STT 465 Homework 1

Derien Weatherspoon

2023-01-10

```
order_details <- read.csv("order_details.csv")
orders <- read.csv("orders.csv")
territories <- read.csv("territories.csv")
regions <- read.csv("regions.csv")
employee_territories <- read.csv("employee_territories.csv")
employees <- read.csv("employees.csv")
customers <- read.csv("customers.csv")
shippers <- read.csv("shippers.csv")
suppliers <- read.csv("suppliers.csv")
products <- read.csv("products.csv")
categories <- read.csv("categories.csv")
```

```
library(sqldf)
```

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

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

```
## Warning in doTryCatch(return(expr), name, parentenv, handler): unable to load shared object '/Library
##   dlopen(/Library/Frameworks/R.framework/Resources/modules//R_X11.so, 0x0006): Library not loaded: /
##   Referenced from: <51CE3E53-B0D5-3EC3-9C0E-347A62D97B61> /Library/Frameworks/R.framework/Versions/4
##   Reason: tried: '/opt/X11/lib/libSM.6.dylib' (no such file), '/System/Volumes/Preboot/Cryptexes/OS/
```

```
## Could not load tcltk.  Will use slower R code instead.
```

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

```
#Perform a sort of orders by employeeID, then by shipVia, and then by freight, for those orders by ship

order_sort <- sqldf("SELECT employeeID, shipVia, freight, shipCountry
                    FROM orders
                    ORDER BY
                      CASE WHEN shipCountry = 'France' THEN 0
                          ELSE 1
                      END,
                      shipCountry")
head(order_sort)
```

```
##   employeeID shipVia freight shipCountry
## 1          5       3   32.38      France
## 2          3       1   41.34      France
## 3          2       1   55.28      France
## 4          6       1    6.01      France
## 5          2       2    1.15      France
## 6          5       2    5.74      France
```

```
#Which shipVia has the largest average cost?
#summary, shipvia, freight,
larg_avg_cost <- sqldf("SELECT avg(freight), shipVia
                       FROM orders
                       ORDER BY freight")
larg_avg_cost
```

```
##   avg(freight) shipVia
## 1      78.2442       3
```

```
#Which product category has the highest average UnitPrice? The Lowest?
high_avg <- sqldf("SELECT avg(UnitPrice), categoryName
                  FROM products, categories
                  WHERE categories.categoryID = products.categoryID
                  GROUP BY UnitPrice, categoryName")

high_avg
```

```
##    avg(UnitPrice)   categoryName
## 1            2.50 Dairy Products
## 2            4.50      Beverages
## 3            6.00        Seafood
## 4            7.00 Grains/Cereals
## 5            7.45   Meat/Poultry
## 6            7.75      Beverages
## 7            9.00 Grains/Cereals
## 8            9.20    Confections
## 9            9.50    Confections
## 10           9.50        Seafood
## 11           9.65        Seafood
## 12          10.00     Condiments
## 13          10.00    Confections
## 14          10.00        Produce
## 15          12.00        Seafood
## 16          12.50    Confections
## 17          12.50 Dairy Products
## 18          12.75    Confections
## 19          13.00     Condiments
## 20          13.25        Seafood
## 21          14.00      Beverages
## 22          14.00    Confections
## 23          14.00 Grains/Cereals
## 24          15.00      Beverages
## 25          15.00        Seafood
## 26          15.50     Condiments
```

```
## 27          16.25    Confections
## 28          17.00     Condiments
## 29          17.45    Confections
## 30          18.00      Beverages
## 31          18.40        Seafood
## 32          19.00      Beverages
## 33          19.00        Seafood
## 34          19.45     Condiments
## 35          19.50 Grains/Cereals
## 36          20.00    Confections
## 37          21.00 Dairy Products
## 38          21.00 Grains/Cereals
## 39          21.05     Condiments
## 40          21.35     Condiments
## 41          21.50 Dairy Products
## 42          22.00     Condiments
## 43          23.25        Produce
## 44          24.00   Meat/Poultry
## 45          25.00     Condiments
## 46          25.89        Seafood
## 47          26.00        Seafood
## 48          28.50     Condiments
## 49          30.00        Produce
## 50          31.00        Seafood
## 51          31.23    Confections
## 52          32.00 Dairy Products
## 53          32.80   Meat/Poultry
## 54          33.25 Grains/Cereals
## 55          34.00 Dairy Products
## 56          34.80 Dairy Products
## 57          36.00 Dairy Products
## 58          38.00 Dairy Products
## 59          38.00 Grains/Cereals
## 60          39.00   Meat/Poultry
## 61          40.00     Condiments
## 62          43.90     Condiments
## 63          43.90    Confections
## 64          45.60        Produce
## 65          46.00      Beverages
## 66          49.30    Confections
## 67          53.00        Produce
## 68          55.00 Dairy Products
## 69          62.50        Seafood
## 70          81.00    Confections
## 71          97.00   Meat/Poultry
## 72         123.79   Meat/Poultry
## 73         263.50      Beverages
```

```
#Dairy has the lowest, beverages has the highest.
```

```
#Which products are supplied by a company in the United States?
supp_USA <- sqldf("SELECT ProductName, Country
                  FROM products, suppliers
                  WHERE products.supplierID = suppliers.supplierID
```

```
                  ORDER BY Country DESC")
head(supp_USA)
```

```
##                       ProductName Country
## 1    Chef Anton's Cajun Seasoning     USA
## 2          Chef Anton's Gumbo Mix     USA
## 3    Grandma's Boysenberry Spread     USA
## 4 Uncle Bob's Organic Dried Pears     USA
## 5      Northwoods Cranberry Sauce     USA
## 6                   Sasquatch Ale     USA
```

```
#Products shown in DESC, 12 products from USA.
```

```
#Which shipper is shipping the largest number of units of product? Answer in terms of units; you do not

ship_ship <- sqldf("SELECT shippers.shipperID, max(UnitsOnOrder), companyName
                   FROM products, shippers
                   WHERE products.supplierID = shippers.shipperID
                   ORDER BY companyName")
ship_ship
```

```
##   shipperID max(UnitsOnOrder)    companyName
## 1         2               100 United Package
```

```
#United Package, ships 100.
```

```
#Which employee is tied to the most sales revenue? Give the name, not the code, along with the total re
employee_rev<- sqldf("SELECT firstName, lastName,
                      unitPrice*quantity
                       AS revenue
                       FROM order_details, employees
                      ORDER BY revenue DESC")

head(employee_rev)
```

```
##   firstName  lastName revenue
## 1     Nancy   Davolio   15810
## 2    Andrew    Fuller   15810
## 3     Janet Leverling   15810
## 4  Margaret   Peacock   15810
## 5    Steven  Buchanan   15810
## 6   Michael    Suyama   15810
```

```
#Find the total revenue for each product category.
total_revenue <- sqldf("SELECT sum(order_details.quantity*order_details.unitPrice), ProductName
                       FROM order_details, products
                       WHERE products.ProductID =order_details.productID

                       GROUP BY ProductName")
total_revenue
```

```
##    sum(order_details.quantity*order_details.unitPrice)
## 1                                             35482.20
## 2                                              3080.00
## 3                                             19048.30
## 4                                             50286.00
## 5                                             31987.50
## 6                                             14277.60
## 7                                             18559.20
## 8                                             13150.80
## 9                                              9424.80
## 10                                             5801.15
## 11                                             1542.75
## 12                                           149984.20
## 13                                             6664.75
## 14                                             3383.80
## 15                                            20876.50
## 16                                             1713.50
## 17                                             1813.50
## 18                                            45121.20
## 19                                            16172.50
## 20                                             7345.00
## 21                                             3047.20
## 22                                             4782.60
## 23                                            24307.20
## 24                                            10524.20
## 25                                            21534.90
## 26                                             7232.40
## 27                                            22140.20
## 28                                            14542.60
## 29                                            25079.20
## 30                                             9098.10
## 31                                             5234.40
## 32                                            16794.00
## 33                                             2562.00
## 34                                             2566.00
## 35                                            14607.00
## 36                                             3519.00
## 37                                            44742.60
## 38                                             9171.20
## 39                                             9500.00
## 40                                             8827.00
## 41                                            25738.80
## 42                                            14775.54
## 43                                            13760.00
## 44                                             4051.60
## 45                                             9685.00
## 46                                            11472.00
## 47                                            18748.05
## 48                                            21510.20
## 49                                            19512.00
## 50                                            13902.00
## 51                                            12866.80
## 52                                            76296.00
## 53                                             7807.80
```

54		8650.55
55		4740.50
56		4200.00
57		26865.60
58		6678.00
59		15231.50
60		9362.50
61		9332.40
62		23635.80
63		9636.00
64		16438.80
65		6144.00
66		14536.80
67		49827.90
68		6159.50
69		87736.40
70		8630.40
71		5121.00
72		4840.20
73		22464.00
74		3510.00
75		17696.30
76		23009.00
77		4358.60

	ProductName
1	Alice Mutton
2	Aniseed Syrup
3	Boston Crab Meat
4	Camembert Pierrot
5	Carnarvon Tigers
6	Chai
7	Chang
8	Chartreuse verte
9	Chef Anton’s Cajun Seasoning
10	Chef Anton’s Gumbo Mix
11	Chocolade
12	Côte de Blaye
13	Escargots de Bourgogne
14	Filo Mix
15	Flotemysost
16	Geitost
17	Genen Shouyu
18	Gnocchi di nonna Alice
19	Gorgonzola Telino
20	Grandma’s Boysenberry Spread
21	Gravad lax
22	Guaraná Fantástica
23	Gudbrandsdalsost
24	Gula Malacca
25	Gumbär Gummibärchen
26	Gustaf’s Knäckebröd
27	Ikura
28	Inlagd Sill
29	Ipoh Coffee

```
## 30  Jack's New England Clam Chowder
## 31                            Konbu
## 32                     Lakkalikööri
## 33        Laughing Lumberjack Lager
## 34                    Longlife Tofu
## 35 Louisiana Fiery Hot Pepper Sauce
## 36        Louisiana Hot Spiced Okra
## 37            Manjimup Dried Apples
## 38               Mascarpone Fabioli
## 39                          Maxilaku
## 40                  Mishi Kobe Niku
## 41           Mozzarella di Giovanni
## 42            Nord-Ost Matjeshering
## 43       Northwoods Cranberry Sauce
## 44          NuNuCa Nuß-Nougat-Creme
## 45  Original Frankfurter grüne Soße
## 46                     Outback Lager
## 47                           Pavlova
## 48                     Perth Pasties
## 49                      Pâté chinois
## 50                    Queso Cabrales
## 51        Queso Manchego La Pastora
## 52             Raclette Courdavault
## 53                   Ravioli Angelo
## 54             Rhönbräu Klosterbier
## 55                       Rogede sild
## 56                        Röd Kaviar
## 57                Rössle Sauerkraut
## 58                     Sasquatch Ale
## 59               Schoggi Schokolade
## 60              Scottish Longbreads
## 61    Singaporean Hokkien Fried Mee
## 62           Sir Rodney's Marmalade
## 63              Sir Rodney's Scones
## 64                   Sirop d'érable
## 65                        Spegesild
## 66                   Steeleye Stout
## 67                   Tarte au sucre
## 68       Teatime Chocolate Biscuits
## 69          Thüringer Rostbratwurst
## 70                              Tofu
## 71                       Tourtière
## 72                        Tunnbröd
## 73  Uncle Bob's Organic Dried Pears
## 74                 Valkoinen suklaa
## 75                     Vegie-spread
## 76        Wimmers gute Semmelknödel
## 77                     Zaanse koeken
```

```
#How do I incorporate productName?
```

```
#Consider the amount of revenue for each customer. If there were no discounts applied, which customer w
```

```
increase_in_cost <- sqldf("SELECT customerID,
```

```
                      unitPrice*quantity-discount
                       AS revenue
                       FROM order_details, customers
                      ORDER BY revenue DESC")
head(increase_in_cost)
```

```
##   customerID revenue
## 1      ALFKI   15810
## 2      ANATR   15810
## 3      ANTON   15810
## 4      AROUT   15810
## 5      BERGS   15810
## 6      BLAUS   15810
```

```
#same as problem 6?
```

```
#Which order(s) has the most number of items (and how many)? Give the orderID for this one.
most_items <- sqldf("SELECT max(quantity), orders.orderID
                    FROM orders, order_details
                    WHERE orders.orderID = order_details.orderID")
most_items
```

```
##   max(quantity) orderID
## 1           130   10764
```

```
#Create a new field called "InventoryOrderRatio" which is, for each product, the UnitsinStock (the inve
```

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

new_inventory
```

```
##                         ProductName InventoryOrderRatio
## 1                              Chai                   0
## 2                             Chang                   0
## 3                     Aniseed Syrup                   0
## 4      Chef Anton's Cajun Seasoning                   0
## 5            Chef Anton's Gumbo Mix                   0
## 6      Grandma's Boysenberry Spread                   0
## 7   Uncle Bob's Organic Dried Pears                   0
## 8        Northwoods Cranberry Sauce                   0
## 9                    Mishi Kobe Niku                   0
## 10                           Ikura                   0
## 11                  Queso Cabrales                   0
## 12       Queso Manchego La Pastora                   0
## 13                           Konbu                   0
```

```
## 14                          Tofu                0
## 15                  Genen Shouyu                0
## 16                       Pavlova                0
## 17                  Alice Mutton                0
## 18               Carnarvon Tigers               0
## 19     Teatime Chocolate Biscuits               0
## 20         Sir Rodney's Marmalade               0
## 21            Sir Rodney's Scones               0
## 22            Gustaf's Knäckebröd               0
## 23                        Tunnbröd               0
## 24             Guaraná Fantástica                0
## 25        NuNuCa Nuß-Nougat-Creme                0
## 26            Gumbär Gummibärchen                0
## 27             Schoggi Schokolade                0
## 28              Rössle Sauerkraut                0
## 29        Thüringer Rostbratwurst                0
## 30          Nord-Ost Matjeshering                0
## 31              Gorgonzola Telino                0
## 32             Mascarpone Fabioli                0
## 33                        Geitost                0
## 34                  Sasquatch Ale                0
## 35                 Steeleye Stout                0
## 36                    Inlagd Sill                0
## 37                     Gravad lax                0
## 38                  Côte de Blaye                0
## 39               Chartreuse verte                0
## 40               Boston Crab Meat                0
## 41  Jack's New England Clam Chowder              0
## 42   Singaporean Hokkien Fried Mee               0
## 43                     Ipoh Coffee               0
## 44                   Gula Malacca                0
## 45                    Rogede sild                0
## 46                      Spegesild                0
## 47                  Zaanse koeken                0
## 48                      Chocolade                0
## 49                       Maxilaku                0
## 50               Valkoinen suklaa                0
## 51           Manjimup Dried Apples               0
## 52                       Filo Mix                0
## 53                  Perth Pasties                0
## 54                      Tourtière                0
## 55                    Pâté chinois               0
## 56          Gnocchi di nonna Alice               0
## 57                  Ravioli Angelo               0
## 58          Escargots de Bourgogne               0
## 59            Raclette Courdavault               0
## 60               Camembert Pierrot               0
## 61                  Sirop d'érable               0
## 62                  Tarte au sucre               0
## 63                    Vegie-spread               0
## 64       Wimmers gute Semmelknödel               0
## 65 Louisiana Fiery Hot Pepper Sauce              0
## 66       Louisiana Hot Spiced Okra               0
## 67       Laughing Lumberjack Lager               0
```

```
## 68              Scottish Longbreads                   0
## 69                 Gudbrandsdalsost                   0
## 70                    Outback Lager                   0
## 71                      Flotemysost                   0
## 72           Mozzarella di Giovanni                   0
## 73                       Röd Kaviar                   0
## 74                   Longlife Tofu                   0
## 75             Rhönbräu Klosterbier                   0
## 76                      Lakkalikööri                   0
## 77  Original Frankfurter grüne Soße                   0
```

```
#A recommender engine looks at which pairs of products tend to be bought by the same customer, so that


frequently_purchased <- sqldf("SELECT order_details.productID, orders.orderID, orders.customerID FROM o
       INNER JOIN orders
       ON orders.orderID = order_details.orderID")
frequently_purchased <- sqldf("SELECT orders.productID, order_details.productID, COUNT(*) as FrequencyO
                              INNER JOIN frequently_purchased order_details ON orders.customerID = orde

frequently_purchased[which.max(frequently_purchased$FrequencyOfPurchase),]
```

```
##     productID productID FrequencyOfPurchase
## 722        21        61                   8
```