FUN = mean)
storage_market = storage_market %>%
  rename(Account_Size = Group.1,
         market = Group.2,
         optimal_price = x)
#Service
Data_service<-Data[Data$BillCodeType=='SERVICE - PERMANENT WITHDRAWAL'|Data$BillCodeType=='SERVICE - RECEIVING AND ENTRY'|Data$BillCodeType=='SERVICE - TRANSPORTATION HANDLING'|Data$BillCodeType=='SERVICE - RETRIEVAL'|Data$BillCodeType=='SERVICE - REFILE'|Data$BillCodeType=='SERVICE - MINIMUM SERVICE CHARGE PER ORDER'|Data$BillCodeType=='Service',]
service = aggregate(Data_service$InvoiceRevenue, by = list(Data_service$CustomerAccountName), FUN = sum)
service = service %>%
  rename(CustomerAccountName = Group.1,
         Size = x)
Data_service = merge(Data_service, service, by = 'CustomerAccountName')
reg_3 = felm(InvoiceQuantity~RateAmount+Size+factor(CountryCode)|factor(GLPeriodQuarter), data = Data_service)
summary(reg_3)
Data_service$Optimal_Price = 0
Data_service$Optimal_Price[Data_service$CountryCode == 'US' ] = -(0.0001768*Data_service$Size+0.6695)/2*(-0.003351)
Data_service$Optimal_Price[Data_service$CountryCode == 'CA' ] = -(0.0001768*Data_service$Size)/2*(-0.003351)
service_res = Data_service[,c(5,7,9,10,11,12,21,22,23)]
summary(service_res$Size[service_res$CountryCode == 'US'])
summary(service_res$Size[service_res$CountryCode == 'CA'])
service_res$Size_account[service_res$Size<681 & service_res$CountryCode == 'US' ] = 'small'
service_res$Size_account[service_res$Size<594.6 & service_res$CountryCode == 'CA' ] = 'small'
service_res$Size_account[service_res$Size>13928 & service_res$CountryCode == 'US'] = 'large'
service_res$Size_account[service_res$Size>9464.6 & service_res$CountryCode == 'CA'] = 'large'
service_res$Size_account[service_res$Size>=681 & service_res$Size<=13928 & service_res$CountryCode == 'US'] = 'median'
service_res$Size_account[service_res$Size>=594.6 & service_res$Size<=9464.6 & service_res$CountryCode == 'US'] = 'median'
service_market = aggregate(service_res$Optimal_Price, by = c(list(service_res$Size_account), list(service_res$MarketName)) , FUN = mean)
service_market = service_market %>%
  rename(Account_Size = Group.1,
         market = Group.2,
         optimal_price = x)
```