HW1

Mikayla Norton

2023-01-10

```
# Load data
library(sqldf)
```

```
## Loading required package: gsubfn

## Loading required package: proto

## Warning in doTryCatch(return(expr), name, parentenv, handler): unable to load shared object '/Librar
##   dlopen(/Library/Frameworks/R.framework/Resources/modules//R_X11.so, 0x0006): Library not loaded: /
##   Referenced from: <05451E21-B5F6-3B2F-9C0F-3EA08D57DC34> /Library/Frameworks/R.framework/Versions/4
##   Reason: tried: '/opt/X11/lib/libSM.6.dylib' (fat file, but missing compatible architecture (have '

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

## Loading required package: RSQLite
```

```
library(tidyverse)
```

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

## v ggplot2 3.4.0      v purrr   1.0.1
## 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("orders/order_details.csv")
orders <- read.csv("orders/orders.csv")
territories <- read.csv("orders/territories.csv")
regions <- read.csv("orders/regions.csv")
employee_territories <- read.csv("orders/employee_territories.csv")
employees <- read.csv("orders/employees.csv")
customers <- read.csv("orders/customers.csv")
shippers <- read.csv("orders/shippers.csv")
suppliers <- read.csv("orders/suppliers.csv")
products <- read.csv("orders/products.csv")
categories <- read.csv("orders/categories.csv")
```

Problem 1

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

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

```
##   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
```

Problem 2

Which shipVia has the largest average cost?

```
order_cost <- sqldf("SELECT order_details.*, unitPrice*quantity*(1-discount) AS cost
                       FROM order_details")
```

```
avgCost <- sqldf("SELECT shipVia, avg(cost) as AvgCost
                 FROM orders
                 INNER JOIN order_cost
                 WHERE orders.orderID = order_cost.orderID
                 GROUP BY shipVia")
maxAvgCost <- sqldf("SELECT shipVia, max(AvgCost)
                    FROM avgCost")
maxAvgCost
```

```
##   shipVia max(AvgCost)
## 1       2      617.532
```

Problem 3

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

```
avgUnitPrice <- sqldf("SELECT CategoryID, avg(UnitPrice) as AvgUnitPrice
                      FROM products
                      GROUP BY CategoryID
                      ORDER BY avg(UnitPrice)")
```

```
maxAvgPrice <- sqldf("SELECT CategoryID, max(AvgUnitPrice)
                     FROM avgUnitPrice")

minAvgPrice <- sqldf("SELECT CategoryID, min(AvgUnitPrice)
                     FROM avgUnitPrice")
maxAvgPrice
```

```
##   CategoryID max(AvgUnitPrice)
## 1          6          54.00667
```

```
minAvgPrice
```

```
##   CategoryID min(AvgUnitPrice)
## 1          5             20.25
```

Problem 4

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

```
productUSA <- sqldf("SELECT DISTINCT ProductName
                     FROM products
                     INNER JOIN suppliers
                     WHERE products.SupplierID = suppliers.SupplierID
                     AND country='USA'")
head(productUSA)
```

```
##                        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
```

Problem 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.

```
BigShipper <- sqldf("SELECT shipperID, sum(quantity) as UnitsShipped
                     FROM orders
                     INNER JOIN order_details
                     INNER JOIN shippers
                     ON orders.orderID = order_details.orderID
                     AND shippers.shipperID = orders.shipVia
                     GROUP BY shipperID
                     ORDER BY UnitsShipped DESC
                     LIMIT 1")
BigShipper
```

```
##   shipperID UnitsShipped
## 1         2        19945
```

Problem 6

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

```
maxEmployee <- sqldf("SELECT firstName, lastName, sum(unitPrice*quantity*(1-discount)) AS revenue
                     FROM orders
                     INNER JOIN order_details
                     INNER JOIN employees
                     ON orders.orderID = order_details.orderID
                     AND orders.employeeID = employees.employeeID
                     GROUP BY orders.employeeID
                     ORDER BY revenue DESC
                     LIMIT 1")
maxEmployee
```

```
##   firstName lastName  revenue
## 1  Margaret  Peacock 232890.8
```

Problem 7

Find the total revenue for each product category.

```
categorical_revenue <- sqldf("SELECT products.CategoryID, order_details.unitPrice*quantity*(1-discount)
                             FROM order_details
                             INNER JOIN products
                             INNER JOIN categories
                             ON order_details.productID = products.ProductID
                             AND products.CategoryID = categories.categoryID
                             GROUP BY products.CategoryID")

head(categorical_revenue)
```

```
##   CategoryID revenue
## 1          1  604.80
## 2          2  214.20
## 3          3 2462.40
## 4          4  168.00
## 5          5   98.00
## 6          6  342.72
```

Problem 8

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

```
CostInc <- sqldf("SELECT customerID, sum(unitPrice*quantity - unitPrice*quantity*(1-discount)) AS CostI
                             FROM order_details
                             INNER JOIN orders
                             WHERE order_details.orderID = orders.orderID
                            GROUP BY customerID
                            ORDER BY CostIncrease DESC
                            LIMIT 1")
CostInc
```

```
##   customerID CostIncrease
## 1      SAVEA     11311.44
```

Problem 9

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

```
mostItems <- sqldf("SELECT orderID, sum(quantity) as totalItems
                   FROM order_details
                   GROUP BY orderID")
mostItems <- sqldf("SELECT orderID, max(totalItems)
                   FROM mostItems")
mostItems
```

```
##   orderID max(totalItems)
## 1   10895             346
```

Problem 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?

```
products2 <- sqldf("SELECT products.*, sum(quantity) as QuantityOrdered, UnitsinStock/sum(quantity) as
                  FROM products
                  INNER JOIN order_details
                  WHERE products.ProductID = order_details.productID
                  GROUP BY order_details.productID
                  ORDER BY InventoryOrderRatio
                  LIMIT 3")
products2
```

```
##   ProductID   ProductName SupplierID CategoryID     QuantityPerUnit UnitPrice
## 1         1          Chai          1          1  10 boxes x 20 bags        18
## 2         2         Chang          1          1  24 - 12 oz bottles        19
## 3         3 Aniseed Syrup          1          2 12 - 550 ml bottles        10
##   UnitsInStock UnitsOnOrder ReorderLevel Discontinued QuantityOrdered
## 1           39            0           10            0             828
## 2           17           40           25            0            1057
## 3           13           70           25            0             328
##   InventoryOrderRatio
## 1                   0
## 2                   0
## 3                   0
```

Problem 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.

```
library(sqldf)
rec <- sqldf("SELECT products.ProductID, ProductName, customers.customerID
             FROM products
             INNER JOIN customers
             INNER JOIN orders
             INNER JOIN order_details
             ON orders.orderID = order_details.orderID
             AND orders.customerID = customers.customerID
             AND order_details.ProductID = products.ProductID")


productsCustomers <- sqldf("SELECT v1.ProductName as Product1, v2.ProductName as Product2, count(DISTIN
               FROM rec v1
               INNER JOIN rec v2
               ON v1.customerID = v2.customerID
               WHERE
                   v1.ProductID < v2.ProductID
               GROUP BY
                   v1.ProductName, v2.ProductName
               ORDER BY totalCustomers DESC")
head(productsCustomers)
```

```
##                 Product1                        Product2 totalCustomers
## 1   Raclette Courdavault                    Lakkalikööri             25
## 2   Raclette Courdavault          Mozzarella di Giovanni             22
## 3 Gnocchi di nonna Alice            Raclette Courdavault             21
## 4      Gorgonzola Telino Jack's New England Clam Chowder             21
## 5      Gorgonzola Telino            Raclette Courdavault             21
## 6                Pavlova            Raclette Courdavault             21
```