hw1

shuangyu_zhao

2023-01-11

```
library(sqldf)
```

```
## Loading required package: gsubfn
```

```
## Loading required package: proto
```

```
## Warning in system2("/usr/bin/otool", c("-L", shQuote(DSO)), stdout = TRUE):
## running command ''/usr/bin/otool' -L '/Library/Frameworks/R.framework/Resources/
## library/tcltk/libs//tcltk.so'' had status 1
```

```
## Loading required package: RSQLite
```

```
order_details <- read.csv("/Users/apple/Desktop/STT811 appl_stat_model/data/northwind/order_details.csv
orders <- read.csv("/Users/apple/Desktop/STT811 appl_stat_model/data/northwind/orders.csv")
territories <- read.csv("/Users/apple/Desktop/STT811 appl_stat_model/data/northwind/territories.csv")
regions <- read.csv("/Users/apple/Desktop/STT811 appl_stat_model/data/northwind/regions.csv")
employee_territories <- read.csv("/Users/apple/Desktop/STT811 appl_stat_model/data/northwind/employee_t
employees <- read.csv("/Users/apple/Desktop/STT811 appl_stat_model/data/northwind/employees.csv")
customers <- read.csv("/Users/apple/Desktop/STT811 appl_stat_model/data/northwind/customers.csv")
shippers <- read.csv("/Users/apple/Desktop/STT811 appl_stat_model/data/northwind/shippers.csv")
suppliers <- read.csv("/Users/apple/Desktop/STT811 appl_stat_model/data/northwind/suppliers.csv")
products <- read.csv("/Users/apple/Desktop/STT811 appl_stat_model/data/northwind/products.csv")
categories <- read.csv("/Users/apple/Desktop/STT811 appl_stat_model/data/northwind/categories.csv")
```

1.Perform a sort of orders by employeeID, then by shipVia, and then by freight, for those orders by shipped to France.

```
ordered_order <- sqldf("SELECT *
                        FROM orders
                        WHERE shipCountry = 'France'
                        ORDER BY employeeID, shipVia, freight")
head(ordered_order)
```

```
##   orderID customerID employeeID               orderDate            requiredDate
## 1   10371      LAMAI          1 1996-12-03 00:00:00.000 1996-12-31 00:00:00.000
## 2   10671      FRANR          1 1997-09-17 00:00:00.000 1997-10-15 00:00:00.000
## 3   10850      VICTE          1 1998-01-23 00:00:00.000 1998-03-06 00:00:00.000
## 4   10525      BONAP          1 1997-05-02 00:00:00.000 1997-05-30 00:00:00.000
## 5   10827      BONAP          1 1998-01-12 00:00:00.000 1998-01-26 00:00:00.000
## 6   10789      FOLIG          1 1997-12-22 00:00:00.000 1998-01-19 00:00:00.000
##                shippedDate shipVia freight             shipName
## 1 1996-12-24 00:00:00.000       1    0.45     La maison d'Asie
## 2 1997-09-24 00:00:00.000       1   30.34  France restauration
## 3 1998-01-30 00:00:00.000       1   49.19 Victuailles en stock
## 4 1997-05-23 00:00:00.000       2   11.06             Bon app'
## 5 1998-02-06 00:00:00.000       2   63.54             Bon app'
## 6 1997-12-31 00:00:00.000       2  100.60    Folies gourmandes
##                shipAddress  shipCity shipRegion shipPostalCode shipCountry
## 1   1 rue Alsace-Lorraine  Toulouse       NULL          31000      France
## 2           54 rue Royale    Nantes       NULL          44000      France
## 3       2 rue du Commerce      Lyon       NULL          69004      France
## 4     12 rue des Bouchers Marseille       NULL          13008      France
## 5     12 rue des Bouchers Marseille       NULL          13008      France
## 6 184 chaussée de Tournai     Lille       NULL          59000      France
```

2. Which shipVia has the largest average cost?

```
ave_cost_max <- sqldf("SELECT shipVia,
                  AVG(freight) AS ave_cost
                  FROM orders
                  GROUP BY shipVia
                  ORDER BY ave_cost DESC
                  LIMIT 2")
ave_cost_max
```

```
##   shipVia ave_cost
## 1       2 86.64064
## 2       3 80.44122
```

shipVia = 2 has the largest average cost.

3. Which product category has the highest average UnitPrice? The Lowest?

```
categorie_unitprice<- sqldf("SELECT AVG(UnitPrice), categories.*
                             FROM products
                             INNER JOIN categories
                             ON categories.categoryID = products.categoryID
                             GROUP BY products.categoryID
                             ORDER BY AVG(UnitPrice)")
head(categorie_unitprice, 2)
```

```
##   AVG(UnitPrice) categoryID   categoryName                      description
## 1        20.2500          5 Grains/Cereals Breads crackers pasta and cereal
## 2        20.6825          8        Seafood                 Seaweed and fish
##         picture
## 1 5.790629e+304
## 2 5.790629e+304
```

```
tail(categorie_unitprice, 2)
```

```
##   AVG(UnitPrice) categoryID categoryName
## 7       37.97917          1    Beverages
## 8       54.00667          6 Meat/Poultry
##                              description       picture
## 7 Soft drinks coffees teas beers and ales 5.790629e+304
## 8                         Prepared meats 5.790629e+304
```

Meat/Poultry(6) has the highest average UnitPrice.

Grains/Cereals(5) has the lowest average UnitPrice.

4. Which products are supplied by a company in the United States?

```
product_usa <- sqldf("SELECT Country, CompanyName, suppliers.SupplierID, products.ProductID, products.P
                    FROM suppliers
                    INNER JOIN products
                    ON suppliers.SupplierID = products.SupplierID
                    GROUP BY products.ProductID
                    HAVING Country = 'USA'
                    ORDER BY suppliers.SupplierID")
product_usa
```

```
##    Country                 CompanyName SupplierID ProductID
## 1      USA  New Orleans Cajun Delights          2         4
## 2      USA  New Orleans Cajun Delights          2         5
## 3      USA  New Orleans Cajun Delights          2        65
## 4      USA  New Orleans Cajun Delights          2        66
## 5      USA   Grandma Kelly’s Homestead          3         6
## 6      USA   Grandma Kelly’s Homestead          3         7
## 7      USA   Grandma Kelly’s Homestead          3         8
## 8      USA           Bigfoot Breweries         16        34
## 9      USA           Bigfoot Breweries         16        35
## 10     USA           Bigfoot Breweries         16        67
## 11     USA New England Seafood Cannery         19        40
## 12     USA New England Seafood Cannery         19        41
##                          ProductName
## 1      Chef Anton’s Cajun Seasoning
## 2            Chef Anton’s Gumbo Mix
## 3  Louisiana Fiery Hot Pepper Sauce
## 4         Louisiana Hot Spiced Okra
## 5      Grandma’s Boysenberry Spread
## 6   Uncle Bob’s Organic Dried Pears
## 7        Northwoods Cranberry Sauce
## 8                     Sasquatch Ale
## 9                    Steeleye Stout
## 10        Laughing Lumberjack Lager
## 11                 Boston Crab Meat
## 12  Jack’s New England Clam Chowder
```

Chef Anton’s Cajun Seasoning, Chef Anton’s Gumbo Mix, Louisiana Fiery Hot Pepper Sauce and Louisiana Hot Spiced Okra are supplied by New Orleans Cajun Delights in USA.

Grandma’s Boysenberry Spread, Uncle Bob’s Organic Dried Pears and Northwoods Cranberry Sauce are supplied by Grandma Kelly’s Homestead in USA.

Sasquatch Ale, Steeleye Stout and Laughing Lumberjack Lager are supplied by Bigfoot Breweries in USA.

Boston Crab Meat and Jack's New England Clam Chowder are supplied by New England Seafood Cannery in USA.

5.Which shipper is shipping the largest number of units of product? Answer in terms of units; you do not need to consider quantityPerUnit here.

```
shipper_largest <- sqldf("SELECT order_details.quantity,
                         orders.shipName, orders.shipVia,
                         shippers.companyName
                         FROM order_details
                         INNER JOIN orders ON orders.orderID = order_details.orderID
                         INNER JOIN shippers ON orders.shipVia = shippers.shipperID
                         ORDER BY order_details.quantity DESC
                         LIMIT 3")
shipper_largest
```

```
##   quantity           shipName shipVia      companyName
## 1      130       Ernst Handel       3 Federal Shipping
## 2      130       Ernst Handel       2   United Package
## 3      120 Save-a-lot Markets       3 Federal Shipping
```

Federal Shipping and United Package are shipping the largest number of units of products.

6. Which employee is tied to the most sales revenue? Give the name, not the code, along with the total revenue for the employee.

```
order_revenue <- sqldf("SELECT orders.orderID, orders.employeeID,
                       order_details.unitPrice * order_details.quantity *( 1- order_details.discount) AS
                       FROM order_details
                       INNER JOIN orders
                       WHERE order_details.orderID = orders.orderID")

best_employee <- sqldf("SELECT revenue, order_revenue.employeeID,
                       employees.firstName, employees.lastName
                       FROM order_revenue
                       INNER JOIN employees
                       WHERE employees.employeeID = order_revenue.employeeID
                       ORDER BY revenue DESC
                       LIMIT 2")
best_employee
```

```
##   revenue employeeID firstName lastName
## 1 15810.0          1     Nancy  Davolio
## 2 15019.5          2    Andrew   Fuller
```

Nancy Davolio is tied to the most sales revenue.

7. Find the total revenue for each product category.

```
product_order_revenue <- sqldf("SELECT productID, orderID,
                              order_details.unitPrice * order_details.quantity *( 1- order_details.disc
                               FROM order_details
                              ORDER BY orderID")

totallrevenue_category <- sqldf("SELECT SUM(product_order_revenue.revenue) AS revenue_category,
                                products.CategoryID, categories.categoryName
                                FROM product_order_revenue
                                INNER JOIN products ON products.productID = product_order_revenue.produ
                                INNER JOIN categories ON products.CategoryID = categories.categoryID
                                GROUP BY products.CategoryID")
totallrevenue_category
```

```
##   revenue_category CategoryID   categoryName
## 1        267868.18          1      Beverages
## 2        106047.09          2     Condiments
## 3        167357.22          3    Confections
## 4        234507.29          4 Dairy Products
## 5         95744.59          5 Grains/Cereals
## 6        163022.36          6   Meat/Poultry
## 7         99984.58          7        Produce
## 8        131261.74          8        Seafood
```

8. Consider the amount of revenue for each customer. If there were no discounts applied, which customer would see the largest increase in cost?

```
order_revenue_incre <- sqldf("SELECT orders.orderID, orders.customerID, customers.companyName,
                       order_details.unitPrice * order_details.quantity * order_details.discount AS reve
                       FROM orders
                       INNER JOIN order_details ON order_details.orderID = orders.orderID
                       INNER JOIN customers ON orders.customerID = customers.customerID
                       ORDER BY revenue_increase DESC")
head(order_revenue_incre)
```

```
##   orderID customerID                  companyName revenue_increase
## 1   10353      PICCO             Piccolo und mehr         2108.000
## 2   10372      QUEEN                Queen Cozinha         2108.000
## 3   10424      MEREP               Mère Paillarde         2065.840
## 4   10912      HUNGO Hungry Owl All-Night Grocers         1856.850
## 5   11030      SAVEA           Save-a-lot Markets         1856.850
## 6   10993      FOLKO               Folk och fä HB         1547.375
```

Piccolo und mehr and Queen Cozinha have largest increase in cost.

9. Which order(s) has the most number of items (and how many)? Give the orderID for this one.

```
order_item <- sqldf("SELECT order_details.orderID,
                    SUM(order_details.quantity) AS item_num
                    FROM order_details
                    GROUP BY order_details.orderID
                    ORDER BY item_num DESC
                    ")

head(order_item)
```

```
##   orderID item_num
## 1   10895      346
## 2   11030      330
## 3   10847      288
## 4   10515      286
## 5   10678      280
## 6   10612      263
```

10895 has the most items.

10. Create a new field called "InventoryOrderRatio" which is, for each product, the UnitsinStock (the inventory) for the product (across all customers) divided by the quantity ordered for that product. A high value represents sufficient product in stock, while a low number represents products that are in danger of running out. What 3 products are most in danger of running out?

```
products_ior <- sqldf("SELECT CAST(products.UnitsInStock AS FLOAT) / CAST(SUM(order_details.quantity) AS
                      products.ProductID, products.ProductName
                      FROM products
                      INNER JOIN order_details ON order_details.productID = products.productID
                      GROUP BY order_details.productID
                      ORDER BY InventoryOrderRatio")
head(products_ior)
```

```
##   InventoryOrderRatio ProductID             ProductName
## 1         0.000000000         5  Chef Anton's Gumbo Mix
## 2         0.000000000        17            Alice Mutton
## 3         0.000000000        29 Thüringer Rostbratwurst
## 4         0.000000000        31       Gorgonzola Telino
## 5         0.000000000        53           Perth Pasties
## 6         0.002952756        21     Sir Rodney's Scones
```

Chef Anton's Gumbo Mix, Alice Mutton, Thüringer Rostbratwurst are most in danger of running out

11. A recommender engine looks at which pairs of products tend to be bought by the same customer, so that if a customer buys one, the recommender engine will recommend they buy the other. Find which product pairs are most likely to be bought by the same customer.

```
orders_sim <- sqldf("SELECT orders.orderID, orders.customerID, order_details.productID
               FROM orders
               INNER JOIN order_details
               ON orders.orderID = order_details.orderID")
head(orders_sim)
```

```
##   orderID customerID productID
## 1   10248      VINET        11
## 2   10248      VINET        42
## 3   10248      VINET        72
## 4   10249      TOMSP        14
## 5   10249      TOMSP        51
## 6   10250      HANAR        41
```

```
order_times <- sqldf("SELECT customerID, COUNT( DISTINCT orderID) AS order_time
                      FROM orders_sim
                      GROUP BY customerID")
head(order_times)
```

```
##   customerID order_time
## 1      ALFKI          6
## 2      ANATR          4
## 3      ANTON          7
## 4      AROUT         13
## 5      BERGS         18
## 6      BLAUS          7
```

```
pairs_order <-  sqldf("SELECT tabel1.customerID, tabel1.productID AS item1, tabel2.productID AS item2
                      FROM orders_sim AS tabel1
                      INNER JOIN orders_sim AS tabel2
                      ON tabel1.orderID = tabel2.orderID
                      AND tabel1.productID < tabel2.productID
                      GROUP BY tabel2.customerID, tabel2.orderID, tabel1.productID, tabel2.productID")
head(pairs_order, 20)
```

```
##    customerID item1 item2
## 1       ALFKI    28    39
## 2       ALFKI    28    46
## 3       ALFKI    39    46
## 4       ALFKI     3    76
## 5       ALFKI    59    77
## 6       ALFKI     6    28
## 7       ALFKI    58    71
## 8       ANATR    69    70
## 9       ANATR    14    42
## 10      ANATR    14    60
## 11      ANATR    42    60
## 12      ANATR    11    13
## 13      ANATR    11    19
## 14      ANATR    11    72
## 15      ANATR    13    19
## 16      ANATR    13    72
## 17      ANATR    19    72
## 18      ANTON    43    48
## 19      ANTON    11    40
## 20      ANTON    11    57
```

```
pair_order_times <- sqldf("SELECT pairs_order.*, COUNT(*) AS times
                          FROM pairs_order
                          GROUP BY customerID, item1, item2
                          ORDER BY times DESC")
head(pair_order_times)
```

```
##   customerID item1 item2 times
## 1      SAVEA     1    71     3
## 2      BERGS    39    54     2
## 3      BOTTM    10    62     2
## 4      ERNSH    12    24     2
## 5      ERNSH    57    64     2
## 6      FOLKO    39    47     2
```

```
prob_pair <- sqldf("SELECT pair_order_times.*, order_times.order_time AS total_order_times,
                   100 * (CAST(pair_order_times.times AS FLOAT) / CAST(order_times.order_time AS FLOAT)
                   FROM pair_order_times
                   INNER JOIN order_times
                   ON pair_order_times.customerID = order_times.customerID
                   ORDER BY prob_percentage DESC")
head(prob_pair)
```

```
##   customerID item1 item2 times total_order_times prob_percentage
## 1      CENTC    21    37     1                 1       100.00000
## 2      GROSR    10    75     1                 2        50.00000
## 3      GROSR    29    72     1                 2        50.00000
## 4      BOLID     4    57     1                 3        33.33333
## 5      BOLID     4    75     1                 3        33.33333
## 6      BOLID    17    29     1                 3        33.33333
```

21 and 37 are more likely to be bought by CENTC