BIS581v

library(tidyr)  
library(tidyverse)

## Warning: package 'ggplot2' was built under R version 4.3.3

## ── 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 3.5.0 ✔ 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')  
  
  
# lowercase column names   
centralData <- centralData %>% rename\_with(tolower)  
westData <- westData %>% rename\_with(tolower)  
eastData <- eastData %>% rename\_with(tolower)  
  
# check column names  
colnames(centralData)

## [1] "row.id" "order.id" "ship.mode" "customer.id"   
## [5] "customer.name" "segment" "country" "city"   
## [9] "state" "postal.code" "product.id" "category"   
## [13] "sub.category" "product" "sales" "quantity"   
## [17] "profit" "order.year" "order.month" "order.day"   
## [21] "ship.year" "ship.month" "ship.day" "discounts"

colnames(westData)

## [1] "row.id" "order.id" "order.date"   
## [4] "ship.date" "ship.mode" "customer.id"   
## [7] "customer.name" "segment" "country"   
## [10] "city" "postal.code" "region"   
## [13] "product.id" "category" "sub.category"   
## [16] "product.name" "sales" "quantity"   
## [19] "discount" "profit" "right\_row.id"   
## [22] "right\_order.date" "right\_ship.date" "right\_ship.mode"   
## [25] "right\_customer.id" "right\_customer.name" "right\_segment"   
## [28] "right\_country" "right\_city" "right\_state2"   
## [31] "right\_postal.code" "right\_region" "right\_product.id"   
## [34] "right\_category" "right\_sub.category" "right\_product.name"   
## [37] "right\_sales" "right\_quantity" "right\_discount"   
## [40] "right\_profit" "state"

colnames(eastData)

## [1] "category" "city" "country" "customer.id"   
## [5] "customer.name" "discount" "order.date" "order.id"   
## [9] "postal.code" "product.id" "product.name" "profit"   
## [13] "quantity" "region" "row.id" "segment"   
## [17] "ship.date" "ship.mode" "state" "sub.category"   
## [21] "sales"

# Add the Region column for 'Central Data'  
centralData <- centralData %>% mutate(region = "central")  
  
# rename 'product' to 'product name'  
centralData <- centralData %>% rename(product.name = product)  
  
# Combine year, month, and day into proper Date columns  
centralData <- centralData %>%  
 mutate(order.date = as.Date(paste(order.year, order.month, order.day, sep = "-"), format = "%Y-%m-%d"),  
 ship.date = as.Date(paste(ship.year, ship.month, ship.day, sep = "-"), format = "%Y-%m-%d"))  
  
# change data type  
centralData <- centralData %>%  
 mutate(discounts = as.numeric(discounts))

## Warning: There was 1 warning in `mutate()`.  
## ℹ In argument: `discounts = as.numeric(discounts)`.  
## Caused by warning:  
## ! NAs introduced by coercion

# Remove columns  
centralData <- centralData %>% select(-order.year, -order.month, -order.day, -ship.year, -ship.month, -ship.day)  
  
  
  
# remove unneccessary column for west data  
westData <- westData %>% select(-starts\_with("right\_"))  
  
# Rename columns  
westData <- westData %>%  
 rename(discounts = discount)  
  
# Rename columns   
eastData <- eastData %>%  
 rename(discounts = discount)  
  
  
# Standardize `order.date` and `ship.date` as Date objects in all datasets  
centralData <- centralData %>%  
 mutate(order.date = as.Date(order.date, format = "%Y-%m-%d"),  
 ship.date = as.Date(ship.date, format = "%Y-%m-%d"))  
  
# Correctly parse order.date and ship.date in westData  
westData <- westData %>%  
 mutate(order.date = as.Date(order.date, format = "%Y-%m-%d %H:%M:%S"),  
 ship.date = as.Date(ship.date, format = "%Y-%m-%d %H:%M:%S"))  
  
eastData <- eastData %>%  
 mutate(order.date = as.Date(order.date, format = "%m/%d/%Y"),  
 ship.date = as.Date(ship.date, format = "%m/%d/%Y"))  
  
#convert Sales column in eastData  
eastData <- eastData %>%  
 mutate(sales = as.numeric(gsub("USD","",sales)))  
  
# Merge the datasets  
merged\_data <- bind\_rows(centralData, westData, eastData)

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.

# 1.which region, on average, ships products faster:

* The **West region** has the fastest average shipping time of **3.90 days** and similarly **East** with **3.91 days** .
* The **Central region** has the slowest average shipping time of **4.06 days**.

merged\_data <- merged\_data %>%  
 mutate(time.to.ship = as.numeric(difftime(ship.date,order.date,units = "days")))  
  
# Remove rows with missing or blank regions  
merged\_data <- merged\_data %>%  
 filter(region != "" & !is.na(region))  
  
average\_shipping\_time <- merged\_data %>%  
 group\_by(region) %>%  
 summarise(avg\_shipping\_time = mean(time.to.ship, na.rm = TRUE)) %>%  
 arrange(avg\_shipping\_time)  
  
print(average\_shipping\_time)

## # A tibble: 3 × 2  
## region avg\_shipping\_time  
## <chr> <dbl>  
## 1 West 3.90  
## 2 East 3.91  
## 3 central 4.06

# 2.Which products ship slowest by region:

* In the **East region**, the product with the slowest shipping time is **Acme Elite Stainless Steel Scissors**, with an average shipping time of **7 days**.
* In the **West region**, the product with the slowest shipping time is **3M Office Air Cleaner**, with an average shipping time of **7 days**.
* In the **Central region**, the product with the slowest shipping time is **#6 3/4 Gummed Flap White Envelopes**, with an average shipping time of **7 days**.

slowest\_shipping\_products <- merged\_data %>%  
 group\_by(region, product.name) %>%  
 summarise(avg\_shipping\_time = mean(time.to.ship, na.rm = TRUE), .groups = "drop") %>%  
 group\_by(region) %>%  
 slice\_max(avg\_shipping\_time, n = 1, with\_ties = FALSE)  
  
print(slowest\_shipping\_products)

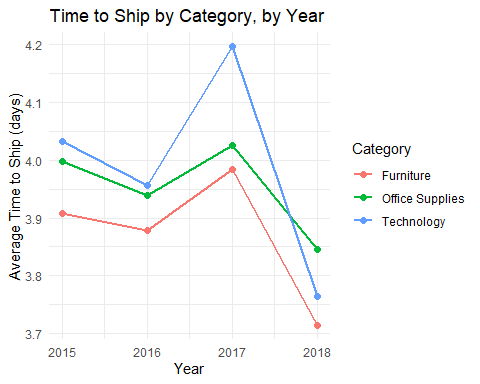
## # A tibble: 3 × 3  
## # Groups: region [3]  
## region product.name avg\_shipping\_time  
## <chr> <chr> <dbl>  
## 1 East Acme Elite Stainless Steel Scissors 7  
## 2 West 3M Office Air Cleaner 7  
## 3 central #6 3/4 Gummed Flap White Envelopes 7

# 3.Plot time to ship by category, by year.

* The **Technology category** consistently had the longest average shipping time over the years, peaking at **4.2 days** in 2017 before dropping significantly in 2018.
* The **Furniture category** showed the shortest average shipping times across most years.
* The **Office Supplies category** maintained moderate shipping times, with minimal variation compared to other categories.

#3.Plot time to ship by category, by year.  
  
merged\_data = merged\_data %>%   
 mutate(order\_year = as.numeric(format(order.date,"%Y")))  
  
time\_to\_ship\_by\_category\_year <- merged\_data %>%  
 group\_by(category,order\_year) %>%  
 summarise(avg\_shipping\_time = mean(time.to.ship, na.rm = TRUE), .groups = "drop")  
  
ggplot(time\_to\_ship\_by\_category\_year, aes(x = order\_year, y = avg\_shipping\_time, color = category)) +  
 geom\_line(size = 1) +  
 geom\_point(size = 2) +  
 labs(title = "Time to Ship by Category, by Year",  
 x = "Year",  
 y = "Average Time to Ship (days)",  
 color = "Category") +  
 theme\_minimal()

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.



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

* The **Office Supplies category** generated the highest total profit of **217,839.04**, making it the most profitable category.
* The **Technology category** follows with a total profit of **197,213.14**.
* The **Furniture category** has the lowest total profit at **23,908.98**.

# Calculate total profit by region and category  
highest\_profit\_categories <- merged\_data %>%  
 group\_by(region, category) %>%  
 summarise(total\_profit = sum(profit, na.rm = TRUE), .groups = "drop") %>%  
 arrange(region, desc(total\_profit)) %>%  
 group\_by(region) %>%  
 slice\_max(total\_profit, n = 1, with\_ties = FALSE)  
  
# Calculate total profit by category (chain-wide)  
highest\_profit\_chain\_wide <- merged\_data %>%  
 group\_by(category) %>%  
 summarise(total\_profit = sum(profit, na.rm = TRUE), .groups = "drop") %>%  
 arrange(desc(total\_profit)) %>%  
 group\_by(category) %>%  
 slice\_max(total\_profit, n = 1, with\_ties = FALSE)  
  
print(highest\_profit\_categories)

## # A tibble: 3 × 3  
## # Groups: region [3]  
## region category total\_profit  
## <chr> <chr> <dbl>  
## 1 East Technology 45696.  
## 2 West Office Supplies 168044.  
## 3 central Technology 33697.

print(highest\_profit\_chain\_wide)

## # A tibble: 3 × 2  
## # Groups: category [3]  
## category total\_profit  
## <chr> <dbl>  
## 1 Furniture 23909.  
## 2 Office Supplies 217839.  
## 3 Technology 197213.

# 5.which segments have the lowest profit by region?

* In the **East region**, the segment with the lowest profit is **Corporate**, with a total profit of **22,716.63**.
* In the **West region**, the segment with the lowest profit is **Home Office**, with a total profit of **33,761.17**.
* In the **Central region**, the segment with the lowest profit is **Consumer**, with a total profit of **8,564.05**.

# Calculate total profit by region and segment  
lowest\_profit\_segments <- merged\_data %>%  
 group\_by(region, segment) %>%  
 summarise(total\_profit = sum(profit, na.rm = TRUE), .groups = "drop") %>%  
 arrange(region, total\_profit) %>%  
 group\_by(region) %>%  
 slice\_min(total\_profit, n = 1, with\_ties = FALSE)  
  
# View the results  
print(lowest\_profit\_segments)

## # A tibble: 3 × 3  
## # Groups: region [3]  
## region segment total\_profit  
## <chr> <chr> <dbl>  
## 1 East Corporate 22717.  
## 2 West Home Office 33761.  
## 3 central Consumer 8564.

# 6.What are yearly sales by region?

* Sales in the **East region** grew steadily from 128,092.7 at the year-end of 2015 to 211,779.5 in 2018
* Growth was rapid in the **West region** (from 442,618.9 sales 2015 to 773,738.5 sales 2018).
* The **Central region** had stable sales, with 103,838.2 in 2015 and 102,874.2 in 2016.

yearly\_sales\_by\_region <- merged\_data %>%  
 group\_by(region, order\_year) %>%  
 summarise(total\_sales = sum(sales, na.rm = TRUE), .groups = "drop") %>%  
 arrange(region, order\_year)  
  
# View the results  
print(yearly\_sales\_by\_region)

## # A tibble: 12 × 3  
## region order\_year total\_sales  
## <chr> <dbl> <dbl>  
## 1 East 2015 128093.  
## 2 East 2016 155998.  
## 3 East 2017 179112.  
## 4 East 2018 211780.  
## 5 West 2015 442619.  
## 6 West 2016 402291.  
## 7 West 2017 533865.  
## 8 West 2018 773739.  
## 9 central 2015 103838.  
## 10 central 2016 102874.  
## 11 central 2017 147429.  
## 12 central 2018 147098.