HW 1 Solutions

Paul Speaker

2023-01-28

```
library(sqldf)
```

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

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

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

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
```

```
## v ggplot2 3.4.0      v purrr   1.0.0
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
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")
```

#1 Perform a sort of orders by employeeID, then by shipVia, and then by freight, # for those orders by shipped to France. Order_France <- sqldf("SELECT * FROM orders WHERE shipCountry = 'France' ORDER BY shipVia, freight")

```
#1 Perform a sort of orders by employeeID, then by shipVia, and then by freight,
# for those orders by shipped to France.
Order_France <- sqldf("SELECT * FROM orders WHERE shipCountry = 'France' ORDER BY shipVia, freight")
glimpse(Order_France)
```

```
## Rows: 77
## Columns: 14
## $ orderID        <int> 10371, 10631, 10738, 10683, 10274, 10826, 10559, 10331,~
## $ customerID     <chr> "LAMAI", "LAMAI", "SPECD", "DUMON", "VINET", "BLONP", "~
## $ employeeID     <int> 1, 8, 2, 2, 6, 6, 6, 9, 8, 3, 5, 4, 5, 8, 1, 7, 3, 6, 1~
```

```
## $ orderDate      <chr> "1996-12-03 00:00:00.000", "1997-08-14 00:00:00.000", "~
## $ requiredDate   <chr> "1996-12-31 00:00:00.000", "1997-09-11 00:00:00.000", "~
## $ shippedDate    <chr> "1996-12-24 00:00:00.000", "1997-08-15 00:00:00.000", "~
## $ shipVia        <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ freight        <dbl> 0.45, 0.87, 2.91, 4.40, 6.01, 7.09, 8.05, 10.19, 11.26,~
## $ shipName       <chr> "La maison d'Asie", "La maison d'Asie", "Spécialités du~
## $ shipAddress    <chr> "1 rue Alsace-Lorraine", "1 rue Alsace-Lorraine", "25 r~
## $ shipCity       <chr> "Toulouse", "Toulouse", "Paris", "Nantes", "Reims", "St~
## $ shipRegion     <chr> "NULL", "NULL", "NULL", "NULL", "NULL", "NULL", "NULL",~
## $ shipPostalCode <chr> "31000", "31000", "75016", "44000", "51100", "67000", "~
## $ shipCountry    <chr> "France", "France", "France", "France", "France", "Fran~
```

```
#2 Which shipVia has the largest average cost?
order_cost <-sqldf("SELECT ShipVia, AVG(freight) FROM orders GROUP BY ShipVia")
glimpse(order_cost)
```

```
## Rows: 3
## Columns: 2
## $ shipVia        <int> 1, 2, 3
## $ `AVG(freight)` <dbl> 65.00133, 86.64064, 80.44122
```

```
# shipVia 2 has the largest cost.  This corresponds to shipper United Package
# (since the field had a different name I will accept either).
```

```
#3 Which product category has the highest average UnitPrice?  The Lowest?
average_cost <-sqldf("SELECT CategoryID, AVG(UnitPrice) FROM products GROUP BY CategoryID")
category_costs <- sqldf("Select average_cost.*, categories.categoryName FROM average_cost INNER JOIN ca
category_costs
```

```
##   CategoryID AVG(UnitPrice)   categoryName
## 1          1       37.97917      Beverages
## 2          2       23.06250     Condiments
## 3          3       25.16000    Confections
## 4          4       28.73000 Dairy Products
## 5          5       20.25000 Grains/Cereals
## 6          6       54.00667   Meat/Poultry
## 7          7       32.37000        Produce
## 8          8       20.68250        Seafood
```

```
# Highest category cost is Meat/Poultry, Lowest is Grain/Cereals
```

```
#4 Which products are supplied by a company in the United States?
USproducts <- sqldf("SELECT products.ProductName FROM products INNER JOIN suppliers WHERE products.Suppl
USproducts
```

```
##                         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        Northwoods Cranberry Sauce
## 7   Uncle Bob's Organic Dried Pears
## 8         Laughing Lumberjack Lager
## 9                     Sasquatch Ale
## 10                  Steeleye Stout
## 11                Boston Crab Meat
```

```
## 12  Jack's New England Clam Chowder
```

```
#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.
full_orders <- sqldf("SELECT orders.*, order_details.productID, order_details.unitPrice, order_details.
total_shipQ <- sqldf("SELECT SUM(full_orders.quantity), shippers.companyName FROM full_orders INNER JOI
total_shipQ
```

```
##   SUM(full_orders.quantity)      companyName
## 1                     15919   Speedy Express
## 2                     19945   United Package
## 3                     15453 Federal Shipping
```

```
# United Package, which is shipping 19,945 units
```

```
#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 *, unitPrice*quantity*(1-discount) as revenue from order_details")
employee_revenue <- sqldf("select orders.employeeID, SUM(order_revenue.revenue) FROM orders INNER JOIN 
revenue_by_name <- sqldf("SELECT employee_revenue.*, employees.lastName, employees.firstName FROM emplo
revenue_by_name
```

```
##   employeeID SUM(order_revenue.revenue)  lastName firstName
## 1          1                  192107.60   Davolio     Nancy
## 2          2                  166537.76    Fuller    Andrew
## 3          3                  202812.84 Leverling     Janet
## 4          4                  232890.85   Peacock  Margaret
## 5          5                   68792.28  Buchanan    Steven
## 6          6                   73913.13    Suyama   Michael
## 7          7                  124568.24      King    Robert
## 8          8                  126862.28  Callahan     Laura
## 9          9                   77308.07 Dodsworth      Anne
```

```
# Margeret Peacock has the most revenue
```

```
#7 Find the total revenue for each product category.
product_category <- sqldf("SELECT categories.categoryName, products.ProductID FROM categories INNER JOI
Cat_revenue <- sqldf("SELECT product_category.categoryName, SUM(order_revenue.revenue) FROM product_cat
Cat_revenue
```

```
##     categoryName SUM(order_revenue.revenue)
## 1      Beverages                  267868.18
## 2     Condiments                  106047.09
## 3    Confections                  167357.22
## 4 Dairy Products                  234507.29
## 5 Grains/Cereals                   95744.59
## 6   Meat/Poultry                  163022.36
## 7        Produce                   99984.58
## 8        Seafood                  131261.74
```

```
#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_disc <- sqldf("SELECT unitPrice*quantity*discount as revenue_disc, orders.customerID FROM order_d
total_disc <- sqldf("SELECT customers.companyName, SUM(order_disc.revenue_disc) FROM customers INNER JO
# Answer is Save-a-lot Markets
```

```
#9 Which order(s) has the most number of items (and how many)?  Give the orderID for this one.
item_count = sqldf("SELECT orderID, COUNT(productID) FROM order_details GROUP BY orderID ORDER BY count
```

```
glimpse(item_count)
```

```
## Rows: 830
## Columns: 2
## $ orderID            <int> 11077, 10979, 10847, 10657, 11064, 11031, 11021, 10~
## $ `COUNT(productID)` <int> 25, 6, 6, 6, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,~
```

```
# OrderID 11077 has 25 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?

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

inventory_data$quant <- as.numeric(inventory_data$total_q)

inventory_ratio <- sqldf("SELECT ProductName, UnitsInStock/quant AS InventoryOrderRatio
                          FROM inventory_data
                          WHERE UnitsInStock > 0 AND total_q > 0
                          ORDER BY UnitsInStock/quant")

inventory_ratio
```

```
##                         ProductName InventoryOrderRatio
## 1               Sir Rodney's Scones         0.002952756
## 2               Scottish Longbreads         0.007509387
## 3                       Rogede sild         0.009842520
## 4                 Camembert Pierrot         0.012048193
## 5                     Longlife Tofu         0.013468013
## 6                    Tarte au sucre         0.015697138
## 7                             Chang         0.016083254
## 8        Northwoods Cranberry Sauce         0.016129032
## 9             Nord-Ost Matjeshering         0.016339869
## 10           Gnocchi di nonna Alice         0.016627078
## 11       Louisiana Hot Spiced Okra         0.016736402
## 12           Mozzarella di Giovanni         0.017369727
## 13               Guaraná Fantástica         0.017777778
## 14                    Outback Lager         0.018359853
```

```
## 15                         Maxilaku        0.019230769
## 16  Uncle Bob's Organic Dried Pears        0.019659240
## 17              Gumbär Gummibärchen        0.019920319
## 18           Manjimup Dried Apples         0.022573363
## 19                  Steeleye Stout        0.022650057
## 20                     Flotemysost        0.024597919
## 21                         Pavlova        0.025043178
## 22                           Konbu        0.026936027
## 23                   Côte de Blaye        0.027287319
## 24                       Tourtière        0.027814570
## 25                     Ipoh Coffee        0.029310345
## 26        Wimmers gute Semmelknödel        0.029729730
## 27               Mascarpone Fabioli        0.030303030
## 28                   Queso Cabrales        0.031161473
## 29       Teatime Chocolate Biscuits        0.034578147
## 30                 Gudbrandsdalsost        0.036414566
## 31    Singaporean Hokkien Fried Mee        0.037302726
## 32                   Aniseed Syrup        0.039634146
## 33  Original Frankfurter grüne Soße        0.040455120
## 34                Rössle Sauerkraut        0.040625000
## 35                            Ikura        0.041778976
## 36                     Gula Malacca        0.044925125
## 37                             Chai        0.047101449
## 38            Raclette Courdavault         0.052807487
## 39                     Vegie-spread        0.053932584
## 40                     Lakkalikööri        0.058103976
## 41                   Zaanse koeken         0.074226804
## 42                         Filo Mix        0.076000000
## 43                 Carnarvon Tigers        0.077922078
## 44                   Ravioli Angelo        0.082949309
## 45                             Tofu        0.086633663
## 46  Jack's New England Clam Chowder        0.086646279
## 47                 Chartreuse verte        0.087011349
## 48                       Gravad lax        0.088000000
## 49 Louisiana Fiery Hot Pepper Sauce        0.102013423
## 50                         Tunnbröd        0.105172414
## 51             Rhönbräu Klosterbier        0.108225108
## 52                         Chocolade        0.108695652
## 53                Boston Crab Meat         0.111514053
## 54           Escargots de Bourgogne        0.116104869
## 55     Chef Anton's Cajun Seasoning        0.116997792
## 56                      Pâté chinois        0.127353267
## 57           Sir Rodney's Marmalade        0.127795527
## 58               Schoggi Schokolade        0.134246575
## 59                       Inlagd Sill        0.139130435
## 60                           Geitost        0.148344371
## 61                       Spegesild          0.173357664
## 62                   Sirop d'érable         0.187396352
## 63                    Sasquatch Ale        0.219367589
## 64          NuNuCa Nuß-Nougat-Creme        0.238993711
## 65        Queso Manchego La Pastora        0.250000000
## 66                 Valkoinen suklaa        0.276595745
## 67        Laughing Lumberjack Lager        0.282608696
## 68              Gustaf's Knäckebröd        0.298850575
```

```
## 69                   Mishi Kobe Niku         0.305263158
## 70                      Genen Shouyu         0.319672131
## 71                        Röd Kaviar         0.344709898
## 72     Grandma's Boysenberry Spread          0.398671096
```

```
# 3 smallest inventory_ratios are for Sir Rodney's Scones, Scottish Longbreads, and Rogede sild
# ok if ones where inventory are 0 are included--these will be a different list.
```

```
# 11 A recommender engine looks at which pairs of products tend to be bought by the same # customer, so
# (Hint: this will require a creative join with the orders data)
full_orders <- sqldf("SELECT orders.customerID, order_details.productID
                     FROM orders
                     INNER JOIN order_details
                     ON orders.orderID = order_details.orderID")

fo2 <- sqldf("SELECT customerID, productID AS prod1 FROM full_orders")
order_combos <- sqldf("SELECT fo2.customerID, fo2.prod1, full_orders.productID AS prod2
                      FROM fo2
                      INNER JOIN full_orders
                      ON fo2.customerID = full_orders.customerID
                      WHERE fo2.prod1 > full_orders.productID")

combo_count <- sqldf("SELECT prod1, prod2, COUNT(customerID) FROM order_combos
                     GROUP BY prod1, prod2
                     ORDER BY COUNT(customerID) DESC")
glimpse(combo_count)
```

```
## Rows: 2,893
## Columns: 3
## $ prod1               <int> 56, 60, 31, 75, 41, 41, 71, 75, 62, 31, 31, 59, 59~
## $ prod2               <int> 31, 2, 2, 2, 2, 31, 31, 41, 56, 17, 24, 2, 24, 31,~
## $ `COUNT(customerID)` <int> 62, 61, 59, 58, 57, 57, 57, 57, 56, 55, 54, 54, 54~
```

```
# Most common order pair is product ID 31 and 56, corresponding to Gorgonzola Telino and Gnocchi di non
# Gnocchi with a Gorgonzola sauce is amazing.
```