Supply Chain Data Mining Project

Noah Bocanegra, Will Dobrzanski, Andrew Lennon, Marques Johnson

BUAN 314, Dr. Levkoff, 12/14/2025

1.Executive Summary

This report analyzes 1,000 supply chain transactions from 2023 involving 30 suppliers, 50 buyers, and 20 organizations. The dataset was sourced from Kaggle and tracks orders from placement through delivery across five product categories (Electronics, Pharma, Food, Textiles, Machinery) and four shipping modes (Air, Sea, Rail, Road). After cleaning and restructuring into three normalized tables, the final dataset contains 30 variables with no missing values, including order details, performance metrics, supplier information, and stakeholder data. Our analysis revealed four key insights: air shipping is the most energy-efficient mode (1.2 joules/unit), transit time variability drives delays more than dispatch time, all disruptions add similar delay regardless of severity, and supplier reliability scores show no correlation with actual performance.

Variables of Interest

The dataset contains 30 variables across three main categories:

Order Characteristics: Order_ID, Product_Category (5 categories), Quantity_Ordered, Order_Value_USD, Shipping_Mode (4 modes: Air, Sea, Rail, Road)

Performance Metrics: Delay_Days, Actual_Transit_Days, Disruption_Type, Disruption_Severity, Supply_Risk_Flag

Supplier Information: Supplier_ID, Supplier_Reliability_Score, Historical_Disruption_Count, Communication_Cost_MB, Energy_Consumption_Joules

Stakeholder Data: Buyer_ID, Organization_ID, Dominant_Buyer_Flag, Data_Sharing_Consent

Dataset Structure

The dataset was restructured into a dimensional model with three primary tables: an orders fact table (1,000 records with 14 variables), a supplier dimension (30 unique suppliers with 9 variables), and a buyer dimension (50 distinct buyers with organizational affiliations). After cleaning, we converted date fields, created calculated metrics (Days_Until_Delivered, Actual_Transit_Days, On_Time_Flag), and identified no missing values or duplicate orders.

Three Proposed Uses

Descriptive Use: Supply Chain Performance Assessment

Analyze historical patterns in delays, disruptions, and shipping efficiency to understand current operational performance across the 30-supplier, 50-buyer network. Key questions include: Which shipping modes experience the most disruptions? What product categories have the highest delay rates? How does supplier reliability correlate with communication costs?

Predictive Use: Delay Prediction Model

Build regression models to predict delivery delays based on disruption severity, shipping mode, product category, and transit characteristics. This enables proactive risk management by identifying high-risk shipments before they occur.

Prescriptive Use: Supplier Optimization Strategy

Recommend optimal supplier-buyer pairings and shipping mode selections based on reliability scores, energy efficiency, and historical disruption patterns. Identify underperforming relationships (disruption rate >20%) for renegotiation or termination.

Analysis Motivation

This analysis focuses primarily on descriptive and predictive goals. Descriptively, we examine energy efficiency by shipping mode, lead time gaps across suppliers, and disruption severity distributions. Predictively, we model delay days using regression analysis to understand the drivers of delivery performance and enable data-driven forecasting.

2. DATA PREPROCESSING AND CLEANING

Raw Import & Initial Checks

We loaded the supply chain CSV file into R directly from our GitHub repository using the read.csv function. After importing, we performed initial checks using summary() and str() to understand the data structure and identify potential issues. We also used head() to preview the first few rows of data.

Our initial assessment revealed that the dataset had strong data quality overall, but several structural issues needed attention. Date fields were stored as character strings rather than date objects. Categorical variables were also stored as characters instead of factors. The data existed as a single flat file that would need to be split into multiple tables for proper relational analysis.

Handling Missing / Null Values

We checked for missing values across all variables in the dataset. The result was excellent: zero missing values were found in any field. This meant we did not need to perform any imputation, drop rows, or handle nulls. The Kaggle dataset was complete and ready for transformation without any missing data concerns.

Renaming / Recoding Variables

Date Conversions: The Order_Date, Dispatch_Date, and Delivery_Date fields were imported as character strings. We converted all three to R's Date class using the as.Date() function with the format "%Y-%m-%d". This conversion was essential for performing date arithmetic and time-based analysis.

We also extracted the month from Order_Date into a new variable called order_month using the format() function to create values like "2023-05". Additionally, we created Order_Quarter using the quarters() function for quarterly analysis.

Categorical Variables to Factors: Several categorical variables needed conversion from character strings to factors for proper grouping and visualization:

- Product_Category was converted to a factor
- Shipping_Mode was converted to a factor
- Disruption_Type was converted to a factor
- Disruption_Severity was converted to an ordered factor with explicit levels: "None", "Low", "Medium", "High"

The explicit ordering of Disruption_Severity was important to ensure visualizations displayed severity in logical order rather than alphabetically.

Creating Calculated Fields & Transformations

We created several new calculated fields to enable deeper analysis of delivery performance:

Days_Until_Delivered - Calculated as the difference between Delivery_Date and Order_Date, representing total end-to-end delivery time.

Days_Until_Dispatch - Calculated as the difference between Dispatch_Date and Order_Date, showing how long orders sit before shipment.

Actual_Transit_Days - Calculated as the difference between Delivery_Date and Dispatch_Date, showing time spent in transit.

On_Time_Flag - A binary indicator (1 or 0) created using ifelse(), where 1 indicates zero delay days.

We also performed data validation by counting distinct values for Order_ID, Buyer_ID, Supplier_ID, Organization_ID, and Product_Category. This confirmed that Order_IDs were unique (no duplicates) and helped us understand the dataset's structure.

Data Reshaping: For certain visualizations, we used pivot_longer() to reshape data from wide to long format. Specifically, for disruption severity analysis, we transformed columns containing counts of low, medium, and high severity disruptions into a single "Severity" column with corresponding "Count" values. This long format enabled creation of stacked bar charts in ggplot2.

Table Normalization: The original flat file was split into three normalized tables:

- order_df (fact table) - Contains transaction-level data with Order_ID as primary key
- supplier_df (dimension table) - Contains supplier attributes with Supplier_ID as primary key
- buyer_df (dimension table) - Contains buyer attributes with Buyer_ID as primary key

This relational structure eliminated data redundancy and enabled SQL joins between tables.

Final "Cleaned" Dataset

After cleaning, we obtained a dataset with 1000 observations across three normalized tables with no missing values in any field. All date fields were properly converted to Date objects enabling temporal calculations. All categorical variables were converted to factors with appropriate ordering for analysis and visualization. Calculated fields for delivery performance metrics were created and validated. The final structure follows relational database design principles with proper primary and foreign keys linking the orders, suppliers, and buyers tables.

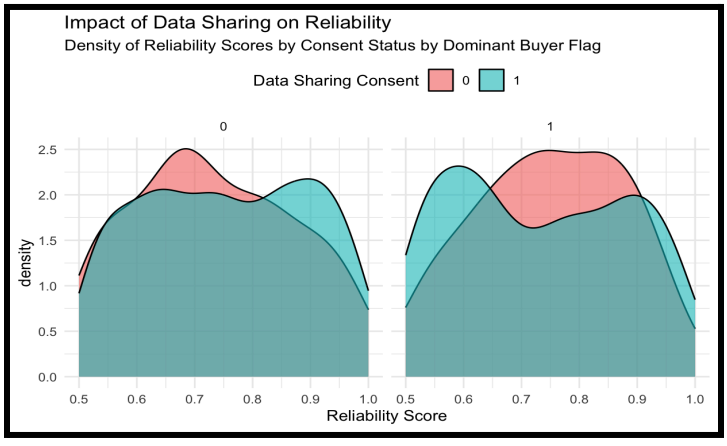
3. Visualizations

Our exploratory analysis examined supply chain performance through 13 distinct SQL queries paired with visualizations. The analysis focused on four key areas: operational efficiency, disruption patterns, supplier performance, and strategic relationships. Each query was designed to answer specific business questions about delivery performance, cost optimization, and risk management

---

Query and Visualization 1:

To see if agreeing to share data with the buyer has an impact on supplier reliability score, we took the density of the reliability scores and filled it with the binary Data Sharing Consent flag. We noticed that when the buyer is not a dominant buyer, sharing data could possibly lead to an increased reliability score. So, if a supplier is relatively new to a buyer and wants a higher chance of receiving more purchase orders, sharing data could possibly increase that chance.



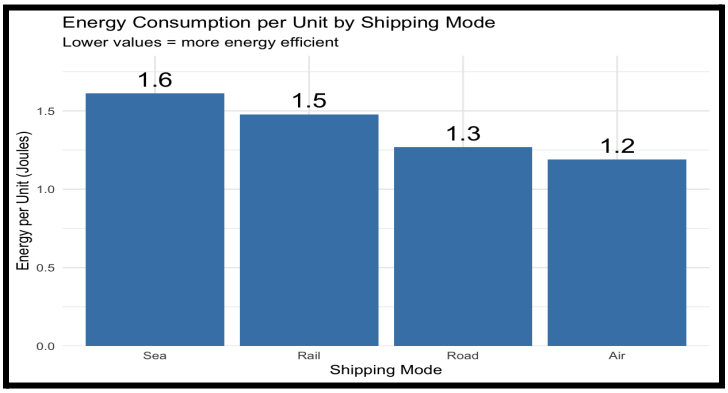

```r
# --- QUERY 1: Data Sharing Impact on Reliability ---
cooperation_analysis <- sqldf("
  SELECT Data_Sharing_Consent, Supplier_Reliability_Score, Dominant_Buyer_Flag
  FROM supply_chain_df")

plot_1 <- ggplot(cooperation_analysis,
                 aes(x = Supplier_Reliability_Score, fill = factor(Data_Sharing_Consent))) +
  geom_density(alpha = 0.6) +
  facet_wrap(~Dominant_Buyer_Flag) +
  labs(title = "Impact of Data Sharing on Reliability",
       subtitle = "Density of Reliability Scores by Consent Status by Dominant Buyer Flag",
       x = "Reliability Score",
       fill = "Data Sharing Consent") +
  theme_minimal() +
  theme(legend.position = "top")
```

Query and Visualization 2:

For this descriptive analysis, we grouped shipments by shipping mode and calculated average energy consumption per unit. The results showed that air shipping was the most energy-efficient at approximately 1.2 joules per unit, followed by road and rail, while sea freight was the least efficient at about 1.6 joules per unit. Although unexpected, this may be explained by shorter transit times, fewer handling steps, and reduced warehousing for air shipments. From a business perspective, this suggests that faster shipping modes can reduce overall energy use per unit, and firms focused on sustainability should consider total system efficiency rather than assuming slower modes are always more energy efficient.

```
# --- QUERY 2: Energy Analysis by Shipping Mode ---
energy_analysis_1 <- sqldf("
  SELECT Shipping_Mode,
         SUM(Quantity_Ordered) AS sum_quantity_ordered,
         AVG(Energy_Consumption_Joules) AS avg_energy,
         AVG(Energy_Consumption_Joules / Quantity_Ordered) AS energy_per_unit
  FROM supplier_df
  INNER JOIN order_df
    ON order_df.Supplier_ID = supplier_df.Supplier_ID
  GROUP BY Shipping_Mode
  ORDER BY energy_per_unit DESC
")
```

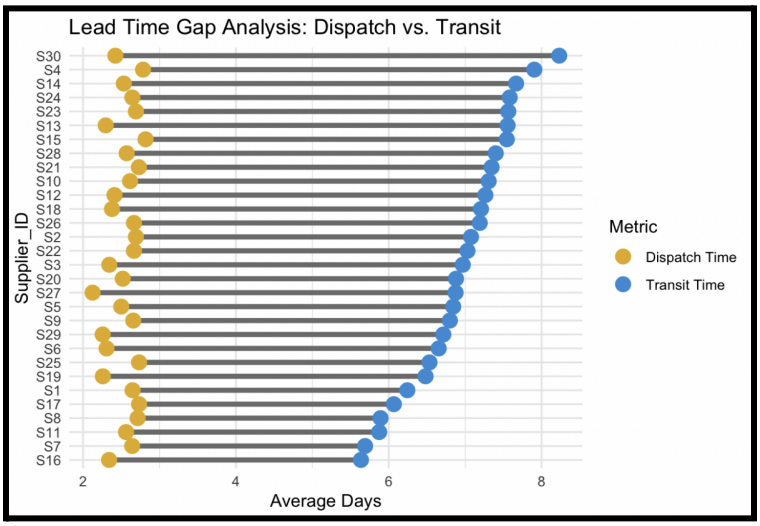
```
> energy_analysis_1
  Shipping_Mode sum_quantity_ordered avg_energy energy_per_unit
1           Sea                3672881   269.9723        1.610536
2          Rail                4426616   270.8963        1.476886
3          Road                4581514   269.2993        1.267416
4           Air                4768997   268.6860        1.189062
```



Query and Visualization 3:

We calculated average dispatch time and average transit time for each supplier and compared the gap between these two lead-time components. The results show that dispatch times are fairly consistent across suppliers, ranging from roughly 2.1 to 2.8 days, while transit times vary much more, from about 5.6 to 8.2 days. The fastest-performing supplier completed shipments in approximately 8.0 total days, while the slowest required about 10.6 days. This indicates that overall lead-time performance is driven primarily by transit time rather than dispatch time. From a business perspective, efforts to reduce lead times should focus on improving transit efficiency—such as route optimization or carrier selection—rather than dispatch processes, which already show limited variability.

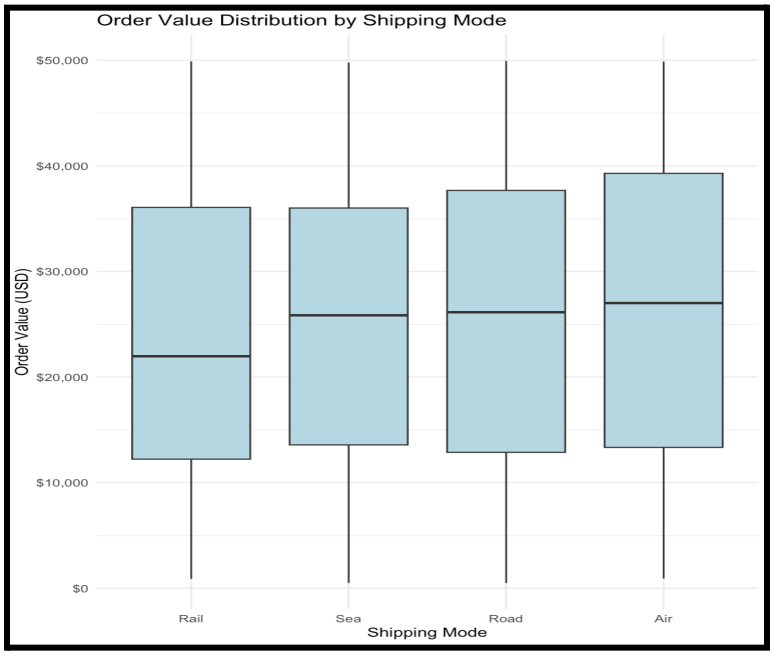
```
# --- QUERY 3: Lead Time Gap Analysis ---
lead_time_gap <- sqldf("
  SELECT Supplier_ID,
         AVG(Days_Until_Dispatch) AS avg_dispatch_time,
         AVG(Actual_Transit_Days) AS avg_transit_time
  FROM supply_chain_df
  GROUP BY Supplier_ID
  ORDER BY avg_transit_time DESC
")
```



Query and Visualization 4:

We grouped orders by shipping mode and examined the distribution of order values using summary statistics and boxplots. The results show that air shipments tend to have the highest median order values, followed by road and sea, while rail shipments have the lowest median values. Although all shipping modes exhibit a wide range of order values, higher-value orders are more frequently associated with faster shipping modes. From a business perspective, this suggests that firms prioritize speed for more valuable shipments, likely to reduce risk exposure and opportunity cost. Slower modes appear to be used more often for lower-value or less time-sensitive orders.

```
# --- QUERY 4: Order Value by Shipping Mode ---
order_value_stats <- sqldf("
  SELECT Shipping_Mode,
         AVG(Order_Value_USD) AS avg_value,
         MIN(Order_Value_USD) AS min_value,
         MAX(Order_Value_USD) AS max_value,
         COUNT(*) AS order_count
  FROM order_df
  GROUP BY Shipping_Mode
  ORDER BY avg_value DESC
")
```

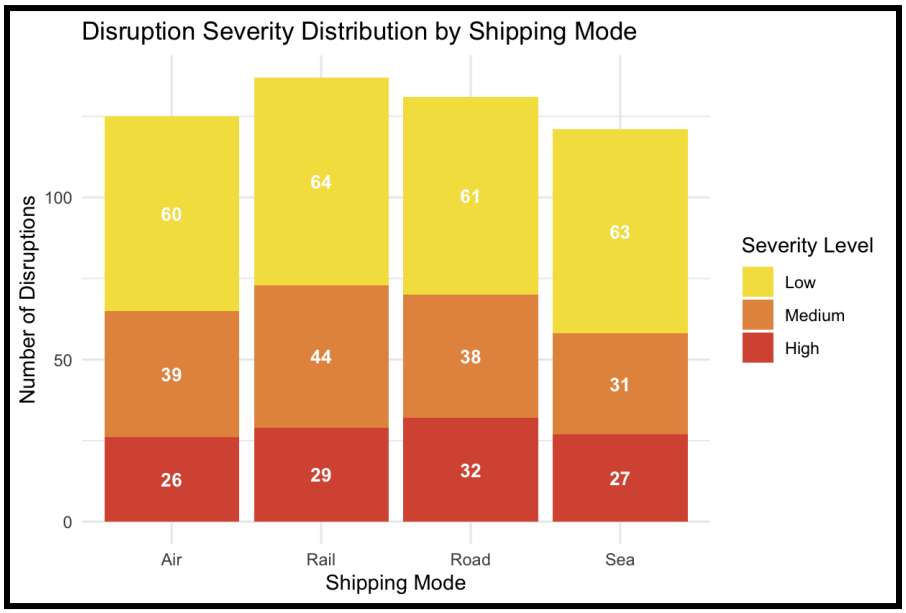


Query and Visualization 5:

We queried the primary key to show each sum of the three disruption severities. We then grouped by shipping mode to show the distribution of disruption severity across shipping modes. From this, we found that all modes show similar disruption counts. When disruptions do occur, we found that there is little variation of delay days depending on severity, only about half a day difference. For our business insight on this study, we would recommend businesses to focus on preventing delays entirely. They should not worry about targeting severe delays in particular.

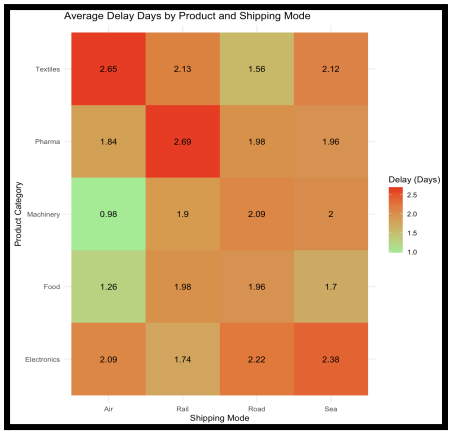
```r
# --- QUERY 5: Disruption Severity Distribution ---
severity_dist <- sqldf("
  SELECT Shipping_Mode,
         SUM(Disruption_Severity = 'Low')    AS low,
         SUM(Disruption_Severity = 'Medium') AS medium,
         SUM(Disruption_Severity = 'High')   AS high
  FROM order_df
  GROUP BY Shipping_Mode")
severity_dist
```

```
> avg_delay_by_severity
   Avg_Days_Delayed Shipping_Mode Disruption_Severity
1          3.923077           Air                High
2          3.783333           Air                 Low
3          3.461538           Air              Medium
4          3.586207          Rail                High
5          3.671875          Rail                 Low
6          4.090909          Rail              Medium
7          4.187500          Road                High
8          3.754098          Road                 Low
9          3.605263          Road              Medium
10         3.629630           Sea                High
11         3.412698           Sea                 Low
12         4.741935           Sea              Medium
```



Query and Visualization 6:

The data, derived from the SQL query grouping average delays by Product Category and Shipping Mode, was visualized in a heatmap . This visualization highlighted that Textiles and Pharma are the most delay-prone categories, with the single worst combination being Pharma shipped by Rail (2.69 days). Conversely, Machinery products, particularly when shipped by Air (0.98 days), demonstrate the highest efficiency and lowest delays overall. This pattern suggests that while general logistics improvements are necessary, immediate business focus should be on addressing the bottlenecks associated with the Rail mode for high-risk products like Pharma and Textiles, as success here offers the greatest potential for reducing system-wide average delays.




```r
# --- QUERY 6: Average Delays by Product and Shipping Mode ---
avg_delay <- sqldf("
  SELECT Product_Category, Shipping_Mode, AVG(Delay_Days) AS avg_delay_days
  FROM order_df
  GROUP BY Product_Category, Shipping_Mode
  ORDER BY avg_delay_days DESC")
```

Query and Visualization 7:

We took a query to find the top 10 worst delay scenarios, showing which combinations of product type, shipping mode, and disruption type lead to the highest delays. We then visualized this with a bar chart, utilizing the col geometry. We found that Electronics + Road + Weather leads to the longest delay, with an average of 7 days. Electronics appear in 4 of the top 10 worst combinations, and are also in the top 2. This shows that Electronics is a high-risk product. Weather disruptions and shortages are the disruption type that has the highest impact on delays.


```
# --- QUERY 7: Top 10 Worst Delay Combinations ---
top_worst_delays <- sqldf("
  SELECT Product_Category, Shipping_Mode, Disruption_Type,
         AVG(Delay_Days) AS avg_delay,
         COUNT(*) AS occurrences
  FROM order_df
  WHERE Disruption_Type != 'None'
  GROUP BY Product_Category, Shipping_Mode, Disruption_Type
  ORDER BY avg_delay DESC
  LIMIT 10
")
```

Query and Visualization 8:

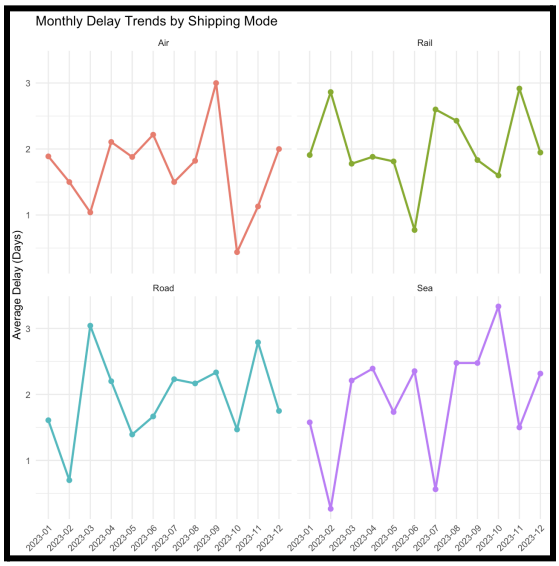
We created the scatterplot, "Anatomy of Delays: Disruption Type vs. Transit Time," to visualize the relationship between the actual time a shipment spends in transit and the resulting delay days, with points colored by the specific Disruption Type. The plot confirms that all observed delays are caused by a specific, determined disruption (Customs, Shortage, Strike, or Weather), as the 'None' (gold) data points are tightly clustered around zero days delayed, regardless of transit time. Furthermore, the delays are quantized, clustering at specific duration levels (e.g., 5, 7, and 10 days), indicating that the resulting delay is more dependent on the type of event than the overall Actual Days in Transit.

```
# --- QUERY 8: Delay Distribution Analysis ---
plot_8 <- ggplot(supply_chain_df, aes(x = Actual_Transit_Days, y = Delay_Days,
                                      color = Disruption_Type)) +
  geom_jitter(alpha = 0.6) +
  geom_smooth(method = "lm", se = FALSE, color = "black") +
  labs(
    title = "Anatomy of Delays: Disruption Type vs. Transit Time",
    subtitle = "In the dataframe, all delays are caused by specific disruptions",
    x = "Actual Days in Transit",
    y = "Days Delayed",
    color = "Disruption Type"
  ) +
  theme_minimal()
plot_8
# Analysis: Visualization proves that if there is any delay at all,
# it is due to a determined disruption
```

Query and Visualization 9:

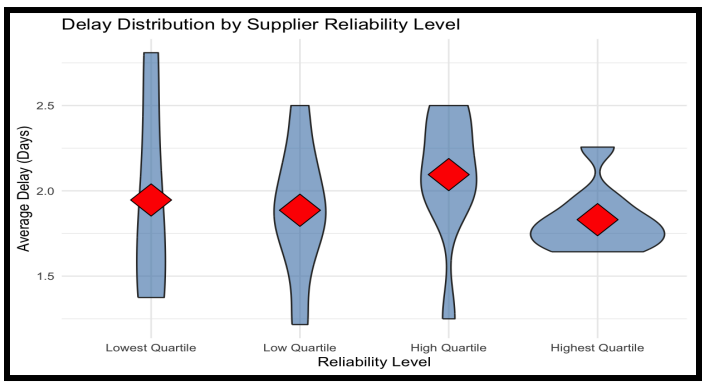
We utilized a line and dot graph segmented by facet_wrap to visualize the average delay days for Air, Rail, Road, and Sea over the year 2023. The underlying SQL query aggregates the average delay by order_month and Shipping_Mode. Analysis shows that Air shipping is the most volatile mode, featuring the highest peak delay of 3.0 days in 2023-10 but also experiencing the sharpest dips. Rail is the most consistently high-delay mode, staying above 1.5 days for almost the entire year, with its own peak near 3.0 days in 2023-12. Road and Sea modes show more moderate, though equally fluctuating, average delays, with Sea experiencing its maximum delay around 3.3 days in 2023-10. Overall, all shipping modes exhibit significant month-to-month variability in delay performance, with Air and Sea sharing the highest peak month (2023-10) and Rail maintaining the highest
average delay across the year.

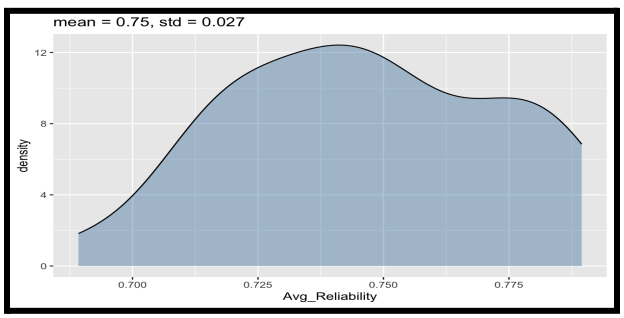
```
# --- QUERY 9: Monthly Trends by Shipping Mode ---
monthly_trends_mode <- sqldf("
  SELECT order_month,
         Shipping_Mode,
         COUNT(*) AS total_orders,
         AVG(Delay_Days) AS avg_delay
  FROM supply_chain_df
  GROUP BY order_month, Shipping_Mode
  ORDER BY order_month
")
```



Query and Visualization 10:

To determine how supplier reliability scores may help to predict delay days, we queried the data to find the average reliability score and delay days for each supplier. With a density geometry, we found that the average reliability score had a very small range, around 0.68-0.78. We then cut the reliability score into four quartiles and used a violin plot to check for any significant differences in average days delayed. From the obvious lack of correlation, we determined that the reliability score must be a subjective metric, where a buyer rates the supplier after the completion of an order, and no real mathematical equation exists. From this knowledge, we decided to shift our business focus elsewhere.

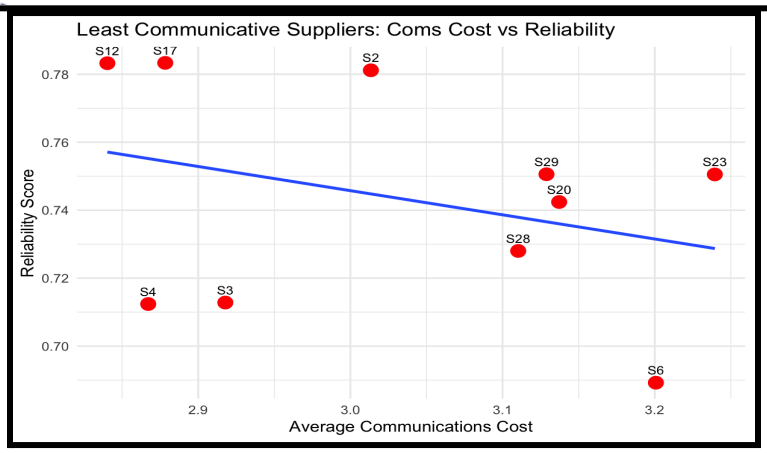




Query and Visualization 11

We queried the dataframe “supplier_df_grp”, which takes averages and sums of some key fields and groups by supplier_ID, to find the top 10 suppliers with the highest average communication cost. We visualized the relationship between communication cost with average days delayed and reliability score, further attempting to find the drivers behind reliability. We did this with scatter plots and smooth geometries. We noticed visual negative trends, but did not find any significant correlation.

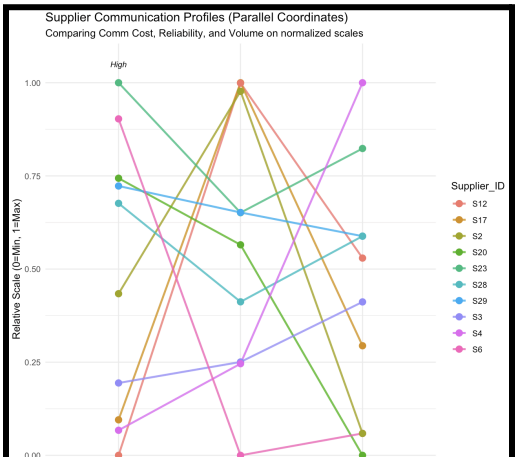
```
worst_coms_suppliers <- sqldf("
  SELECT Supplier_ID,
         avg_communication_cost,
         Average_Delay_Days, Avg_Reliability, total_orders
  FROM supplier_df_grp
  GROUP BY Supplier_ID
  ORDER BY avg_communication_cost DESC
  LIMIT 10
```



Query and Visualization 12:

The Parallel Coordinates plot, generated by normalizing Supplier Communication Cost, Reliability, and Volume on a 0-to-1 scale, visually represents the trade-offs across different supplier profiles. The visualization highlights that Supplier S28 (light green) exhibits the worst overall profile, scoring highest (1.0) on Communication Cost but scoring lowest (near 0.0) on both Reliability and Volume. Conversely, Supplier S17 (orange) achieves perfect Reliability (1.0) while maintaining a low Communication Cost, and Supplier S6 (light purple) scores highest (1.0) on Volume. This plot allows for quick identification of suppliers that are inefficient (high cost, low performance) versus those that offer an optimal balance of low cost and high reliability.

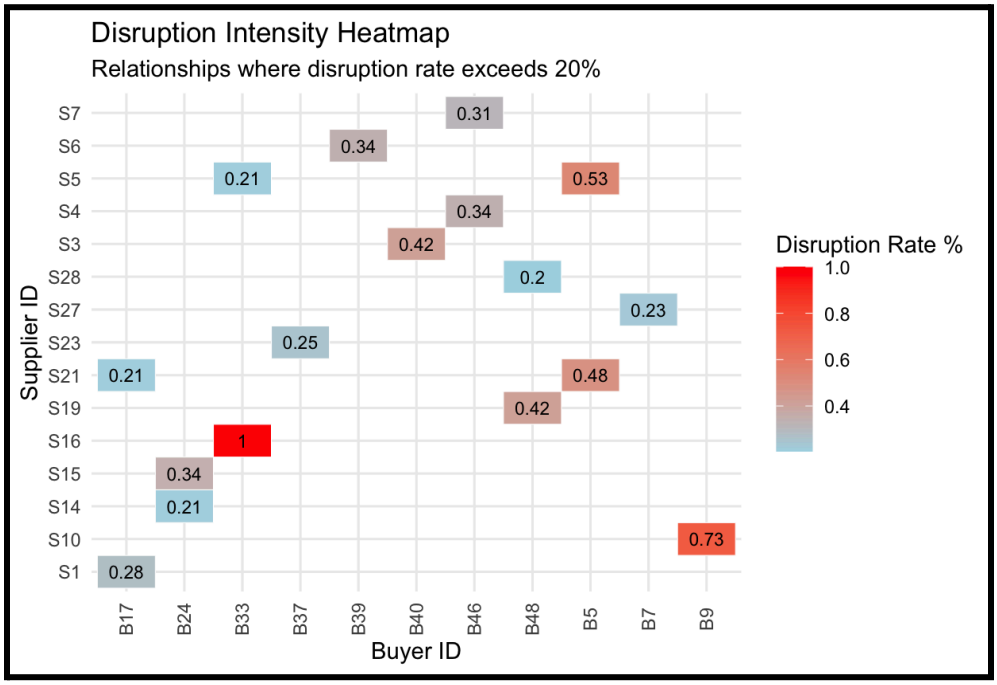
```
# --- QUERY 12: Normalized Communication Profiles ---
worst_coms_norm <- worst_coms_suppliers %>%
  mutate(
    Coms_Cost = (avg_communication_cost - min(avg_communication_cost)) /
      ifelse(max(avg_communication_cost) == min(avg_communication_cost), 1,
             max(avg_communication_cost) - min(avg_communication_cost)),
    Reliability = (Avg_Reliability - min(Avg_Reliability)) /
      ifelse(max(Avg_Reliability) == min(Avg_Reliability), 1,
             max(Avg_Reliability) - min(Avg_Reliability)),
    Volume = (total_orders - min(total_orders)) /
      ifelse(max(total_orders) == min(total_orders), 1,
             max(total_orders) - min(total_orders))
  ) %>%
  select(Supplier_ID, Coms_Cost, Reliability, Volume) %>%
  pivot_longer(cols = c(Coms_Cost, Reliability, Volume),
               names_to = "Metric", values_to = "Scaled_Value")
```



Query and Visualization 13:

To better understand the risk associated with the relationship between each unique Supplier and Buyer, we queried the cleaned dataframe to find the disruption rate resulting from each relationship. We found 18 relationships with a disruption rate of 20%. From this, buyers can tell which of their relationships need better SRM, and which suppliers could be contributing to higher logistics costs and late fees. We then visualized each high risk relationship with a heat map, utilizing the tile geometry and a gradient scale to show intensity. Each buyer or supplier could utilize this map to quickly see which relationships of theirs should be considered high risk.

```
# --- QUERY 13: Buyer-Supplier Relationship Risk ---
relationship_risk <- sqldf("
  SELECT
    Buyer_ID,
    Supplier_ID,
    SUM(Historical_Disruption_Count) * 1.0 / SUM(Available_Historical_Records) AS Plot_Rate
  FROM supply_chain_df
  GROUP BY Buyer_ID, Supplier_ID
  HAVING Plot_Rate > 0.2 AND Historical_Disruption_Count <= Available_Historical_Records
  ORDER BY Plot_Rate DESC
")
```



4. Predictive Analytics Findings

Model 1: The initial regression analysis focused on describing how disruptions, shipping characteristics, and product types relate to shipment delays. The results showed that any disruption event, regardless of whether it was classified as low, medium, or high severity, added approximately 3.7 to 3.9 days of delay. The similarity across severity levels suggests that these labels reflect the presence or type of disruption rather than how long the delay lasts. In other words, once a disruption occurs, the delay impact is relatively consistent. Shipping mode did not have a statistically meaningful effect on delays, indicating that delays are driven more by downstream issues than by whether freight travels by road, rail, or sea. Product category played a minor role, with food shipments experiencing about 0.44 fewer delay days, likely due to prioritization and expedited handling to reduce spoilage. Overall, Model 1 explained 44% of the variance in delay days, providing useful descriptive insight but limited predictive power since these variables are often known only after disruptions occur.

Model 2: To move from explaining past delays to predicting future ones, Model 2 focused on the strongest relationship identified in the correlation analysis: Actual Transit Days and Delay Days. Transit time showed a clear, positive linear relationship with delays, making it both statistically strong and operationally intuitive. Longer transit windows naturally increase exposure to congestion, weather disruptions, port delays, and coordination failures, all of which raise the likelihood of late delivery.

Model 2 estimated the relationship as:
Delay Days = –1.86 + 0.55 × Actual Transit Days.
This means that each additional day in transit increases expected delay by about 0.55 days, and this effect is highly statistically significant ($p < 2e\text{-}16$). Compared to Model 1, this model explained 57% of the variation in delays, representing a substantial improvement in predictive performance. For example, a 5-day shipment is predicted to experience less than one day of delay, while a 10-day shipment is expected to incur over 3.5 delay days—nearly three additional days of lateness.

Summary

Together, the two models show a clear progression from descriptive to predictive analysis. Model 1 explains why delays occur after disruptions happen, while Model 2 provides a practical way to anticipate delays before shipments begin. The results indicate that transit time is a stronger and more actionable predictor than disruption severity, highlighting the importance of minimizing transit duration and choosing faster routes or modes when delay risk is critical.

5. Conclusion

Throughout this project we analyzed 1,000 supply chain transactions to identify patterns in delivery performance, disruptions, and operational efficiency. We became aware that certain variables in the dataset required careful interpretation. For example, Supplier Reliability Scores appeared to be potentially meaningful, but after our correlation analysis they had virtually no relationship with actual performance, suggesting that they were subjective assessments rather than objective measures.

The most important information we found relates to critical drivers of delivery delays. Our analysis revealed that Actual Transit Days is the strongest predictor of delays, explaining about 57% of variance in the model:

Delay = -1.86 + 0.55 × Transit Days. We also found that disruption severity explained 44% of the data, though all severity levels added similar delay amounts (3.7-3.9 days), indicating that prevention is more important than severity management.

Other queries we ran revealed insights on shipping performance and product-specific risks. We learned that air shipping is the most energy efficient (1.2 joules/unit) with the lowest risk profile. Electronics and pharmaceutical products experience the highest delay days, especially via rail transport, with Pharma+Rail averaging 2.69 days.

We identified 18 high-risk buyer-supplier relationships with disruption rates exceeding 20%, providing clear targets for process improvement. The heatmap visualization made these problem relationships immediately visible. We also discovered that transit time variability drives lead-time performance much more than dispatch time, showing that carrier selection can have greater improvement potential than warehouse operations.

Lessons learned

Working through this project taught us valuable skills in integrating ggplot2 and sqldf for supply chain analysis. We learned how diverse visualization types, heatmaps, violin plots, and faceting, reveal different patterns that single approaches would miss. The biggest surprise was discovering that Supplier Reliability Scores had no predictive merit, requiring us to get creative in how we wanted to approach the rest of this project.

Recommendations

Based on our analysis, we recommend the following actions. First, prioritize air shipping for time-sensitive and high-value shipments, such as pharmaceuticals, when able. Second, immediately address the 18 buyer-supplier relationships with disruption rates exceeding 20% through enhanced protocols or supplier replacement. Third, focus operational improvements on transit time reduction through route optimization and carrier selection rather than dispatch process improvements, as transit variability drives the majority of delays. Fourth and finally, replace, track, and implement proper and objective reliability scores based on true and correct metrics of supplier performance.