Relational Database Design

Using MS Access to help a start-up prosper

Tatsu Hobby
The Collector's Hobby Shop
<table>
<thead>
<tr>
<th>Team 5 (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ashley-Mengyi Zhang</strong></td>
</tr>
<tr>
<td>- IEOR Masters and ORMS Bachelors</td>
</tr>
<tr>
<td>- New Product Development Engineer at SanDisk</td>
</tr>
<tr>
<td><strong>Chengwei Wong</strong></td>
</tr>
<tr>
<td>- Applied Mathematics</td>
</tr>
<tr>
<td>- Actuarial Analyst at Esurance</td>
</tr>
<tr>
<td><strong>Dongwei Han</strong></td>
</tr>
<tr>
<td>- Economics</td>
</tr>
<tr>
<td><strong>Eric-Shiqi Zhang</strong></td>
</tr>
<tr>
<td>- Mathematics and Statistics</td>
</tr>
<tr>
<td><strong>Jolie-Jiayu Tan</strong></td>
</tr>
<tr>
<td>- Statistics and IEOR Minor, Regent’s Scholar</td>
</tr>
<tr>
<td>- Data Analyst at AbsolutData Analytics</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Introduction</th>
<th>Revised EER</th>
<th>Schema</th>
<th>Interesting Queries</th>
<th>Conclusion</th>
</tr>
</thead>
</table>

Introduction

Revised EER

Schema

Interesting Queries

Conclusion
<table>
<thead>
<tr>
<th>Team 5 (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Marty Ren</strong></td>
</tr>
<tr>
<td>- Statistics</td>
</tr>
<tr>
<td><strong>Patrick Hop</strong></td>
</tr>
<tr>
<td>- Applied Mathematics</td>
</tr>
<tr>
<td>- Machine Learning Engineer at the AMPLab</td>
</tr>
<tr>
<td><strong>Raymond Ma</strong></td>
</tr>
<tr>
<td>- IEOR</td>
</tr>
<tr>
<td>- User Operations at Dropbox</td>
</tr>
<tr>
<td><strong>Ryan Runchey</strong></td>
</tr>
<tr>
<td>- ORMS &amp; Haas Business</td>
</tr>
<tr>
<td>- Corporate Finance at Agilent Technologies</td>
</tr>
</tbody>
</table>
Our Client **Tatsu Hobby**

Tatsu Hobby is a San Jose based multi-channel retailer selling collectible toys and model kits. Since its launch in 2009, the company has served thousands of satisfied customers, with one goal in mind:

“To bring the best selection of collectible toys, tagged with highly competitive pricing, and serve them to the customers with a smile.” – Tatsu Hobby

Currently operates two retail channels – online and events – with plans to open a storefront by year’s end. The company relies on both domestic and international suppliers to stock its warehouse.
Help staff store and manage data more efficiently - forecast demand, purchase optimal order quantities, and deliver customer value - with storefront opening.

Our Project Goals

I. Mineable Database
II. Reduced Data Entry Time
III. Automated Analysis w/Dashboards
IV. Efficient Business Cycle
V. Reduce Tatsu’s Operating Costs

These five benefits will help Tatsu Hobby expand their business.

Integrated Systems Management

Minimal Employee Hours Spent
How Tatsu Hobby Benefits

We make their data available, organized, and actionable.

- **Product Line Expansion**
- **Customer Programs Driving Recurring Revenues**
- **Employee Work Scheduling**
- **Supply Chain Optimization: Identifying Previously Unknown Drivers of Financial Information**
- **Enhanced Customer Browsing Experience**
- **Analytics and Machine Learning**
Revised EER Diagram
Super Class:
1. **PERSON** (PID, phone, email, Fname, MI, Lname)
2. **ORDER** (order_ID, placed_by_PID\(^1\), handled_by_PID\(^1\), quantity, price, order_date)
3. **LOCATION** (loc_ID, addr_ID\(^2\))

Website:
24. **EVALUATION** (eval_ID, customer_PID\(^1\), order_ID\(^2\), prod_ID\(^4\), ship_score, pack_score, prod_score, review, add_points_ID\(^{44}\))
25. **ESTIMATED_PRODUCT_RATING** (rating_ID, for_PID\(^1\), prod_ID\(^4\), score)
26. **WISHLIST** (wishlist_ID, PID\(^1\), date_added, remarks)
27. **CATEGORY** (cat_ID, cat_name, description)
28. **CATEGORY_PAGE** (page_ID\(^{36}\), cat_ID\(^{27}\))
29. **THEME** (theme_ID, theme_name, description)
30. **THEME_PAGE** (page_ID\(^{36}\), theme_ID\(^{29}\))
31. **PRODUCT_LINE** (prod_line_ID, prod_line_name, manufacturer, description)
32. **PRODUCT_LINE_PAGE** (page_ID\(^{36}\), prod_line_ID\(^{31}\))
33. **SIZE** (size_ID, description)
34. **SIZE_PAGE** (page_ID\(^{36}\), size_ID\(^{33}\))
35. **PRODUCT_PAGE** (page_ID\(^{36}\), prod_ID\(^4\))
36. **WEBPAGE** (page_ID, url, hits)
37. **VISITING_IP** (IP_addr, page_ID)
38. **PERSON_HAS_IP** (PID\(^1\), IP_addr\(^{37}\))
Operations:

4. **PRODUCT** (prod_ID, cat_ID, prod_line_ID, page_ID, product_name, buy_price, sell_price, release_date, weight)

5. **PRODUCT_INVENTORY** (prod_ID, prod_IID, loc_ID, prod_inv_OID, order_ID, quantity)

6. **PRODUCT_INVENTORY_ORDER** (prod_inv_OID, date_placed, fixed_cost, buy_price, total_order, supplier_ID, status, lead_time)

7. **PRODUCT_WISHLIST** (prod_ID, wishlist_ID)

8. **TRANSACTION** (trans_ID, prod_inv_OID, ship_mtl_OID, OID, time, amount)

9. **EMPLOYEE** (PID, SSN, supervisor_PID, hourly_rate, discount)

10. **PAYROLL** (payroll_ID, PID, trans_ID, aggregate_work_hr_ID, hourly_rate)

11. **EMPLOYEE_ORDER** (order_ID, placed_by_PID, discount, order_date)

12. **WORK_HOUR** (work_hr_ID, payroll_ID, cal_date, clock_in, clock_out, hours)

13. **SUPPLIER** (PID, liason_PID, supplier_name, catalog)

14. **SHIPPING_SERVICE** (shipserv_ID, shipper_name, phone, email)

15. **SHIPMENT** (ship_ID, ship_fee, trans_ID, from_addr_ID, to_addr_ID, est_trans_time, dispatch_date, receive_date)

16. **SHIPPING_MATERIAL_INVENTORY** (ship_mtl_ID, ship_mtl_OID, ship_ID, item_name, quantity)

17. **SHIPPING_MATERIAL_ORDER** (ship_mtl_OID, supplier_ID, ship_ID, trans_ID, quantity, price)

18. **SHIPMENT_ORDER** (OID, ship_ID)

19. **WAREHOUSE** (loc_ID, capacity, rent, remarks)

20. **STOREFRONT** (loc_ID, capacity, rent, remarks)

21. **CONVENTION_EVENT** (loc_ID, event_name, cal_date, event_attendance, booth_attendance)

22. **ADDRESS** (addr_ID, street, city, state_prov, country, zip_post)

23. **PERSON_ADDRESS** (PID, addr_ID)
### Schema

**Customer Loyalty Programs:**

39. **CUSTOMER** (PID\(^1\), tastu_reward_ID\(^{43}\))

40. **SALES_PROMOTION** (promo_ID, prod_ID\(^4\), discount, start_date, end_date)

41. **CUSTOMER_ORDER** (order_ID\(^2\), placed_by_PID\(^1\), promo_ID\(^{40}\), points_ID\(^{44}\), discount, add_points_ID, used_points_ID, order_date)

42. **SERVICE_REQUEST** (req_ID, req_by_PID\(^1\), date_req, handled_by_PID\(^1\), date_handled, description, status)

43. **TASTU_REWARD_ACCOUNT** (reward_ID, owner_PID\(^1\), total_points_amount)

44. **TATSU_POINT_TRANSACTION** (points_ID, reward_ID\(^{43}\), time, amount)

**Contingent:**

45. **CATEGORY_PRODUCT** (cat_ID\(^{27}\), prod_ID\(^4\))

46. **THEME_PRODUCT** (theme_ID\(^{29}\), prod_ID\(^4\))

47. **SIZE_PRODUCT** (size_ID\(^{33}\), prod_ID\(^4\))

48. **EVALUATION_RATING** (eval_ID\(^{24}\), rating_ID\(^{25}\))

49. **VISITING_IP_WEBPAGE** (IP_addr\(^{37}\), page_ID\(^{36}\))

50. **SHIPPING_MATERIAL_ORDER_SHIPMENT** (ship_mtl_OID\(^{17}\), ship_ID\(^{15}\), received_by_PID\(^1\))
1. Granular Financial Analysis Visualization
2. Predict Profitability of New Products
3. Optimal Inventory Policy Management
4. Collaborative Filtering via Distributed Matrix Factorization
5. Ideal Warehouse Location for Future Expansion
Granular Financial Analysis Visualization

Understanding financial drivers by identifying most and least profitable products

Compute profit on a per-product basis
Compute revenue on a per-product basis
Visualize results
Query (1) Granular Financial Analysis

Goal: Turning raw data into insight by providing easy to understand visualization of pricing financials.

- VBA grabs results from Access and then feeds it into Excel
- Excel cleans raw data, calculates gross profits, and displays various graphs displaying profitability over time
SELECT prod_ID, Sum(O.quantity) * (sell_price-buy_price) as Gross_Profit
FROM Order as O, Product as P
WHERE year(order_date) = '2012' and P.prod_ID=O.pro_ID
GROUP by month(order_date), prod_ID
ORDER by prod_ID, month(order_date)
Simply Click on an button I created in Excel

- VBA will grab data from Access and feed it to Excel
- Excel will polish data and display various graphs includes

- Each Product’s Gross Profit By Month
- Company’s Gross Profit by Month
- Gross profit by product
- Past 3 years’ Cash Flow
Query (1) Visualization

Product Monthly Gross Profit

Total Gross Profit by Month

Gross Profit by Product

Company Cash Flow

Introduction
Revised EER
Schema
Interesting Queries
Conclusion
Profit Contribution Distribution

Find the contribution of each product to the gross profit

Discover Distribution Type

Identify Most Profitable Products

Separate winning and losing products
glmnet: Lasso and elastic-net regularized generalized linear models

MATLAB
The Language of Technical Computing

MLbase
SELECT p.prod_ID, p.release_date, p.prod_lineID, release_date,
(o.quantity * (o.price - p.buy_price))
FROM Order as o, Product as p
WHERE o.prod_ID = p.prod_ID;

Data-Generating Distribution

Suppose $x \sim N(\mu, \Sigma)^*$, $\mu = 0$, $\Sigma_{i,j} = 0$ iff $i = j$

* Can be tuned for non-gaussian distributions
Theory

Find the maximizer of the penalized log-likelihood

$$\max_{(\beta_0, \beta) \in \mathbb{R}^{d+1}} \left[ \frac{1}{N} \sum_{i=1}^{N} \{ I(g_i = 1) \log p(x_i) + I(g_i = 2) \log(1 - p(x_i)) \} - \lambda P_\alpha(\beta) \right]$$

Grid-Search

Perform brute-force search in hyper-parameter space. Finds the optimal* hyperparameter, as measured by performance on k-fold validation.

K-Fold Cross Validation

Randomly distribute datapoints into $k$ partitions. Train on first $k - 1$ partitions, and test on the $k^{th}$ partition. Permute test partition $k$-times.

* Hyperparameter space is non-convex
## Results

<table>
<thead>
<tr>
<th></th>
<th>Raw</th>
<th>Normalized</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initial Features</strong></td>
<td>86</td>
<td>86</td>
</tr>
<tr>
<td><strong>Valid Features</strong></td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td><strong>Observations</strong></td>
<td>35042</td>
<td>35042</td>
</tr>
<tr>
<td><strong>Hyperparameter</strong></td>
<td>.0001</td>
<td>1</td>
</tr>
<tr>
<td><strong>Test Error</strong></td>
<td>18%</td>
<td>18%</td>
</tr>
<tr>
<td><strong>Top 5 Features’ Contribution</strong></td>
<td>56%</td>
<td>57%</td>
</tr>
</tbody>
</table>
A Continuous Review Inventory Model

Advantages Over Traditional EOQ:
Captures Demand Variability, Order Lead Time & Profit Contribution

MS Access
SQL extract data from Access and export to Excel

Excel
Automatically calculate and display results
For each product j, select
K: Fixed ordering cost, i.e. transportation cost and administrative cost
D: Average demand per unit time
σ: Standard deviation of demand per unit time
C: Buy price
P: Sell price
b: Backorder penalty cost
i: Interest rate (opportunity cost)
LT: Order lead time

Decision Variables:
Q: Optimal order quantity
R: Reorder point

**Order Q whenever the on-hand inventory falls to R**
Query (3) Mathematical Model (2)

For Q, use the EOQ quantity $\sqrt{2KD/iC}$

To determine R?

- First need to determine an optimal service level
  Recall in Newsvendor Model
  Critical fractile:
  $$\frac{c_u}{c_u + c_o} = \frac{b + (P \ C) \ast (1 \ sub\%)}{b + (P \ C) \ast (1 \ sub\%) + i \ c(O/D)}$$

  $ic \frac{O}{D} =$ cost of holding one unit in inventory for one cycle (plays the role of $c_o$)

- Find Z score corresponding with critical fractile
- $R = \text{mean demand during lead time} + Z \ast \text{standard deviation of demand during lead time}$

$R = D \ast LT + Z \ast \sqrt{LT} \ast$
MONTHLY_SALE Table
SELECT o.prod_ID, Month(o.order_date) AS [Month], Year(o.order_date) AS [Year], Sum(o.quantity) AS quantity INTO MONTHLY_SALE FROM 02_ORDER AS o GROUP BY o.prod_ID, Month(o.order_date), Year(o.order_date) ORDER BY Sum(o.quantity) ASC;

EXPORT Table
SELECT MS.prod_ID, AVG(MS.quantity), STDEV(MS.quantity), AVG(PI.fixed_cost), P.buy_price, P.sell_price FROM 06_PRODUCT_INVENTORY_ORDER AS PI, MONTHLY_SALE AS MS, 04_PRODUCT AS P WHERE PI.prod_ID=MS.prod_ID AND P.prod_ID=MS.prod_ID GROUP BY MS.prod_ID;
## Query Sample Output

### Output Table

<table>
<thead>
<tr>
<th>Optimal Order Q</th>
<th>Optimal Service Level</th>
<th>Reorder Point</th>
<th>Status</th>
<th>Supplier Name</th>
<th>Contact info</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>94%</td>
<td>20</td>
<td>Normal</td>
<td>X-strip</td>
<td><a href="mailto:ZippyGamer@gmail.com">ZippyGamer@gmail.com</a></td>
</tr>
<tr>
<td>65</td>
<td>68%</td>
<td>86</td>
<td>Normal</td>
<td>Tempplex</td>
<td><a href="mailto:DerangedGamer@gmail.com">DerangedGamer@gmail.com</a></td>
</tr>
<tr>
<td>76</td>
<td>89%</td>
<td>45</td>
<td>Normal</td>
<td>Warentechnology</td>
<td><a href="mailto:MellowGamer@gmail.com">MellowGamer@gmail.com</a></td>
</tr>
<tr>
<td>219</td>
<td>93%</td>
<td>46</td>
<td>Normal</td>
<td>Betacone</td>
<td><a href="mailto:LearnedGamer@gmail.com">LearnedGamer@gmail.com</a></td>
</tr>
<tr>
<td>123</td>
<td>94%</td>
<td>92</td>
<td>Normal</td>
<td>bigware</td>
<td><a href="mailto:GodlyGamer@gmail.com">GodlyGamer@gmail.com</a></td>
</tr>
<tr>
<td>151</td>
<td>96%</td>
<td>54</td>
<td>Normal</td>
<td>Techiplex</td>
<td><a href="mailto:DearGamer@gmail.com">DearGamer@gmail.com</a></td>
</tr>
<tr>
<td>132</td>
<td>90%</td>
<td>153</td>
<td>Normal</td>
<td>Voltsailbam</td>
<td><a href="mailto:HumdrumGamer@gmail.com">HumdrumGamer@gmail.com</a></td>
</tr>
<tr>
<td>227</td>
<td>96%</td>
<td>16</td>
<td>Normal</td>
<td>Scotdrill</td>
<td><a href="mailto:FearlessGamer@gmail.com">FearlessGamer@gmail.com</a></td>
</tr>
<tr>
<td>143</td>
<td>94%</td>
<td>103</td>
<td>Stockout</td>
<td>Zimtech</td>
<td><a href="mailto:TranquilGamer@gmail.com">TranquilGamer@gmail.com</a></td>
</tr>
<tr>
<td>121</td>
<td>97%</td>
<td>20</td>
<td>Normal</td>
<td>Vollothew</td>
<td><a href="mailto:PastoralGamer@gmail.com">PastoralGamer@gmail.com</a></td>
</tr>
<tr>
<td>99</td>
<td>93%</td>
<td>92</td>
<td>Stockout</td>
<td>Conelectrics</td>
<td><a href="mailto:HaltingGamer@gmail.com">HaltingGamer@gmail.com</a></td>
</tr>
<tr>
<td>80</td>
<td>99%</td>
<td>17</td>
<td>Normal</td>
<td>Tansaotech</td>
<td><a href="mailto:ProductiveGamer@gmail.com">ProductiveGamer@gmail.com</a></td>
</tr>
<tr>
<td>87</td>
<td>94%</td>
<td>15</td>
<td>Normal</td>
<td>Jaylab</td>
<td><a href="mailto:VenomousGamer@gmail.com">VenomousGamer@gmail.com</a></td>
</tr>
<tr>
<td>75</td>
<td>86%</td>
<td>56</td>
<td>Normal</td>
<td>Lam-kix</td>
<td><a href="mailto:SableGamer@gmail.com">SableGamer@gmail.com</a></td>
</tr>
<tr>
<td>150</td>
<td>88%</td>
<td>140</td>
<td>Stockout</td>
<td>Dingtech</td>
<td><a href="mailto:TheCeaselessGamer@gmail.com">TheCeaselessGamer@gmail.com</a></td>
</tr>
<tr>
<td>125</td>
<td>95%</td>
<td>92</td>
<td>Normal</td>
<td>Faxquote</td>
<td><a href="mailto:Black-and-whiteGamer@gmail.com">Black-and-whiteGamer@gmail.com</a></td>
</tr>
<tr>
<td>53</td>
<td>88%</td>
<td>26</td>
<td>Normal</td>
<td>Acertrans</td>
<td><a href="mailto:EliteGamer@gmail.com">EliteGamer@gmail.com</a></td>
</tr>
<tr>
<td>87</td>
<td>96%</td>
<td>35</td>
<td>Normal</td>
<td>Truefase</td>
<td><a href="mailto:OppositeGamer@gmail.com">OppositeGamer@gmail.com</a></td>
</tr>
<tr>
<td>177</td>
<td>97%</td>
<td>82</td>
<td>Stockout</td>
<td>Latquote</td>
<td><a href="mailto:TartGamer@gmail.com">TartGamer@gmail.com</a></td>
</tr>
<tr>
<td>130</td>
<td>93%</td>
<td>71</td>
<td>Normal</td>
<td>Betabase</td>
<td><a href="mailto:RoomyGamer@gmail.com">RoomyGamer@gmail.com</a></td>
</tr>
<tr>
<td>206</td>
<td>97%</td>
<td>132</td>
<td>Normal</td>
<td>Zoocode</td>
<td><a href="mailto:EagerGamer@gmail.com">EagerGamer@gmail.com</a></td>
</tr>
<tr>
<td>38</td>
<td>92%</td>
<td>30</td>
<td>Normal</td>
<td>haydax</td>
<td><a href="mailto:WildGamer@gmail.com">WildGamer@gmail.com</a></td>
</tr>
<tr>
<td>83</td>
<td>93%</td>
<td>130</td>
<td>Normal</td>
<td>Voyaelectronics</td>
<td><a href="mailto:PlacidGamer@gmail.com">PlacidGamer@gmail.com</a></td>
</tr>
<tr>
<td>52</td>
<td>99%</td>
<td>4</td>
<td>Normal</td>
<td>Labgreen</td>
<td><a href="mailto:JumbledGamer@gmail.com">JumbledGamer@gmail.com</a></td>
</tr>
</tbody>
</table>
# Intelligent Product Recommendations

Find the sparse user-product matrix for collaborative-filtering

---

**Increase Customer Engagement**

**Increase Volume**

**Product Discovery**
Query Four Technology

- Distributed Collaborative Filtering via Matrix Factorization
- One machine, four cores, eight parallel threads.

Spark
Lightning-Fast Cluster Computing

Scala
SELECT pr.PID, pr.prod_ID, score
FROM Product_Rating as pr;

Data-Generating Distribution
Suppose \( n_i \sim N(\mu, \Sigma) \), \( \mu = 0 \), \( \Sigma_{i,j} = 0 \) iff \( i = j \), \( n_i \in N \) IID
Theory

Definition
\[ ||.||_F \] is the Frobenius Norm

Definition
\( X^* \) is the conjugate transpose of a matrix \( X \). \( X^* = X^t \) over \( R \)

Definition
\( <.,.> \) is the euclidean inner product

Definition
\( P_E \) is the projection onto a feasible set \( E \)

Definition
\( x^* \) is an optimal solution
Theory

Can be thought of as minimizing a distance in matrix-space

\[ N, X, Y \in M(R), \quad \lambda \in F \]

\[
\min F(X, Y) := \frac{1}{2} \| P_E(N - XY^*) \|_F^2 + \frac{\lambda}{2} \| X \|_F^2 + \frac{\lambda}{2} \| Y \|_F^2 \\
= \frac{1}{2} \sum_{(i,j) \in E} (N_{i,j} - \langle x_i, y_j \rangle)^2 + \frac{\lambda}{2} \| x_i \|_2^2 + \frac{\lambda}{2} \| y_j \|_2^2
\]

Alternating Least Squares (ALS)[2]

Problem is non-convex in \( X \) and \( Y \). Fix one of the unknowns, and alternate. Result is a convex QP. Can be cast as an SDP using lifting.
Training Error = 0.40372
What is the ideal warehouse location for future expansion?

Find all the total sales sorted by cities

Computed Weighted distance and get optimized warehouse location

Visualized it based on the Google map
Select a.city AS city, sum([o.quantity] *[o.price]) AS Sales_Performance
From 02_ORDER AS o, 22_ADDRESS AS a
where o.placed_by_PID IN
(select p.PID
From 23_person_address as p
Where a.addr_ID = p.addr_ID)
group by a.city;

(this table has two columns: city, and sales performance)
Limited-Memory BFGS: in the family of quasi-Newton methods

Objective function:
minimize the sum of weighted distance

Compute distances from each customers to current warehouses and potential warehouse

Select the distances that potential warehouse is less than current warehouse

Box Constraint: US long and lat

```r
fr <- function(x){
  R <- 6371
  p0rad <- c(-121.895,37.339)*pi/180
  p1rad <- x* pi/180
  p2rad <- Tatsu[,c(4,5)] * pi/180
  D <- NULL
  for(i in 1:nrow(CT)){
    d0 <- sin(p0rad[2])*sin(p2rad[i,2])
    +cos(p0rad[2])*cos(p2rad[i,2])*cos(abs(p0rad[1]-p2rad[i,1]))
    d1 <- sin(p1rad[2])*sin(p2rad[i,2])
    +cos(p1rad[2])*cos(p2rad[i,2])*cos(abs(p1rad[1]-p2rad[i,1]))
    if (d1 >= d0 ){
      d = d1
    }else{
      d = d1
      D[i] <- acos(d)*R*0.62137*ratio[i]
    }
  }
  return(sum(D))
}

optm<-optim(c(-100,40),fr,method="L-BFGS-B",
  lower=c(-130,20),upper=c(-70,50),
  control = list(maxit = 30000, temp = 2000, trace = TRUE,REPORT = 500))
```
Query (5) Visualization

Some important functions:

- **GPS**: enable retrieve of longitude and latitude from Google of any addresses.
- **lmRsd**: compare Gross State Product for past ten years and fit a linear model to it.
- **plotUSBaseMap**: plot the base map from the google maps and plot US borders of each states.
- **plotSalesMap**: plot sales performance of each cities on and optimal warehouse on the google map with reference of gross state product as an economic measure of each state.
- **SalesMap**: plot current and optimal warehouses with arrows pointing to customer locations.

All in one single package!
> final()
iter  0 value 370.701250
final  value 320.406500
converged
$par
[1] -85.80996  38.03796
$value
[1] 320.4065
$counts
function gradient
  10     10
$convergence
[1] 0

Optimized Location
Optimized sum of weighted distance
Number of calls to objective function and computing hessain
Integer code. 0 indicates successful completion
Decomposing to 1NF

SUPPLIER\( (\text{supplier\_ID}, \text{liason\_PID}, \text{supplier\_name}, \text{catalog}) \)

Normalize to 1NF:

SUPPLIER\( (\text{supplier\_PID}, \text{liason\_PID}, \text{supplier\_name}) \)

SUPPLIER\_CATALOG\( (\text{supplier\_PID}, \text{catalog}) \)
Decomposing to 2NF

EVALUATION (eval_ID, customer_PID, order_ID, prod_ID, add_points_ID, ship_score, pack_score, prod_score, review)

eval_ID → {ship_score, pack_score, prod_score, review}

Normalize to 2NF:

EVALUATION (eval_ID, customer_PID, order_ID, prod_ID, add_points_ID)

EVAL_SCORE (eval_ID, ship_score, pack_score, prod_score, review)
Decomposing to 2NF

PAYROLL (payroll_ID, employee_PID, trans_ID, aggregate_work_hr, hourly_rate, claimed_status)

 Normalize to 2NF:

PAYROLL (payroll_ID, employee_PID, trans_ID, aggregate_work_hr, hourly_rate, claimed_status)

SALARY_CHART (employee_PID, hourly_rate)
Decomposing to 3NF&BCNF (1):

SHIPMENT (ship_ID, ship_fee, trans_ID, shipserver_ID, from_addr_ID, to_addr_ID, est_trans_time, dispatch_date, receive_date)

\{from_addr_ID, to_addr_ID\} → est_trans_time

Normalize to 3NF & BCNF:

SHIPMENT (ship_ID, ship_fee, trans_ID, shipserver_ID, from_addr_ID, to_addr_ID, est_trans_time)

SHIPPING_TIME_ESTIMATE(from_addr_ID, to_addr_ID, est_trans_time)
Decomposing to 3NF&BCNF (2):

SHIPMENT (\texttt{ship\_ID}, \texttt{ship\_fee}, \texttt{trans\_ID}, \texttt{shipserver\_ID},
\texttt{from\_addr\_ID}, \texttt{to\_addr\_ID}, \texttt{est\_trans\_time}, \texttt{dispatch\_date},
\texttt{receive\_date})

{\texttt{est\_trans\_time}, \texttt{dispatch\_date}} \rightarrow \texttt{receive\_date}

Normalize to 3NF & BCNF:

SHIPMENT (\texttt{ship\_ID}, \texttt{ship\_fee}, \texttt{trans\_ID}, \texttt{shipserver\_ID}, \texttt{from\_addr\_ID},
\texttt{to\_addr\_ID}, \texttt{est\_trans\_time})

\texttt{SHIPPING\_RECEIVING\_TIME} (\texttt{est\_trans\_time}, \texttt{dispatch\_date},
\texttt{receive\_date})
Thank You for Your Time!
Appendix (1)

Links to Video Demonstrations in YouTube

Data

- Data Input: How to Use Forms
  - link

Information

- Query 1: Conversion Funnel
  - link
- Query 2: Profit Contribution Distribution
  - link

Insight

- Query 3: Optimal Inventory Policy Management
  - link
- Query 4: Users and Ratings for Collaborative Filtering
  - link
- Query 5: Facility Location: Where are our Order Coming From Geographically?
  - link
Appendix (2)

Works Cited:

Further Reading on Mathematical Modeling

Query 1: Conversion Funnel
- link
Query 2: Profit Contribution Distribution
- link
Query 3: Optimal Inventory Policy Management
- link
Query 4: Users and Ratings for Collaborative Filtering
- link
Query 5: Facility Location: Where are our Order Coming From Geographically?
- Link