Getting the file path and loading the datasets with respective of their path mentioned.

## please add your first name below  
##Student First Name: Poojitha  
   
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

library(readxl)  
  
# loading the file path to the respective objects created  
Orders\_Central\_path <- "C:/Users/pooja/Documents/581/Orders\_Central.csv"  
Orders\_west\_path <- "C:/Users/pooja/Documents/581/Orders\_West.csv"  
Orders\_East\_path <- "C:/Users/pooja/Documents/581/Orders\_East.txt"  
  
#loading the data from csv  
  
centralData <- read.csv(Orders\_Central\_path, header=TRUE)  
westData <- read.csv(Orders\_west\_path, header=TRUE)  
eastData <- read.delim(Orders\_East\_path, header=TRUE, sep='\t')

Viewing the loaded data

View(centralData)  
View(westData)  
View(eastData)

\*\* Data wrangling and data cleaning \*\*

* First combining the ship date, year,day ,month as single column for better understanding

library(dplyr)  
  
centralData <- centralData %>% unite(Ship.Date,  
 Ship.Year,  
 Ship.Month,  
 Ship.Day,  
 sep = "-")

* Changing the date format as it helps to plot graph easily

centralData <- centralData %>%  
 mutate(Ship.Date = as.Date(Ship.Date, format = "%Y-%m-%d"))

-Same combing the Order date into one column and changing the format

centralData <- centralData %>% unite(Order.Date,  
 Order.Year,  
 Order.Month,  
 Order.Day,  
 sep = "-") %>%  
 mutate(Order.Date = as.Date(Order.Date, format = "%Y-%m-%d"))

* Handling the negative values \*
* Replacing the negative value with mean of profit

mean\_profit <- mean(centralData$Profit[centralData$Profit >= 0], na.rm = TRUE)  
  
  
centralData$Profit[centralData$Profit < 0] <- mean\_profit

* Converting the format of the ship.date \*
* It is in format of “Y-M-D H:M:S” so converting it into “Y-M-D” making sure for all data sets the format is same.

library(lubridate)  
westData$Ship.Date <- as.POSIXct(westData$Ship.Date, format = "%Y-%m-%d %H:%M:%S",  
 tz = "UTC")  
  
# Format Ship Date to "%Y-%m-%d"  
westData$Ship.Date <- format(westData$Ship.Date, "%Y-%m-%d")  
  
westData$Order.Date <- as.POSIXct(westData$Order.Date, format = "%Y-%m-%d %H:%M:%S",  
 tz = "UTC")  
westData$Order.Date <- format(westData$Order.Date, "%Y-%m-%d")  
  
westData$Right\_Ship.Date <- as.POSIXct(westData$Right\_Ship.Date, format = "%Y-%m-%d %H:%M:%S",  
 tz = "UTC")  
  
westData$Right\_Ship.Date <- format(westData$Right\_Ship.Date, "%Y-%m-%d")  
  
westData$Right\_Order.Date <- as.POSIXct(westData$Right\_Order.Date, format = "%Y-%m-%d %H:%M:%S", tz = "UTC")  
westData$Right\_Order.Date <- format(westData$Right\_Order.Date, "%Y-%m-%d")

-Here we are compare ship.date column with Right\_ship.date and get the count of them as we have we know if both columns are same we can delete Right\_Ship.Date column which helps to clean the data.

# Compare Ship\_Date and Right\_Ship\_Date columns  
different\_dates <- westData[westData$Ship\_Date != westData$Right\_Ship\_Date, ]  
  
# Count rows with different dates  
different\_dates\_count <- nrow(different\_dates)  
  
# Print the count  
print(different\_dates\_count)

## [1] 0

-As we can see the count is “0” which both column has same value so we can delete the unwanted column

* Similarly we are comparing all the columns which Prefix as Right with the original ones according to that we can delete the unnecessary data.

# List of columns to compare (without Right\_ prefix)  
columns\_to\_compare <- c("Row.ID", "Order.Date", "Ship.Date", "Ship.Mode",   
 "Customer.ID", "Customer.Name", "Segment", "Country",   
 "City", "Postal.Code", "Region", "Product.ID",   
 "Category", "Sub.Category", "Product.Name", "Sales",   
 "Quantity", "Discount", "Profit")  
  
# Initialize a vector to store counts  
different\_counts <- numeric(length(columns\_to\_compare))  
  
# Loop through each column and compare  
for (col\_name in columns\_to\_compare) {  
   
 # Construct column names with and without Right\_ prefix  
 col\_name\_right <- paste0("Right\_", col\_name)  
 different\_rows <- westData[westData[[col\_name\_right]] != westData[[col\_name]], ]  
 different\_counts[col\_name] <- nrow(different\_rows)  
   
 # Print the count for each pair  
 cat("Count of differences in", col\_name, "and", col\_name\_right, ":", different\_counts[col\_name], "\n")  
}

## Count of differences in Row.ID and Right\_Row.ID : 6350   
## Count of differences in Order.Date and Right\_Order.Date : 0   
## Count of differences in Ship.Date and Right\_Ship.Date : 0   
## Count of differences in Ship.Mode and Right\_Ship.Mode : 0   
## Count of differences in Customer.ID and Right\_Customer.ID : 0   
## Count of differences in Customer.Name and Right\_Customer.Name : 0   
## Count of differences in Segment and Right\_Segment : 0   
## Count of differences in Country and Right\_Country : 0   
## Count of differences in City and Right\_City : 0   
## Count of differences in Postal.Code and Right\_Postal.Code : 0   
## Count of differences in Region and Right\_Region : 0   
## Count of differences in Product.ID and Right\_Product.ID : 6348   
## Count of differences in Category and Right\_Category : 3458   
## Count of differences in Sub.Category and Right\_Sub.Category : 5766   
## Count of differences in Product.Name and Right\_Product.Name : 6348   
## Count of differences in Sales and Right\_Sales : 6350   
## Count of differences in Quantity and Right\_Quantity : 5292   
## Count of differences in Discount and Right\_Discount : 2882   
## Count of differences in Profit and Right\_Profit : 6350

cat("\nTotal differences across columns are :\n")

##   
## Total differences across columns are :

cat(sum(different\_counts), "differences found in total.\n")

## 49144 differences found in total.

-So by considering the output we can delete the unwanted data later we will handle the different valued columns

-Deleting the columns using within and rm

westData <- within(westData, rm(Right\_Order.Date,  
 Right\_Ship.Date,  
 Right\_Ship.Mode,  
 Right\_Customer.ID,  
 Right\_Customer.Name,  
 Right\_Segment,  
 Right\_Postal.Code,  
 Right\_Region,  
 Right\_Country,  
 Right\_City))

-Now the no.of columns have been reduced to 31 from 41 columns.

-As the state has shortform of state names and we need to convert that to fullform to compare with Right\_state2

-so we are using state.abb to save the abbreviation and replacing them.

Full\_name <- match(westData$State, state.abb)  
westData$State <- state.name[Full\_name]

-After converting comparing them and getting count of difference observed.

different\_states <- westData[westData$State != westData$Right\_State2, ]  
different\_states\_count <- nrow(different\_states)  
print(different\_states\_count)

## [1] 0

-The difference between states and right\_states is 0 so we can delete the right\_state2 column.

* Deleting the Right\_State2 column

westData <- within(westData, rm(Right\_State2))

* Ship.Date was an extra column so have to delete the additional column

#centralData <- within(centralData, rm(Ship.Date))  
#westData <- within(westData, rm(Region))  
#eastData <- within(eastData, rm(Region))

* To work with different data sets we need to make sure that the column names should be similar.
* So change the column names according while compared with others central data doesn’t have proper column name.

colnames(centralData)[colnames(centralData) == "Product"] <- "Product.Name"

* Converting the format of the ship.date and order.date \*
* Similarly to the westData , eastData has same format
* It is in format of “Y-M-D H:M:S” so converting it into “Y-M-D” making sure for all data sets the format is same.

eastData$Order.Date <- as.POSIXct(eastData$Order.Date, format = "%m/%d/%Y %H:%M", tz = "UTC")  
eastData$Order.Date <- format(eastData$Order.Date, "%Y-%m-%d")  
  
  
eastData$Ship.Date <- as.POSIXct(eastData$Ship.Date, format = "%m/%d/%Y %H:%M", tz = "UTC")  
eastData$Ship.Date <- format(eastData$Ship.Date, "%Y-%m-%d")

* Handling negative value in EastData \* Replacing the negative value with mean of profit

mean\_profit <- mean(eastData$Profit[eastData$Profit >= 0], na.rm = TRUE)  
  
  
eastData$Profit[eastData$Profit < 0] <- mean\_profit

-As we have observed the difference between few columns .Assume Right\_ columns data to be correct columns and deleting the other column

* After renaming the column smiliar to the other data sets column names

westData <- within(westData , rm(Row.ID,  
 Product.ID,  
 Category,  
 Sub.Category,  
 Product.Name,  
 Sales,  
 Discount,  
 Profit,  
 Quantity))  
  
  
colnames(westData)[colnames(westData) %in% c("Right\_Row.ID", "Right\_Product.ID",  
 "Right\_Category",  
 "Right\_Sub.Category",  
 "Right\_Product.Name",  
 "Right\_Sales",  
 "Right\_Quantity",  
 "Right\_Discount", "Right\_Profit")] <- c("Row.ID",  
 "Product.ID", "Category", "Sub.Category", "Product.Name", "Sales", "Quantity", "Discounts", "Profit")

* Handling negative value in westData \*
* Replacing the negative value with mean of profit

mean\_profit <- mean(westData$Profit[westData$Profit >= 0], na.rm = TRUE)  
westData$Profit[westData$Profit < 0] <- mean\_profit

* Handling the missing values\*
* Counting the missing values

sum(is.na(westData))

## [1] 0

sum(is.na(centralData))

## [1] 0

sum(is.na(eastData))

## [1] 291

* As we saw east data has the missing data , after view the data we got to know it has all empty rows
* So using na.omit , deleting all the na values rows

eastData<-eastData %>%  
 na.omit()  
View(eastData)

* When observed the data carefully central data doesn’t have region column listed ,so adding the region column to the data as other also have the region column listed.
* By doing this it would be help full for us to perform analysis when data is combined

centralData$Region <- "Central"

* Here we are changing the datatype of few columns .Make sure all the datasets have same datatype columns
* Changing the column names if any difference is observed.

colnames(eastData)[colnames(eastData) == "Discount"] <- "Discounts"

westData$Discounts <- as.character(westData$Discounts)  
eastData$Discounts <- as.character(eastData$Discounts)  
  
#The east data Sales column is in USD so changing it to numeric  
eastData$Sales <- gsub("USD", "", eastData$Sales)  
eastData$Sales <- as.numeric(eastData$Sales)

-Using str checking the data type of the columns

#str(westData)  
#str(centralData)  
#str(eastData)

* Checking the column names of all the three data sets by listing them

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.Name" "Sales" "Quantity"   
## [17] "Profit" "Order.Date" "Ship.Date" "Discounts"   
## [21] "Region"

colnames(westData)

## [1] "Order.ID" "Order.Date" "Ship.Date" "Ship.Mode"   
## [5] "Customer.ID" "Customer.Name" "Segment" "Country"   
## [9] "City" "Postal.Code" "Region" "Row.ID"   
## [13] "Product.ID" "Category" "Sub.Category" "Product.Name"   
## [17] "Sales" "Quantity" "Discounts" "Profit"   
## [21] "State"

colnames(eastData)

## [1] "Category" "City" "Country" "Customer.ID"   
## [5] "Customer.Name" "Discounts" "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"

* To combine the data all the column should be in similar order sorting the colnames using sort function.

library(dplyr)  
sorted\_colnames <- sort(colnames(centralData))  
  
# Reorder columns in each data set using dplyr  
centralData <- centralData %>%  
 select(sorted\_colnames)

## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.  
## ℹ Please use `all\_of()` or `any\_of()` instead.  
## # Was:  
## data %>% select(sorted\_colnames)  
##   
## # Now:  
## data %>% select(all\_of(sorted\_colnames))  
##   
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

westData <- westData %>%  
 select(sorted\_colnames)  
  
eastData <- eastData %>%  
 select(sorted\_colnames)

1. Which region, on average, ships products faster?

#1.Answer  
##Student Last Name: Velthuri  
#Converting the order date and ship date as date which helps for calculations  
centralData$Order.Date <- as.Date(centralData$Order.Date)  
centralData$Ship.Date <- as.Date(centralData$Ship.Date)  
  
# Calculate shipping time (in days)   
#Subtracting ship date with order date through which we can get the product shipping time in days  
centralData$Shipping.Time <- as.numeric(centralData$Ship.Date - centralData$Order.Date)  
  
# calculating for westData  
westData$Order.Date <- as.Date(westData$Order.Date)  
westData$Ship.Date <- as.Date(westData$Ship.Date)  
westData$Shipping.Time <- as.numeric(westData$Ship.Date - westData$Order.Date)  
  
# Calculating for eastData  
eastData$Order.Date <- as.Date(eastData$Order.Date)  
eastData$Ship.Date <- as.Date(eastData$Ship.Date)  
eastData$Shipping.Time <- as.numeric(eastData$Ship.Date - eastData$Order.Date)

* Here, we store the shipping time in days as a separate column in each data set.
* We then calculate the mean shipping time for each data set and store the results in respective objects.
* Using the min function, we retrieve the minimum value and display it alongside the region.”

##Student Last Name: Velthuri  
#Calculating Shipping Time Average for 3 region data set   
centralData\_Average\_shipping.Time <- mean(centralData$Shipping.Time, na.rm = TRUE)  
westData\_Average\_shipping.Time <- mean(westData$Shipping.Time, na.rm = TRUE)  
eastData\_Average\_shipping.Time <- mean(eastData$Shipping.Time, na.rm = TRUE)  
  
output <- paste("The Average Shipping Time in days according to the dataset are:",  
 "\n",  
 "centralData : ", centralData\_Average\_shipping.Time, " days\n",  
 "westData : ", westData\_Average\_shipping.Time, " days\n",  
 "eastData : ", eastData\_Average\_shipping.Time, " days\n")  
  
# Print the output  
#cat(output)  
  
Average\_Shipping\_Time <- c(  
 central = centralData\_Average\_shipping.Time,  
 west = westData\_Average\_shipping.Time,  
 east = eastData\_Average\_shipping.Time  
)  
  
  
Faster\_shipping <- min(Average\_Shipping\_Time)  
  
cat("Region with faster shipping time :", names(Average\_Shipping\_Time)[which.min(Average\_Shipping\_Time)], " - ", Faster\_shipping, " days\n")

## Region with faster shipping time : west - 3.900136 days

1. Which products ship slowest by region? Please display the top 9 if you have multiple products meeting the criteria.

#2.Answer  
##Student Whole Name: Poojitha Velthuri  
  
library(dplyr)  
#Combining the datasets using bind\_rows , we can use rbind as well and storing as combinedData  
combinedData <- bind\_rows(centralData, westData, eastData)  
  
#Using the group and summarise getting the details of top 9 slowest shipping products with the help of shipping time  
top\_slowest <- combinedData %>%  
 group\_by(Product.Name, Region) %>%  
 summarise(AvgShippingTime = mean(Shipping.Time)) %>%  
 arrange(Region, desc(AvgShippingTime)) %>%  
 group\_by(Region) %>%  
 top\_n(3, AvgShippingTime) %>%   
#Here we gave 3 because we are group data by region so based on the we will be getting output from all the three regions  
#After that we are arranging the data in descending order with respective of avgshippingtime.  
 ungroup() %>%  
 arrange(Region, desc(AvgShippingTime)) %>%  
 slice\_head(n = 9)

## `summarise()` has grouped output by 'Product.Name'. You can override using the  
## `.groups` argument.

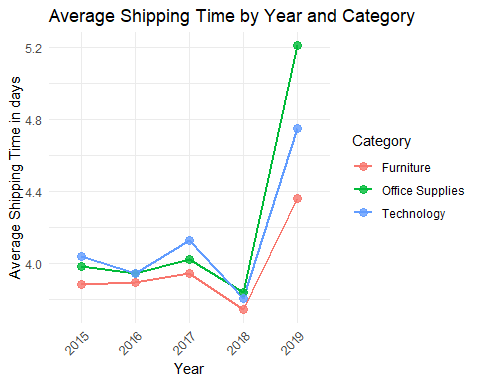
#using the slice \_head which gives the top 9 entries of data and displaying it using print  
  
# Print the top 9 slowest shipping products by region  
print(top\_slowest)

## # A tibble: 9 × 3  
## Product.Name Region AvgShippingTime  
## <chr> <chr> <dbl>  
## 1 #6 3/4 Gummed Flap White Envelopes Centr… 7  
## 2 Ames Color-File Green Diamond Border X-ray Mailers Centr… 7  
## 3 Avery 479 Centr… 7  
## 4 Avery 50 Centr… 7  
## 5 Avery Durable Slant Ring Binders Centr… 7  
## 6 Avery Non-Stick Heavy Duty View Round Locking Ring Bin… Centr… 7  
## 7 Balt Split Level Computer Training Table Centr… 7  
## 8 Bionaire Personal Warm Mist Humidifier/Vaporizer Centr… 7  
## 9 Boston 1799 Powerhouse Electric Pencil Sharpener Centr… 7

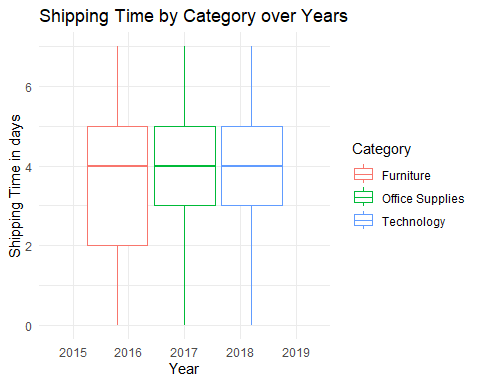
1. Plot time to ship by category, by year.

#3.Answer  
##Student Fist Name: Poojitha  
  
#here i have plotted 3 different graphs   
#1.Using average shipping time and ploting the graph  
#2. Using the ship time by year and category ploting box plot  
#3. Using few aesthetics and modifying the 2 graph  
  
library(ggplot2)  
#To plot the graph by year we are storing the year from ship date into the object as converting them into factor for better ploting  
combinedData$Ship.Year <- format(combinedData$Ship.Date, "%Y")  
combinedData$Ship.Year <- factor(combinedData$Ship.Year)  
combinedData$Category <- factor(combinedData$Category)  
  
# Line plot - using average of shipping time  
mean\_data <- aggregate(Shipping.Time ~ Ship.Year + Category, data = combinedData, FUN = mean)  
#I am ploting the graph using average of shipping time  
  
ggplot(mean\_data, aes(x = Ship.Year, y = Shipping.Time, color = Category, group = Category)) +  
 geom\_line(size = 1) +   
 geom\_point(size = 3, alpha = 0.8) +   
 labs(x = "Year", y = "Average Shipping Time in days",   
 title = "Average Shipping Time by Year and Category") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))

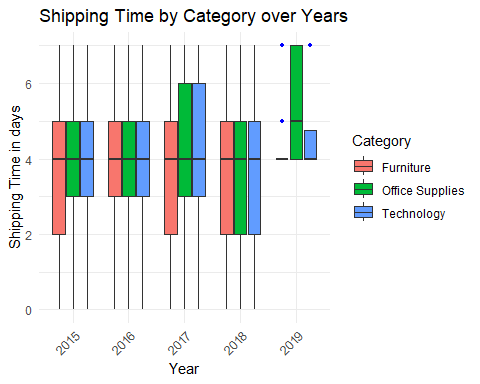
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.



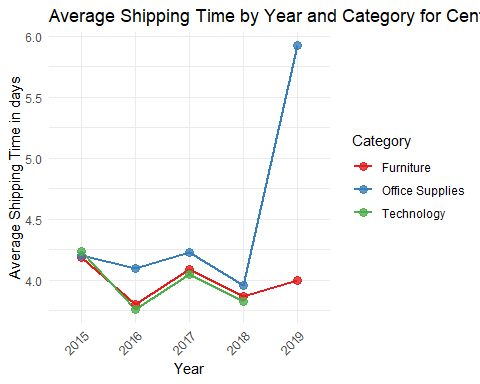
#Theme helps as to visualize the parameters properly , like this will rotate the axis in mentioned angle  
  
#Box Plot  
ggplot(combinedData, aes(x = Ship.Year, y = Shipping.Time, color = Category, group = Category)) +  
 geom\_boxplot() +  
 labs(x = "Year", y = "Shipping Time in days", title = "Shipping Time by Category over Years") +   
 theme\_minimal()



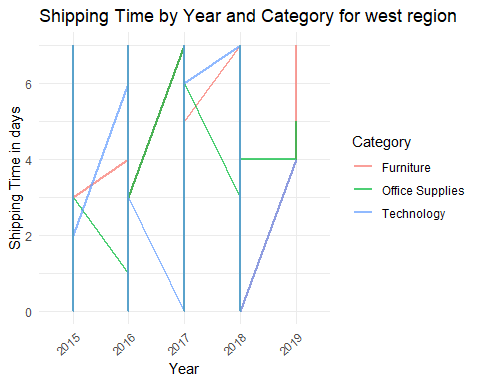
# Extract year from Ship.Date and add it as a new column  
combinedData$Ship.Year <- format(combinedData$Ship.Date, "%Y")  
  
# Create the box plot using ggplot2 for more customization  
ggplot(combinedData, aes(x = Ship.Year, y = Shipping.Time, fill = Category)) +  
 geom\_boxplot(outlier.size = 1, outlier.colour = "blue") +   
 labs(x = "Year", y = "Shipping Time in days",   
 title = "Shipping Time by Category over Years") +  
 theme\_minimal() +   
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))

 Below I have plotted graph for single data sets as well like for central , west and east .I have plotted different graphs to see the insights for different regions

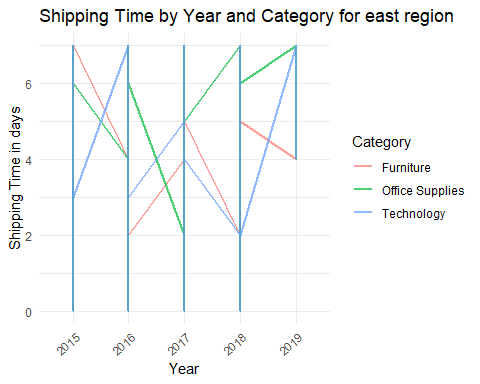
#3.Answer (Considering Central Data)  
# Graph for central data only  
##Student first Name: Poojitha  
  
# here am using the average of shipping time and ploting   
centralData$Ship.Year <- format(centralData$Ship.Date, "%Y")  
  
library(ggplot2)  
mean <- aggregate(Shipping.Time ~ Ship.Year + Category, data = centralData, FUN = mean)  
  
# Convert Ship.Year and Category to factors for better plotting order  
mean$Ship.Year <- factor(mean$Ship.Year)  
mean$Category <- factor(mean$Category)  
  
# Plot as a line graph  
ggplot(mean, aes(x = Ship.Year, y = Shipping.Time, color = Category, group = Category)) +  
 geom\_line(size = 1) +  
 geom\_point(size = 3, alpha = 0.8) +  
 labs(x = "Year", y = "Average Shipping Time in days",   
 title = "Average Shipping Time by Year and Category for Central Region") +  
 scale\_color\_brewer(palette = "Set1") + # Set color palette (adjust as needed)  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1),   
 # Rotate x-axis labels for better readability  
 legend.position = "right")



#3.Answer (Considering West Data)  
# Graph for west data only  
##Student First Name: Poojitha  
  
# Where as for this data am not using average shipping time , just ploting using the ship time with year and category  
westData$Ship.Year <- format(westData$Ship.Date, "%Y")  
  
ggplot(westData, aes(x = Ship.Year, y = Shipping.Time, color = Category, group = Category)) +  
 geom\_line(size = 1, alpha = 0.7) +  
 labs(x = "Year", y = "Shipping Time in days", title = "Shipping Time by Year and Category for west region") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 40, hjust = 1))



#3.Answer (Considering East Data)  
##Student First Name: Poojitha  
# Graph for East data only  
  
# Where as for this data am not using average shipping time , just ploting using the ship time with year and category  
eastData$Ship.Year <- format(eastData$Ship.Date, "%Y")  
  
ggplot(eastData, aes(x = Ship.Year, y = Shipping.Time, color = Category, group = Category)) +  
 geom\_line(size = 1, alpha = 0.7) +  
 labs(x = "Year", y = "Shipping Time in days", title = "Shipping Time by Year and Category for east region") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 40, hjust = 1))



1. Which categories have highest profit by region, chain-wide?

#4.Answer  
##Student Last Name: Velthuri  
# To find the highest profit be region , am using group by to calculate sum of profit and then arrange it in descending order after that displaying the total profit  
library(dplyr)  
profit<- combinedData %>%  
 group\_by(Region, Category) %>%  
 summarise(Total\_Profit = sum(Profit)) %>%  
 arrange(desc(Total\_Profit))

## `summarise()` has grouped output by 'Region'. You can override using the  
## `.groups` argument.

# View the results  
print(profit)

## # A tibble: 9 × 3  
## # Groups: Region [3]  
## Region Category Total\_Profit  
## <chr> <fct> <dbl>  
## 1 West Office Supplies 193910.  
## 2 West Technology 154504.  
## 3 West Furniture 85854.  
## 4 East Technology 74574.  
## 5 Central Office Supplies 65125.  
## 6 East Office Supplies 64480.  
## 7 Central Technology 39948.  
## 8 Central Furniture 35924.  
## 9 East Furniture 31864.

1. Which segments have the lowest profit by region?

#5.Answer  
##Student Whole Name: Poojitha Velthuri  
# In similar way to the 4th question we will no calculate the profit which is grouped by region and segment and arranged, by default it will be arranged as lowest to highest.  
library(dplyr)  
profit\_summary <- combinedData %>%  
 group\_by(Region, Segment) %>%  
 summarise(Total\_Profit = sum(Profit)) %>%  
 arrange(Total\_Profit)

## `summarise()` has grouped output by 'Region'. You can override using the  
## `.groups` argument.

# View the results  
print(profit\_summary)

## # A tibble: 9 × 3  
## # Groups: Region [3]  
## Region Segment Total\_Profit  
## <chr> <chr> <dbl>  
## 1 Central Home Office 27252.  
## 2 East Home Office 37650.  
## 3 Central Corporate 44997.  
## 4 East Corporate 48012.  
## 5 Central Consumer 68748.  
## 6 West Home Office 74372.  
## 7 East Consumer 85257.  
## 8 West Corporate 127935.  
## 9 West Consumer 231962.

1. What are yearly sales by region?

#6.Answer  
##Student First Name: Poojitha  
#To calculate the yearly sales by region using group and summarise which calculates the sales based on the give group  
# Calculate yearly sales by region  
yearly\_sales <- combinedData %>%  
 group\_by(Region, Ship.Year) %>%  
 summarise(Total\_Sales = sum(Sales)) %>%  
 arrange(Region, Ship.Year)

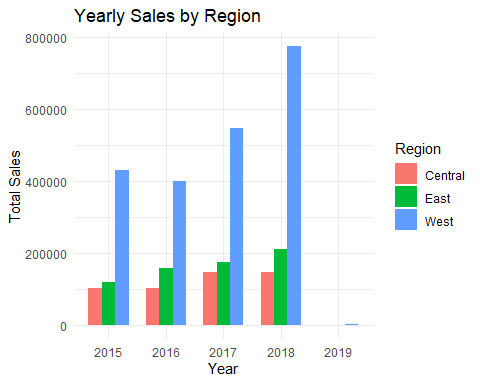
## `summarise()` has grouped output by 'Region'. You can override using the  
## `.groups` argument.

#Here the output is displayed for regions yearly based sales.  
print(yearly\_sales)

## # A tibble: 15 × 3  
## # Groups: Region [3]  
## Region Ship.Year Total\_Sales  
## <chr> <chr> <dbl>  
## 1 Central 2015 102900.  
## 2 Central 2016 103223.  
## 3 Central 2017 147071.  
## 4 Central 2018 147146.  
## 5 Central 2019 900.  
## 6 East 2015 119405.  
## 7 East 2016 159537.  
## 8 East 2017 175550.  
## 9 East 2018 210960.  
## 10 East 2019 601.  
## 11 West 2015 428989.  
## 12 West 2016 399176.  
## 13 West 2017 547331.  
## 14 West 2018 774297.  
## 15 West 2019 2720.

To get better understanding I have plotted bar graph for the respective scenario

# Plotting graph to get better insights of sales according to the year with respective of region.  
##Student First Name: Poojitha  
library(ggplot2)  
  
ggplot(yearly\_sales, aes(x = factor(Ship.Year), y = Total\_Sales, fill = Region)) +  
 geom\_bar(stat = "identity", position = "dodge", width = 0.7) +  
 labs(  
 title = "Yearly Sales by Region",  
 x = "Year",  
 y = "Total Sales",  
 fill = "Region"  
 ) +  
 scale\_y\_continuous(labels = function(x) format(x, scientific = FALSE)) +  
 theme\_minimal()

 7. Which customer purchased the most (by dollar amount, in a single order) store-wide and by region and what did they buy?

# To find the customer who purchased the most ,I have used filter and mentioned max sales which gives the details of customers who had purchased the most. this based on store wide.  
# Using slice(1) gives top one customer as output.  
  
library(dplyr)  
##Student Last Name: Velthuri  
# Store-wide: Find the customer with the maximum sales amount in a single order  
max\_customer\_storewide <- combinedData %>%  
 filter(Sales == max(Sales)) %>%  
 slice(1)   
  
# As the sales were asked in USD so converting sales into USD using paste0 and format  
max\_customer\_storewide$Sales <- paste0("$", format(max\_customer\_storewide$Sales, big.mark = ",", decimal.mark = ".", scientific = FALSE))  
  
cat("Store-wide:\n")

## Store-wide:

print(max\_customer\_storewide[, c("Customer.ID", "Customer.Name", "Sales","Product.Name")])

## Customer.ID Customer.Name Sales Product.Name  
## 1 TC-20980 Tamara Chand $17,499.95 Canon imageCLASS 2200 Advanced Copier

##Student Last Name: Velthuri  
#In order to get the details of customer by region here am using arrange which sets data by region and sales  
# By Region: Find the customer with the maximum sales amount in each region  
max\_customer\_by\_region <- combinedData %>%  
 group\_by(Region) %>%  
 filter(Sales == max(Sales)) %>%  
 arrange(Region, desc(Sales)) %>%  
 slice(1)  
  
  
cat("\nBy Region:\n")

##   
## By Region:

print(max\_customer\_by\_region[, c("Customer.Name", "Region", "Sales","Product.Name")])

## # A tibble: 3 × 4  
## # Groups: Region [3]  
## Customer.Name Region Sales Product.Name   
## <chr> <chr> <dbl> <chr>   
## 1 Tamara Chand Central 17500. Canon imageCLASS 2200 Advanced Copier  
## 2 Tom Ashbrook East 11200. Canon imageCLASS 2200 Advanced Copier  
## 3 Raymond Buch West 14000. Canon imageCLASS 2200 Advanced Copier