

Appendix – Data Analysis – Luca Zacharias

Query 1 (In the following, I will provide an excerpt of a data set. Can you write R code to visualize the given data in a meaningful way. “<data>”?):

ChatGPT-3.5:

Sure, assuming the data is stored in a CSV file named "data.csv", you can use the following R code to read the data and create visualizations:

```
# Load required libraries
install.packages("tidyverse")
library(tidyverse)

# Read the data from CSV file
data <- read.csv("data.csv")

# Check the structure of the data
str(data)

# Summary statistics of numerical columns
summary(data)

# Visualization 1: Bar plot for Categories
ggplot(data, aes(x = Category)) +
  geom_bar(fill = "skyblue") +
  ggtitle("Number of Orders by Category") +
  xlab("Category") +
  ylab("Number of Orders")

# Visualization 2: Scatter plot for Sales vs. Profit
ggplot(data, aes(x = Sales, y = Profit, color = Category)) +
  geom_point() +
  ggtitle("Sales vs. Profit") +
  xlab("Sales") +
  ylab("Profit")

# Visualization 3: Box plot for Sales by Region
```

```
ggplot(data, aes(x = Region, y = Sales, fill = Region)) +  
  geom_boxplot() +  
  ggtitle("Sales by Region") +  
  xlab("Region") +  
  ylab("Sales")
```

Visualization 4: Pie chart for Ship Modes

```
ggplot(data, aes(x = "", fill = ShipMode)) +  
  geom_bar(width = 1) +  
  coord_polar("y") +  
  ggtitle("Distribution of Ship Modes")
```

Visualization 5: Histogram for Discounts

```
ggplot(data, aes(x = Discount)) +  
  geom_histogram(binwidth = 0.05, fill = "skyblue", color = "black") +  
  ggtitle("Distribution of Discounts") +  
  xlab("Discount") +  
  ylab("Frequency")
```

In this code:

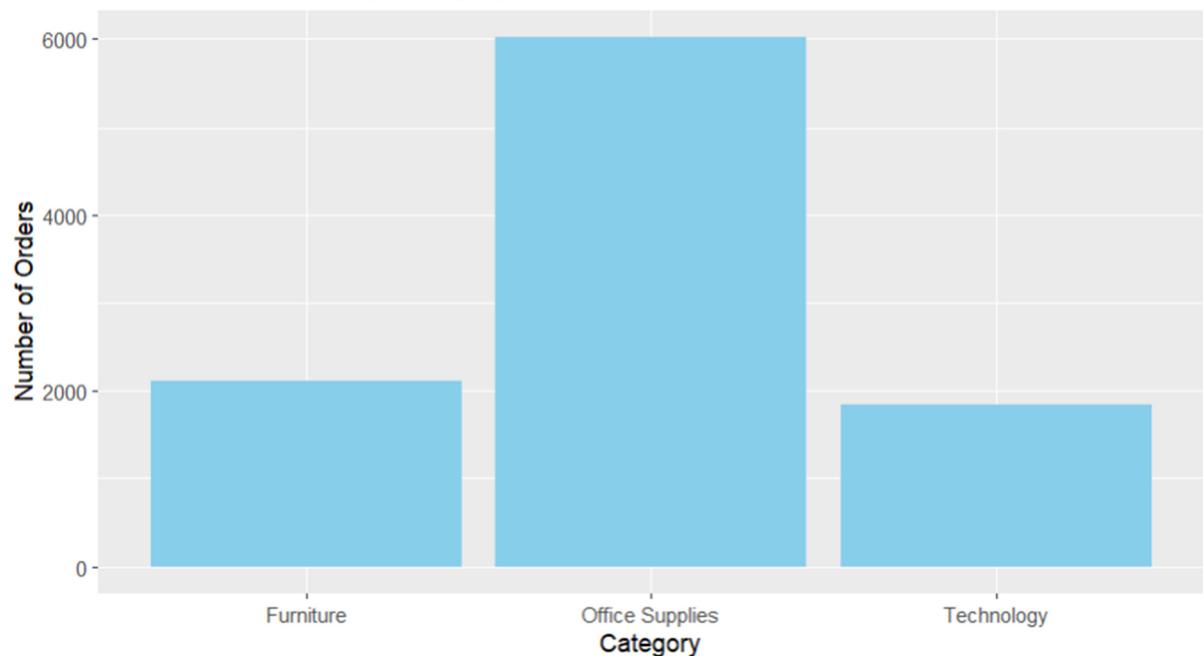
1. **The data is read from the CSV file.**
2. **Visualizations are created:**

- A bar plot shows the number of orders for each category.
- A scatter plot displays the relationship between sales and profit, with colors indicating different categories.
- A box plot illustrates sales distribution across different regions.
- A pie chart represents the distribution of ship modes.
- A histogram displays the distribution of discounts.

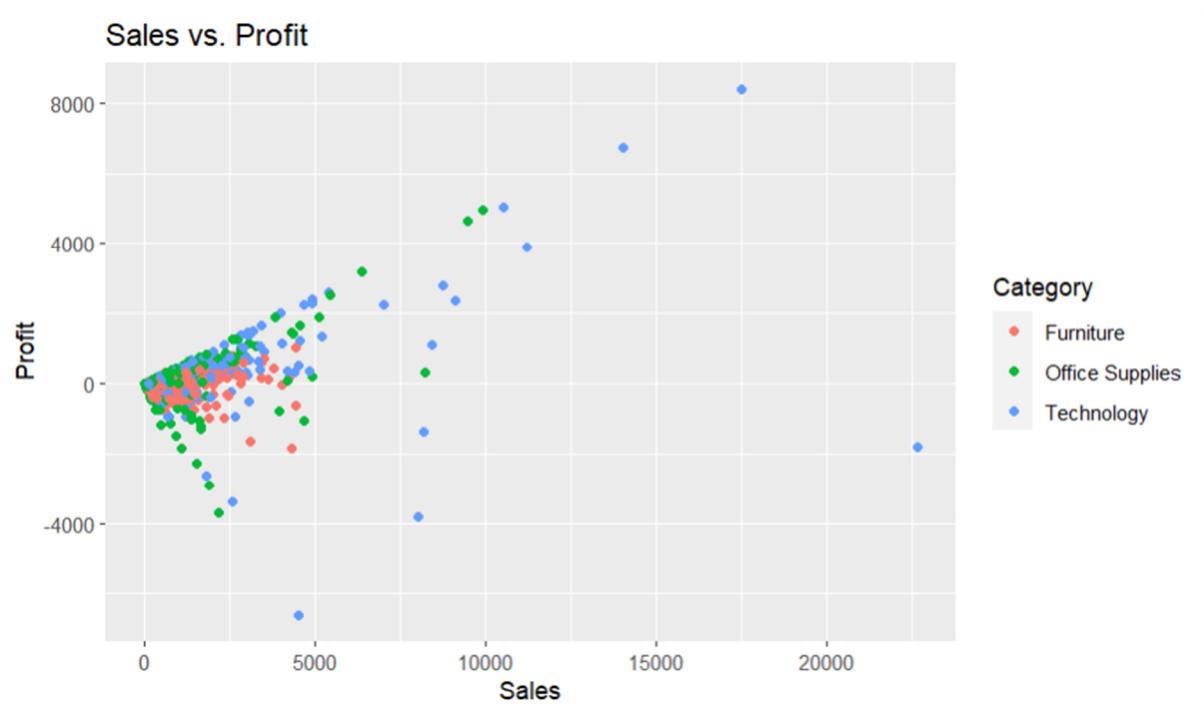
Make sure to customize the visualizations further according to your specific needs. You can modify the plot titles, axis labels, color schemes, and other plot aesthetics as desired.

Output of the provided Code:

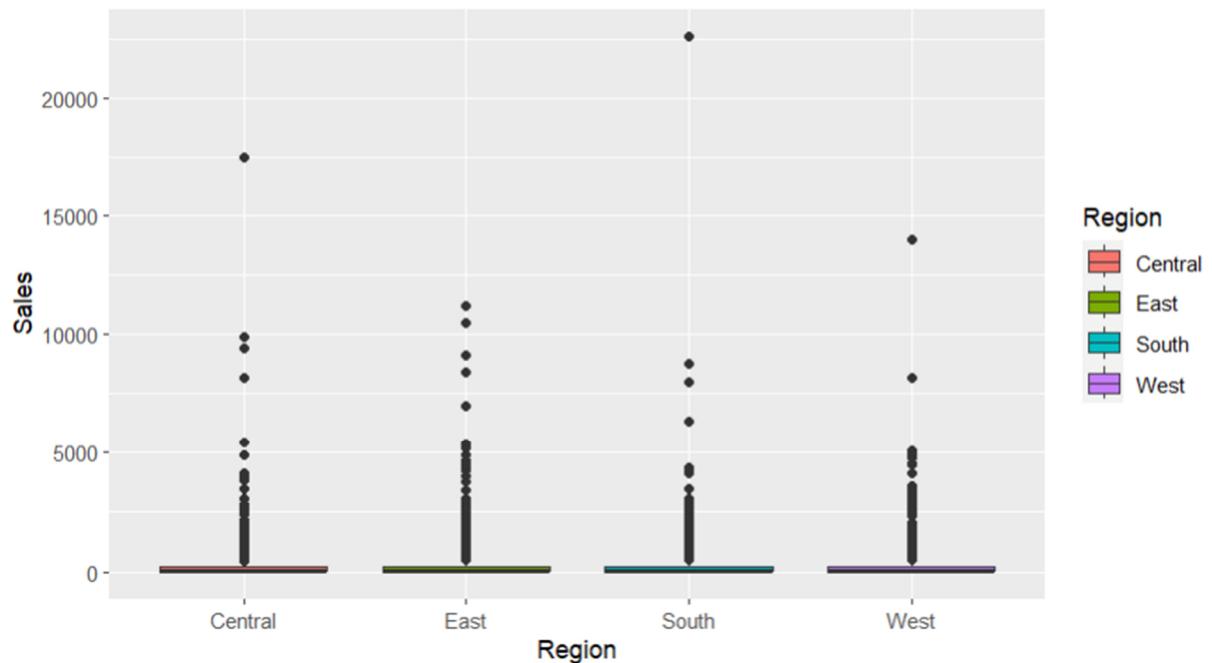
Number of Orders by Category



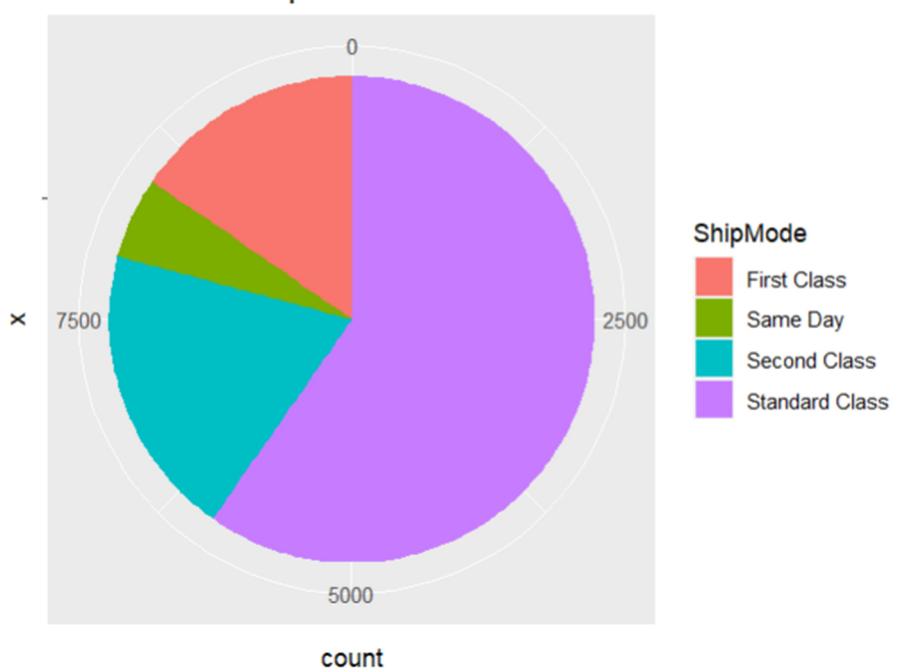
Sales vs. Profit

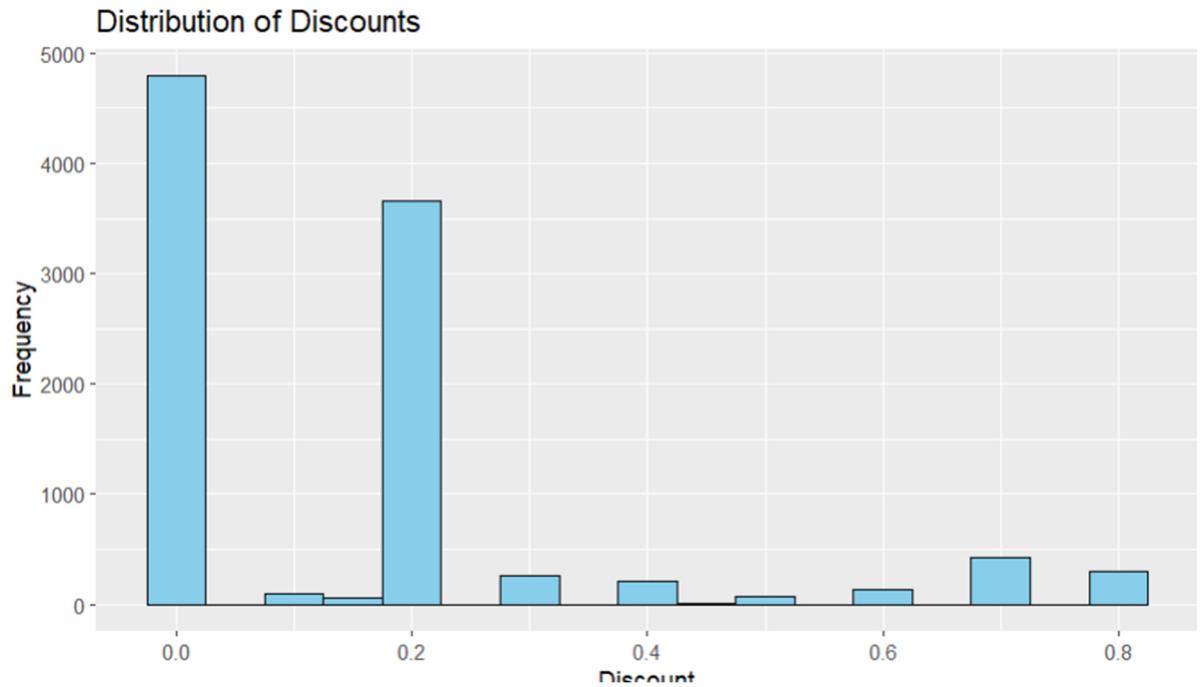


Sales by Region



Distribution of Ship Modes





ChatGPT-4.0:

Got it! Let's start by loading the data into R from the CSV. Once the data is loaded, we can proceed with some basic visualization.

Given the rich nature of the dataset, there are countless visualizations we can produce. However, I'll focus on a few:

1. **Sales by Category:** A bar plot showcasing the sales in each product category.
2. **Profit Ratio by Region:** A bar plot to understand the profit ratio in different regions.
3. **Sales and Profit on a Map:** Scatter plot on a US map showing sales and profit.
4. **Orders Over Time:** A time series plot showcasing sales over time.

```
# Load necessary libraries
install.packages(c("ggplot2", "dplyr", "ggmap", "lubridate"))

library(ggplot2)
library(dplyr)
library(ggmap)
library(lubridate)
```

```
# Load the data
data <- read.csv("your_data_file_name.csv")
```

```

# Convert OrderDate to a Date object
data$OrderDate <- as.Date(data$OrderDate, format="%Y-%m-%dT%H:%M:%S")

# 1. Sales by Category
ggplot(data, aes(x=Category, y=Sales, fill=Category)) +
  geom_bar(stat="identity") +
  ggtitle("Sales by Category") +
  theme_minimal()

# 2. Profit Ratio by Region
ggplot(data, aes(x=Region, y=ProfitRatio, fill=Region)) +
  geom_bar(stat="identity") +
  ggtitle("Profit Ratio by Region") +
  theme_minimal()

# 3. Sales and Profit on a US Map
# Note: Fetching the map can require an API key and can take time depending on your internet connection
us_map <- get_map(location="united states", zoom=4)
ggplot(data) +
  geom_point(aes(x=longitude, y=latitude, size=Sales, color=Profit), alpha=0.5) +
  ggtitle("Sales and Profit on US Map") +
  theme_minimal() +
  coord_fixed(ratio = 1.3) +
  borders("state") +
  theme(legend.position="bottom")

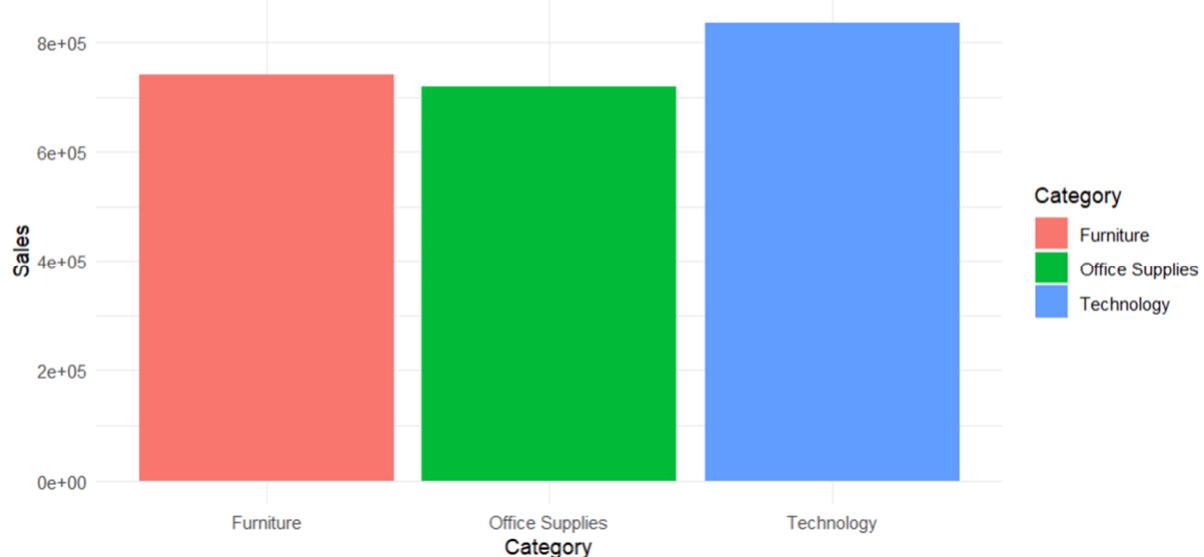
# 4. Orders Over Time
ggplot(data, aes(x=OrderDate, y=Sales, group=1)) +
  geom_line(color="blue") +
  ggtitle("Orders Over Time") +
  theme_minimal()

```

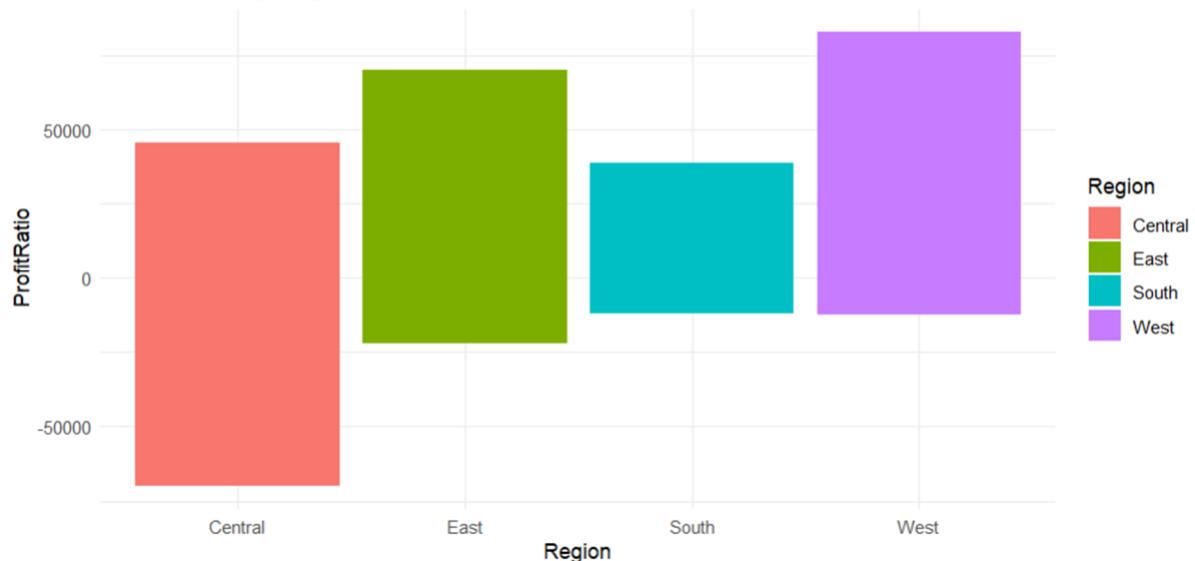
Remember to replace "your_data_file_name.csv" with the actual path of your CSV file.

Output of the provided Code:

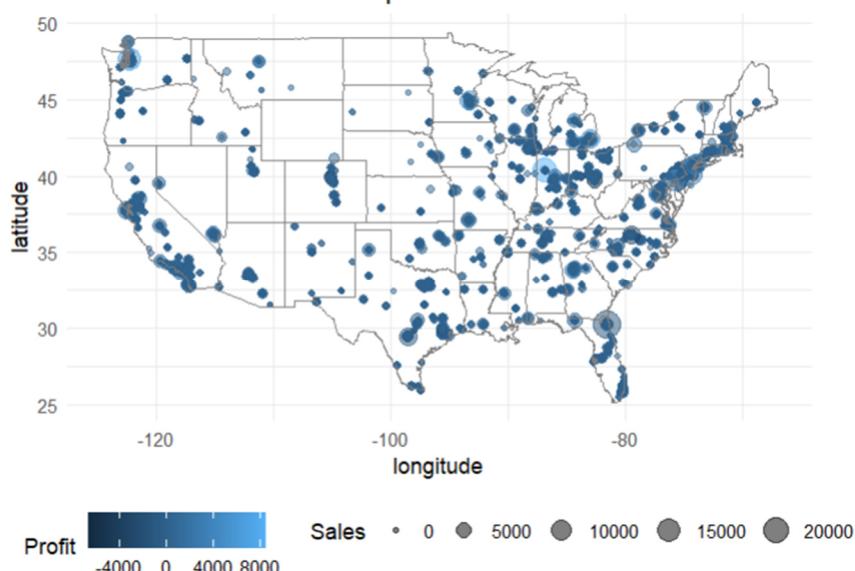
Sales by Category

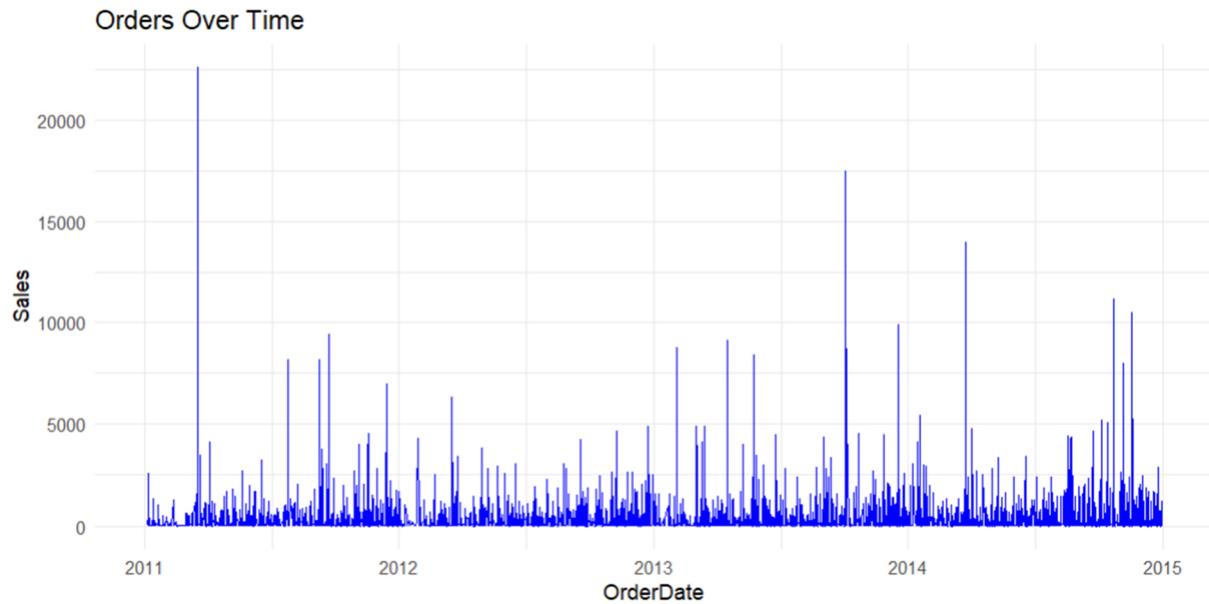


Profit Ratio by Region



Sales and Profit on US Map





Bing Chat:

Sure, I can help with that. Here's an example of how you might use R to visualize this data. This code will create a bar plot of the total sales by category.

```
# Load necessary libraries
library(ggplot2)
library(readr)

# Assuming the data is in a CSV file named 'data.csv'
data <- read_csv('data.csv')

# Convert OrderDate to Date type
data$OrderDate <- as.Date(data$OrderDate)

# Summarize Sales by Category
sales_by_category <- aggregate(data$Sales, by = list(Category = data$Category), FUN = sum)

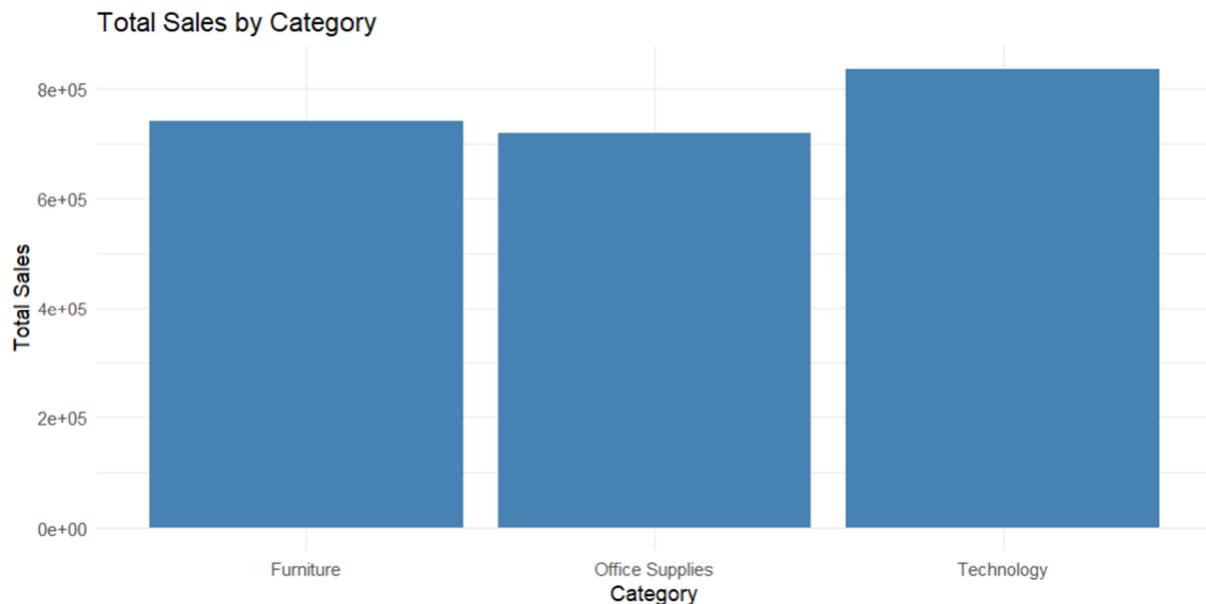
# Create a bar plot
ggplot(sales_by_category, aes(x = Category, y = x)) +
  geom_bar(stat = 'identity', fill = 'steelblue') +
  labs(x = 'Category', y = 'Total Sales', title = 'Total Sales by Category') +
  theme_minimal()
```

Please replace '`data.csv`' with the actual path to your CSV file. This code assumes that your date is in the ISO 8601 format (yyyy-mm-dd). If it's not, you may need to adjust the date conversion line.

This is just a basic example. Depending on what you're interested in, you might want to create different visualizations or perform additional data processing steps. For

example, you could look at sales over time, sales by region, etc. Let me know if you need help with something more specific!

Output of the provided Code:



Google Bard:

```
# Load the data
sales_data <- read.csv("sales_data.csv")

# Create a bar chart of sales by category
barplot(sales_data$Sales, names.arg = sales_data$Category, col = sales_data$Color, main = "Sales by Category")

# Create a line chart of sales by day
ggplot(sales_data, aes(x = OrderDate, y = Sales)) +
  geom_line() +
  labs(title = "Sales by Day")

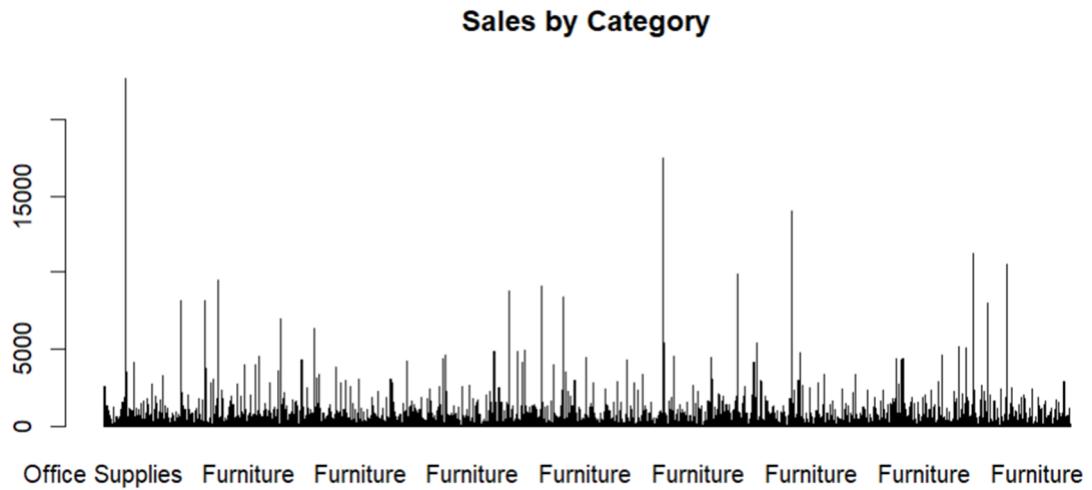
# Create a scatter plot of sales by profit
ggplot(sales_data, aes(x = Profit, y = Sales)) +
  geom_point() +
  labs(title = "Sales by Profit")
```

```
# Create a map of sales by state  
ggplot(sales_data, aes(x = longitude, y = latitude, color = State, size = Sales)) +  
  geom_point() +  
  labs(title = "Sales by State")
```

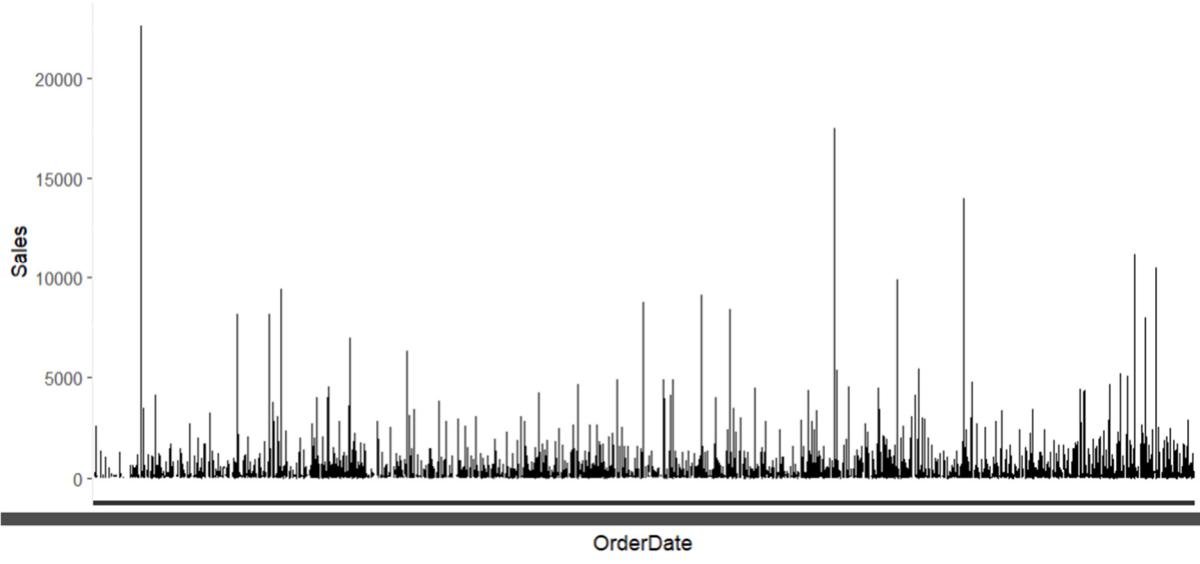
These visualizations provide different insights into the sales data. The bar chart shows that the "Technology" category has the highest sales, followed by "Furniture" and "Office Supplies." The line chart shows that sales are generally increasing over time, with some spikes and dips. The scatter plot shows that there is a positive correlation between sales and profit, meaning that products that generate higher profits also tend to generate higher sales. The map shows that sales are highest in the most populous states, such as California and New York.

You can also use other R packages to create more complex and interactive data visualizations. For example, you could use the `shiny` package to create a web app that allows users to explore the data in different ways.

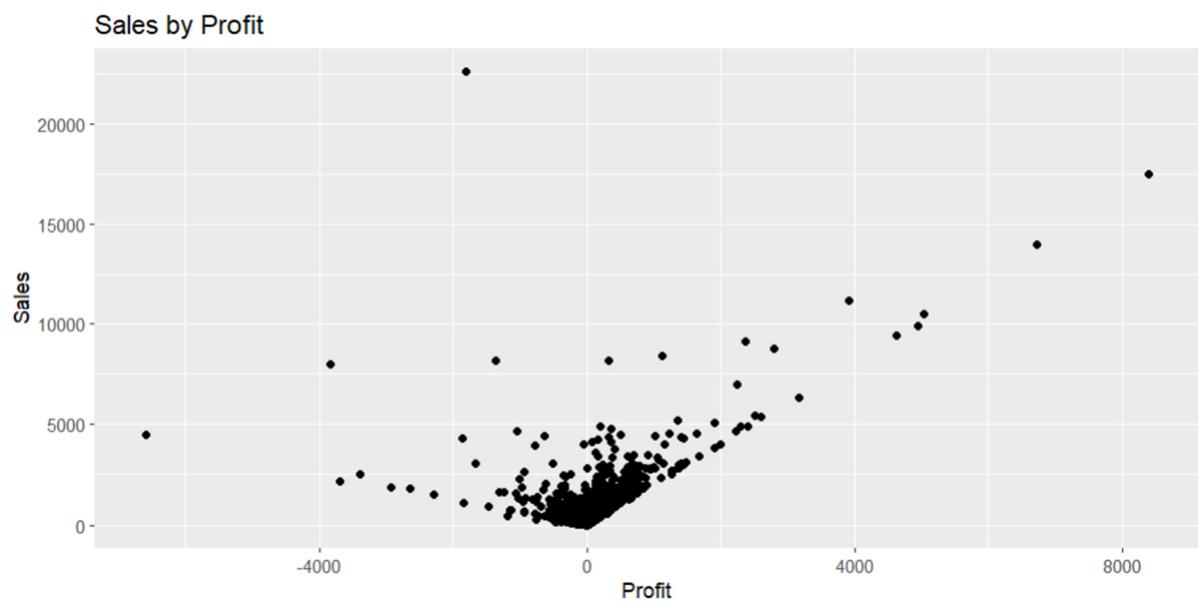
Output of the provided Code:

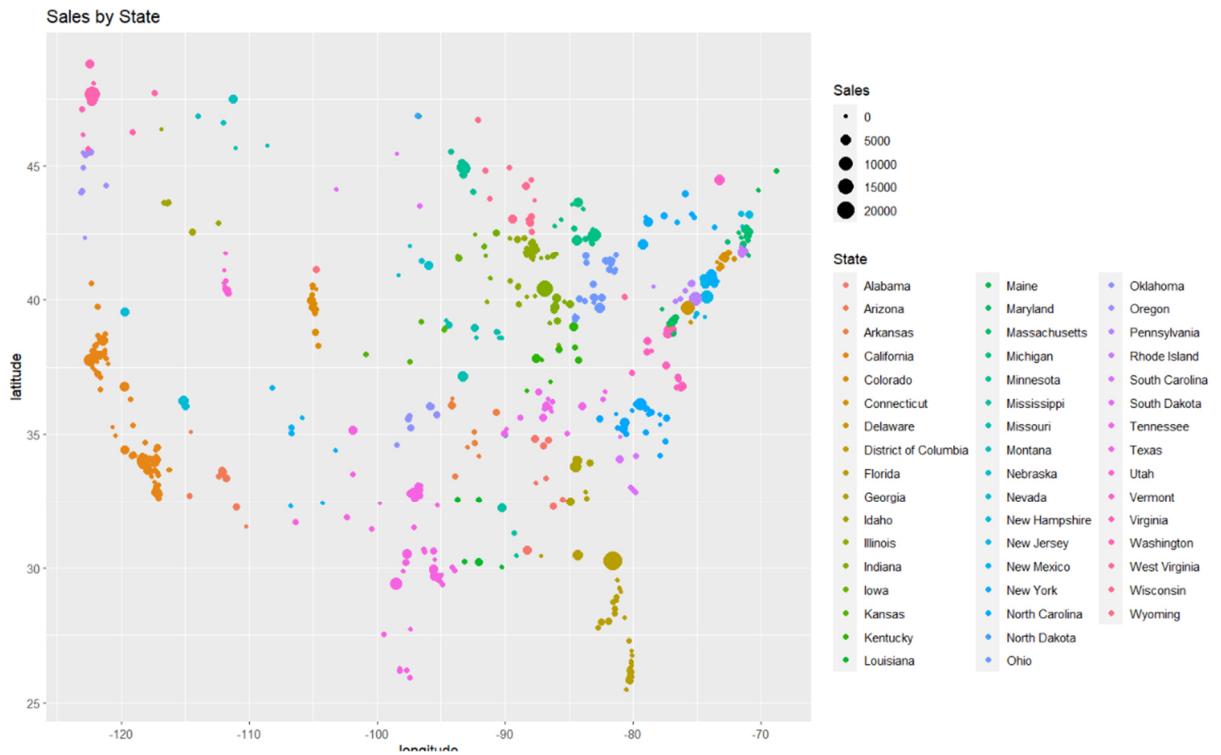


Sales by Day



Sales by Profit





YouChat:

```
# Load the data.frame called orders
orders <- read.csv("orders.csv")

# Plot a bar chart of total sales by category
library(ggplot2)
ggplot(orders, aes(x = Category, y = Sales, fill = Category)) +
  geom_bar(stat = "identity")

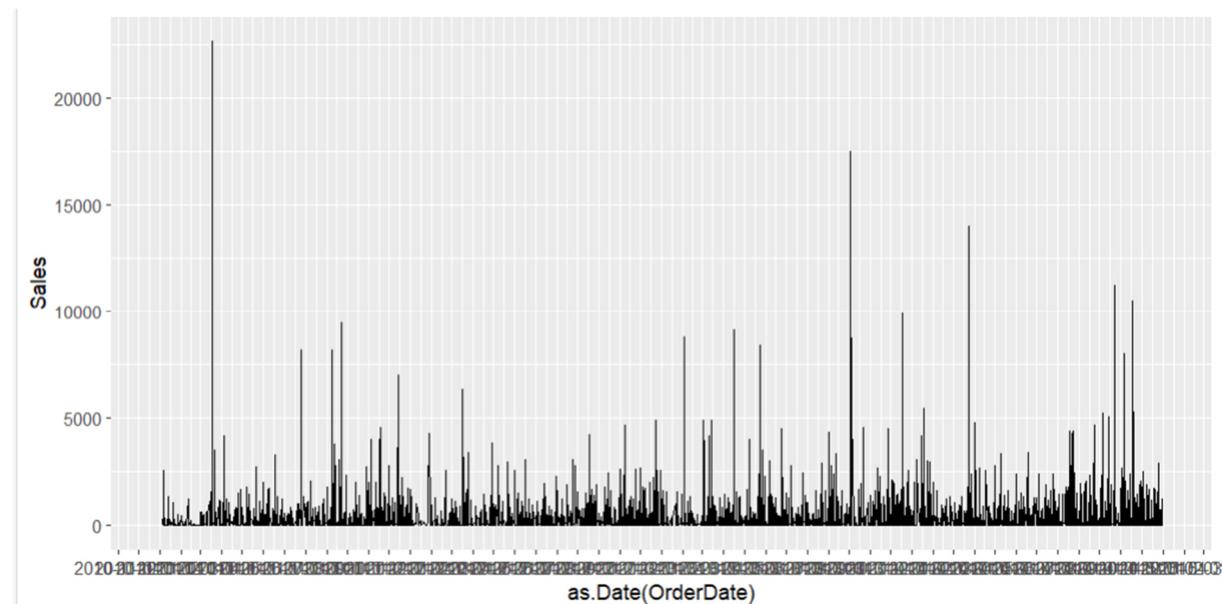
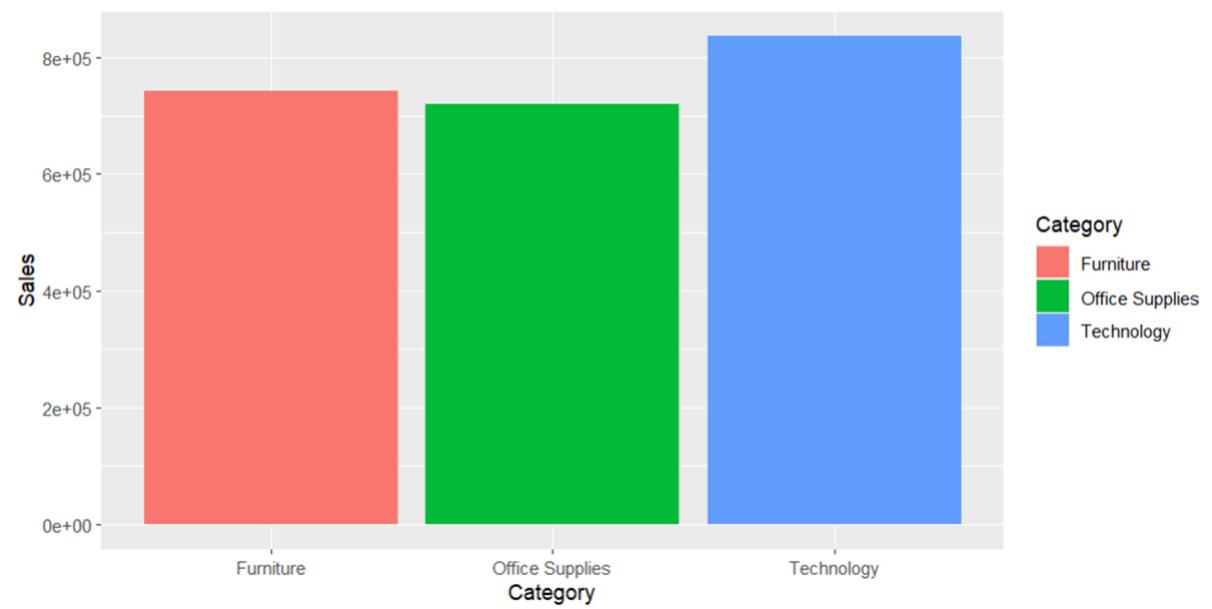
# Plot a line chart of total sales over time
ggplot(orders, aes(x = OrderDate, y = Sales)) +
  geom_line() +
  scale_x_date(date_breaks = "1 month")

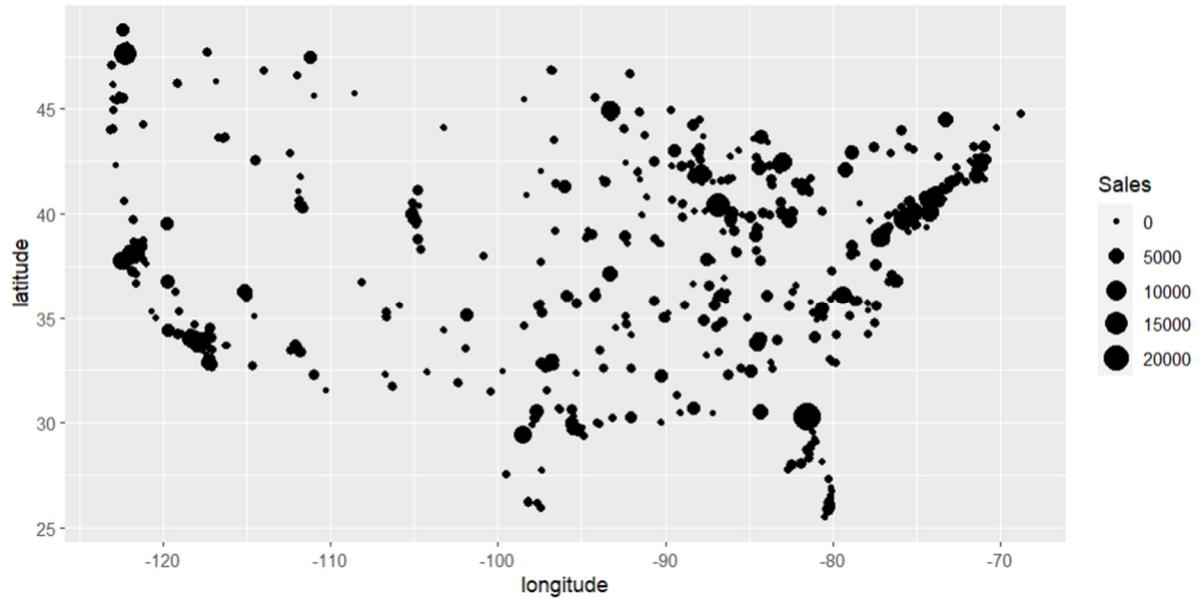
# Plot a map of total sales by city
library(ggmap)
ggplot(orders, aes(x = longitude, y = latitude, size = Sales)) +
  geom_point()
```

The bar chart by category will show you which product categories generate the most revenue. The line chart over time will reveal sales trends. And the map plot will highlight the most profitable cities.

Hope this helps! Let me know if you have any other questions.

Output of the provided Code:





Query 2 (Can you create a diagram which shows the possible correlation between ShipMode and ShipStatus?):

ChatGPT-3.5:

To visualize the possible correlation between `ShipMode` and `ShipStatus`, you can create a stacked bar plot. Each bar represents a `ShipMode`, and the segments within the bar represent different `ShipStatus` categories. Here's how you can do it using R:

```
# Load required libraries if not already loaded
install.packages("tidyverse")
library(tidyverse)

# Read the data from CSV file
data <- read.csv("data.csv")

# Create a stacked bar plot for ShipMode and ShipStatus
ggplot(data, aes(x = ShipMode, fill = ShipStatus)) +
  geom_bar() +
  ggtitle("ShipMode vs. ShipStatus") +
  xlab("ShipMode") +
  ylab("Count") +
  scale_fill_manual(values = c("#66c2a5", "#fc8d62", "#8da0cb")) +
  theme_minimal()
```

In this plot:

- `x = ShipMode` sets the x-axis to represent different shipping modes.
- `fill = ShipStatus` divides the bars into segments based on different shipping statuses.
- `geom_bar()` creates the stacked bar plot.
- `scale_fill_manual(values = c("#66c2a5", "#fc8d62", "#8da0cb"))` sets custom colors for the different `ShipStatus` categories.

You can customize the colors, titles, and other plot elements according to your preference. This plot will give you a visual representation of how different `ShipStatus` categories are distributed across various `ShipMode` values.

Output of the provided Code:



ChatGPT-4.0:

To visualize the correlation between `ShipMode` and `ShipStatus`, you may want to look at their distribution together, probably using a mosaic plot or a stacked bar plot. This will show you the distribution of `ShipStatus` for each `ShipMode` and might help to indicate if a particular `ShipStatus` is more common for a particular `ShipMode`.

However, keep in mind that these plots show associations, not correlations. Correlation typically refers to a linear relationship between two numeric variables, whereas `ShipMode` and `ShipStatus` are both categorical.

Here's how to create a stacked bar plot to visualize the relationship:

```

# Load necessary libraries

install.packages(c("ggplot2", "dplyr"))

library(ggplot2)

library(dplyr)

# Load the data

data <- read.csv("your_data_file_name.csv")

# Create a stacked bar plot

ggplot(data, aes(x=ShipMode, fill=ShipStatus)) +
  geom_bar(position="stack") +
  ggtitle("Distribution of ShipStatus by ShipMode") +
  theme_minimal() +
  labs(y="Number of Orders", x="Ship Mode", fill="Ship Status")

```

For a mosaic plot, you can use the **vcd** package:

```

# Load necessary libraries

install.packages("vcde")

library(vcd)

# Load the data

data <- read.csv("your_data_file_name.csv")

# Create a mosaic plot

mosaic(~ ShipMode + ShipStatus, data=data, main="Mosaic Plot of ShipMode vs ShipStatus")

```

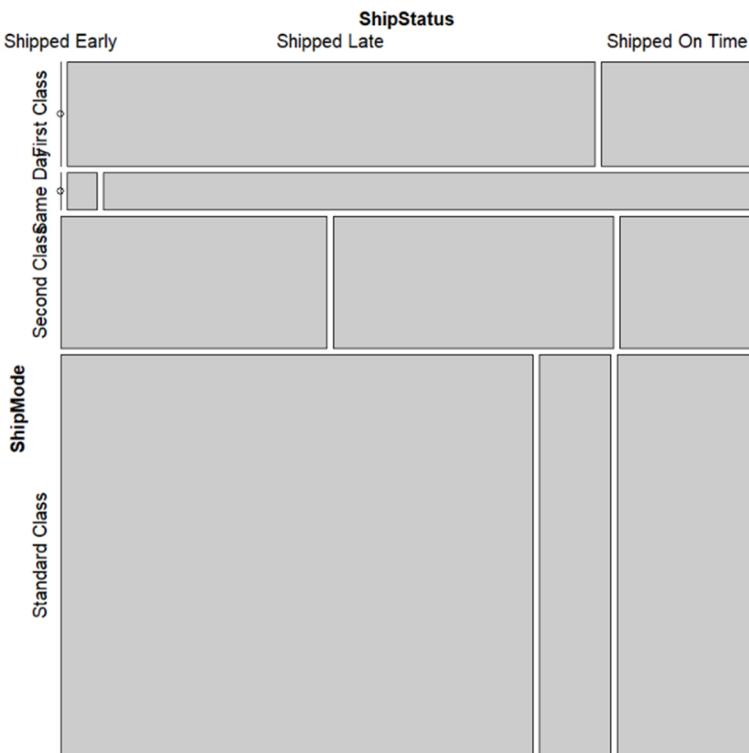
Again, remember to replace "your_data_file_name.csv" with the actual path of your CSV file. These plots will help you visualize the distribution of **ShipStatus** across different **ShipModes**. If you notice that a particular **ShipStatus** is common for a specific **ShipMode**, this might indicate an association between the two variables.

Output of the provided Code:

Distribution of ShipStatus by ShipMode



Mosaic Plot of ShipMode vs ShipStatus



Bing Chat:

Sure, I can help with that. Here's an example of how you might use R to visualize the correlation between `ShipMode` and `ShipStatus`. This code will create a mosaic plot which is a graphical display of the cell frequencies of a contingency table.

```
# Load necessary libraries
library(vcd)
library(readr)
```

```

# Assuming the data is in a CSV file named 'data.csv'
data <- read_csv('data.csv')

# Create a contingency table of ShipMode and ShipStatus
contingency_table <- table(data$ShipMode, data$ShipStatus)

# Create a mosaic plot
mosaic(contingency_table, main="Mosaic Plot of ShipMode vs ShipStatus")

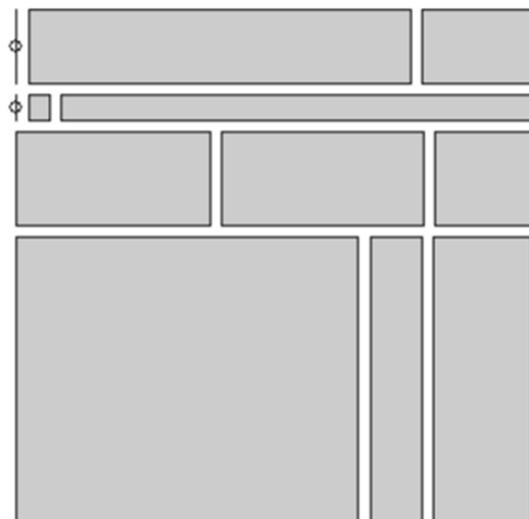
```

Please replace '`data.csv`' with the actual path to your CSV file. This code assumes that `ShipMode` and `ShipStatus` are categorical variables. If they're not, you may need to convert them to factors using the `as.factor()` function.

This is just a basic example. Depending on what you're interested in, you might want to create different visualizations or perform additional data processing steps. Let me know if you need help with something more specific!

Output of the provided Code:

Mosaic Plot of ShipMode vs ShipStatus



Ask to add labels:

```
contingency_table <- table(data$ShipMode, data$ShipStatus)
```

```
# Create a mosaic plot with labels
```

```
mosaic(contingency_table, main="Mosaic Plot of ShipMode vs ShipStatus", labeling =
labeling_values)
```

Error in if (split_vertical[i]) { : missing value where TRUE/FALSE needed

Suggested fix:

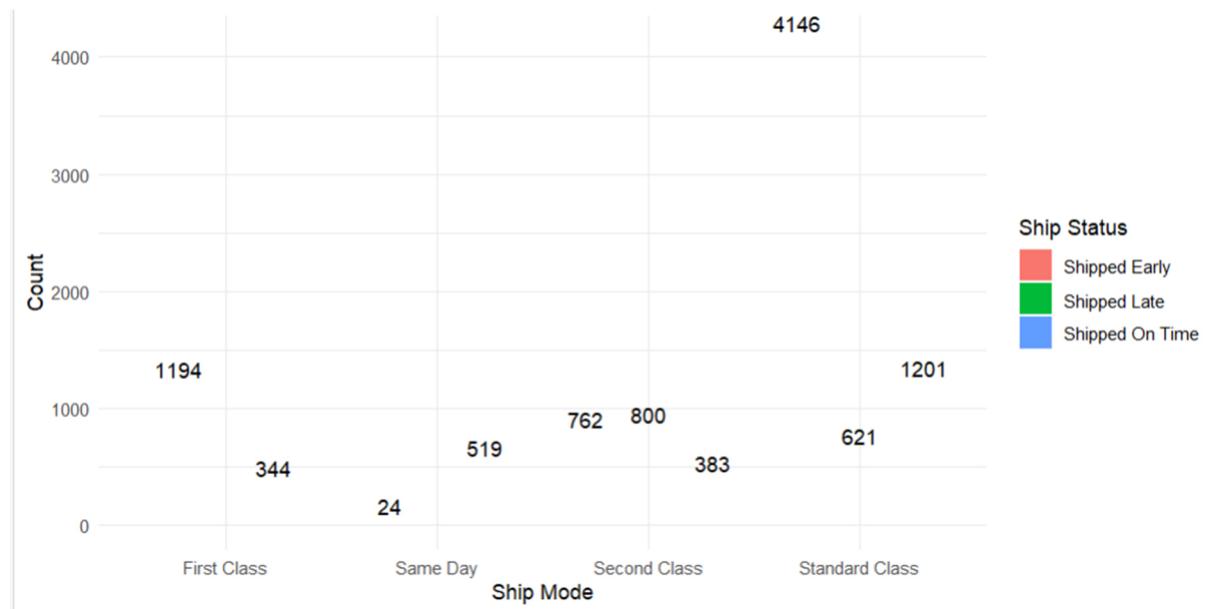
```
if (!is.na(split_vertical[i]) && split_vertical[i]) {
```

```
  # Your code here
```

```
}
```

Error: object 'split_vertical' not found

- To further work on the following Querys. I found a working diagram after a few queries, which still was a very poor diagram.



Google Bard:

```
# Create a data frame of the possible ShipMode and ShipStatus combinations
ship_mode_status_df <- data.frame(
  ShipMode = c("Standard Class", "Second Class", "First Class"),
  ShipStatus = c("Shipped Early", "Shipped On Time", "Shipped Late")
)
```

```
# Create a plot of the data frame
```

```
ggplot(ship_mode_status_df, aes(x = ShipMode, y = ShipStatus)) +
  geom_point() +
  labs(title = "Correlation between ShipMode and ShipStatus")
```

```

# Add labels to the data points
geom_label(
  data = ship_mode_status_df,
  aes(x = ShipMode, y = ShipStatus, label = ShipStatus),
  hjust = c(-0.5, 0, 0.5),
  vjust = c(0, 0, 0)
)

```

This code will create a scatter plot with the ShipMode on the x-axis and the ShipStatus on the y-axis. The data points will be labeled with the ShipStatus.

You can also use other R packages to create more complex and interactive diagrams. For example, you could use the `graphviz` package to create a directed acyclic graph (DAG) that shows the possible causal relationships between ShipMode and ShipStatus.

Output of the provided Code:



YouChat:

Certainly! Here's an example of R code that you can use to create a diagram showing the possible correlation between ShipMode and ShipStatus using the `ggplot2` package:

```

library(ggplot2)

# Assuming you have a data frame named 'df' with ShipMode and ShipStatus columns

```

```
# Create a stacked bar chart  
ggplot(df, aes(x = ShipMode, fill = ShipStatus)) +  
  geom_bar()  
  
# Create a heatmap  
ggplot(df, aes(x = ShipMode, y = ShipStatus, fill = ShipStatus)) +  
  geom_tile()  
  
# Create a cross-tabulation table  
table_df <- table(df$ShipMode, df$ShipStatus)  
heatmap(table_df)
```

In the first example, `geom_bar()` creates a stacked bar chart where each bar represents a `ShipMode` category, and the heights of the bars represent the count or frequency of each `ShipStatus` category within each `ShipMode`.

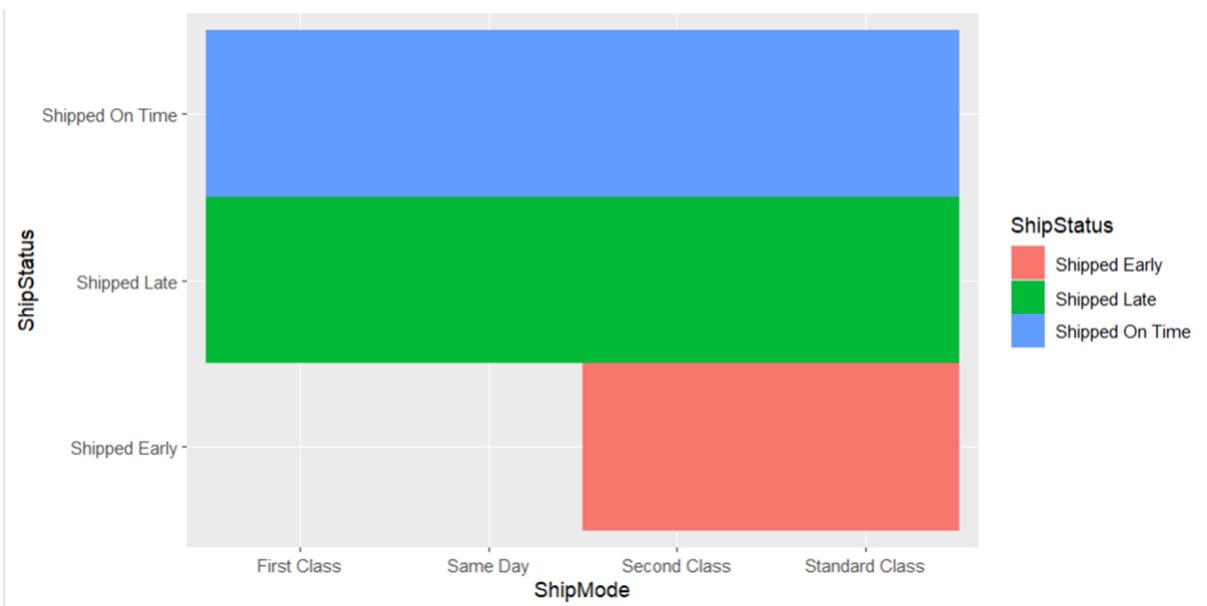
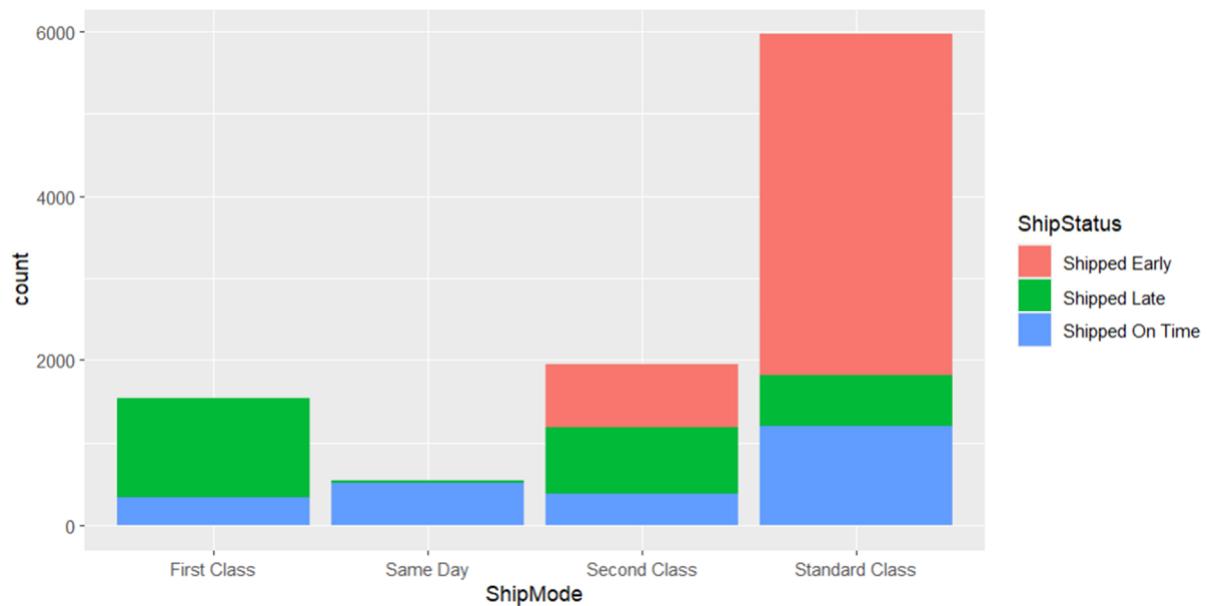
In the second example, `geom_tile()` creates a heatmap where each cell represents a combination of `ShipMode` and `ShipStatus`, and the fill color of each cell indicates the count or frequency of that combination.

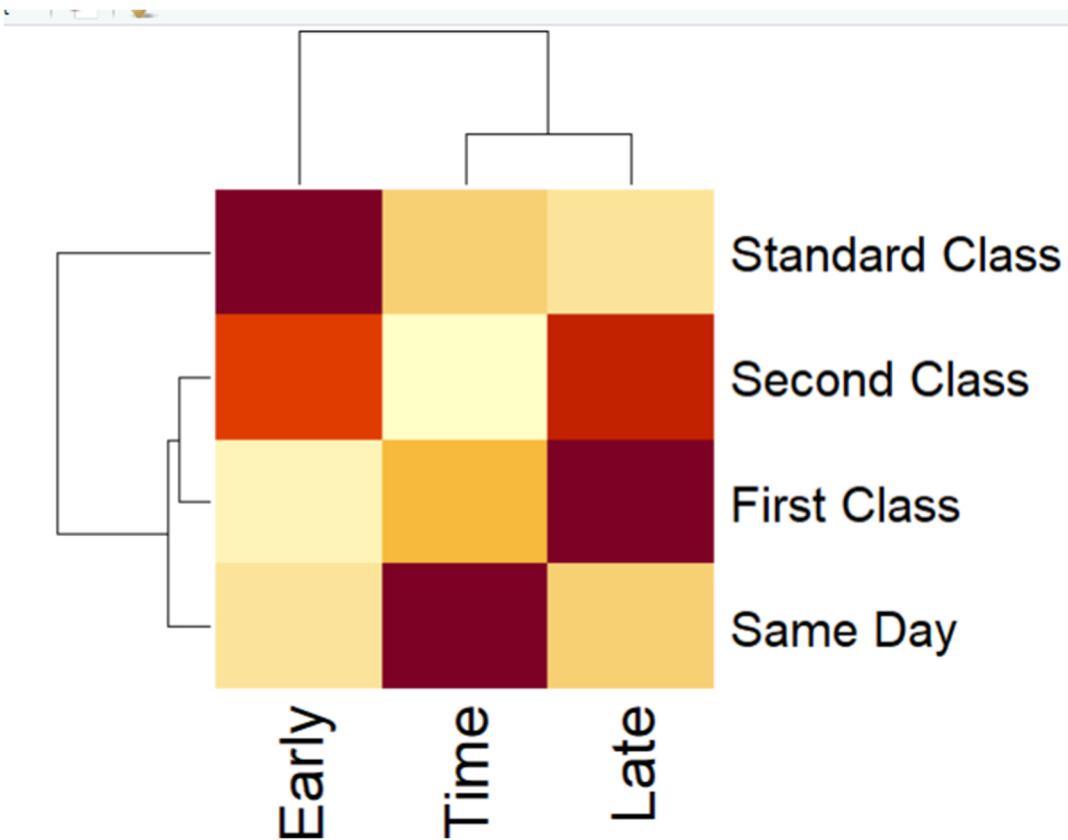
In the third example, `heatmap()` creates a cross-tabulation table and displays it as a heatmap, where each row represents a `ShipMode` category, each column represents a `ShipStatus` category, and the color indicates the count or frequency of each combination.

You can choose the visualization type that best suits your data and analysis goals. Make sure to adjust the code based on your specific data frame name and column names.

I hope this helps! Let me know if you have any further questions.

Output of the provided Code:





Query 3 (Can you also include the time it took to deliver (timespan between OrderDate and Shipdate) in this diagram?):

ChatGPT-3.5:

Certainly! To include the time it took to deliver (timespan between OrderDate and ShipDate) in the diagram, you can create a grouped bar plot where each bar represents a **ShipMode**, and within each bar, there are grouped bars for different **ShipStatus** categories. The height of the grouped bars will represent the count of orders, and the color can represent different **ShipStatus** categories. Here's how you can modify the R code to create this grouped bar plot:

```
data$OrderDate <- as.Date(data$OrderDate)
```

```
data$ShipDate <- as.Date(data$ShipDate)
```

```
# Calculate the time it took to deliver (in days)
```

```
data$DeliveryTime <- as.numeric(difftime(data$ShipDate, data$OrderDate, units = "days"))
```

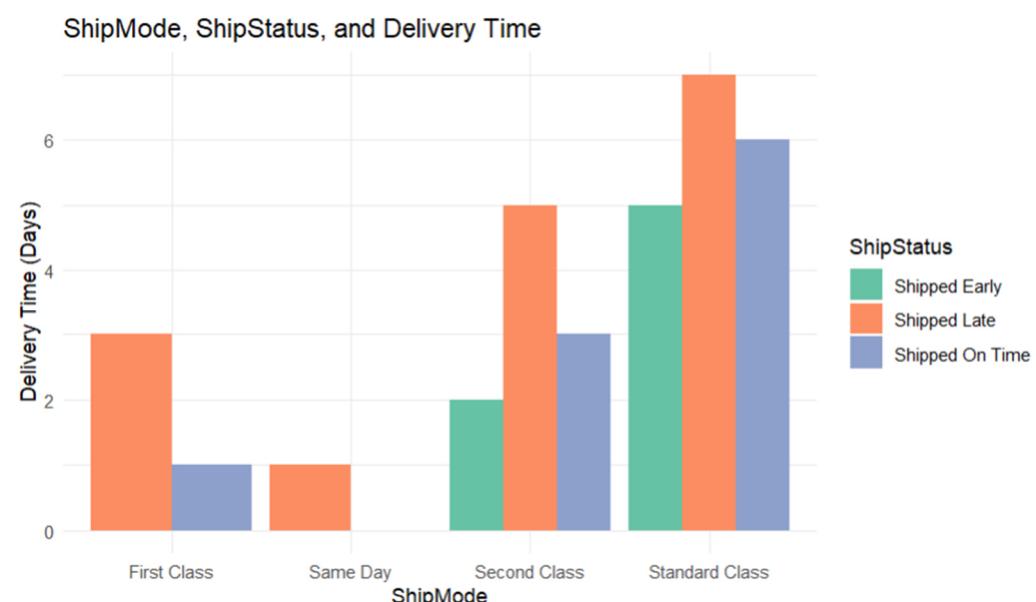
```
# Create a grouped bar plot for ShipMode, ShipStatus, and DeliveryTime
ggplot(data, aes(x = ShipMode, y = DeliveryTime, fill = ShipStatus)) +
  geom_bar(stat = "identity", position = "dodge") +
  ggtitle("ShipMode, ShipStatus, and Delivery Time") +
  xlab("ShipMode") +
  ylab("Delivery Time (Days)") +
  scale_fill_manual(values = c("#66c2a5", "#fc8d62", "#8da0cb")) +
  theme_minimal()
```

In this modified plot:

- `x = ShipMode` sets the x-axis to represent different shipping modes.
- `y = DeliveryTime` sets the y-axis to represent the delivery time (in days).
- `fill = ShipStatus` divides the grouped bars into segments based on different shipping statuses.
- `geom_bar(stat = "identity", position = "dodge")` creates the grouped bar plot with delivery time represented by the height of the bars.

This plot will give you a visual representation of how delivery time varies across different `ShipMode` values and different `ShipStatus` categories.

Output of the provided Code:



ChatGPT-4.0:

Certainly! To visualize the timespan between `OrderDate` and `ShipDate` alongside the relationship between `ShipMode` and `ShipStatus`, a suitable approach would be to use a faceted boxplot or violin plot. This way, for each `ShipMode` and `ShipStatus` combination, you can see the distribution of delivery timespans.

First, you'll need to compute the delivery timespan. Here's how to do this:

```
# Load necessary libraries
install.packages(c("ggplot2", "dplyr", "lubridate"))

library(ggplot2)
library(dplyr)
library(lubridate)

# Load the data
data <- read.csv("your_data_file_name.csv")

# Compute delivery timespan in days
data$DeliveryTimespan <- as.numeric(difftime(ymd(data$ShipDate), ymd(data$OrderDate), units =
"days"))

Once the timespan is computed, you can create a faceted boxplot or violin plot:

# Boxplot
ggplot(data, aes(x=ShipMode, y=DeliveryTimespan, fill=ShipStatus)) +
  geom_boxplot() +
  facet_wrap(~ ShipStatus, scales = "free_x") +
  ggtitle("Delivery Timespan by ShipMode and ShipStatus") +
  theme_minimal() +
  labs(y="Delivery Timespan (days)", x="Ship Mode", fill="Ship Status")

# If you prefer a violin plot, replace `geom_boxplot()` with `geom_violin()`
```

This plot will show you the distribution of delivery timespans for each combination of `ShipMode` and `ShipStatus`. A boxplot captures the median, quartiles, and potential outliers, while a violin plot additionally provides insight into the overall distribution shape.

Again, replace "your_data_file_name.csv" with the actual path of your CSV file. This approach helps you visualize if certain shipping modes or statuses are associated with longer or shorter delivery timespans.

- Has an error but is able to fix it when asked to

Warning messages: 1: All formats failed to parse. No formats found. 2: All formats failed to parse. No formats found."

If the dates have a timestamp, like "YYYY-MM-DD HH:MM:SS":

```
data$DeliveryTimespan <- as.numeric(difftime(ymd_hms(data$ShipDate), ymd_hms(data$OrderDate),  
units = "days"))
```

Output of the provided Code:



Bing Chat:

Sure, you can include the delivery time in the plot by creating a new variable that represents the number of days between OrderDate and ShipDate. Here's how you can do it:

```
# Load necessary libraries  
library(dplyr)  
library(ggplot2)  
library(readr)  
  
# Assuming the data is in a CSV file named 'data.csv'  
data <- read_csv('data.csv')  
  
# Convert OrderDate and ShipDate to Date type  
data$OrderDate <- as.Date(data$OrderDate)
```

```

data$ShipDate <- as.Date(data$ShipDate)

# Calculate the delivery time in days
data$DeliveryTime <- as.numeric(data$ShipDate - data$OrderDate)

# Count the number of each combination of ShipMode and ShipStatus
data_count <- data %>%
  group_by(ShipMode, ShipStatus) %>%
  summarise(count = n(), avg_delivery_time = mean(DeliveryTime))

# Create a bar plot with labels and color the bars by average delivery time
ggplot(data_count, aes(x = ShipMode, fill = ShipStatus)) +
  geom_bar(aes(y = count), position = "dodge") +
  geom_text(aes(label = count, y = count), vjust = -0.5, position = position_dodge(0.9)) +
  scale_fill_gradient(low = "blue", high = "red") +
  labs(x = 'Ship Mode', y = 'Count', fill = 'Ship Status') +
  theme_minimal()

```

In this code, I've added a new variable `DeliveryTime` to the data frame that represents the number of days between `OrderDate` and `ShipDate`. I've also modified the `summarise()` function to calculate the average delivery time for each combination of `ShipMode` and `ShipStatus`. The bars in the plot are then colored according to this average delivery time, with shorter times in blue and longer times in red. Please replace '`data.csv`' with the actual path to your CSV file. This code assumes that `ShipMode` and `ShipStatus` are categorical variables. If they're not, you may need to convert them to factors using the `as.factor()` function.

Let me know if this helps or if you need help with something more specific!

Error: Caused by error in `setup_params()`: !`stat_count()` must only have an x or y aesthetic.

- Ask to fix the error:

Error: Discrete value supplied to continuous scale

- Is not able to fix the code

Google Bard:

```

# Load the data
sales_data <- read.csv("sales_data.csv")

# Create a new dataframe with the timespan between OrderDate and ShipDate
sales_data$TimetoDeliver <- sales_data$ShipDate - sales_data$OrderDate

```

```
# Create a scatter plot of the timespan between OrderDate and ShipDate by ShipMode and ShipStatus
ggplot(sales_data, aes(x = TimetoDeliver, y = ShipStatus, color = ShipMode)) +
  geom_point() +
  labs(title = "Time to Deliver by ShipMode and ShipStatus")
```

This code will create a scatter plot with the timespan between OrderDate and ShipDate on the x-axis and the ShipStatus on the y-axis. The data points will be colored by the ShipMode.

You can also use other R packages to create more complex and interactive diagrams. For example, you could use the `graphviz` package to create a directed acyclic graph (DAG) that shows the possible causal relationships between ShipMode, ShipStatus, and the time it took to deliver.

However, it is important to note that a DAG is only a theoretical model and the actual relationships between these variables may be more complex.

Error in sales_data\$ShipDate - sales_data\$OrderDate :

non-numeric argument to binary operator

- Asked to fix the error

You are correct, I cannot just take the difference between the Dates because they are not numeric values but date formats. To calculate the timespan between OrderDate and ShipDate, we need to use the `difftime()` function in R.

Here is some R code to calculate the timespan between OrderDate and ShipDate:

```
# Calculate the timespan between OrderDate and ShipDate
sales_data$TimetoDeliver <- difftime(sales_data$ShipDate, sales_data$OrderDate, units = "days")
```

This code will create a new column in the `sales_data` dataframe called `TimetoDeliver`, which will contain the timespan between OrderDate and ShipDate in days.

We can then use this `TimetoDeliver` column to create the diagram:

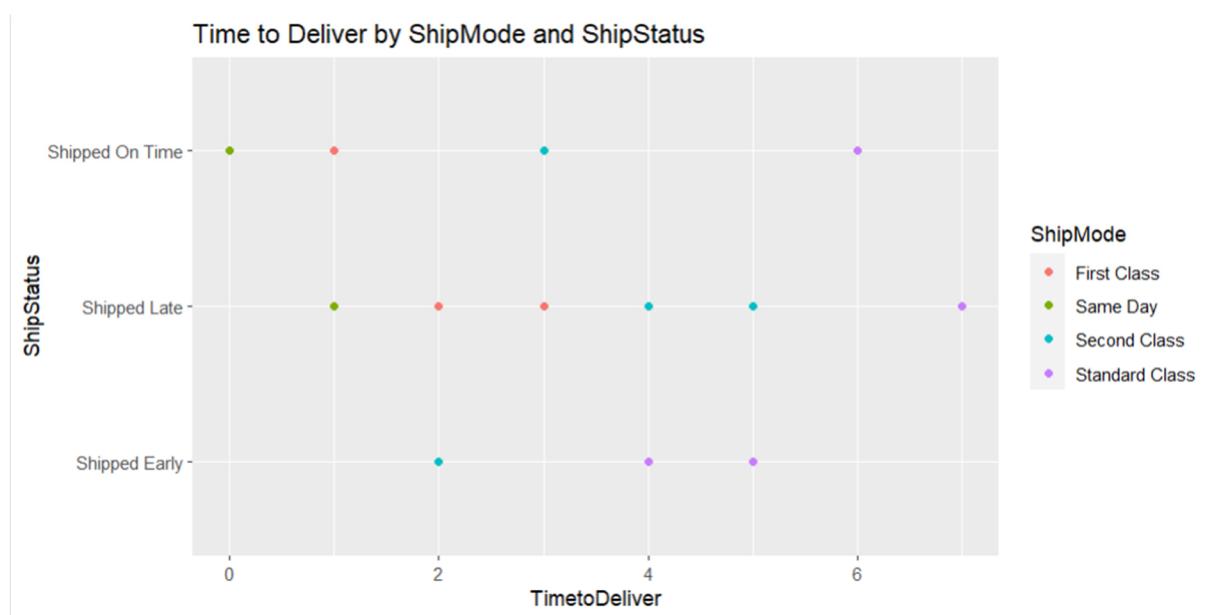
```
# Create a scatter plot of the timespan between OrderDate and ShipDate by ShipMode and ShipStatus
ggplot(sales_data, aes(x = TimetoDeliver, y = ShipStatus, color = ShipMode)) +
  geom_point() +
  labs(title = "Time to Deliver by ShipMode and ShipStatus")
```

This code will create a scatter plot with the timespan between OrderDate and ShipDate on the x-axis and the ShipStatus on the y-axis. The data points will be colored by the ShipMode.

I apologize for my previous mistake. I am still under development and I am always learning.

Don't know how to automatically pick scale for object of type <difftime>. Defaulting to continuous.

Output of the provided Code:



YouChat:

To include the time it took to deliver (timespan between OrderDate and ShipDate) in the diagram, you can follow these steps:

1. Calculate the timespan: Create a new column in your dataset that calculates the timespan between the OrderDate and ShipDate. You can use the `difftime()` function in R to calculate the difference between two dates.

```
df$DeliveryTime <- difftime(df$ShipDate, df$OrderDate, units = "days")
```

This will create a new column named "DeliveryTime" in your dataset, which represents the timespan between the OrderDate and ShipDate in days.

2. Modify the visualization: Update the visualization code to include the DeliveryTime variable. Depending on the type of visualization you choose, you can map the DeliveryTime to the size, color, or length of the bars or tiles in the diagram.

For example, if you are using a stacked bar chart:

```
ggplot(df, aes(x = ShipMode, fill = ShipStatus, y = DeliveryTime)) +  
  geom_bar(stat = "identity")
```

This will create a stacked bar chart where the height of each bar represents the DeliveryTime, and the fill color represents the ShipStatus.

Alternatively, if you are using a heatmap:

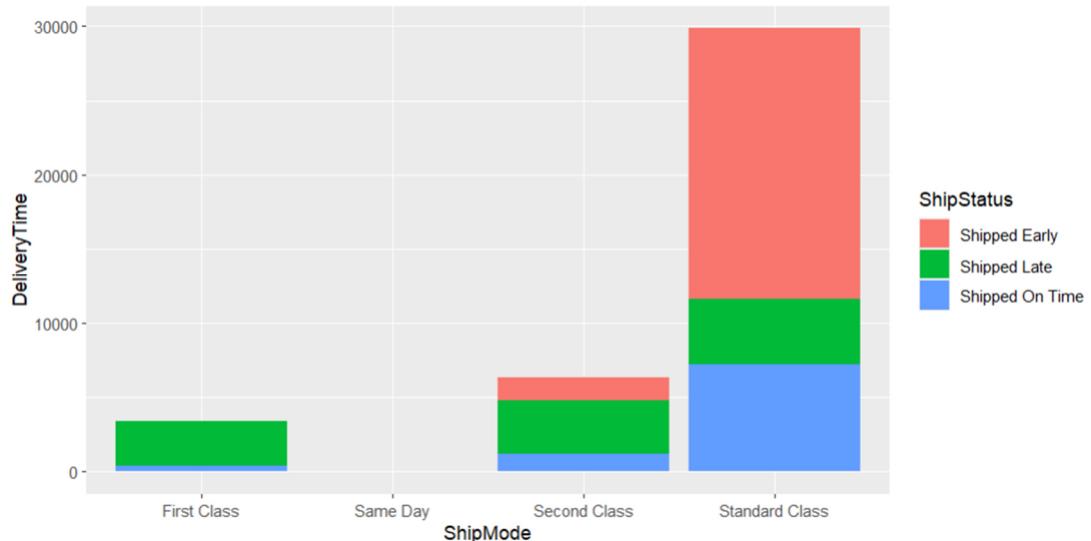
```
ggplot(df, aes(x = ShipMode, y = ShipStatus, fill = DeliveryTime)) +  
  geom_tile()
```

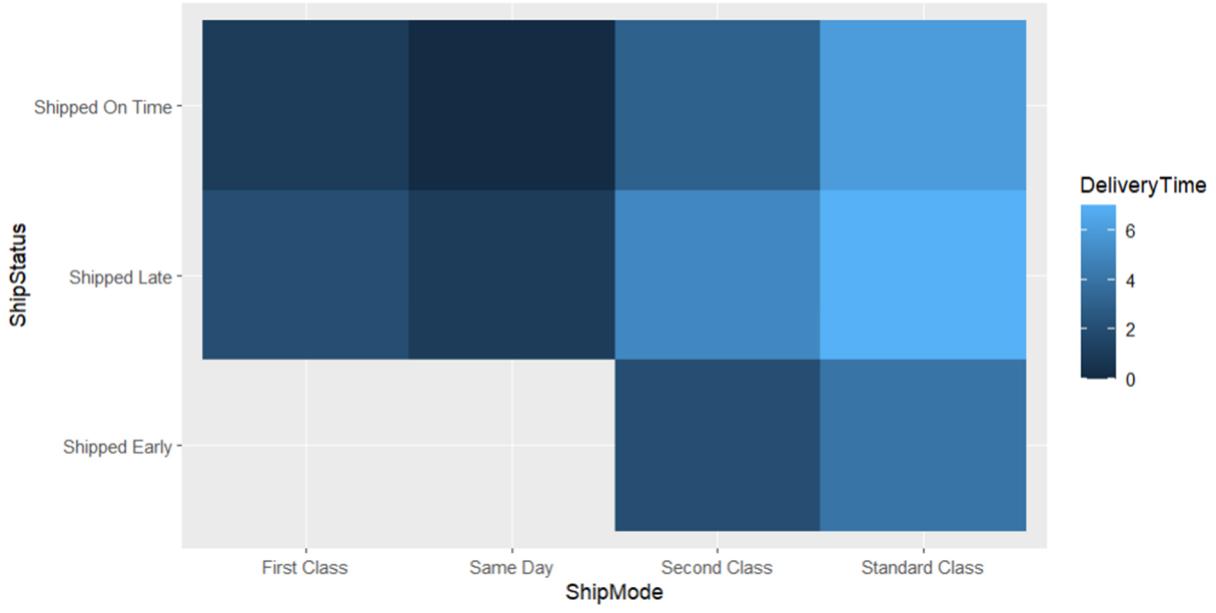
This will create a heatmap where the color of each tile represents the DeliveryTime. Remember to adjust the code based on your specific dataset and visualization preferences.

By following these steps, you can include the time it took to deliver (timespan between OrderDate and ShipDate) in the diagram, providing additional insights into the correlation between ShipMode, ShipStatus, and DeliveryTime.

Let me know if you have any further questions!

Output of the provided Code:





Query 4 (Okay but can you include both the number of early/late/on time deliveries and the time it took to deliver?):

ChatGPT-3.5:

Certainly! To create a comprehensive visualization that includes both the count of early/late/on-time deliveries and the delivery time (timespan between OrderDate and ShipDate), you can use a combination of grouped bar plots and box plots. Here's how you can modify the R code to create this visualization:

```
# Load required libraries if not already loaded
install.packages("tidyverse")
library(tidyverse)

# Read the data from CSV file
data <- read.csv("data.csv")

# Convert OrderDate and ShipDate to Date objects
data$OrderDate <- as.Date(data$OrderDate)
data$ShipDate <- as.Date(data$ShipDate)

# Calculate the time it took to deliver (in days)
data$DeliveryTime <- as.numeric(difftime(data$ShipDate, data$OrderDate, units = "days"))

# Calculate the delivery status (Early, Late, On Time)
```

```

data$DeliveryStatus <- ifelse(data$DeliveryTime < 0, "Late",
                             ifelse(data$DeliveryTime > 0, "Early", "On Time"))

# Create a grouped bar plot for ShipMode, ShipStatus, and Delivery Status (Early/Late/On Time)
ggplot(data, aes(x = ShipMode, fill = DeliveryStatus)) +
  geom_bar(position = "dodge") +
  ggtitle("Delivery Status and Delivery Time by ShipMode") +
  xlab("ShipMode") +
  ylab("Count") +
  scale_fill_manual(values = c("#66c2a5", "#fc8d62", "#8da0cb")) +
  theme_minimal() +
  facet_wrap(~ShipStatus, scales = "free_y")

# Create a box plot for Delivery Time by ShipMode and ShipStatus
ggplot(data, aes(x = ShipMode, y = DeliveryTime, fill = ShipStatus)) +
  geom_boxplot() +
  ggtitle("Delivery Time Distribution by ShipMode and ShipStatus") +
  xlab("ShipMode") +
  ylab("Delivery Time (Days)") +
  scale_fill_manual(values = c("#66c2a5", "#fc8d62", "#8da0cb")) +
  theme_minimal()

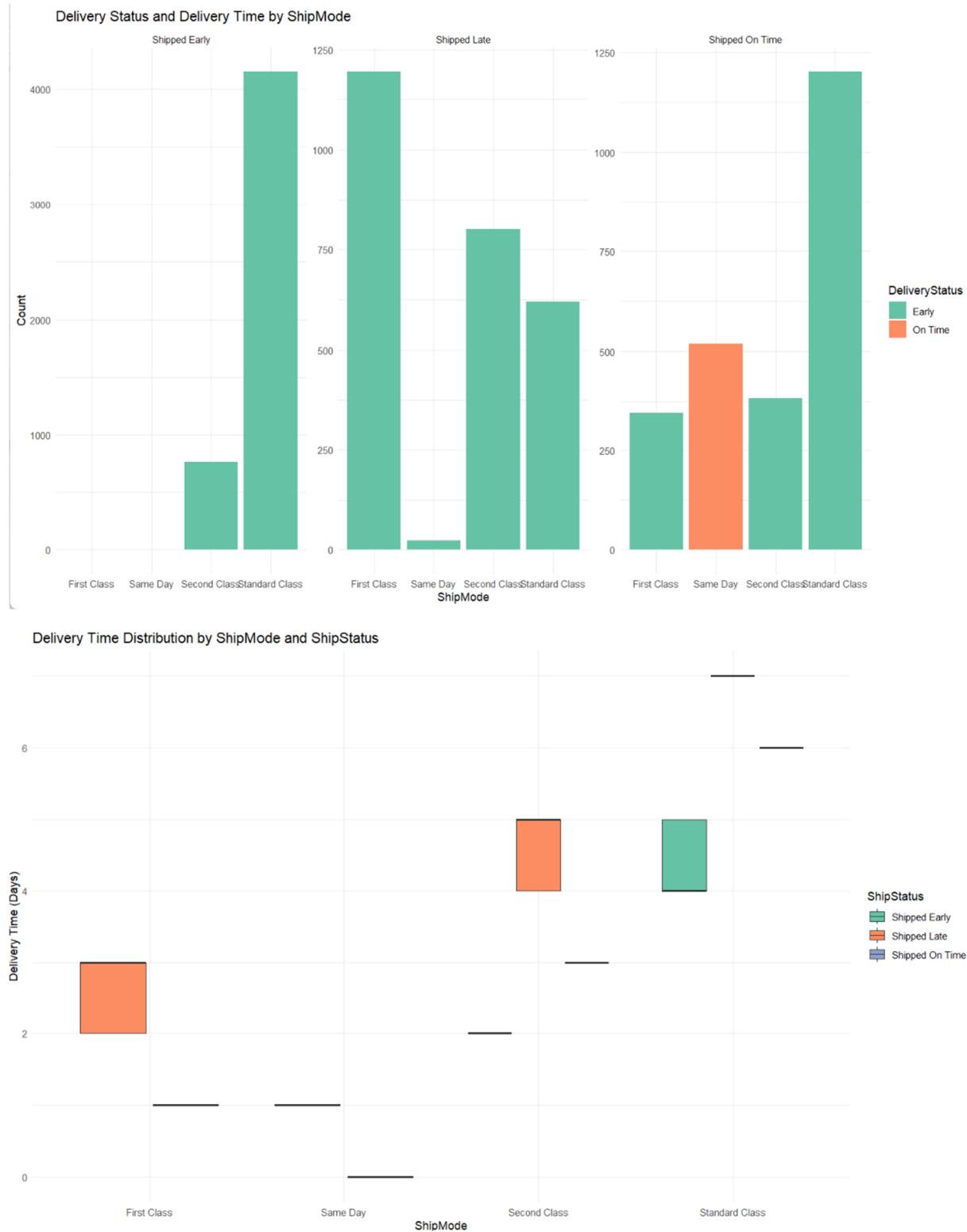
```

In this code:

- The first plot (`ggplot` with `geom_bar`) shows the count of early, late, and on-time deliveries for each `ShipMode` and `ShipStatus`. Different colors represent different delivery statuses (Early, Late, On Time), and bars are grouped by `ShipMode`. The `facet_wrap` function allows you to create separate plots for each `ShipStatus`.
- The second plot (`ggplot` with `geom_boxplot`) shows the distribution of delivery times (in days) for each combination of `ShipMode` and `ShipStatus`. The box plot provides a summary of the delivery time distribution, including median, quartiles, and potential outliers. Different colors represent different `ShipStatus` categories.

These visualizations will provide a comprehensive overview of both the delivery status and delivery time across different `ShipMode` values and `ShipStatus` categories.

Output of the provided Code:



- Does not show the intended information.

ChatGPT-4.0:

Certainly! Here's how you can visualize the correlation between `ShipMode`, `ShipStatus`, and `DeliveryTimeSpan` in R using the `ggplot2` package:

1. Calculate DeliveryTimeSpan:

Assuming `OrderDate` and `ShipDate` are in a date format, we'll subtract them to get the `DeliveryTimeSpan`.

If they are not in a date format, you'll first need to convert them:

```
data$OrderDate <- as.Date(data$OrderDate, format="%Y-%m-%d") # Adjust the format as per your data  
data$ShipDate <- as.Date(data$ShipDate, format="%Y-%m-%d")
```

```
data$DeliveryTimeSpan <- as.numeric(difftime(data$ShipDate, data$OrderDate, units = "days"))
```

2. Visualize the correlation using ggplot2:

Here we'll create a boxplot which can show us the distribution of `DeliveryTimeSpan` for each `ShipMode`, separated by the `ShipStatus`.

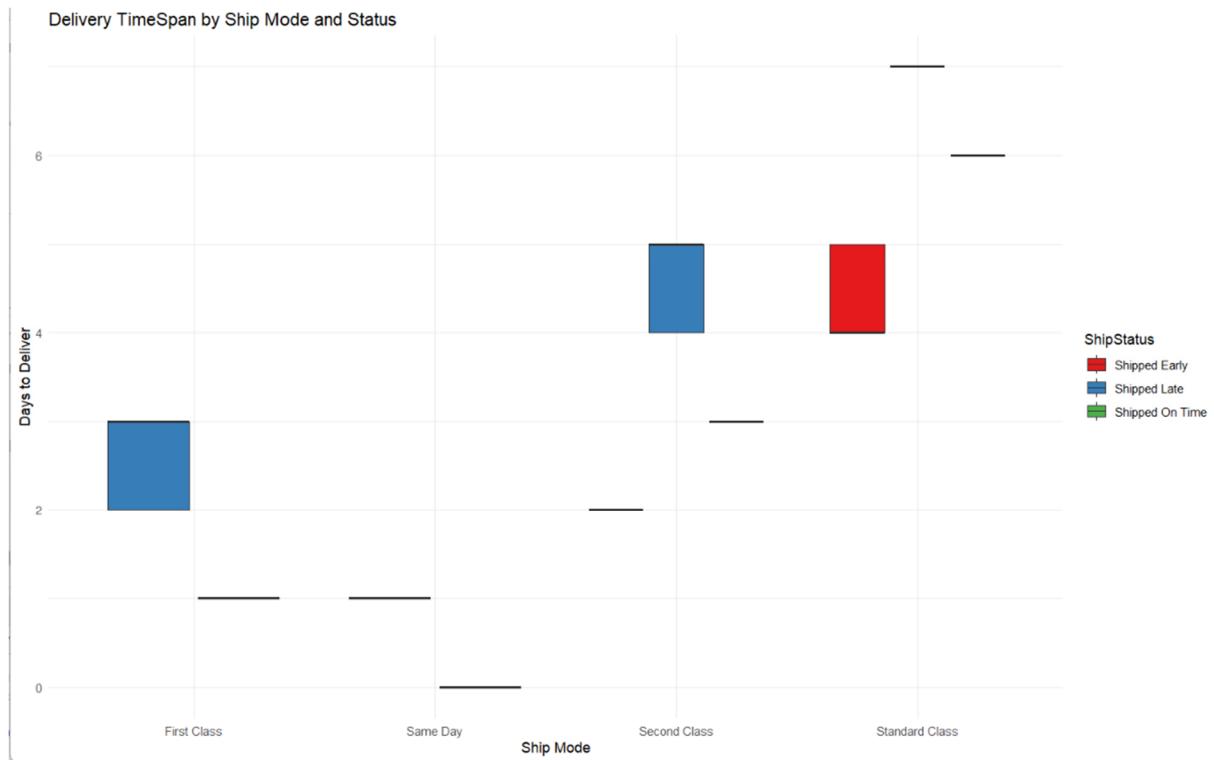
```
ggplot(data, aes(x=ShipMode, y=DeliveryTimeSpan, fill=ShipStatus)) +  
  geom_boxplot() +  
  labs(title="Delivery TimeSpan by Ship Mode and Status",  
       y="Days to Deliver",  
       x="Ship Mode") +  
  theme_minimal() +  
  scale_fill_brewer(palette="Set1")
```

Display the plot:

```
dev.off()
```

This will give you a boxplot where you can visually inspect the distribution of delivery timespans for different ship modes, and see how they differ based on whether they were on time, early, or late. The boxplots will give you a sense of the median, quartiles, and possible outliers in the data.

Output of the provided Code:



- Does not show the intended information.

Bing Chat:

- Was not able to produce working code for 3. use case. So I could not proceed with trying out the 4. use case

Google Bard:

```
# Load the data
sales_data <- read.csv("sales_data.csv")

# Calculate the timespan between OrderDate and ShipDate
sales_data$TimetoDeliver <- difftime(sales_data$ShipDate, sales_data$OrderDate, units = "days")

# Create a new dataframe with the counts of early/late/on time deliveries by ShipMode
ship_mode_delivery_status_df <- sales_data %>%
  group_by(ShipMode, ShipStatus) %>%
  summarize(count = n())

# Create a bar plot of the counts of early/late/on time deliveries by ShipMode
ggplot(ship_mode_delivery_status_df, aes(x = ShipMode, y = count, fill = ShipStatus)) +
```

```

geom_bar(stat = "sum", position = position_dodge()) +
  labs(title = "Number of Early/Late/On Time Deliveries by ShipMode")

# Add labels to the data points
geom_label(
  data = ship_mode_delivery_status_df,
  aes(x = ShipMode, y = count, label = count),
  hjust = ifelse(ship_mode_delivery_status_df$ShipStatus == "Shipped Early", -0.5, 0.5),
  vjust = 0
)

# Add a secondary y-axis to show the time it took to deliver
ggplot(ship_mode_delivery_status_df, aes(x = ShipMode, y = TimetoDeliver, fill = ShipStatus)) +
  geom_bar(stat = "mean", position = position_dodge()) +
  labs(title = "Time to Deliver by ShipMode")

# Add labels to the data points
geom_label(
  data = ship_mode_delivery_status_df,
  aes(x = ShipMode, y = TimetoDeliver, label = TimetoDeliver),
  hjust = ifelse(ship_mode_delivery_status_df$ShipStatus == "Shipped Early", -0.5, 0.5),
  vjust = 0
)

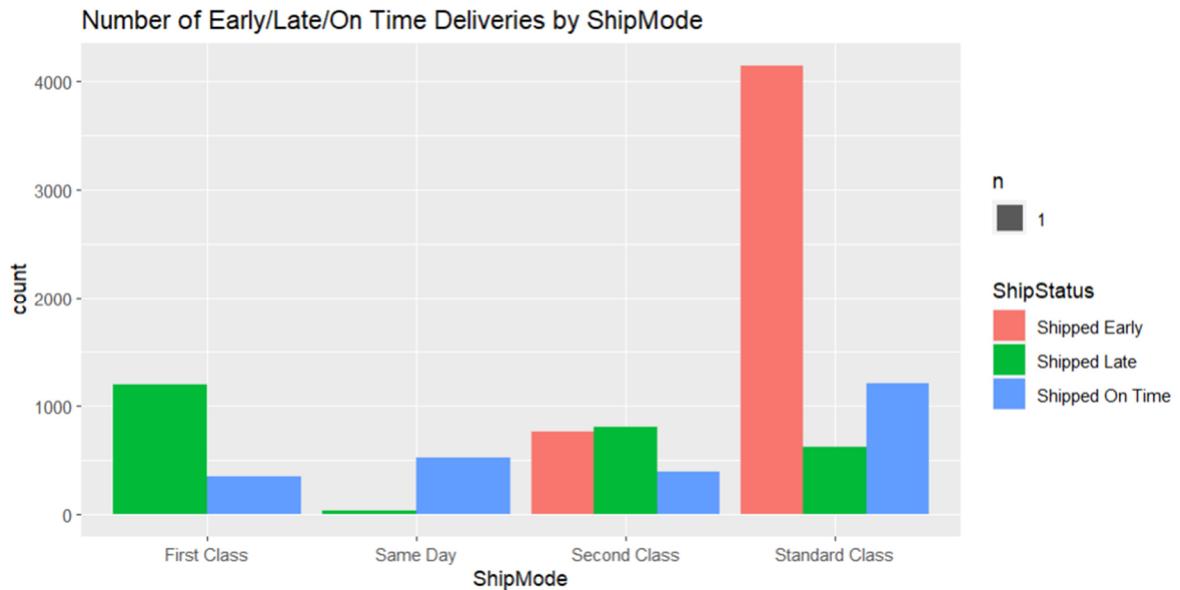
# Add a legend
scale_fill_manual(values = c("green", "yellow", "red")) +
  legend(x = "right", y = "bottom", title = "Ship Status")

```

This code will create a bar plot with the ShipMode on the x-axis and the number of early/late/on time deliveries on the primary y-axis. The data points will be colored by the ShipStatus. The time it took to deliver will be shown on the secondary y-axis.

You can also use other R packages to create more complex and interactive diagrams. For example, you could use the `plotly` package to create a stacked bar chart that allows users to hover over the different ShipModes to see the time it took to deliver for each ShipStatus.

Output of the provided Code:



Can't find stat called "mean"

- Is not able to fix this error

YouChat:

Certainly! To include both the number of early/late/on-time deliveries and the time it took to deliver in the diagram, you can follow these steps:

1. Calculate the delivery status: Create a new column in your dataset that categorizes the deliveries as early, late, or on time based on the desired criteria. For example, you can define a delivery as early if it was delivered before the expected delivery date, late if it was delivered after the expected delivery date, and on time if it was delivered on the expected delivery date.
2. Calculate the delivery time: Create a new column in your dataset that calculates the time it took to deliver each package. You can calculate this by subtracting the expected delivery date from the actual delivery date.
3. Modify the visualization: Update the visualization code to include both the number of early/late/on-time deliveries and the delivery time. Depending on the type of visualization you choose, you can map the number of deliveries to the size or color of the bars or tiles, and map the delivery time to the length or color of the bars or tiles in the diagram.

For example, if you are using a stacked bar chart:

```
ggplot(df, aes(x = ShipMode, fill = DeliveryStatus)) +  
  geom_bar() +  
  geom_text(aes(label = DeliveryTime), position = position_stack(vjust = 0.5))
```

This will create a stacked bar chart where the height of each bar represents the number of deliveries, the fill color represents the delivery status (early/late/on time), and the text labels represent the delivery time.

Alternatively, if you are using a heatmap:

```
ggplot(df, aes(x = ShipMode, y = DeliveryStatus, fill = DeliveryTime)) +  
  geom_tile() +  
  geom_text(aes(label = DeliveryTime), color = "white")
```

This will create a heatmap where the color of each tile represents the delivery time, and the text labels represent the delivery time.

Remember to adjust the code based on your specific dataset, column names, and criteria for categorizing deliveries as early, late, or on time.

By following these steps, you can include both the number of early/late/on-time deliveries and the time it took to deliver in the diagram, providing a comprehensive view of the correlation between ShipMode, DeliveryStatus, and DeliveryTime.

Let me know if you have any further questions!

- Code for first diagram does not work. And YouChat is not able to fix

Output of the provided Code:

