BIS581v

BIS 581v Mid project. You have three data sets: central, western and eastern region. You may need to edit the code below for the path to the files.

library(tidyr)  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ purrr 1.0.2  
## ✔ forcats 1.0.0 ✔ readr 2.1.5  
## ✔ ggplot2 3.5.1 ✔ stringr 1.5.1  
## ✔ lubridate 1.9.3 ✔ tibble 3.2.1  
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

#load from CSV  
centralData <- read.csv("Orders\_Central.csv", header=TRUE)  
westData <- read.csv("orders\_west.csv", header=TRUE)  
eastData <- read.delim("Orders\_East.txt", header=TRUE, sep='\t')

For this assignment, answer the following questions, but please note: the data may or may not be setup such that you can just answer. You may need to perform some “data wrangling” before you can answer. Include the code you use for any wrangling/EDA in your submission. Also include the code you use to answer each question along with text indicating what the answer is. You must provide proof of your answer by showing your R code. You may use online resources and talk to others, but what you submit must be your own work.

#Wrangling the data  
# Combine Year, Month, and Day columns into a Date format  
centralData$Order.Date <- as.Date(sprintf("%s-%s-%s", centralData$Order.Year,centralData$Order.Month,centralData$Order.Day))  
#The sprintf function in R formats strings with placeholders, allowing precise control over numbers, text, and alignment,   
  
centralData$Ship.Date <- as.Date(sprintf("%s-%s-%s", centralData$Ship.Year, centralData$Ship.Month,centralData$Ship.Day))  
  
# Convert 'Order.Date' and 'Ship.Date' columns from character to Date, ignoring the time  
eastData$Order.Date <- as.Date(eastData$Order.Date, format = "%m/%d/%Y")  
eastData$Ship.Date <- as.Date(eastData$Ship.Date, format = "%m/%d/%Y")  
  
# Convert 'Order.Date' and 'Ship.Date' columns to Date, ignoring the time part  
westData$Order.Date <- as.Date(substr(westData$Order.Date, 1, 10))  
westData$Ship.Date <- as.Date(substr(westData$Ship.Date, 1, 10))  
  
# Set the 'Region' for the central data  
centralData$Region <- "Central"  
  
# Remove non-numeric characters from 'Sales' and convert to numeric for calculations  
eastData$Sales <- as.numeric(gsub("[^0-9.]", "", eastData$Sales))  
#The gsub function in R is used to replace all occurrences of a specified pattern within a string with a replacement value, using regular expressions for pattern matching.  
  
  
# Combining all data using bind\_rows  
combined\_orders <- bind\_rows(centralData, eastData, westData)

which region, on average, ships products faster:

# Calculate the average shipping time for each region  
average\_ship\_time <- aggregate((combined\_orders$Ship.Date - combined\_orders$Order.Date), by = list(combined\_orders$Region), mean)  
  
# Rename the columns  
colnames(average\_ship\_time) <- c("Region", "Average\_Shipping\_Time")  
  
# View the result  
average\_ship\_time

## Region Average\_Shipping\_Time  
## 1 NA days  
## 2 Central 4.057254 days  
## 3 East 3.913400 days  
## 4 West 3.900136 days

# Find the region with the fastest average shipping time  
fastest\_region <- average\_ship\_time[which.min(average\_ship\_time$Average\_Shipping\_Time), ]  
fastest\_region

## Region Average\_Shipping\_Time  
## 4 West 3.900136 days

Which products ship slowest by region:

# Central region: Find the slowest-shipped product  
central\_slowestproducts <- centralData %>%  
 mutate(Shipping\_Time = as.numeric(Ship.Date - Order.Date)) %>%  
 group\_by(Product) %>%  
 summarize(slowest\_central = max(Shipping\_Time, na.rm = TRUE)) %>%  
 filter(slowest\_central == max(slowest\_central))  
  
# East region: Find the slowest-shipped product  
east\_slowestproducts <- eastData %>%  
 mutate(Shipping\_Time = as.numeric(Ship.Date - Order.Date)) %>%  
 group\_by(Product.Name) %>%  
 summarize(slowest\_east = max(Shipping\_Time, na.rm = TRUE)) %>%  
 filter(slowest\_east == max(slowest\_east))

## Warning: There was 1 warning in `summarize()`.  
## ℹ In argument: `slowest\_east = max(Shipping\_Time, na.rm = TRUE)`.  
## ℹ In group 1: `Product.Name = ""`.  
## Caused by warning in `max()`:  
## ! no non-missing arguments to max; returning -Inf

# West region: Find the slowest-shipped product  
west\_slowestproducts <- westData %>%  
 mutate(Shipping\_Time = as.numeric(Ship.Date - Order.Date)) %>%  
 group\_by(Product.Name) %>%  
 summarize(slowest\_west = max(Shipping\_Time, na.rm = TRUE)) %>%  
 filter(slowest\_west == max(slowest\_west))  
  
# Print the slowest-shipped products for each region  
print(central\_slowestproducts)

## # A tibble: 126 × 2  
## Product slowest\_central  
## <chr> <dbl>  
## 1 "#10 Gummed Flap White Envelopes, 100/Box" 7  
## 2 "#6 3/4 Gummed Flap White Envelopes" 7  
## 3 "36X48 HARDFLOOR CHAIRMAT" 7  
## 4 "Acco Hot Clips Clips to Go" 7  
## 5 "Acco Perma 4000 Stacking Storage Drawers" 7  
## 6 "Acco Pressboard Covers with Storage Hooks, 14 7/8\" x 11\",… 7  
## 7 "Acco Side-Punched Conventional Columnar Pads" 7  
## 8 "Acme Forged Steel Scissors with Black Enamel Handles" 7  
## 9 "Acme Tagit Stainless Steel Antibacterial Scissors" 7  
## 10 "Advantus Plastic Paper Clips" 7  
## # ℹ 116 more rows

print(east\_slowestproducts)

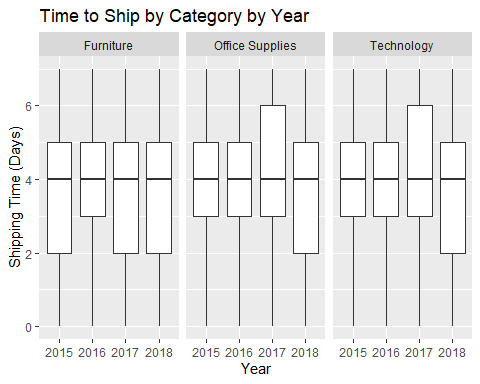
## # A tibble: 144 × 2  
## Product.Name slowest\_east  
## <chr> <dbl>  
## 1 "3M Hangers With Command Adhesive" 7  
## 2 "Acco Data Flex Cable Posts For Top & Bottom Load Binders, 6\" … 7  
## 3 "Acco Perma 2700 Stacking Storage Drawers" 7  
## 4 "Acme Elite Stainless Steel Scissors" 7  
## 5 "Acme Hot Forged Carbon Steel Scissors with Nickel-Plated Handl… 7  
## 6 "Acme Kleen Earth Office Shears" 7  
## 7 "Acme Serrated Blade Letter Opener" 7  
## 8 "Adjustable Depth Letter/Legal Cart" 7  
## 9 "Advantus Push Pins" 7  
## 10 "Ames Color-File Green Diamond Border X-ray Mailers" 7  
## # ℹ 134 more rows

print(west\_slowestproducts)

## # A tibble: 218 × 2  
## Product.Name slowest\_west  
## <chr> <dbl>  
## 1 24-Hour Round Wall Clock 7  
## 2 36X48 HARDFLOOR CHAIRMAT 7  
## 3 3M Office Air Cleaner 7  
## 4 3M Replacement Filter for Office Air Cleaner for 20' x 33' Room 7  
## 5 4009 Highlighters by Sanford 7  
## 6 50 Colored Long Pencils 7  
## 7 APC 7 Outlet Network SurgeArrest Surge Protector 7  
## 8 AT&T 1070 Corded Phone 7  
## 9 AT&T 1080 Corded phone 7  
## 10 AT&T 841000 Phone 7  
## # ℹ 208 more rows

Plot time to ship by category, by year.

# Load required library  
library(ggplot2)  
  
# Ensure Order.Date and Ship.Date are in Date format  
combined\_orders$Order.Date <- as.Date(combined\_orders$Order.Date)  
combined\_orders$Ship.Date <- as.Date(combined\_orders$Ship.Date)  
  
# Calculate shipping time in days  
combined\_orders$Shipping.Time <- as.numeric(difftime(combined\_orders$Ship.Date, combined\_orders$Order.Date, units = "days"))  
  
# Remove rows with non-finite values in Shipping.Time (e.g., NA or Inf)  
combined\_orders <- combined\_orders[is.finite(combined\_orders$Shipping.Time), ]  
  
# Create a new column for the year from Order.Date  
combined\_orders$Year <- format(combined\_orders$Order.Date, "%Y")  
  
# Plot shipping time by category, grouped by year  
ggplot(combined\_orders, aes(x = Year, y = Shipping.Time)) +  
 geom\_boxplot() + # Create boxplot  
 facet\_wrap(~ Category) + # Separate plots for each category  
 labs(title = "Time to Ship by Category by Year", x = "Year", y = "Shipping Time (Days)")



which categories have highest profit by region, chain-wide?

# Calculate the total profit for each product category by region  
Sum\_profit <- aggregate(Profit ~ Region + Category, data = combined\_orders, sum)  
  
# Print the total profit for each category by region  
print(Sum\_profit)

## Region Category Profit  
## 1 Central Furniture -2871.049  
## 2 East Furniture 2899.566  
## 3 West Furniture 23880.468  
## 4 Central Office Supplies 8879.980  
## 5 East Office Supplies 40915.469  
## 6 West Office Supplies 168043.587  
## 7 Central Technology 33697.432  
## 8 East Technology 45695.665  
## 9 West Technology 117820.045

# Find the category with the highest profit chain-wide (across all regions)  
highest\_chained\_profit <- Sum\_profit[which.max(Sum\_profit$Profit), ]  
  
# Print the category and region with the highest total profit  
print(highest\_chained\_profit)

## Region Category Profit  
## 6 West Office Supplies 168043.6

which segments have the lowest profit by region?

# Calculate the average profit for each segment by region  
Avg\_profit <- aggregate(Profit ~ Region + Segment, data = combined\_orders, mean)  
  
# Print the average profit for each segment by region  
print(Avg\_profit)

## Region Segment Profit  
## 1 Central Consumer 7.066046  
## 2 East Consumer 27.798478  
## 3 West Consumer 34.887099  
## 4 Central Corporate 27.791831  
## 5 East Corporate 26.384006  
## 6 West Corporate 34.052692  
## 7 Central Home Office 28.398202  
## 8 East Home Office 53.455800  
## 9 West Home Office 21.435666

# Find the segment with the lowest average profit chain-wide (across all regions)  
lowest\_profit\_by\_region <- Avg\_profit[which.min(Avg\_profit$Profit), ]  
  
# Print the segment with the lowest average profit  
print(lowest\_profit\_by\_region)

## Region Segment Profit  
## 1 Central Consumer 7.066046

What are yearly sales by region?

# Convert Order.Date to Date format if not already done  
combined\_orders$Order.Date <- as.Date(combined\_orders$Order.Date)  
  
# Extract the year from Order.Date  
combined\_orders$Year <- format(combined\_orders$Order.Date, "%Y")  
  
# Calculate yearly sales by region  
Yearly\_sales <- aggregate(Sales ~ Region + Year, data = combined\_orders, sum)  
  
# Print the yearly sales for each region  
print(Yearly\_sales)

## Region Year Sales  
## 1 Central 2015 103838.2  
## 2 East 2015 128092.7  
## 3 West 2015 442618.9  
## 4 Central 2016 102874.2  
## 5 East 2016 155997.6  
## 6 West 2016 402291.2  
## 7 Central 2017 147429.4  
## 8 East 2017 179112.4  
## 9 West 2017 533864.8  
## 10 Central 2018 147098.1  
## 11 East 2018 211779.5  
## 12 West 2018 773738.5