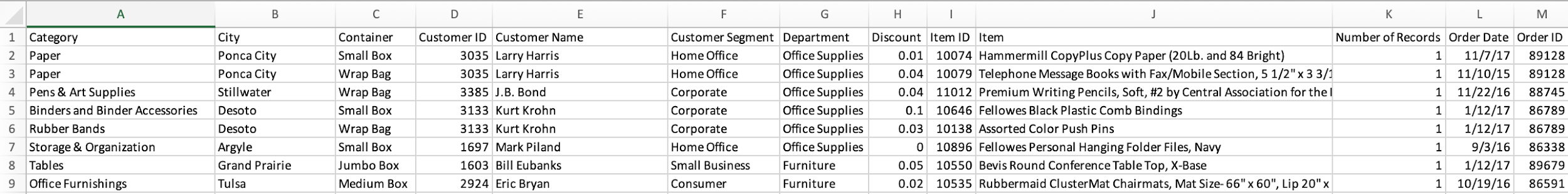
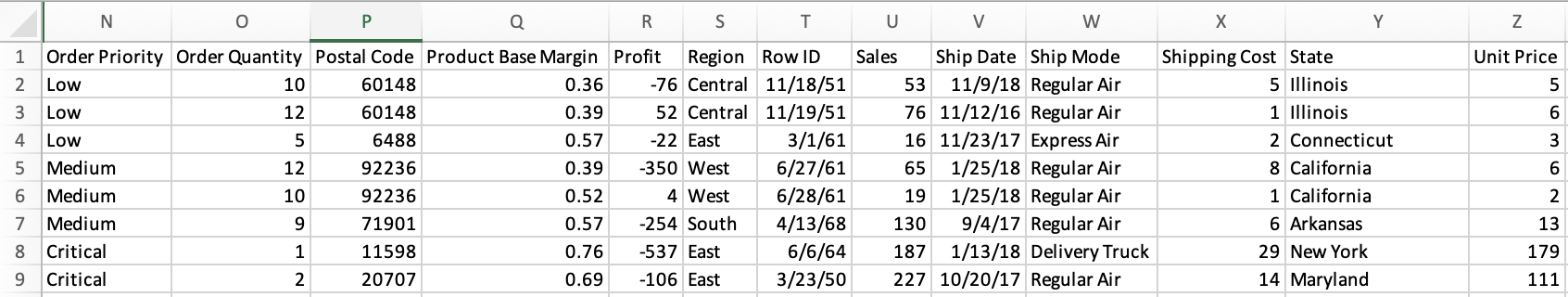
# **R for Data Science – Chapters 1 to 7 Summative Lab**

## *Analyzing Superstore Data*



For this lab, we’ll be using a fictional dataset containing information from orders at a superstore. This same dataset was used in your early days of learning Tableau last semester. The following screenshots includes a sample of the dataset, including all column names:





For each exercise below, paste the code and any outputs that it creates below each question.

1. Import the Superstore.xlsx file.

library(readxl)

Superstore <- read\_excel("data/Superstore.xlsx")

1. Create an R script for the exercises below. Call it “Superstore Lab – YourLastName.R”.

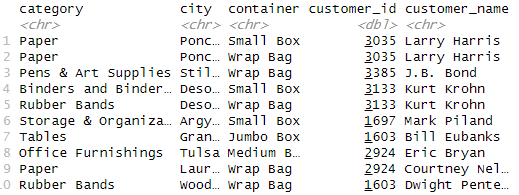


1. Is this dataset “tidy” according to the definition above figure 5.1 (the usual data science meaning)?

Yes, this dataset is “tidy.” There are no missing values or value type discrepancies.

1. Use the clean\_names function in the janitor package to make all column names snake case (Shipping Cost -> shipping\_cost).

Superstore <- clean\_names(Superstore)



1. Keep only the following columns: “category”, “department”, “discount”, “order\_date”, “order\_priority”, “order\_quantity”, “profit”, “region”, “sales”, “ship\_mode”, “shipping\_cost”, “state”, “customer\_name”.

Superstore <- Superstore |>

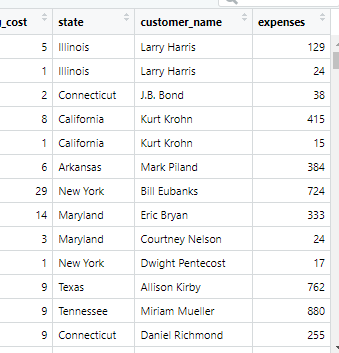
select(category, department, discount, order\_date, order\_priority, order\_quantity, profit, region, sales, ship\_mode, shipping\_cost, state, customer\_name)

1. Create a new column called “expenses” that uses the formula “sales” – “profit”. Save this new column to the existing dataframe.

Superstore <- mutate(Superstore,

expenses = sales - profit

)



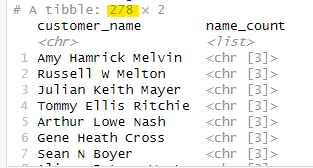
1. Create a new column called “large\_discount” that returns “Yes” if the discount > 0.10 and “No” otherwise.

Superstore <- mutate(Superstore,

large\_discount = ifelse(discount > 0.1, large\_discount <- "Yes", large\_discount <- "No"))



1. You’ve been informed that a customer with a middle name is making fraudulent orders. Determine the number of customers that have used a middle name in their order. Do not count the same customer twice.



Superstore |>

distinct(customer\_name) |>

mutate(

name\_count = strsplit(customer\_name, split = " ")

) |>

filter(lengths(name\_count) > 2)

1. Remove all customers with a middle name from the dataframe and save the change.

df <- Superstore |>

distinct(customer\_name) |>

mutate(

name\_count = strsplit(customer\_name, split = " ")

) |>

filter(lengths(name\_count) > 2)

Superstore <- anti\_join(Superstore, df, by = "customer\_name")

1. Use relocate to display the categorical columns first and save this change to the dataframe.

Superstore <- Superstore |>

relocate(where(is.character))

1. Do each of the following and save the results to a new dataframe.
   1. For the “East” region, calculate the sum of the sales for each state.

elevendf <- Superstore |>

filter(region == "East") |>

group\_by(state) |>

summarize(

sales\_sum = sum(sales)

)

* 1. For each state, calculate the average profit by department.

elevendfb <- Superstore |>

group\_by(state, department) |>

summarize(

avg\_profit = mean(profit)

)

* 1. For each month, calculate the total order quantity and sort it from highest to lowest.

> Superstore <- Superstore |>

+ mutate(

+ date\_split = strsplit(as.character(order\_date), split = "-")

+ )

> Superstore <- Superstore |>

+ mutate(

+ month = sapply(date\_split, function(x) x[[2]])

+ )

> elevendfc <- Superstore |>

+ group\_by(month) |>

+ summarise(

+ desc(sum(order\_quantity))

+ )

> elevendfc <- Superstore |>

+ group\_by(month) |>

+ summarise(

+ total\_order\_quantity = sum(order\_quantity)

+ ) |>

+ arrange(desc(total\_order\_quantity))

* 1. For each department, calculate the average, lowest, and highest discount along with the number of data points used to calculate each.

elevendfd <- Superstore |>

group\_by(department) |>

summarize(

avg\_discount = mean(discount),

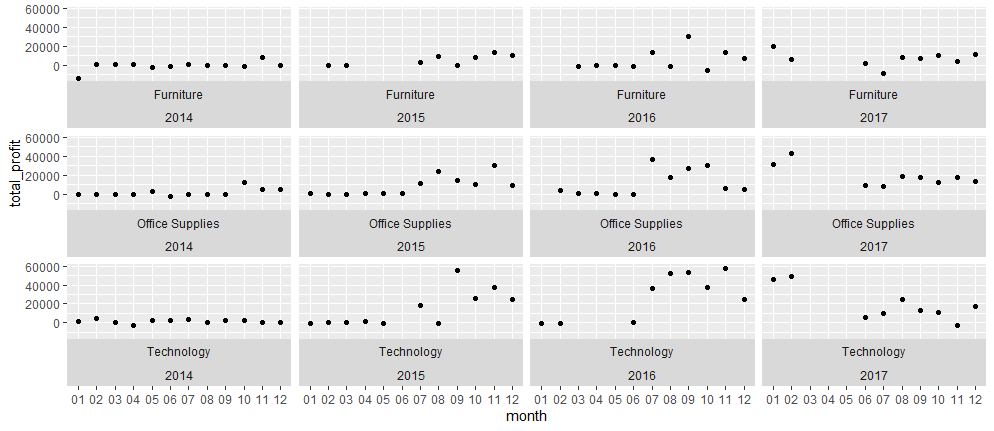
low\_discount = min(discount),

high\_discount = max(discount),

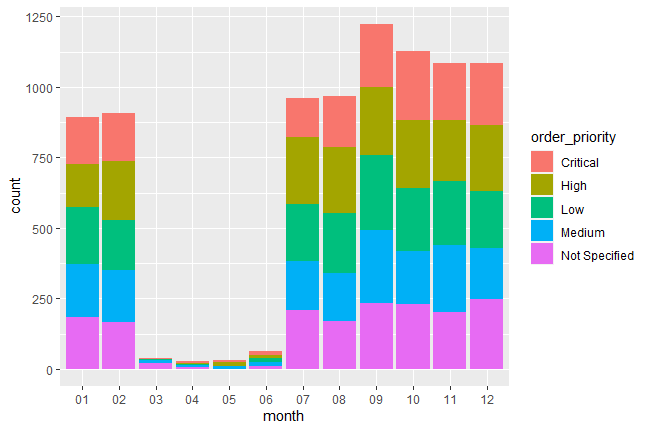
num\_points = length(department)

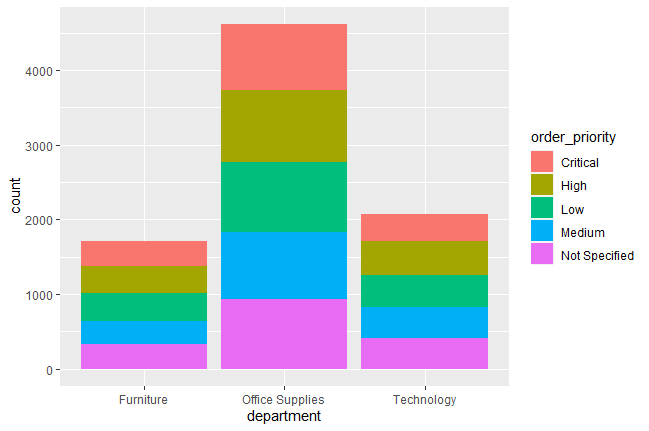
)

1. Create visuals to answer the following. It may be helpful to do the calculations and first save those to a new dataframe:
   1. How does the total profit change by month and year based on department? In other words, your plot should have month, year on the x-axis, total profit on the y-axis, and each department should be a separate bar or line series.



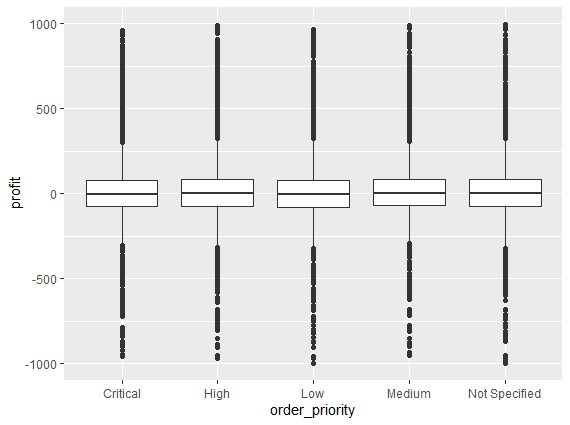
* 1. What proportion of orders have a given order\_priority by month? By department?





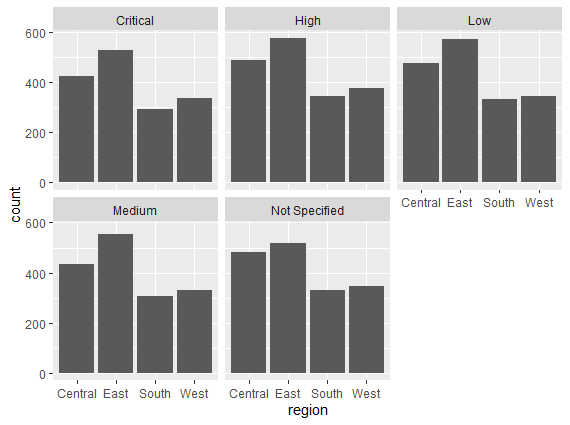
1. Create a hypothesis about the data and run an appropriate statistical test of your choosing in R to test your assertion.

I hypothesize that orders with a higher priority will have a higher profit. The higher profit variable makes the order a higher priority.



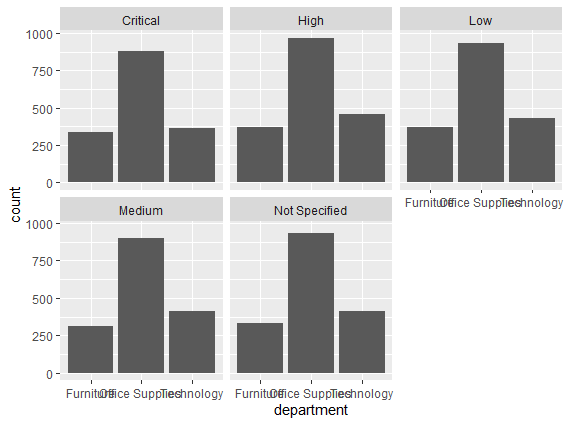
Somehow, every order\_priority has a similar boxplot set on 0 profit.

Is order priority instead dictated by the region in which the order takes place?



No, the region in which the order is made does not dictate the order’s priority

What about department?



Nope.

Order\_priority might instead be dictated by the customer ordering the items.

Superstore |>

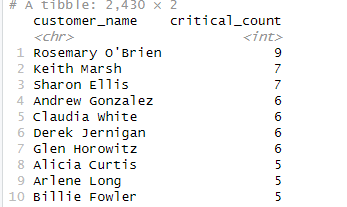
group\_by(customer\_name) |>

summarize(

critical\_count = sum(order\_priority == "Critical")

) |>

arrange(desc(critical\_count))



Superstore |>

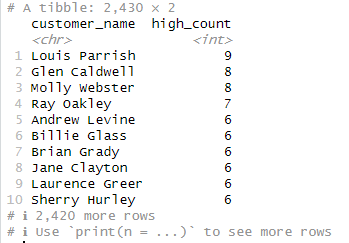
group\_by(customer\_name) |>

summarize(

high\_count = sum(order\_priority == "High")

) |>

arrange(desc(high\_count))



Superstore |>

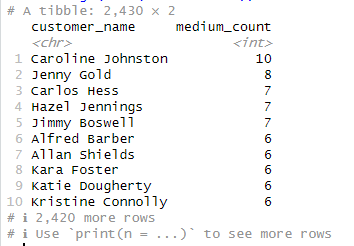
group\_by(customer\_name) |>

summarize(

medium\_count = sum(order\_priority == "Medium")

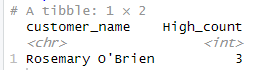
) |>

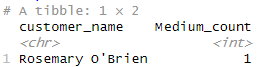
arrange(desc(medium\_count))

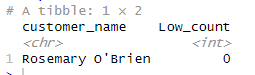


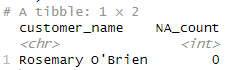
Different customers might be marked as critical, high, medium, low, or not specified priority

The customer with the highest amount of critical priority orders was “Rosemary O’Brien” with 9









This customer is skewed as a more critical customer with more of her orders leaning towards the higher end of priority.