Data Warehousing

Homework 2 and 3
Grocery Chain Data Enhancements

Breitzman 9/16/2017

Hopefully…

- Hopefully, everyone has a grocery store version 1 completed where you have a large file with date, customer #, sku, price for however many transactions per customer and about 1000 customers per day

- In version 2 we will add an inventory management system to your program and re-run it. (Don’t worry you don’t have to start from scratch, version 1 just needs a couple of modifications)

Simplifying Assumptions

- We'll assume Milk is delivered 7 days a week
- All other items are delivered 3 days per week (if your last name starts with A-M your deliveries are Monday-Wednesday-Friday, otherwise your deliveries are Tuesday-Thursday-Saturday.
- We'll assume all deliveries occur before the store opens and we'll assume we can order any item at any time before the delivery and it will magically appear

Homework 2

- This one is easy
- Take a week or two weeks worth of your existing data and compute the average sales per day of each item
- Obviously the special items like Milk, bread, cereal etc. will have different totals then the others. Remaining items should be roughly equal.
- For example if we have a thousand customers and 70% buy milk we expect to see 700 containers of milk per day. Since we have 6 kinds of milk we would expect to sell about 115 per day. Your mileage may vary, but if your numbers are too far off you probably have a bug
- Similarly on random items if we have roughly 2000 items, 1000 customers per day and each customer buys roughly 50 items we would expect to sell 50,000 items per day and since they're randomly distributed we would expect to sell about 25 of each non-special item per day (special items have their own rules like Milk, cereal, peanut butter, etc.)

Homework 2 – due next week

- So the actual deliverable is as follows
- Take each of the special items Milk, baby food etc. and compute how many should sell per day given your parameters.
- Do the same for the non-special items (you can lump these together. Don’t do a separate line for 2000 items)
- Next take a week or two of your data and calculate daily sales for each of the above items
- Make sure your numbers are close (if they’re too far off then something is wrong. If they’re too close, then you are probably faking your data – don’t do that either.)
- Put all of your data into a spreadsheet (Excel or LibreOffice Calc. If you use something more exotic turn it into a pdf to submit it.)

Homework 3 – due in 2 weeks

- Now we use the data collected in HW 2 to modify your program recording transactions
- Let’s assume your current program is working correctly. (And if it’s not, I will attempt to fix your programs after class tonight)
- We need to make the following changes
  - For each SKU we need to keep a running inventory. Assume on January 1 we have a 3 day supply of every product and a 1.5 day supply of milk.
  - Every time an item is bought we reduce the inventory by 1
  - If it’s a delivery day, then before we open the store we run through each product (except milk) and if there is less than a 3 day supply of anything we order n cases where n * 12 brings us just above a 3 day supply. For milk we try to maintain a 1.5 day supply at all times
  - The transaction program will also need the following change. If you attempt to buy an item and there are 0 left, then the customer will have to buy a different product
  - If we’re going to manage inventories as well as sales we will have to have the following outputs: Date|Customer#|SKU|Price|ItemsLeft|TotalCasesOrdered

Homework 3 – Examples

- So for example on January 1 let’s suppose we have 7 cases of SKU: 44112001 (Atomic Malted Milk Balls) and that is the first transaction. The output looks like the following
- 20170101|1|44112001|$1.96|83|7
- Note we had 84 items, and we reduce it to 83 on the transaction
- We had 7 cases on hand
- Let’s say after during the next 2 days we sold 44 more so that on the 980th customer of 20170102 we have the following
- 20170102|980|44112001|$1.96|39|7
- Now before we open the store we get a delivery. We need to get the count back over 84 so we add 4 cases so that we have a total of 8 7 packages.
- Now the next transaction that buys this product is customer 5 so we have the following
- 20170103|5|44112001|$1.96|86|11
- Note that the last column is a running total of the number of cases bought for the year, so it goes up from 7 to 11
- There’s a lot of moving parts here, but each one is fairly simple

Homework 3 – Implementation Details

- To implement this, you actually need to keep track of 2 running tallies for each SKU (the count left in store, and the count of cases purchased so far)
- Each time a case is incremented, we increment the count by store by 12
- Each time an item is purchased we decrement the count by store by 1 but we don’t touch the case count
- So these items just present minor changes to your existing code
- A slightly more difficult implementation detail is that if you go to process a transaction and the item count is 0 then you will need a substitute (e.g. if we run out of 1 kind of baby food we need to pick a random baby food replacement and repeat until we find one that is in stock. Same with milk, same with malted milk balls etc.)
- Third detail is when we get to a new day we need to find out if it’s a delivery day. We need to then run through our 2000 non-milk items and if we have a less than 3 day supply we increment the cases by however many we need and increment the items by 12*cases.
- That’s for non-milk items. For milk we need to do it every day, but we’re only trying to maintain a 1.5 day supply

Homework 3 – Any Questions?

- I promise this is the last modification we will need to make to your programs (unless I didn't think of something)
- From these transactions we will be able to actually build our data warehouse (which we will design in class tonight)

Rules of the game (same as before)

- You may use any language you want: Python, C, C++, C# Scheme?, Visual Basic, (even Java!)
- You may work in pairs or threes if you want for generating the code to do the simulations, however each of you should compile your own version with your own parameters so each person is represented in the final data warehouse with his/her own grocery store
- If you work together the source code should have both of your names on it, but each turn in a version with their own constants/parameters