Project 1: Occupational Hazard Analysis Using Industry Data

Objective

This project aims to analyze occupational safety data to identify trends in workplace hazards across various industries. The analysis will focus on metrics such as reported incidents, injuries, and fatalities.

Libraries Used

We'll use the following libraries:

- **Numpy**: For numerical operations.
- **Pandas**: For data manipulation.
- **Matplotlib**: For data visualization.

Step 1: Import Libraries

We start by importing the necessary libraries.

```
# Importing necessary libraries for data analysis
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

plt.style.use('seaborn-v0_8-darkgrid')
plt.rcParams["figure.figsize"] = (10, 6)
```

Step 2: Load the Data

In this project phase, we'll load the data into a DataFrame. Here, we are using a simulated dataset, which will be replaced with actual data upon fetching it from an industry source like the Indian government's data portal.

```
columns = ["Industry", "Year", "Reported_Incidents", "Injuries",
"Fatalities"]
data = {
    "Industry": ["Manufacturing", "Construction", "Healthcare",
"Transportation", "Mining"],
    "Year": [2020, 2020, 2020, 2020, 2020],
    "Reported_Incidents": [250, 300, 150, 200, 350],
    "Injuries": [120, 180, 90, 110, 250],
    "Fatalities": [5, 12, 3, 8, 20]
}
```

```
# Creating DataFrame
df = pd.DataFrame(data)
df.head()

          Industry  Year  Reported_Incidents  Injuries  Fatalities
0    Manufacturing  2020                 250       120           5
1     Construction  2020                 300       180          12
2       Healthcare  2020                 150        90           3
3   Transportation  2020                 200       110           8
4           Mining  2020                 350       250          20
```

Step 3: Data Overview and Cleaning

We'll check for any missing values or anomalies in the data. Since this is a simulated dataset, it's clean; however, in real-world data, missing values and inconsistencies are common.

```
missing_values = df.isnull().sum()
missing_values

Industry              0
Year                  0
Reported_Incidents    0
Injuries              0
Fatalities            0
dtype: int64
```

Step 4: Data Analysis

In this section, we calculate key metrics for each industry, such as:

- **Incident Rate**: Ratio of reported incidents to the total number of records.
- **Injury Severity**: Ratio of injuries to reported incidents.
- **Fatality Rate**: Ratio of fatalities to reported incidents.

```
# Calculating additional metrics
df['Incident_Rate'] = df['Reported_Incidents'] /
df['Reported_Incidents'].sum()
df['Injury_Severity'] = df['Injuries'] / df['Reported_Incidents']
df['Fatality_Rate'] = df['Fatalities'] / df['Reported_Incidents']

# Displaying the updated DataFrame with calculated metrics
df[['Industry', 'Reported_Incidents', 'Injury_Severity',
'Fatality_Rate']]

          Industry  Reported_Incidents  Injury_Severity  Fatality_Rate
0    Manufacturing                 250         0.480000       0.020000
1     Construction                 300         0.600000       0.040000
2       Healthcare                 150         0.600000       0.020000
```

```
3  Transportation                      200          0.550000        0.040000
4          Mining                      350          0.714286        0.057143
```

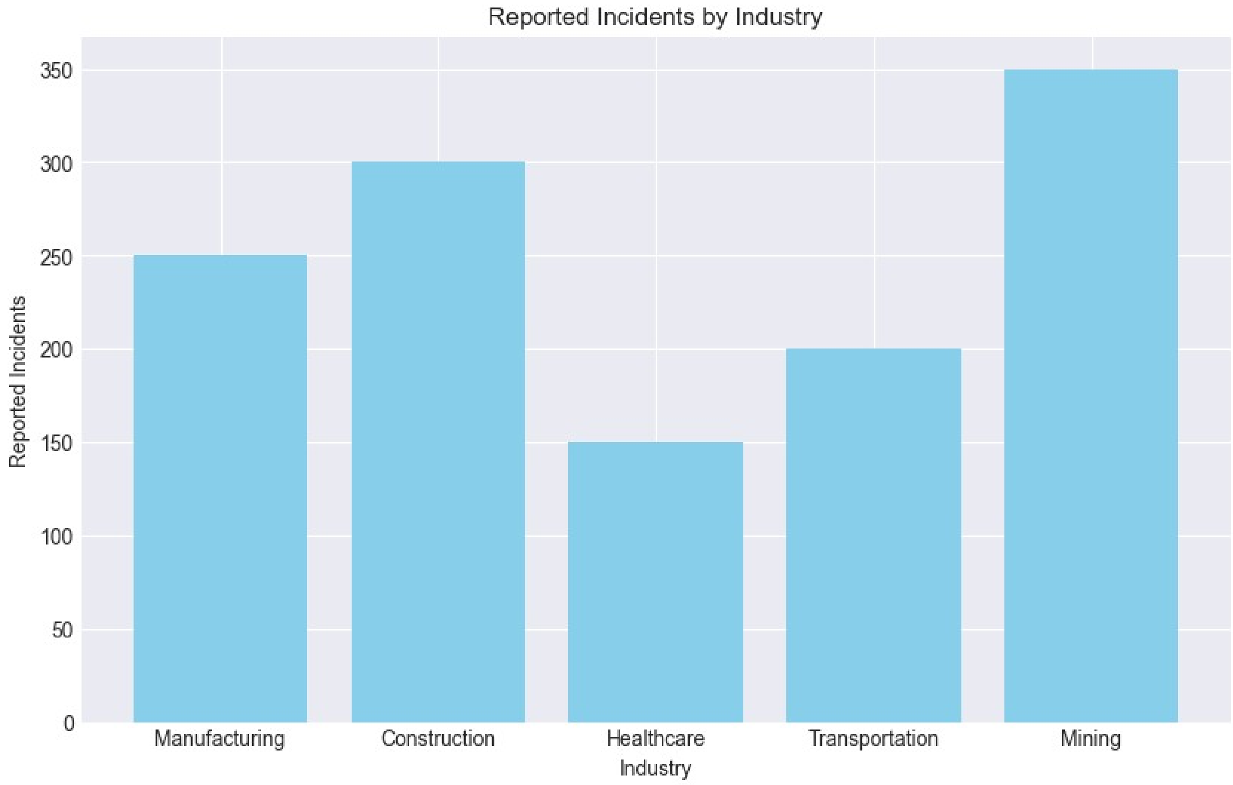
Step 5: Data Visualization

To better understand the distribution and trends, we’ll create visualizations:

1. **Incident Comparison by Industry**: A bar chart to show reported incidents per industry.
2. **Injury Severity Across Industries**: A bar chart representing injury severity per industry.
3. **Fatality Rate by Industry**: A bar chart comparing the fatality rates.

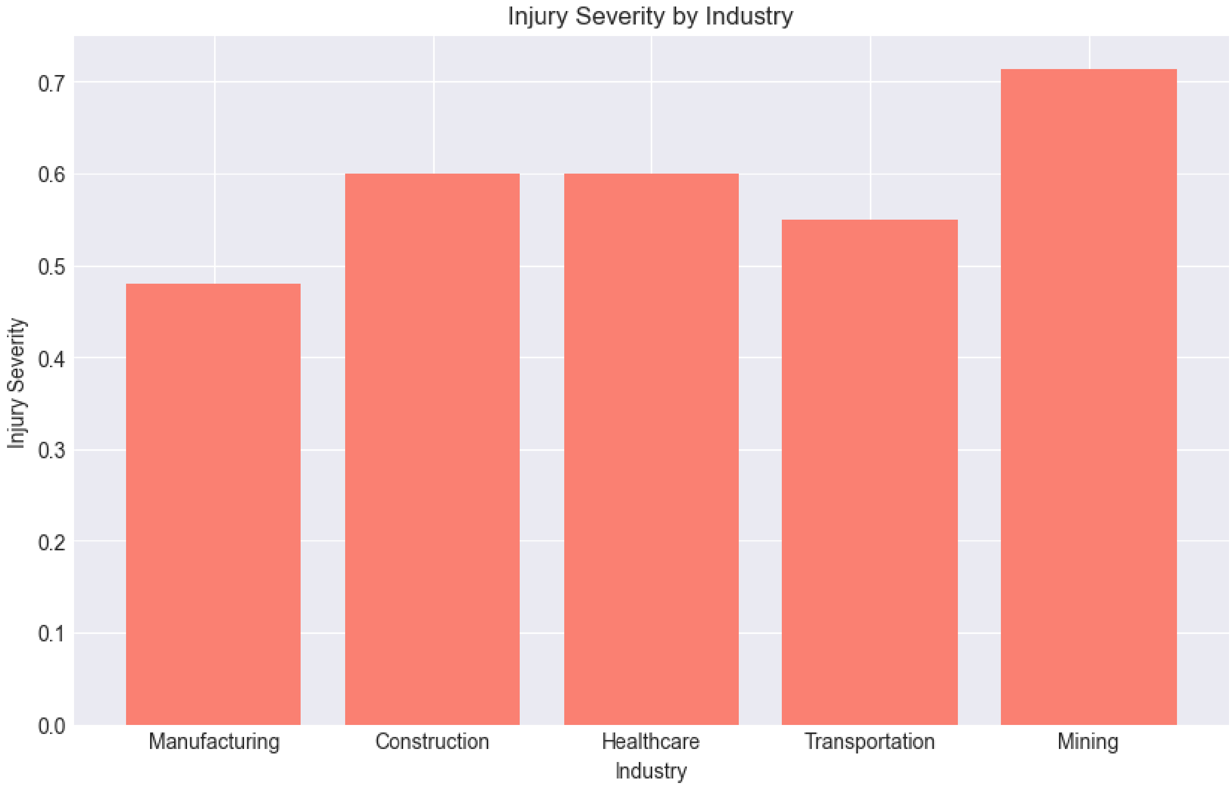
Visualization 1: Reported Incidents by Industry

```
# Plotting Reported Incidents by Industry
plt.figure()
plt.bar(df['Industry'], df['Reported_Incidents'], color='skyblue')
plt.xlabel('Industry')
plt.ylabel('Reported Incidents')
plt.title('Reported Incidents by Industry')
plt.show()
```



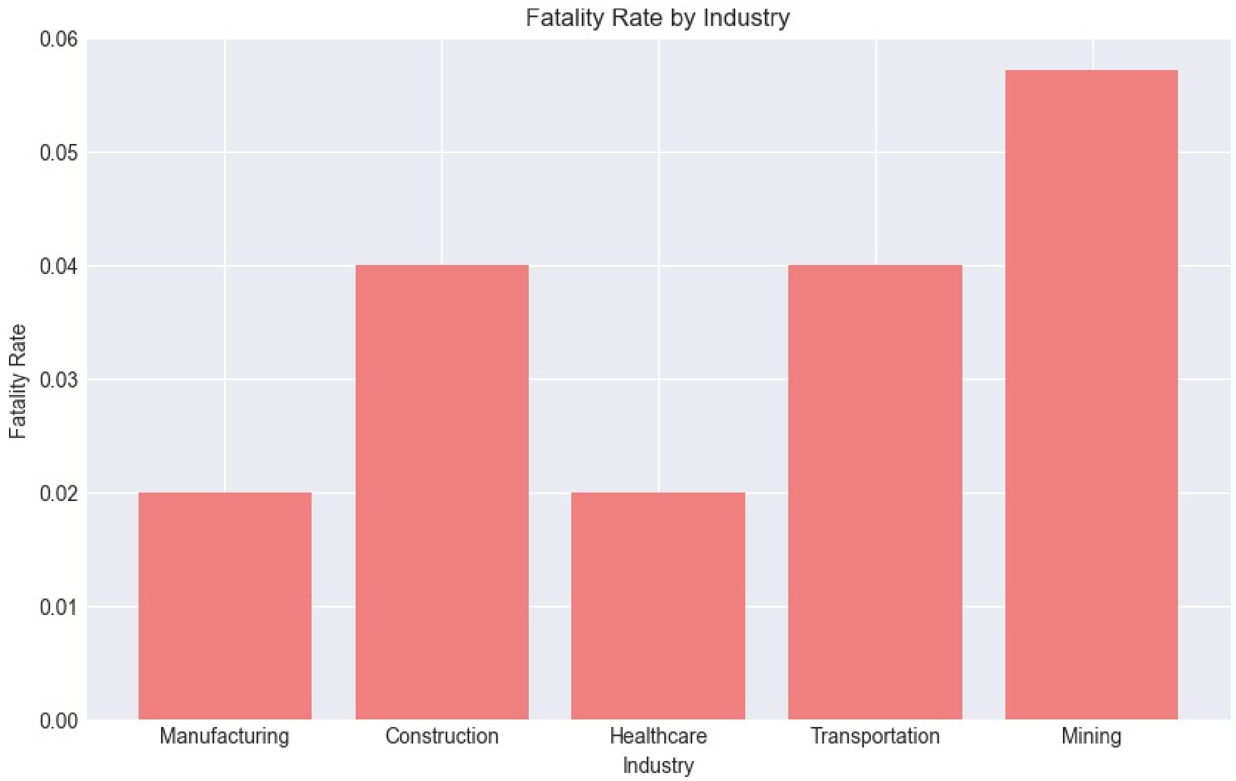
Visualization 2: Injury Severity by Industry

```
# Plotting Injury Severity by Industry
plt.figure()
plt.bar(df['Industry'], df['Injury_Severity'], color='salmon')
plt.xlabel('Industry')
plt.ylabel('Injury Severity')
plt.title('Injury Severity by Industry')
plt.show()
```



Visualization 3: Fatality Rate by Industry

```
# Plotting Fatality Rate by Industry
plt.figure()
plt.bar(df['Industry'], df['Fatality_Rate'], color='lightcoral')
plt.xlabel('Industry')
plt.ylabel('Fatality Rate')
plt.title('Fatality Rate by Industry')
plt.show()
```



Step 6: Conclusion and Insights

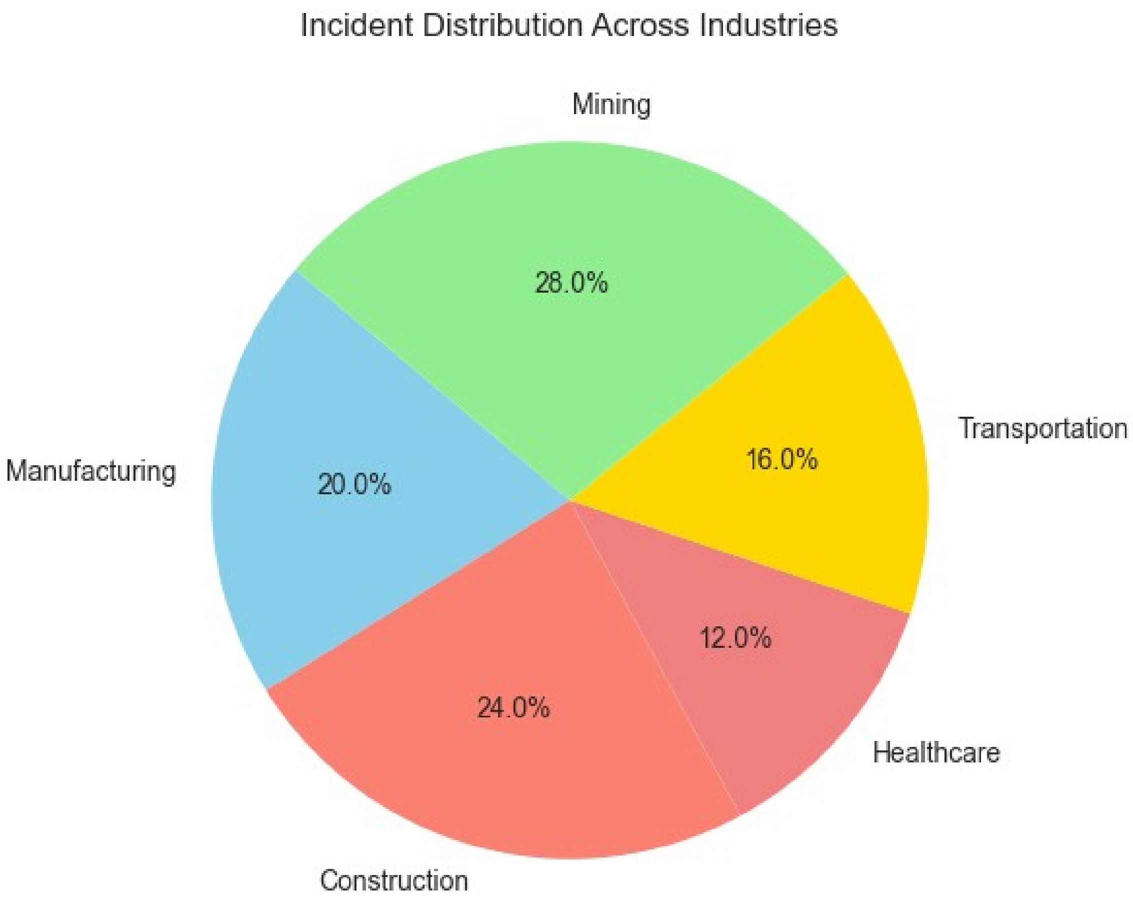
From the analysis, we observe:

- Industries like *Mining* have the highest reported incidents, injury severity, and fatality rate.
- *Construction* also shows a high level of risk, indicating the need for further safety interventions.
- These insights can help prioritize safety training and resource allocation to high-risk industries.

Visualization 4: Incident Distribution Across Industries (Pie Chart)

This visualization provides a quick overview of how incidents are distributed across different industries, helping us understand which industries contribute the most to total incidents.

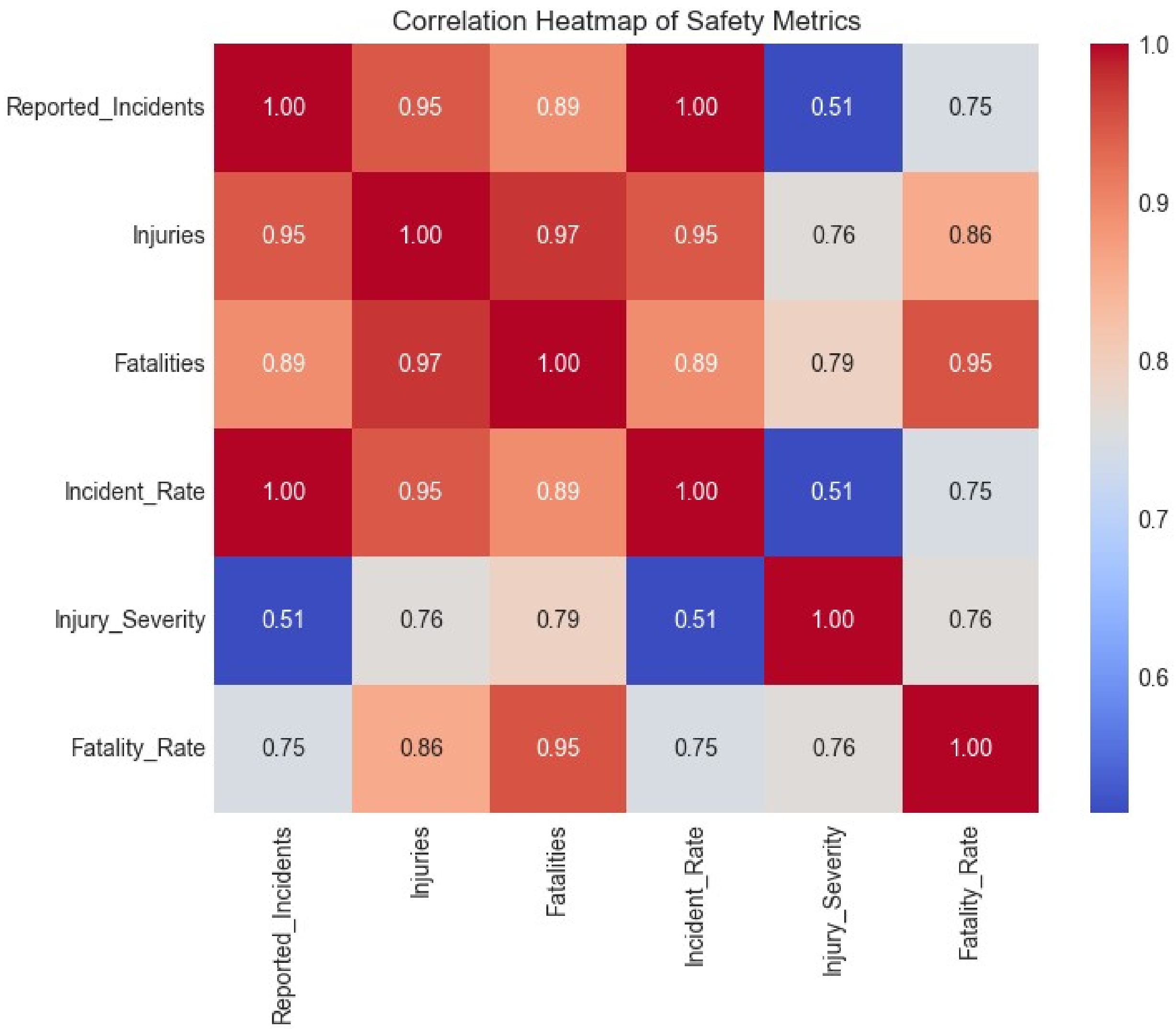
```
# Pie chart for Incident Distribution Across Industries
plt.figure()
plt.pie(df['Reported_Incidents'], labels=df['Industry'],
autopct='%1.1f%%', startangle=140, colors=['skyblue', 'salmon',
'lightcoral', 'gold', 'lightgreen'])
plt.title('Incident Distribution Across Industries')
plt.show()
```



Visualization 5: Correlation Heatmap

To identify relationships between variables, we plot a correlation heatmap. This helps us understand how metrics like incidents, injuries, and fatalities are interrelated.

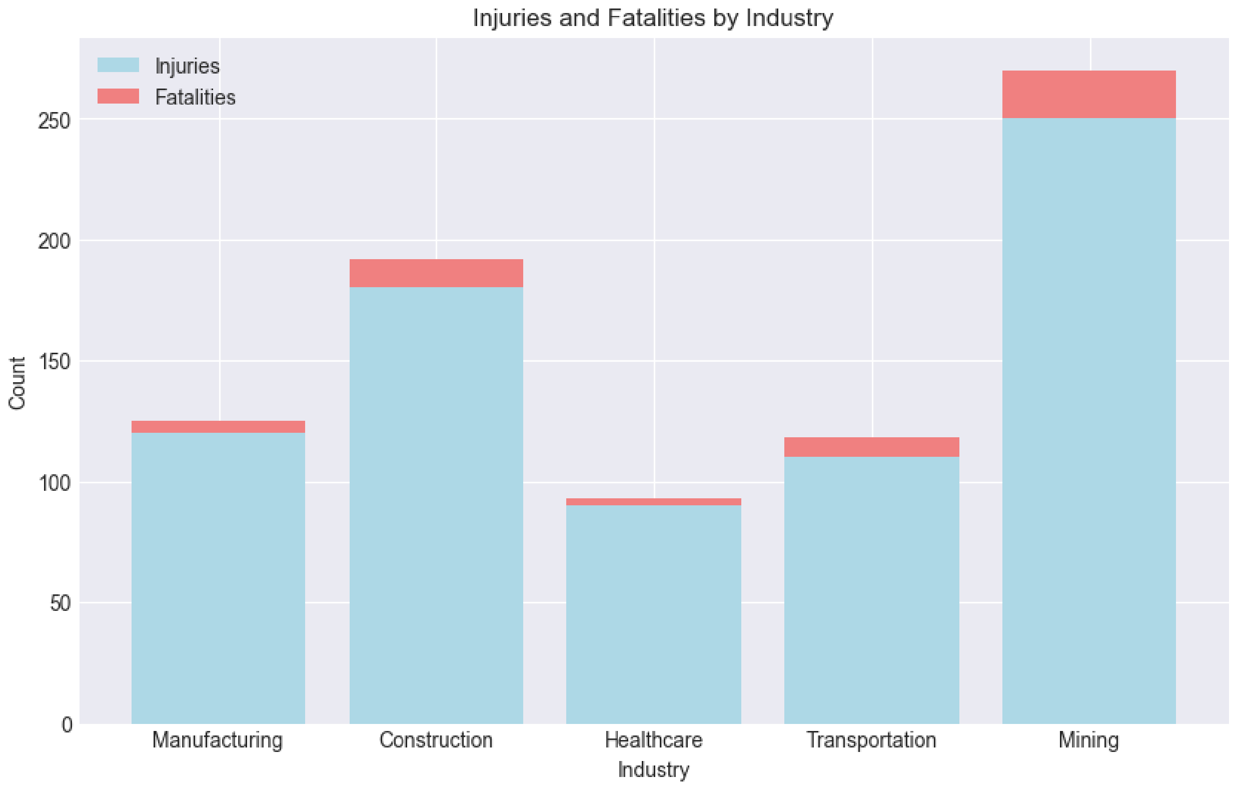
```
import seaborn as sns

# Calculating correlation matrix for numeric columns
correlation_matrix = df[['Reported_Incidents', 'Injuries',
'Fatalities', 'Incident_Rate', 'Injury_Severity',
'Fatality_Rate']].corr()

# Plotting the correlation heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
fmt='.2f')
plt.title('Correlation Heatmap of Safety Metrics')
plt.show()
```



Visualization 6: Stacked Bar Chart for Severity Metrics by Industry

To get a comprehensive view of both injuries and fatalities across industries, we can use a stacked bar chart, showing both metrics in one plot.

```
# Stacked bar chart for injuries and fatalities by industry
plt.figure()
plt.bar(df['Industry'], df['Injuries'], color='lightblue',
label='Injuries')
plt.bar(df['Industry'], df['Fatalities'], bottom=df['Injuries'],
color='lightcoral', label='Fatalities')
plt.xlabel('Industry')
plt.ylabel('Count')
plt.title('Injuries and Fatalities by Industry')
plt.legend()
plt.show()
```



Insights from Analysis

1. **Incident Distribution (Pie Chart)**: The industry with the largest share of reported incidents can quickly be identified, guiding resource allocation.
2. **Correlation Heatmap**: Positive correlations indicate that as reported incidents increase, injuries and fatalities tend to increase, suggesting safety interventions are crucial where high incidents are observed.
3. **Severity Metrics by Industry (Stacked Bar Chart)**: The stacked bar chart provides a visual comparison of both injuries and fatalities, indicating which industries might require higher preventive measures for both injury reduction and fatality prevention.

This analysis adds depth to our understanding of workplace hazards, offering insights that can be used to prioritize and tailor safety measures.