# **Suppliers Quality Analysis Project**

**Using**

**Excel & Power Query & Data modeling & Tableau & Python**

**Group Code:** **CLS\_CAI-DAT1-G3e**

**Team**

| **ID** | **Name** |
| --- | --- |
| **1120138778** | **Gamal Mahmoud Zakaria Abdallah** |
| **21020102** | **Samaa Alieldien Hussien Ali** |
| **21042325** | **Nada Abdelmalek Hassan Ali** |
| **21044791** | **Ahmed Mohamed Abdelgawad AboElil** |
| **21022656** | **Yousif Mohamed AbdelAtti sayeed** |
| **21038630** | **Hassan Mahmoud Hassan Elleithy** |

**2024 - 2025**

## **Data Cleaning using Power Query**

**Introduction:**

The dataset provided for this analysis required cleaning and preprocessing to ensure accuracy and consistency. Power Query in Excel was used for this purpose, offering an efficient way to clean and transform the data. This section details the steps taken to clean the data using Power Query.

**Steps:**

1. **Import Data into Power Query:**
   * The dataset was downloaded locally and all sheets were converted into tables.
   * A new Excel sheet was opened, and the data was imported from the **"Supplier Quality Analysis"** Excel sheet using the **"Get Data"** function in the **"Data" tab.**
   * All Excel tables within the file were selected.
   * The **"Transform Data"** option was clicked to initiate the cleaning process.
2. **Rename Tables:**
   * Excel sheet names were modified to indicate their purpose, with **"D"** representing **dimension tables** and **"F"** representing **fact tables.**
3. **Enable Data View Options:**
   * In the **"Data View" tab**, the following options were enabled:
   * "Enable Column Profile"
   * "Show Column Quality Details"
   * "Show Column Value Distribution"
4. **Clean Dimension Tables:**
   * In the **"DCategory"** table, the **"Sub-Category"** column was deleted.
   * In the **"FDefected Items"** table, the **"Sub-Category"** column was renamed to **"Category."**
   * **Data formats** were checked and corrected in all tables.
   * All ID formats were set to **"ABC text."**
5. **Clean Fact Table:**
   * The date/time format in the **"FDefected Items"** table was changed to **"Date."**
   * In the **"DDefect"** table, the following changes were made to correct.

**inconsistencies:**

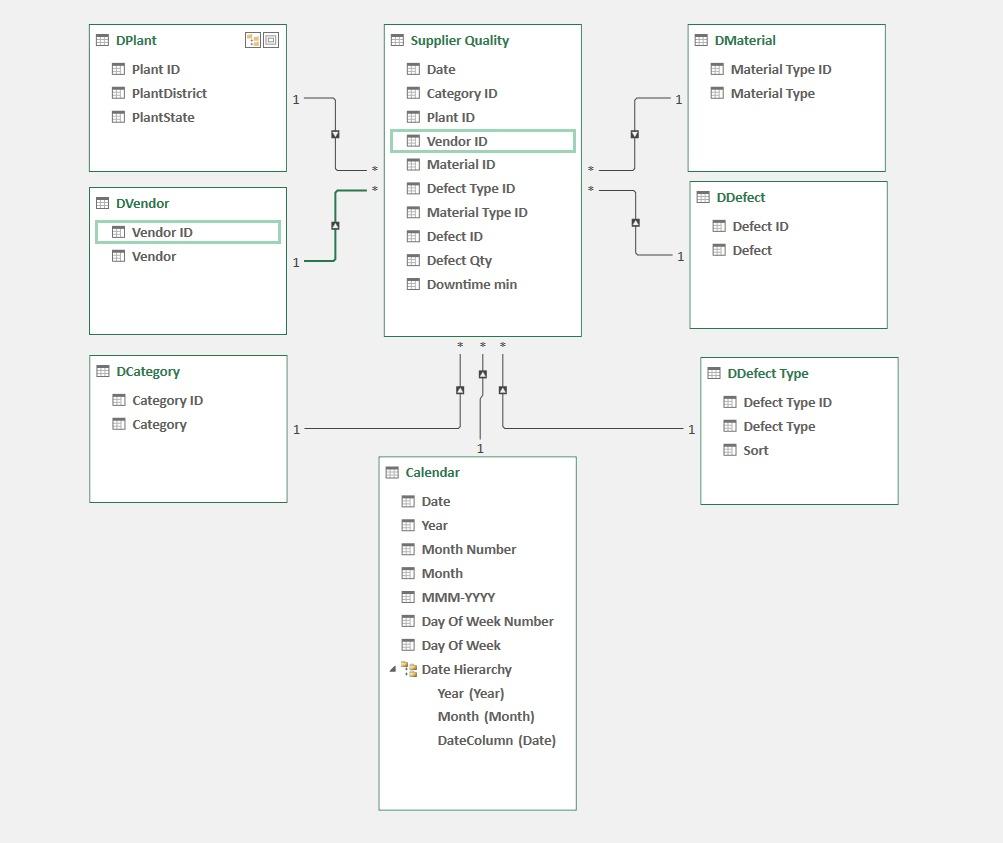
* + - **"Flaps - Incorrect Gap"** changed to **"Flaps and Incorrect Gap"**
    - **"Dimensions - Bad Finishing"** changed to **"Dimensions and Bad Finishing"**
    - **"Material Handling/ Shipping Requirements Error"** changed to **"Material Handling or Shipping Requirements Error"**
    - **"Wrinkles / Scratches/ Scuffing"** changed to **"Material Handling or Shipping Requirements Error"**
    - **"Folding - Wrong Crease"** changed to **"Folding and Wrong Crease"**
    - **"Creases / Wrinkles"** changed to **"Creases or Wrinkles"**
    - **"Bowed/Warped"** changed to **"Bowed or Warped"**
    - **"Over/Under Fill"** changed to **"Over or Under Fill"**
  + The **"Plant"** column was split into two columns: **"PlantDistrict"** and **"PlantState"**

1. **Merge and Clean Data:**
   * All data in the fact table was merged using column IDs.
   * Duplicate entries were removed from all tables.
   * That helped us to see and understand the data as complete rows.
   * we removed the row associated with Defect ID No. 19580 because there were no IDs like this in each dimension table.
   * After understanding the data, I made data normalization.

**Conclusion**

These steps ensured the data was clean, consistent, and ready for analysis. By using Power Query, the cleaning process was efficient and repeatable, allowing for easy updates to the data in the future. The cleaned data will be used to generate accurate and reliable insights in the subsequent analysis.

Data Modeling



## **Data Modeling Report**

**1. Executive Summary**

This report outlines the data modeling process for the Supplier Quality Analysis project. The primary goal is to design a robust and efficient data model to facilitate the analysis and reporting of supplier-quality data. The model utilizes a star schema design, with a central fact table ("Supplier Quality") connected to multiple dimension tables. This structure enables efficient querying and analysis of data related to defects, suppliers, materials, plants, and time.

**2. Introduction**

The data model was created in Excel using Power Pivot, leveraging the relationships between the different tables to facilitate analysis and reporting. The model consists of six dimension tables and one fact table, connected through unique identifiers.

**3. Data Exploration and Preparation**

**The dataset comprises six tables:**

* **Supplier Quality (Fact Table):**

Contains detailed information about each defect instance, including date, supplier, material, defect type, plant, and downtime.

* **DPlant (Dimension Table):**

Provides information about the plant where the defect occurred, including plant ID, district, and state.

* **DVendor (Dimension Table):**

Contains details about the supplier, including vendor ID and name.

* **DMaterial (Dimension Table):**

Stores information about the material involved in the defect, including material type ID and name.

* **DDefect (Dimension Table):**

Provides details about the type of defect, including defect ID and description.

* **DDefect Type (Dimension Table):**

Contains information about the category of the defect, including defect type ID and name.

* **Calendar (Dimension Table):**

Contains date-related information for time intelligence analysis.

**4. Data Modeling**

A star schema design was chosen for its simplicity and efficiency in handling analytical queries. The "Supplier Quality" table serves as the central fact table, connected to the dimension tables through one-to-many relationships.

**Relationships:**

* **Supplier Quality - DPlant:**

One-to-many relationship based on "Plant ID."

* **Supplier Quality - DVendor:**

One-to-many relationship based on "Vendor ID."

* **Supplier Quality - DMaterial:**

One-to-many relationship based on "Material ID."

* **Supplier Quality - DDefect:**

One-to-many relationship based on "Defect ID."

* **Supplier Quality - DDefect Type:**

One-to-many relationship based on "Defect Type ID."

* **Supplier Quality - Calendar:**

One-to-many relationship based on "Date."

**Data Types:**

Appropriate data types were assigned to each column in the tables to ensure data integrity and efficient querying. For example, ID columns were set as whole numbers and date columns were set as date/time data types.

**5. Model Validation**

The data model was validated by performing test queries and verifying the accuracy and consistency of the results. The relationships between tables were checked to ensure that data could be effectively joined and analyzed.

**6. Conclusion**

The developed data model provides a robust and efficient foundation for analyzing supplier-quality data. The star schema design enables easy querying and analysis of data from various perspectives, such as supplier performance, defect trends, and material quality. This model will support the creation of insightful reports and dashboards to drive data-driven decisions for improving supplier quality.

**Supplier Quality Analysis steps and results using Tableau**

1. Create a calculated field to do measure aggregation.
2. Create a dimension Group to distribute the dimensions into categories
3. Create a measure Bin to distribute the measure into categories → to show the Histogram Viz

Made three dashboards →

1- Defect quantity dashboard:

* Defect quantity Histogram Viz to show → defect qty distribution
* Defected quantity by category Treemap Viz to show→ to show which the highest category defected: the mechanical category
* Defected Quantity by defected name Bar Viz to show → Which defected name caused defected qty: Leaking seals
* Vendors performance for defected Qty Box Plot Viz to show → three risk level groups (High risk & medium risk & low risk) by vendors
* Defected type distribution Pie Viz to show →defected Qty & downtime min for defected types every year
* Defected Quantity by vendor Bar Viz to show → Defected Qty per every vendor.
* Defected Quantity by the material Bar Viz to show → Defected Qty per every material.

2- Downtime min dashboard:

* Downtime Min Histogram Viz to show → Downtime Min distribution
* Defected quantity by category Treemap Viz to show→ to show which the highest category defected: the logistics category
* Vendors downtime min performance Box plot Viz to show → three risk level groups (High risk & medium risk & low risk) by vendors
* Downtime min by the material Line Viz to show → Downtime min per every material
* Defected type distribution Pie Viz to show →defected Qty & downtime min for defected types every year
* Downtime Min by vendor Bar Viz to show → Downtime Min per every vendor.
* Downtime min by defect Bar Viz to show → Downtime Min per every Defect.

3- Summary Dashboard:

* The correlation between Defected Qty & Downtime for every defect by Scatter plot Viz
* Defect Qty & Downtime min Pattern Per month
* The correlation between Defected Qty & Downtime for every Plant by Scatter plot Viz
* Defect Qty & Downtime min per Defect Type Per Vendor

**Section 1: Summary of Key Performance Indicators (KPIs)**

| **KPI** | **Value** |
| --- | --- |
| Average Defects Qty | 9166 |
| Total Defects Qty | 54539216 |
| Total Downtime Minutes | 138251 |

**Section 2: Downtime Analysis**

1. **Most Common Downtime Periods**

| **Downtime Category** | **Total Downtime Minutes** |
| --- | --- |
| Logistics | 55849 |
| Mechanical Issues | 34208 |
| Packaging | 29326 |
| Goods and Services | 11964 |
| Material Components | 4946 |
| Electrical | 1958 |

1. **Total Downtime Minutes by Vendors**

| **Vendor** | **Total Downtime Minutes** |
| --- | --- |
| Reddot | 26185 |
| Sanlab | 10275 |
| Plustax | 10270 |
| XXWay | 9966 |
| Quotelane | 5831 |

**Section 3: Defect Analysis**

1. **Average Downtime by Defect Type**

| **Defect Type** | **Average Downtime (Minutes)** |
| --- | --- |
| Impact | 54.75 |
| No Impact | 2.15 |
| Rejected | 17.24 |

1. **Total Downtime Minutes by Material Type**

| **Material Type** | **Total Downtime Minutes** |
| --- | --- |
| Corrugated | 52726 |
| Raw Materials | 23568 |
| Cartons | 12869 |
| Film | 8608 |
| Controllers | 8555 |

1. **Defect Type Distribution**

| **Defect Type** | **Quantity** |
| --- | --- |
| Impact | 216 |
| No Impact | 212 |
| Rejected | 217 |

## **Supplier Quality Analysis Dashboard**

**1. Executive Summary**

This report details the development and insights derived from an interactive Tableau dashboard designed to analyze supplier quality data. The dashboard facilitates the identification of trends, high-risk vendors, and areas for improvement in the manufacturing process.

**Key Findings:**

* **'Logistics'** emerged as the category with the highest defect quantity and downtime.
* **'Leaking seals'** were the most frequent defect, significantly impacting overall defect quantity.
* **'Corrugated materials'** experienced the highest downtime.
* **'Reddot'**, **'Sanlab'**, and **'Plustax'** were identified as the top vendors contributing to downtime.
* **'Impact defects'** caused considerably more downtime compared to **'No Impact defects'**.

**2. Introduction**

The primary objective of this project was to develop an interactive Tableau dashboard to analyze supplier quality data and provide actionable insights to improve product quality, reduce downtime, and optimize supplier relationships. The dashboard enables users to explore defect trends, downtime drivers, and vendor performance through various visualizations and filters.

**3. Data Source and Preparation**

The dataset used for this analysis was sourced from an Excel file containing seven interconnected tables: DPlant, DVendor, DMaterial, DDefect, DDefect Type, DCategory, and Supplier Quality. The data was cleaned and preprocessed using Power Query in Excel to ensure accuracy and consistency.

**4. Dashboard Development**

**Calculated Fields and Groups:**

* **Calculated fields** were created to aggregate measures such as total defect quantity and total downtime.
* **Dimension groups** were created to categorize dimensions into relevant groups, such as defect categories and material types.
* **Measure bins** were created to distribute measures into categories for histogram visualizations.

**Dashboards:**

**Three interconnected dashboards were created:**

1. **Defect Quantity Dashboard:**

**This dashboard focuses on defect quantity analysis, featuring visualizations such as:**

* + **Histogram**: Shows the distribution of defect quantities.
  + **Treemap**: Illustrates defect quantity by category.
  + **Bar chart**: Identifies the leading defect by name.
  + **Box plot**: Benchmarks vendor performance based on defect quantity.
  + **Pie chart**: Shows defect type distribution.

1. **Downtime Dashboard:**

**This dashboard focuses on downtime analysis, featuring visualizations such as:**

* + **Histogram**: Shows the distribution of downtime minutes.
  + **Treemap**: Shows downtime by category.
  + **Box plot**: Categorizes vendors based on downtime performance.
  + **Line chart**: Illustrates downtime trends for different materials over time.
  + **Pie chart**: Shows defect type distribution.

1. **Summary Dashboard:**

**This dashboard provides an overview of key insights, featuring visualizations such as:**

* + **Scatter plot**: Shows the correlation between defect quantity and downtime for each defect and plant.
  + **Line chart**: Illustrates defect quantity and downtime patterns per month.
  + **Heatmap**: Shows defect quantity and downtime per defect type per vendor.

**5. Dashboard Interactivity**

**The dashboards were designed to be interactive, allowing users to:**

* Filter data by various dimensions, such as vendor, material, defect type, and date.
* Drill down into specific data points to gain more detailed insights.
* Hover over data points to view tooltips with additional information.
* Switch between different visualizations to explore data from various perspectives.

**6. Key Insights and Recommendations**

**The interactive dashboards revealed several key insights:**

* Logistics emerged as a critical area for improvement, contributing significantly to both defect quantity and downtime.
* The high frequency of Leaking seals indicates a need for focused quality control measures for this specific defect.
* The significant downtime associated with Corrugated materials warrants further investigation into potential material quality issues or handling procedures.
* Identifying high-risk vendors like Reddot, Sanlab, and Plustax allows for targeted interventions to improve their performance.

**Recommendations:**

* Enhance quality inspection processes, particularly for incoming materials.
* Invest in employee training programs focused on defect prevention and handling.
* Strengthen collaboration and communication with suppliers to address quality issues proactively.

**7. Conclusion**

The interactive Tableau dashboards provide a powerful tool for analyzing supplier quality data and identifying areas for improvement. By leveraging the insights and recommendations presented in this report, the organization can enhance product quality, reduce downtime, and optimize supplier relationships.

**8. Future Work**

Future work could involve incorporating additional data sources, such as employee feedback and machine sensor data, to gain a more holistic view of quality issues. Additionally, predictive models could be developed to forecast potential quality issues and proactively implement preventive measures.

**Supplier Quality Analysis steps and results using Python**

Supplier Defects and Downtime

Purpose

The purpose of this analysis is to gain insights into the defect quantities and downtime associated with different suppliers, materials, and plants. By understanding the patterns and correlations between these variables, the analysis aims to identify key areas of improvement to optimize quality control, reduce downtime, and improve supplier performance.

import pandas as pd

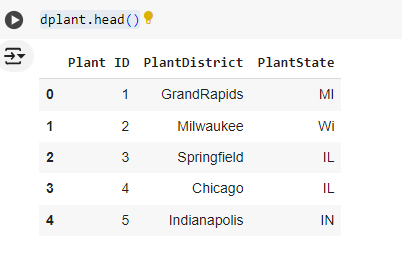
import plotly.express as px

import matplotlib.pyplot as plt

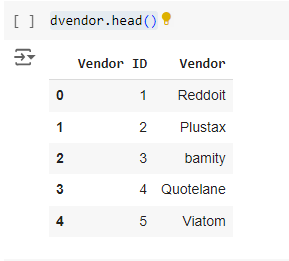
import seaborn as sns

| import pandas as pd  # Read all sheets from a single Excel file file\_path = "D:\Programs\Smart\my\Data analysis\DEPI\Final Project\cleaned un-joined.xlsx" # Replace with your actual file path  # Assuming each sheet is named after the table name dplant = pd.read\_excel(file\_path, sheet\_name='DPlant') dvendor = pd.read\_excel(file\_path, sheet\_name='DVendor') dcategory = pd.read\_excel(file\_path, sheet\_name='DCategory') dmaterial = pd.read\_excel(file\_path, sheet\_name='DMaterial') ddefect = pd.read\_excel(file\_path, sheet\_name='DDefect') ddefect\_type = pd.read\_excel(file\_path, sheet\_name='DDefect Type') #calendar = pd.read\_excel(file\_path, sheet\_name='Calendar') supplier\_quality = pd.read\_excel(file\_path, sheet\_name='Supplier Quality') |
| --- |

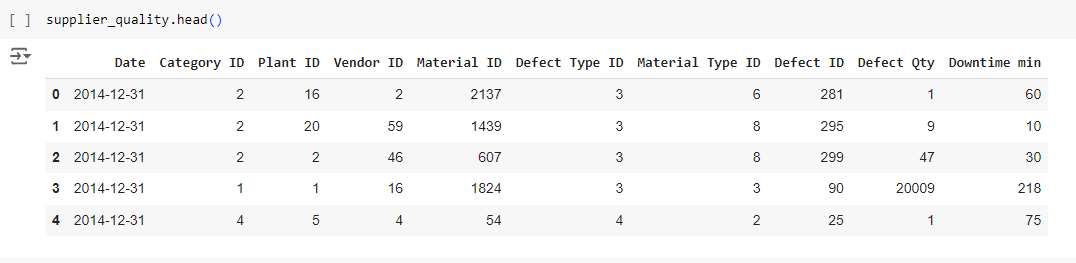
| dplant.head() |
| --- |



| dvendor.head() |
| --- |

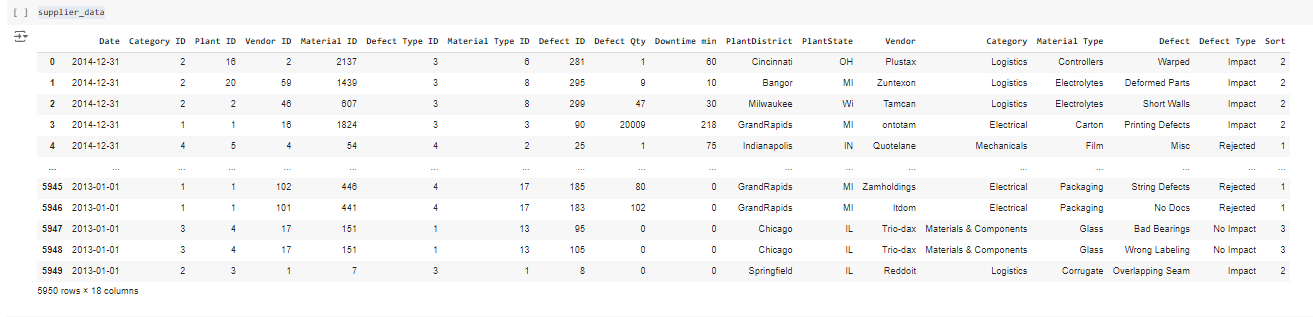


| supplier\_quality.head() |
| --- |

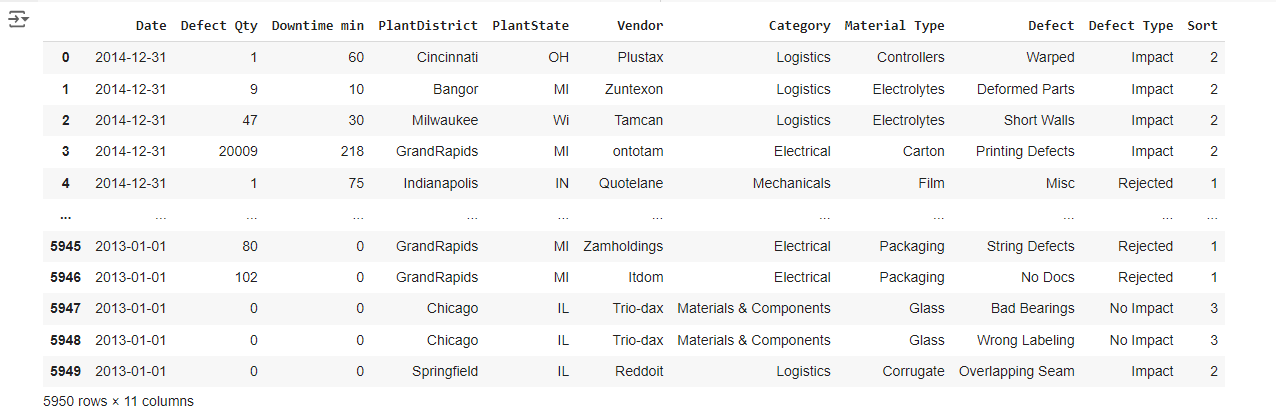


| # Merge the Supplier Quality table with DPlant and DVendor supplier\_data = pd.merge(supplier\_quality, dplant, on='Plant ID', how='left') supplier\_data = pd.merge(supplier\_data, dvendor, on='Vendor ID', how='left') supplier\_data = pd.merge(supplier\_data, dcategory, on='Category ID', how='left') supplier\_data = pd.merge(supplier\_data, dmaterial, on='Material Type ID', how='left') supplier\_data = pd.merge(supplier\_data, ddefect, on='Defect ID', how='left') supplier\_data = pd.merge(supplier\_data, ddefect\_type, on='Defect Type ID', how='left') #supplier\_data = pd.merge(supplier\_data, calendar, on='Date', how='left') |
| --- |

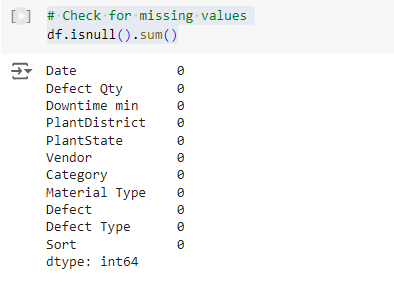
| supplier\_data |
| --- |



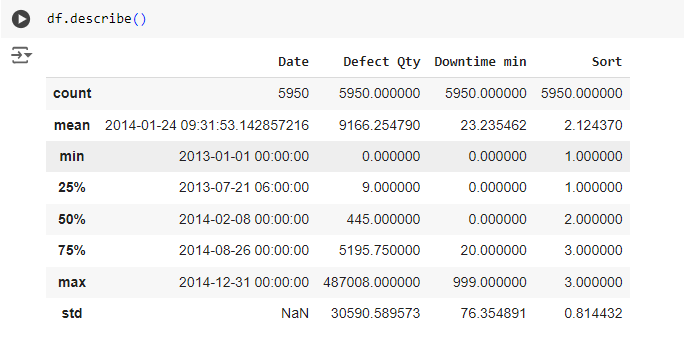
| # Drop all columns ending with 'ID' from the DataFrame df\_cleaned = supplier\_data.drop(supplier\_data.filter(regex='ID$').columns, axis=1)  # Display the cleaned DataFrame df\_cleaned df=df\_cleaned df |
| --- |



| # Check for missing values df.isnull().sum() |
| --- |



| df.describe() |
| --- |

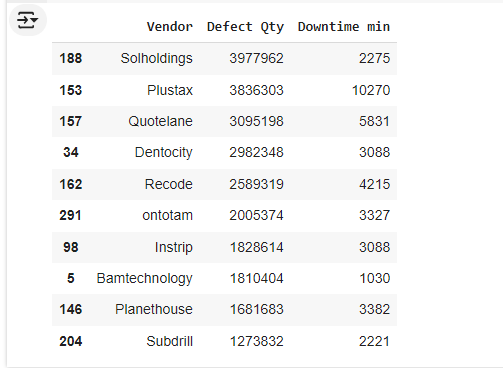


| # Convert columns to appropriate data types (if necessary) supplier\_data['Date'] = pd.to\_datetime(supplier\_data['Date']) |
| --- |

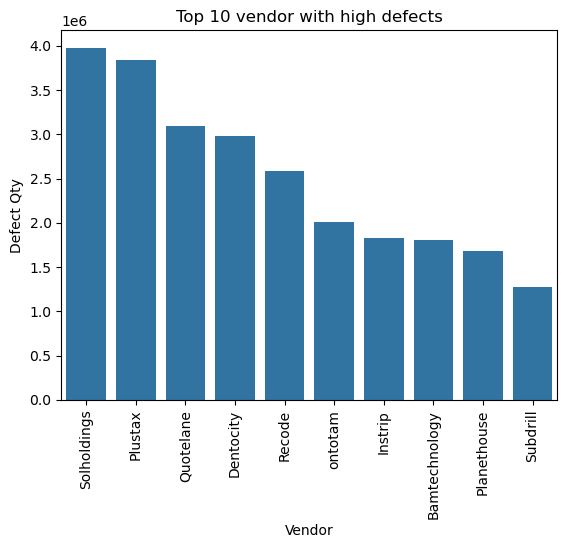
# **Vendor Analysis**

**Top 10 vendors with high defects**

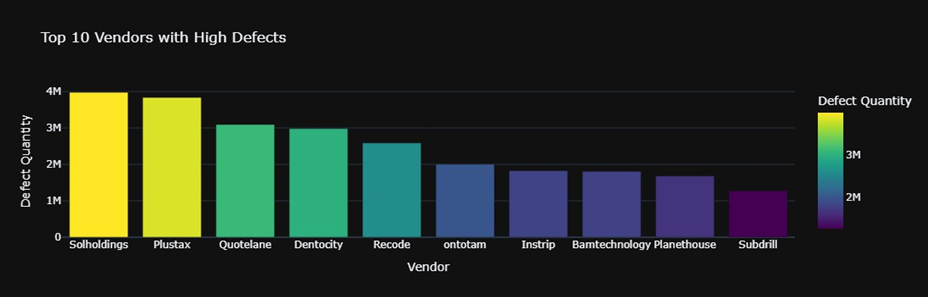
| # Group by Vendor and calculate total defects and downtime vendor\_analysis = supplier\_data.groupby('Vendor')[['Defect Qty', 'Downtime min']].sum().reset\_index() # Sort vendor\_analysis in ascending order by 'Defect Qty' and 'Downtime min' vendor\_analysis\_sorted = vendor\_analysis.sort\_values(by=['Defect Qty', 'Downtime min'], ascending=False)  # Display the sorted DataFrame vendor\_analysis\_sorted.head(10) |
| --- |



| # Plot total defects by vendor import matplotlib.pyplot as plt import seaborn as sns  sns.barplot(data=vendor\_analysis\_sorted.head(10), x='Vendor', y='Defect Qty') plt.xticks(rotation=90) plt.title('Top 10 vendor with high defects') plt.show() |
| --- |

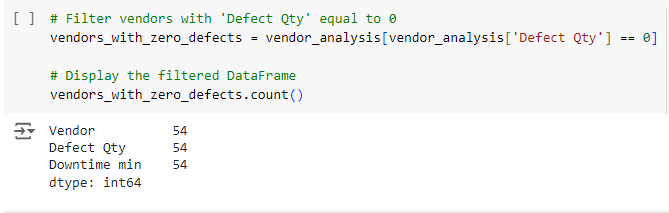


| # Create a bar chart for top 10 vendors with high defect quantities using plotly fig = px.bar(vendor\_analysis\_sorted.head(10),   x='Vendor',   y='Defect Qty',  title='Top 10 Vendors with High Defects',  labels={'Defect Qty': 'Defect Quantity'},  template='plotly\_dark',  color='Defect Qty',  color\_continuous\_scale='Viridis')   # Display the plot fig.show() |
| --- |



## Vendors with a Zero Defect Qty

| # Filter vendors with 'Defect Qty' equal to 0 vendors\_with\_zero\_defects = vendor\_analysis[vendor\_analysis['Defect Qty'] == 0]  # Display the filtered DataFrame vendors\_with\_zero\_defects.count() |
| --- |
|  |

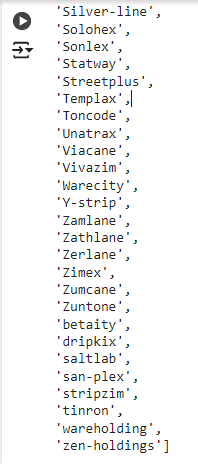


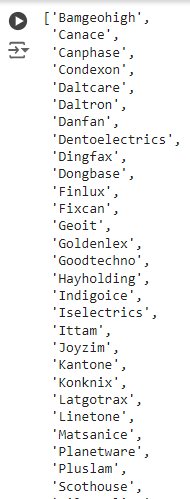
# Get a list of vendors with 'Defect Qty' equal to 0

vendors\_with\_zero\_defects\_list = vendors\_with\_zero\_defects['Vendor'].tolist()

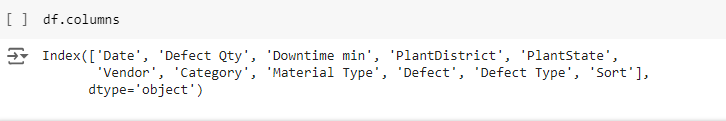
# Display the list

vendors\_with\_zero\_defects\_list



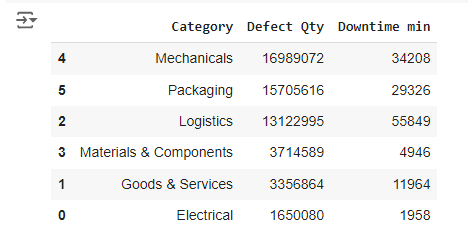


| df.columns |
| --- |

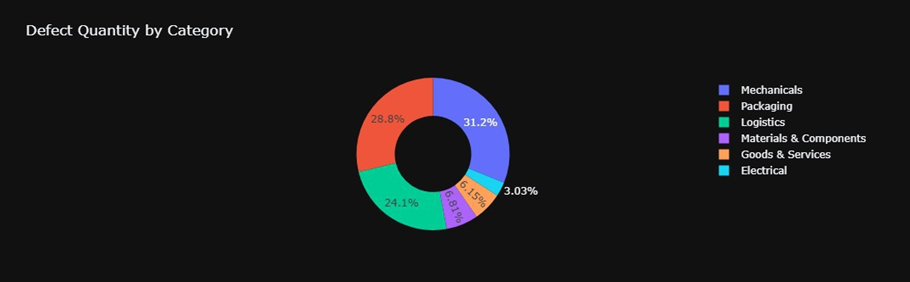


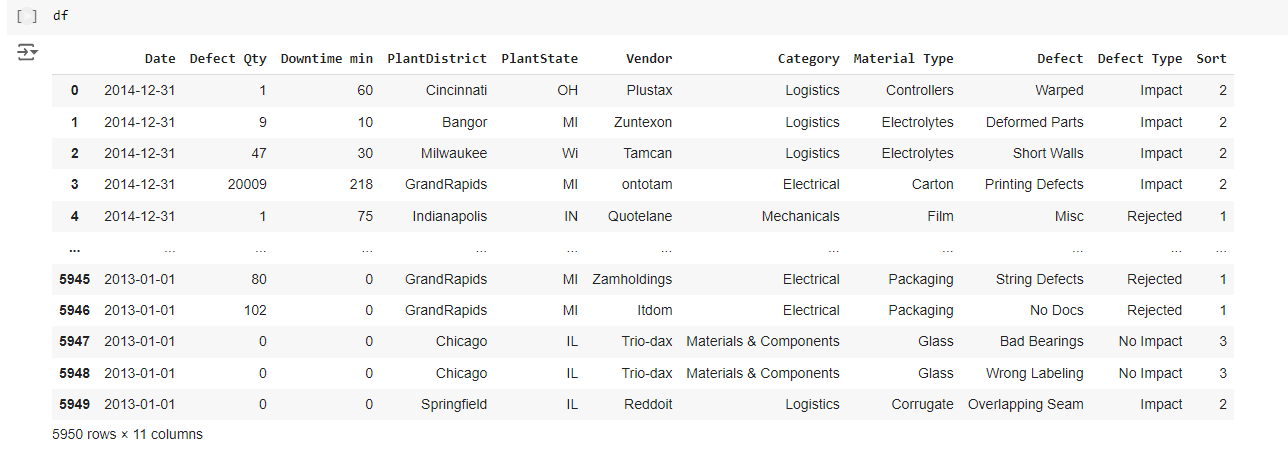
**Category Analysis**

| **# Group by Category and calculate total defects and downtime category\_analysis = supplier\_data.groupby('Category')[['Defect Qty', 'Downtime min']].sum().reset\_index()  # Display the analysis results category\_analysis\_sorted = category\_analysis.sort\_values(by='Defect Qty', ascending=False)  category\_analysis\_sorted** |
| --- |

****

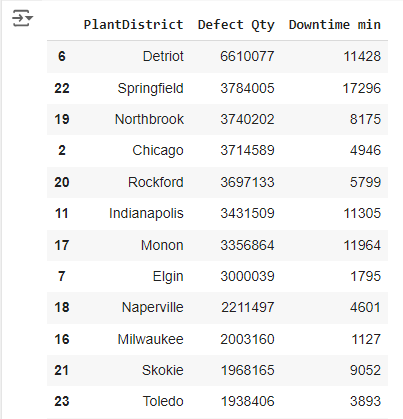
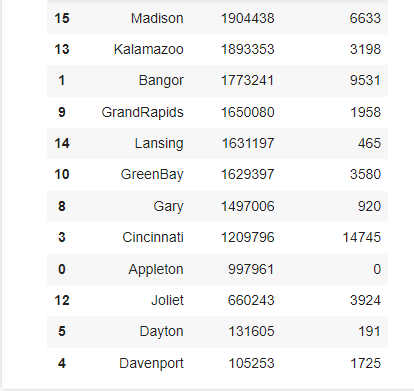
| import plotly.express as px # Create a pie chart using plotly to visualize the defect quantity by category fig = px.pie(category\_analysis\_sorted,   values='Defect Qty',   names='Category',  title='Defect Quantity by Category',  hole=0.5,  template='plotly\_dark')  # Show the pie chart fig.show() |
| --- |



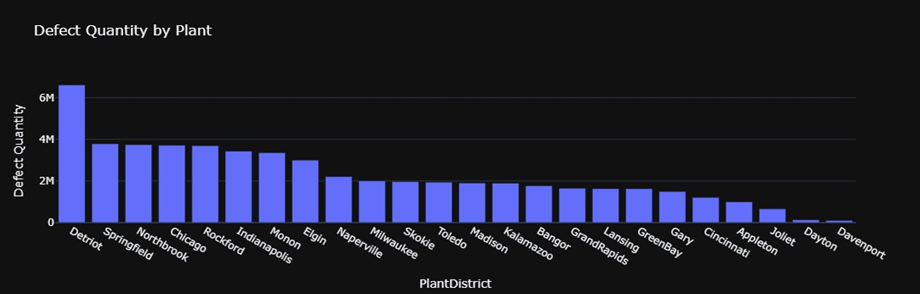


# **plant Analysis**

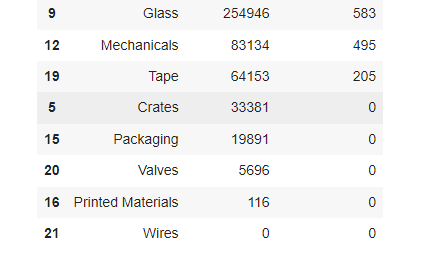
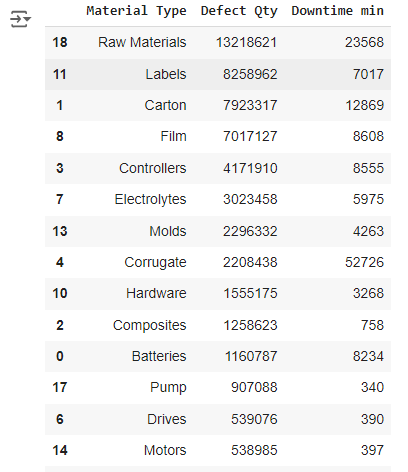
| # Group by Plant and calculate total defects and downtime plant\_analysis = supplier\_data.groupby('PlantDistrict')[['Defect Qty', 'Downtime min']].sum().reset\_index()  # Sort the results by total defects plant\_analysis\_sorted = plant\_analysis.sort\_values(by='Defect Qty', ascending=False)  # Display the analysis results plant\_analysis\_sorted |
| --- |



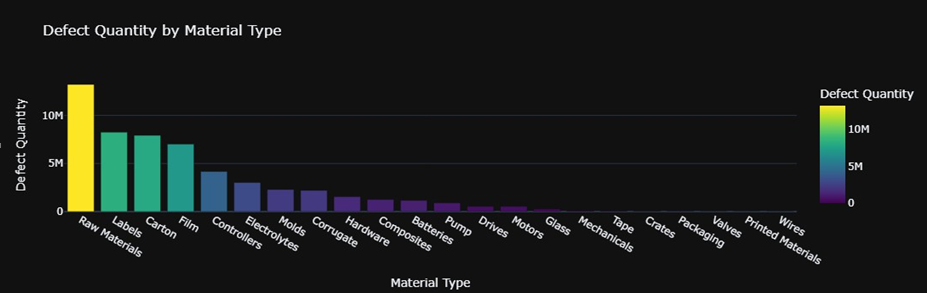
| import plotly.express as px  # Create a bar chart for defect quantity by plant fig = px.bar(plant\_analysis\_sorted,   x='PlantDistrict',   y='Defect Qty',  title='Defect Quantity by Plant',  labels={'Defect Qty': 'Defect Quantity'},  template='plotly\_dark')  # Show the plot fig.show() |
| --- |

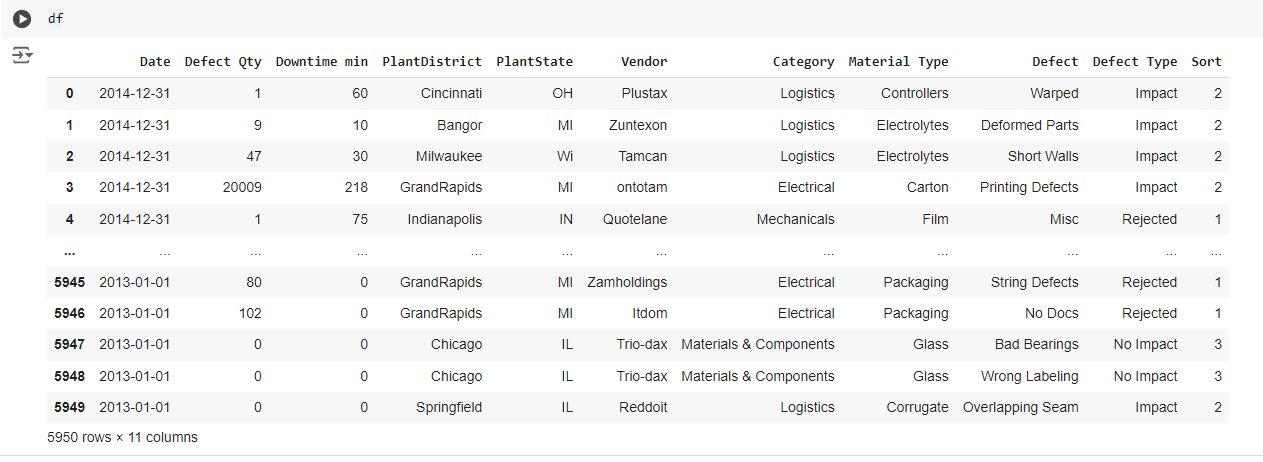


| # Group by Material Type and calculate total defects and downtime material\_type\_analysis = supplier\_data.groupby('Material Type')[['Defect Qty', 'Downtime min']].sum().reset\_index()  # Sort the results by total defect quantity material\_type\_analysis\_sorted = material\_type\_analysis.sort\_values(by='Defect Qty', ascending=False)  # Display the analysis results material\_type\_analysis\_sorted |
| --- |



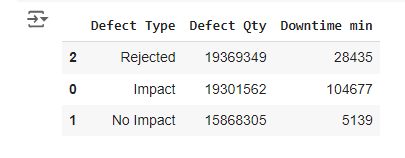
| import plotly.express as px  # Create a bar chart for defect quantity by material type fig = px.bar(material\_type\_analysis\_sorted,   x='Material Type',   y='Defect Qty',  title='Defect Quantity by Material Type',  labels={'Defect Qty': 'Defect Quantity'},  template='plotly\_dark',  color='Defect Qty',  color\_continuous\_scale='Viridis') # Choose a color palette  # Display the plot fig.show() |
| --- |



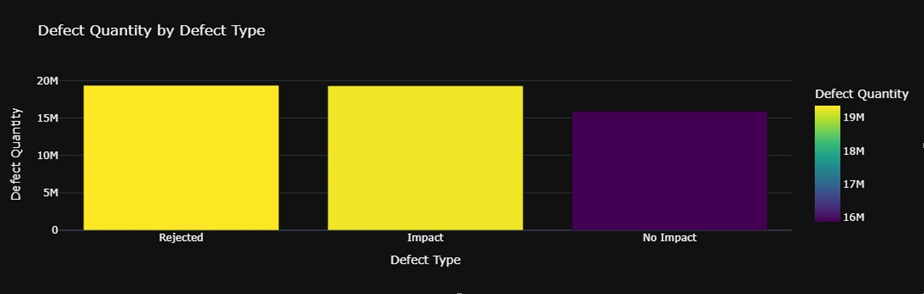


**Defect Type Analysis**

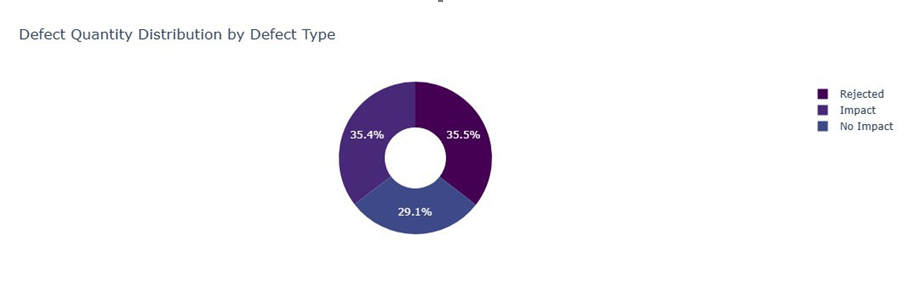
| # Group by Defect Type and calculate total defect quantity and downtime defect\_analysis = supplier\_data.groupby('Defect Type')[['Defect Qty', 'Downtime min']].sum().reset\_index()  # Sort the results by total defect quantity defect\_analysis\_sorted = defect\_analysis.sort\_values(by='Defect Qty', ascending=False)  # Display the analysis results defect\_analysis\_sorted |
| --- |

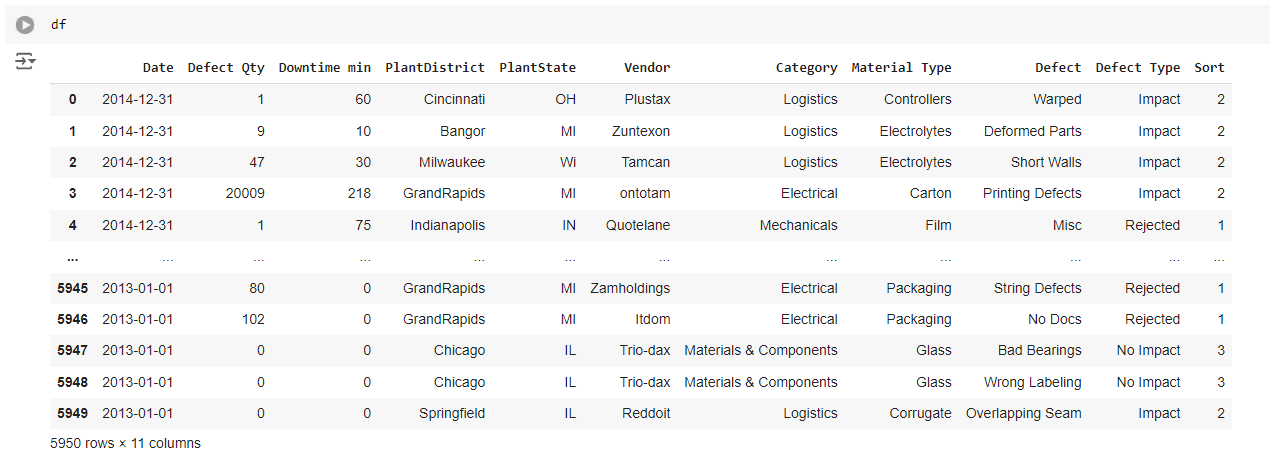
****

| import plotly.express as px  # Create a bar chart for defect quantity by defect type fig = px.bar(defect\_analysis\_sorted,   x='Defect Type',   y='Defect Qty',  title='Defect Quantity by Defect Type',  labels={'Defect Qty': 'Defect Quantity'},  template='plotly\_dark',  color='Defect Qty',  color\_continuous\_scale='Viridis')  # Display the plot fig.show() |
| --- |



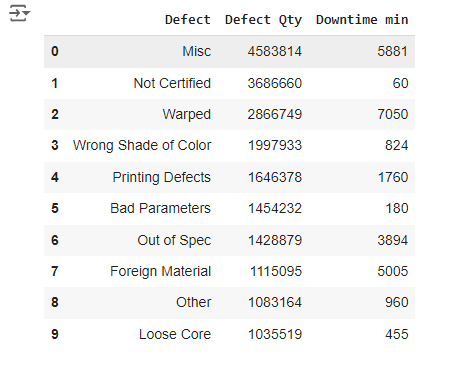
| # Create a pie chart for defect quantity by defect type fig = px.pie(defect\_analysis\_sorted,   values='Defect Qty',   names='Defect Type',   title='Defect Quantity Distribution by Defect Type',  hole=0.4, # Creates a donut chart  color\_discrete\_sequence=px.colors.sequential.Viridis)  # Display the plot fig.show() |
| --- |



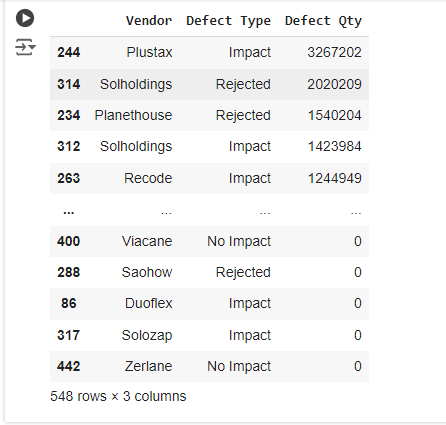


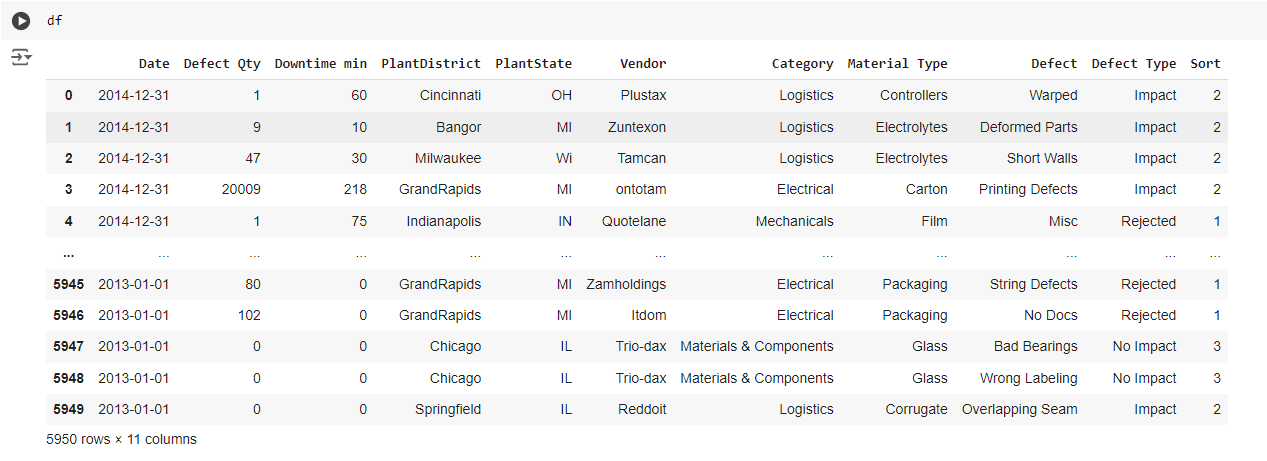
# **Top 10 Defect Quantities by Defects**

| # Group the data by Defect to get total defect quantities and associated downtime defect\_analysis = supplier\_data.groupby('Defect')[['Defect Qty', 'Downtime min']].sum().reset\_index()  # Sort by the total defect quantity for better visualization defect\_analysis\_sorted = defect\_analysis.sort\_values(by='Defect Qty', ascending=False).reset\_index(drop=True)  # Display the top 10 results without an extra index defect\_analysis\_sorted=defect\_analysis\_sorted.head(10) defect\_analysis\_sorted |
| --- |

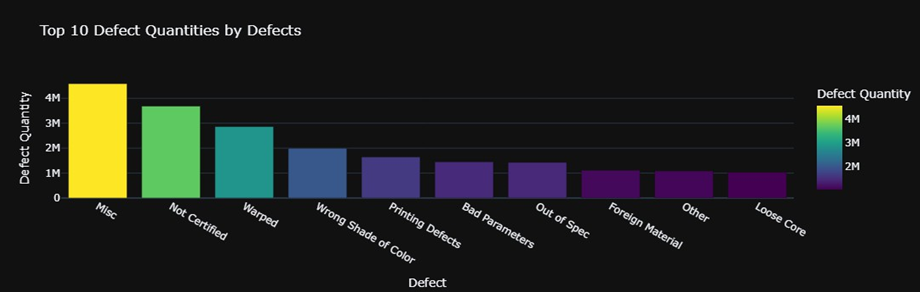


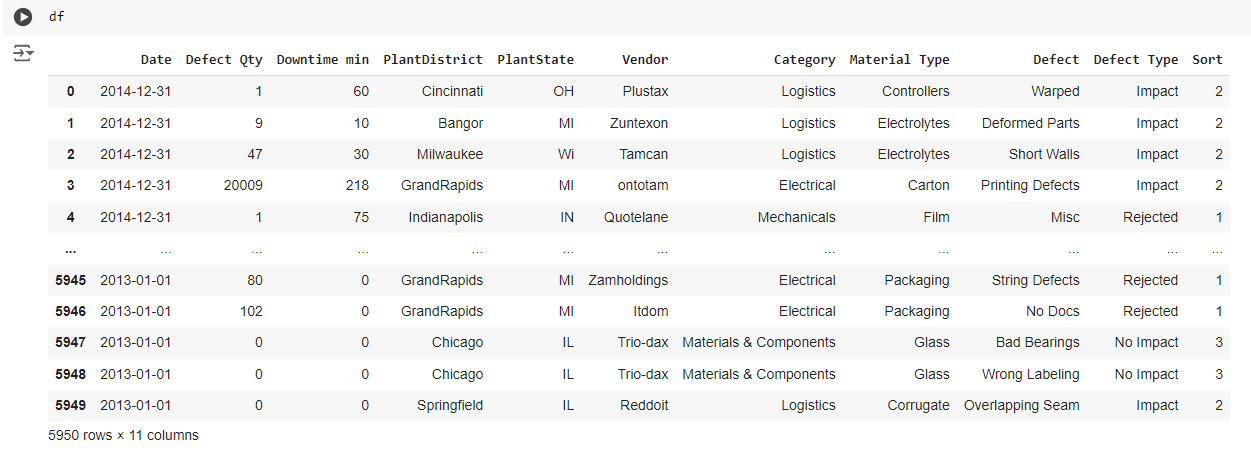
| # Group by Vendor and Defect Type to see how defects are spread across vendors vendor\_defect\_analysis = supplier\_data.groupby(['Vendor', 'Defect Type'])[['Defect Qty']].sum().reset\_index()  # Display the top entries for insight vendor\_defect\_analysis.sort\_values(by='Defect Qty', ascending=False) |
| --- |





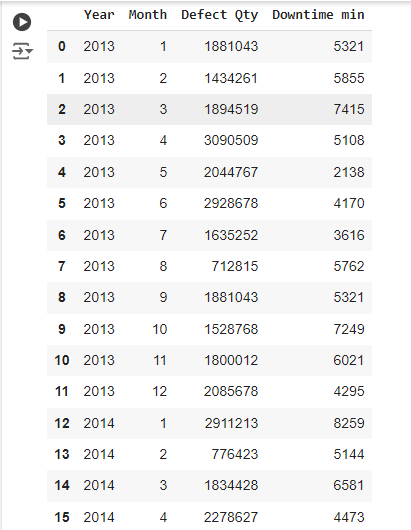
| # Create a bar chart to show total defect quantities by defect type fig = px.bar(defect\_analysis\_sorted,   x='Defect',   y='Defect Qty',  title='Top 10 Defect Quantities by Defects',  labels={'Defect Qty': 'Defect Quantity'},  template='plotly\_dark',  color='Defect Qty',  color\_continuous\_scale='Viridis')   fig.update\_layout(xaxis={'categoryorder':'total descending'})  # Display the plot fig.show() |
| --- |

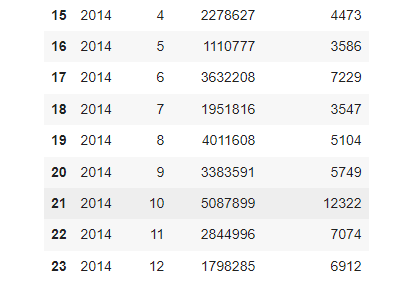




| # Convert 'Date' column to datetime if not already df['Date'] = pd.to\_datetime(supplier\_data['Date']) |
| --- |

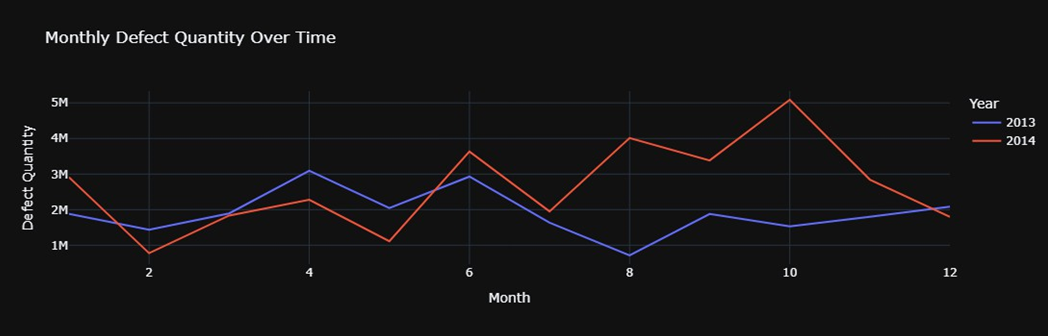
| # Group by month and year to analyze defects over time df['Year'] = df['Date'].dt.year df['Month'] = df['Date'].dt.month  # Group by year and month to calculate total defects and downtime date\_analysis = df.groupby(['Year', 'Month'])[['Defect Qty', 'Downtime min']].sum().reset\_index()  # Display the analysis results date\_analysis |
| --- |





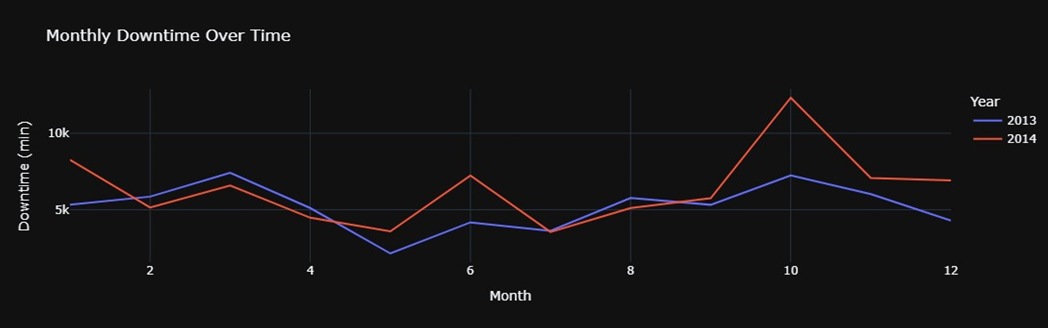
# **Defect Quantities Over Time**

| import plotly.express as px  # Create a line chart for defect quantities over time fig = px.line(date\_analysis,   x='Month',   y='Defect Qty',   color='Year',  title='Monthly Defect Quantity Over Time',  labels={'Defect Qty': 'Defect Quantity'},  template='plotly\_dark')  fig.show() |
| --- |



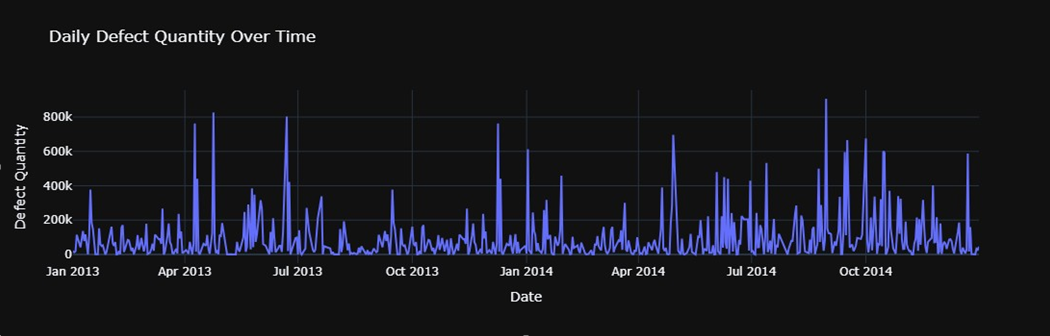
**Downtime Over Time**

| # Create a line chart for downtime over time fig = px.line(date\_analysis,   x='Month',   y='Downtime min',   color='Year',  title='Monthly Downtime Over Time',  labels={'Downtime min': 'Downtime (min)'},  template='plotly\_dark')  fig.show() |
| --- |



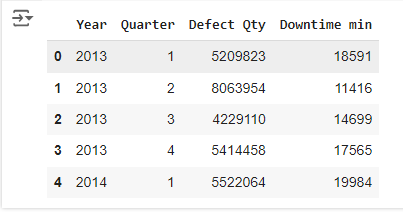
# **Daily or Weekly Analysis**

| # Group by day to see daily trends daily\_analysis = supplier\_data.groupby('Date')[['Defect Qty', 'Downtime min']].sum().reset\_index()  # Create a line chart for daily defect quantities fig = px.line(daily\_analysis,   x='Date',   y='Defect Qty',   title='Daily Defect Quantity Over Time',  labels={'Defect Qty': 'Defect Quantity'},  template='plotly\_dark')  fig.show() |
| --- |

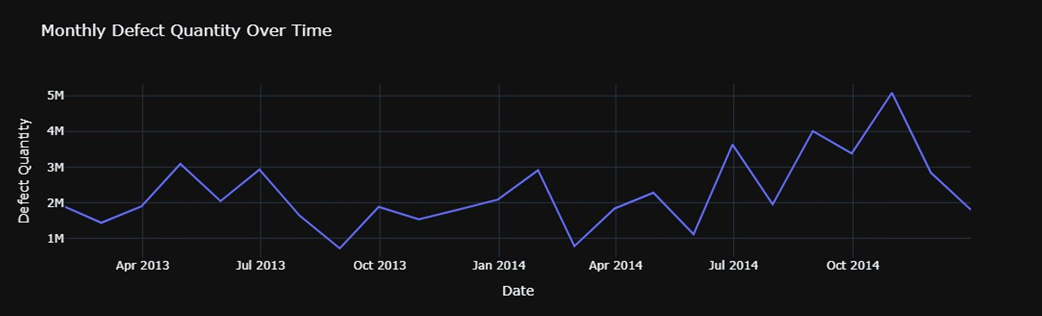


# **Seasonal or Quarter Analysis**

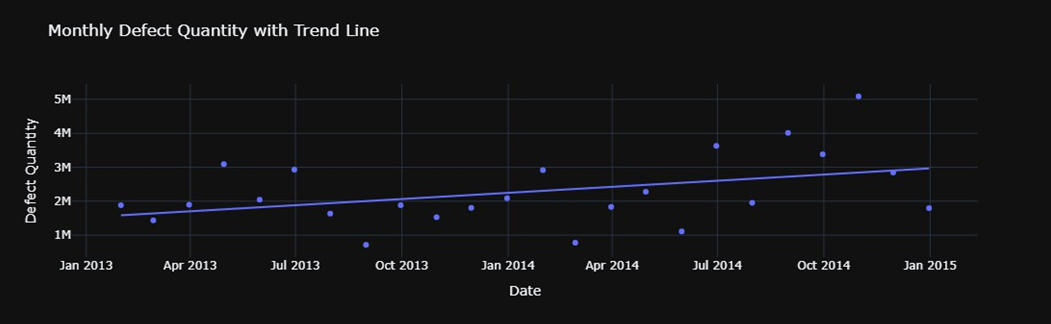
| # Ensure the 'Date' column is in datetime format supplier\_data['Date'] = pd.to\_datetime(supplier\_data['Date'])  # Extract Year and Quarter supplier\_data['Year'] = supplier\_data['Date'].dt.year supplier\_data['Quarter'] = supplier\_data['Date'].dt.quarter  # Group by 'Year' and 'Quarter' for analysis quarterly\_analysis = supplier\_data.groupby(['Year', 'Quarter'])[['Defect Qty', 'Downtime min']].sum().reset\_index()  # Display the analysis quarterly\_analysis.head() |
| --- |



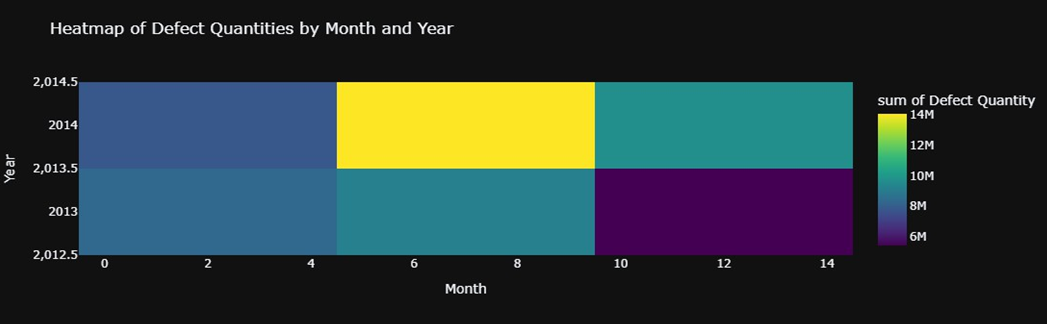
| import plotly.express as px  # Ensure 'Date' column is in datetime format supplier\_data['Date'] = pd.to\_datetime(supplier\_data['Date'])  # Aggregate defect quantity by month monthly\_defects = supplier\_data.resample('M', on='Date').sum().reset\_index()  # Create a line chart for defect quantity over time fig = px.line(monthly\_defects,  x='Date',  y='Defect Qty',  title='Monthly Defect Quantity Over Time',  labels={'Defect Qty': 'Defect Quantity'},  template='plotly\_dark')  # Display the line chart fig.show() |
| --- |



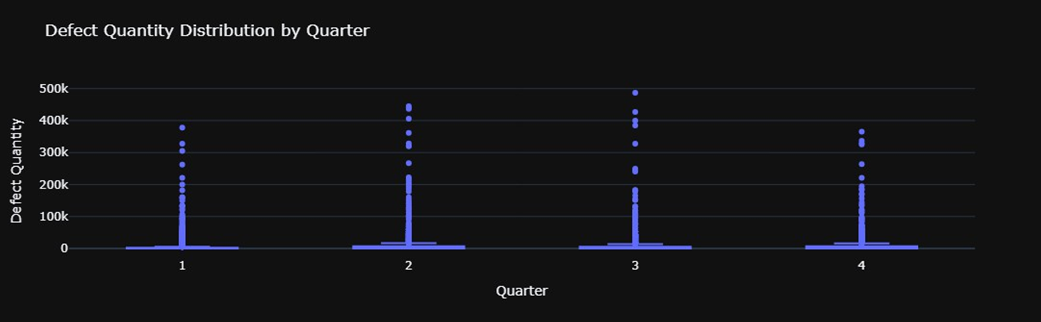
| # Create a scatter plot for defect quantity with trend line fig = px.scatter(monthly\_defects,  x='Date',  y='Defect Qty',  title='Monthly Defect Quantity with Trend Line',  labels={'Defect Qty': 'Defect Quantity'},  trendline='ols', # Add an Ordinary Least Squares (OLS) trendline  template='plotly\_dark')  # Display the scatter plot fig.show() |
| --- |



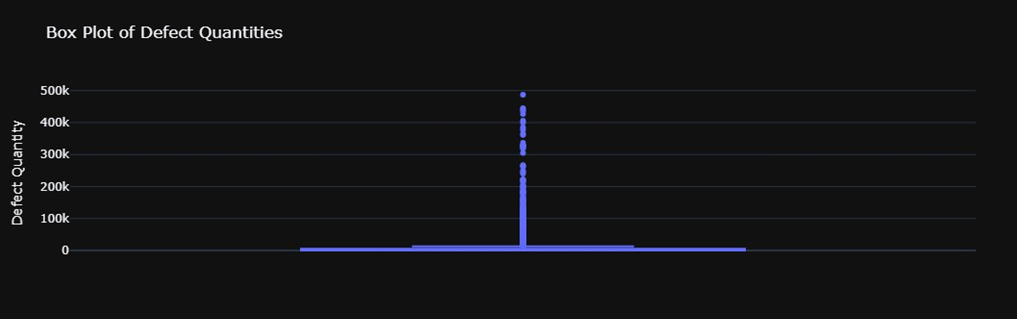
| # Create 'Year' and 'Month' columns supplier\_data['Year'] = supplier\_data['Date'].dt.year supplier\_data['Month'] = supplier\_data['Date'].dt.month  # Group by month and year monthly\_defect\_summary = supplier\_data.groupby(['Year', 'Month'])['Defect Qty'].sum().reset\_index()  # Create a heatmap to visualize defects over time fig = px.density\_heatmap(monthly\_defect\_summary,  x='Month',  y='Year',  z='Defect Qty',  title='Heatmap of Defect Quantities by Month and Year',  labels={'Defect Qty': 'Defect Quantity'},  template='plotly\_dark',  color\_continuous\_scale='Viridis')  # Display the heatmap fig.show() |
| --- |

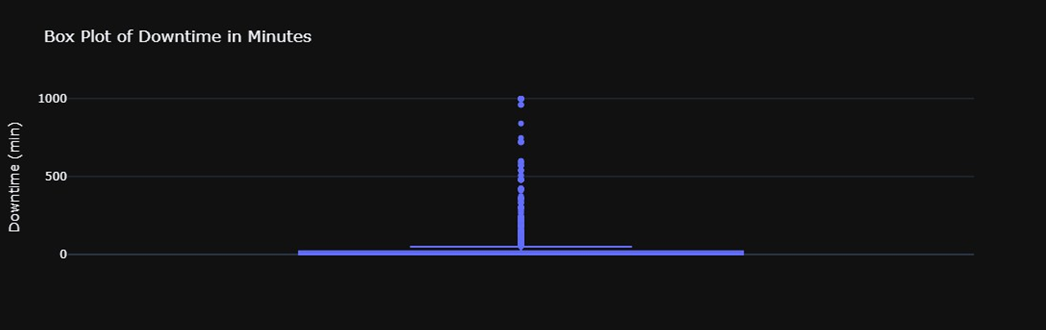


| # Create a 'Quarter' column supplier\_data['Quarter'] = supplier\_data['Date'].dt.quarter  # Create a box plot to visualize defect distribution by quarter fig = px.box(supplier\_data,   x='Quarter',   y='Defect Qty',  title='Defect Quantity Distribution by Quarter',  labels={'Defect Qty': 'Defect Quantity'},  template='plotly\_dark')  # Display the box plot fig.show() |
| --- |

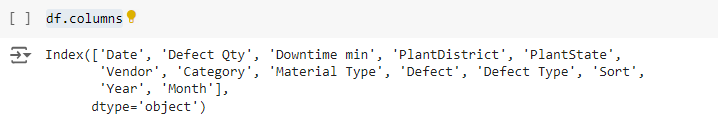


| import plotly.express as px  # Create a box plot for 'Defect Qty' fig\_qty = px.box(supplier\_data,  y='Defect Qty',  title='Box Plot of Defect Quantities',  labels={'Defect Qty': 'Defect Quantity'},  template='plotly\_dark')  # Display the box plot for 'Defect Qty' fig\_qty.show()  # Create a box plot for 'Downtime min' fig\_downtime = px.box(supplier\_data,  y='Downtime min',  title='Box Plot of Downtime in Minutes',  labels={'Downtime min': 'Downtime (min)'},  template='plotly\_dark')  # Display the box plot for 'Downtime min' fig\_downtime.show() |
| --- |

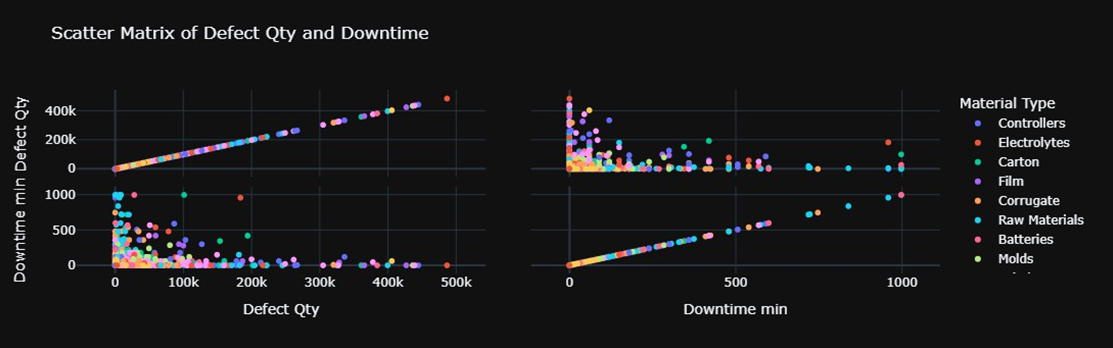




| df.columns |
| --- |



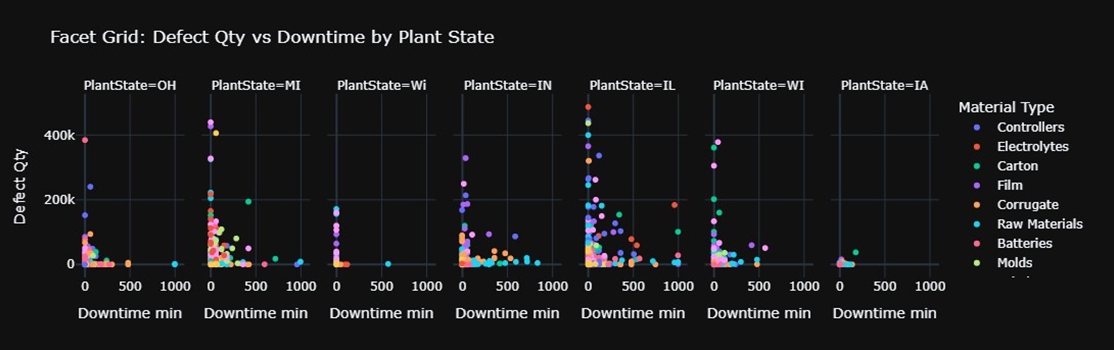
| import plotly.express as px  # Select relevant numerical columns for multivariate analysis numerical\_columns = supplier\_data[['Defect Qty', 'Downtime min']]  # Add categorical columns for color coding numerical\_columns['Material Type'] = supplier\_data['Material Type']  # Create a pair plot fig = px.scatter\_matrix(numerical\_columns,   dimensions=['Defect Qty', 'Downtime min'],  color='Material Type',  title='Scatter Matrix of Defect Qty and Downtime',  template='plotly\_dark')  # Display the pair plot fig.show() |
| --- |



| import numpy as np  # Calculate the correlation matrix corr\_matrix = supplier\_data[['Defect Qty', 'Downtime min']].corr()  # Create a heatmap for the correlation matrix fig = px.imshow(corr\_matrix,   text\_auto=True,   title='Heatmap of Correlation Between Defect Qty and Downtime',  template='plotly\_dark')  # Display the heatmap fig.show() |
| --- |



| # Create a facet grid plot for Defect Qty by PlantState fig = px.scatter(supplier\_data,   x='Downtime min',   y='Defect Qty',  color='Material Type',  facet\_col='PlantState',  title='Facet Grid: Defect Qty vs Downtime by Plant State',  template='plotly\_dark')  # Display the facet grid plot fig.show() |
| --- |



## Summary

The analysis was performed on a dataset containing information about defect quantities (Defect Qty), downtime (Downtime min), associated Vendors, Material Types, Plants, and Defect Types. Several visualizations were used to explore relationships between these variables:

* **Defect Analysis**: Visualized defect quantities and their associated downtime by defect type, revealing key contributors to quality issues.
* **Date Analysis**: Assessed the distribution of defects and downtime over time (daily, monthly, and quarterly) to uncover any seasonal or periodic patterns.
* **Multivariate Analysis**: Used scatter plots, heatmaps, and facet grids to investigate how multiple factors such as PlantDistrict, Material Type, and Vendor impact defects and downtime.

The key findings indicate that certain defect types are strongly associated with longer downtime, and some vendors have disproportionately high defect rates compared to others. There are also seasonal patterns, with some periods experiencing significantly higher defect rates.

## Conclusion

The analysis reveals that defect rates and downtime are not evenly distributed across all vendors, materials, and plants. Several insights were identified:

* **High Impact Defects**: A small number of defect types contribute to the majority of quality issues and associated downtime.
* **Vendor Performance**: Some vendors have consistently higher defect rates, indicating potential quality control issues that need to be addressed.
* **Time-Based Patterns**: Defect rates and downtime fluctuate throughout the year, with noticeable peaks at certain times, possibly due to production cycles, seasonality, or operational changes.
* **Plant and Material-Specific Issues**: Certain plants or materials are more susceptible to defects, suggesting the need for targeted proce, and supplier performance.