```
import pandas as pd  # For data manipulation and analysis
import numpy as np   # For numerical computations
import matplotlib.pyplot as plt  # For data visualization
import seaborn as sns  # For statistical data visualization
import plotly.express as px
```

```
# Load the dataset
df = pd.read_csv('/content/Nuclear Incidents - Sheet3.csv')
```

```
df.dtypes
```

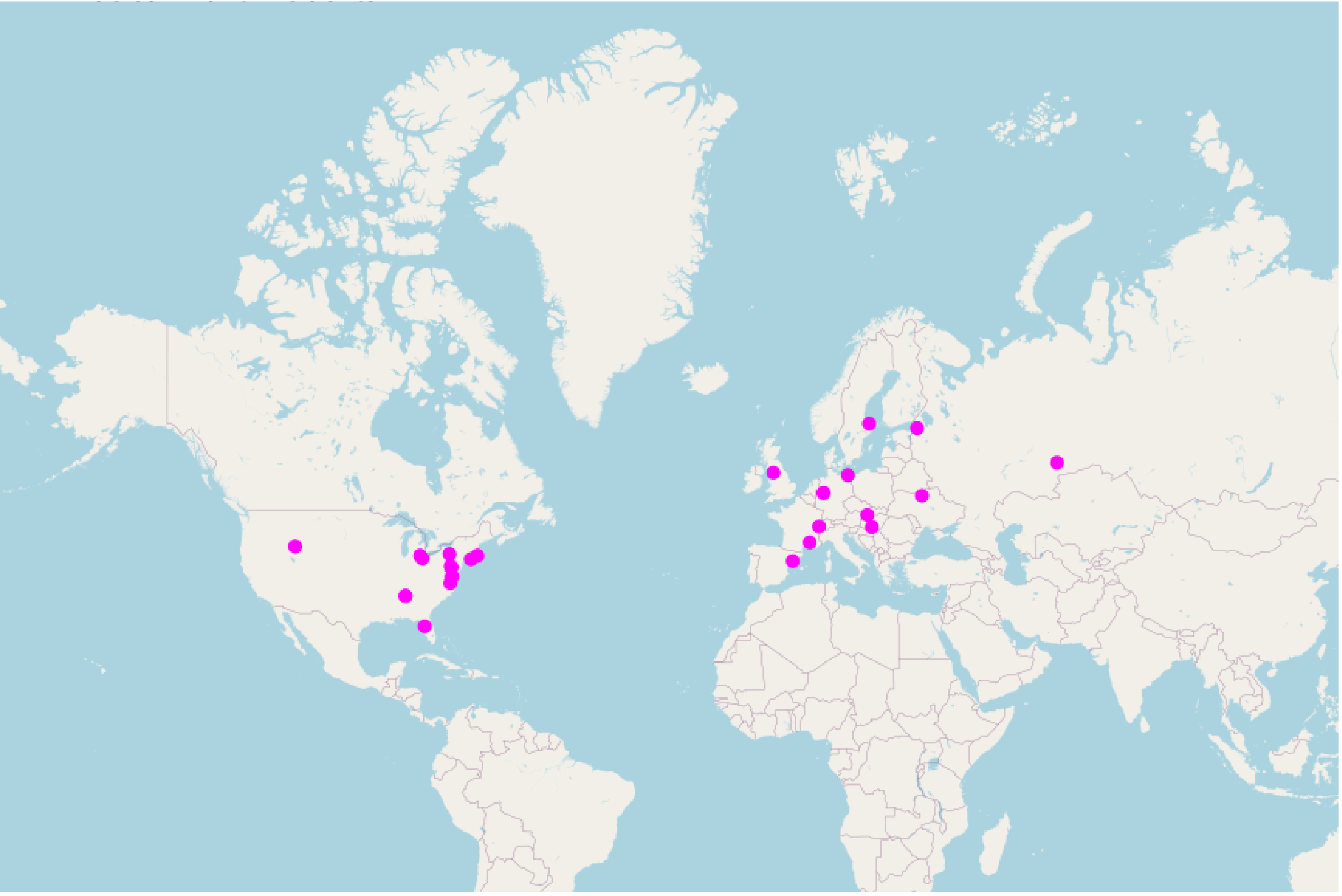
```
Location                        object
Incident                        object
Category                        object
Numbers of Direct Deaths       float64
Numbers of InDirect Deaths      object
INES\nlevel                    float64
Latitude                       float64
Longitude                      float64
dtype: object
```

```
df.rename(columns={"INES\nlevel": "INES LEVEL"},inplace = True)
df.head()
```

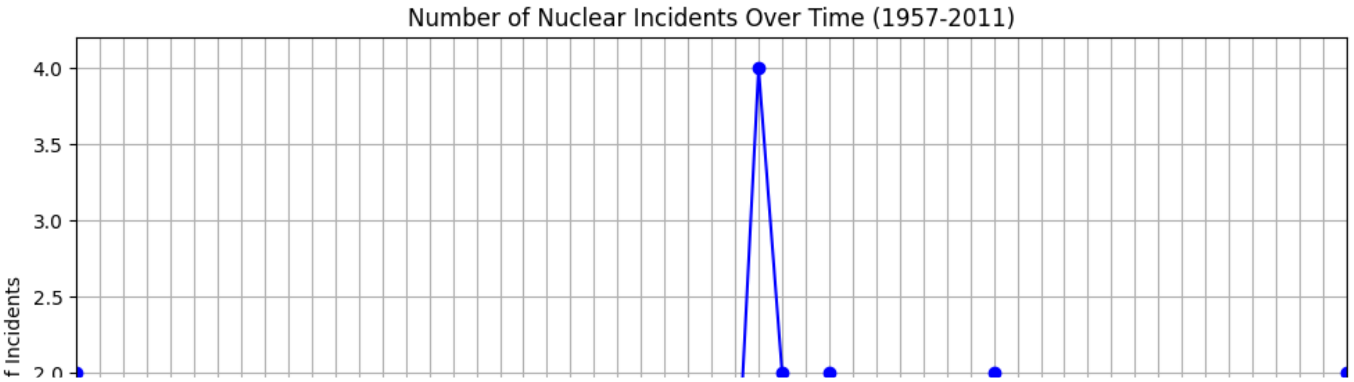
	Location	Incident	Category	Numbers of Direct Deaths	Numbers of InDirect Deaths	INES LEVEL	Latitude	Longitude
Date								
1957-09-29	Mayak, Kyshtym, Soviet Union	Kyshtym disaster	Storage/Handling	NaN	200	6.0	55.7131	60.8526
1957-10-10	Sellafield, Cumberland, United Kingdom	Windscale fire	Operational/Safety Measures	0.0	240	5.0	54.4167	-3.4833
1961-01-03	Idaho Falls, Idaho, United States	SL-1 prototype explosion	Operator Error	3.0	NaN	4.0	43.4920	-112.0401
1966-10-05	Frenchtown Charter Township, Michigan, United ...	Fermi 1 Reactor meltdown	Technical Flaws	0.0	NaN	4.0	41.9562	-83.6639
1969	Lucens reactor,	Loss-of-Coolant						

```
print('Hover on dots to see more infromation about nuclear plant incidents and scroll towards left-right to see whole
fig = px.scatter_mapbox(data_frame=df,lat="Latitude", lon="Longitude", hover_name="Location ", hover_data=["Incident"
                        color_discrete_sequence=["fuchsia"], zoom=1, height=700,title = 'Nuclear Plant Incidents')
fig.update_layout(mapbox_style="open-street-map")
fig.update_traces(marker=dict(size=10))
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```

```
Hover on dots to see more infromation about nuclear plant incidents and scroll towards left-right to see whole ma
```



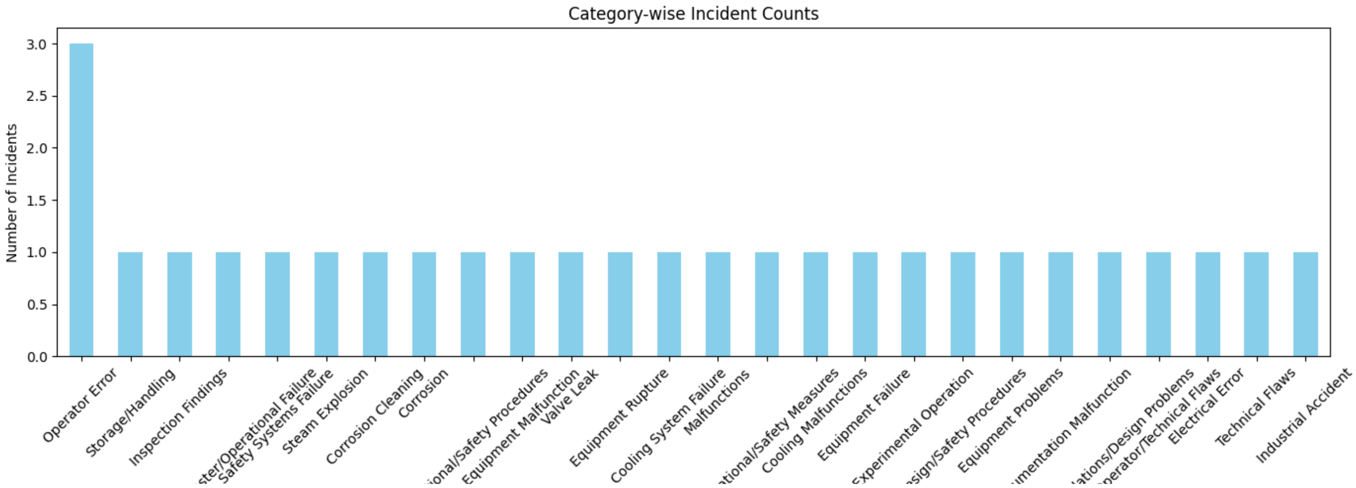
```
# Resample the data by year and count the number of incidents per year
incidents_per_year = df.resample('Y').size()

# Plotting the temporal analysis
plt.figure(figsize=(10, 6))
incidents_per_year.plot(kind='line', marker='o', color='blue')

plt.title('Number of Nuclear Incidents Over Time (1957-2011)')
plt.xlabel('Year')
plt.ylabel('Number of Incidents')
plt.grid(True)
plt.xticks(incidents_per_year.index, rotation=45)
plt.tight_layout()

plt.show()
```



```
# Group incidents by Category and count the occurrences
category_counts = df['Category'].value_counts()

# Display the counts
print("Category-wise Incident Counts:")
print(category_counts)
```

```
Category-wise Incident Counts:
Operator Error                          3
Storage/Handling                        1
Inspection Findings                     1
Natural Disaster/Operational Failure    1
Safety Systems Failure                  1
Steam Explosion                         1
Corrosion Cleaning                      1
Corrosion                               1
Operational/Safety Procedures           1
Equipment Malfunction                   1
Valve Leak                              1
Equipment Rupture                       1
Cooling System Failure                  1
Malfunctions                            1
Operational/Safety Measures             1
Cooling Malfunctions                    1
Equipment Failure                       1
Experimental Operation                  1
Design/Safety Procedures                1
Equipment Problems                      1
Instrumentation Malfunction             1
Safety Violations/Design Problems       1
Operator/Technical Flaws                1
Electrical Error                        1
Technical Flaws                         1
Industrial Accident                     1
Name: Category, dtype: int64
```

```
# Plotting the category-wise incident counts
plt.figure(figsize=(14, 6))
category_counts.plot(kind='bar', color='skyblue')

plt.title('Category-wise Incident Counts')
plt.xlabel('Category')
plt.ylabel('Number of Incidents')
plt.xticks(rotation=45)
plt.tight_layout()

plt.show()
```



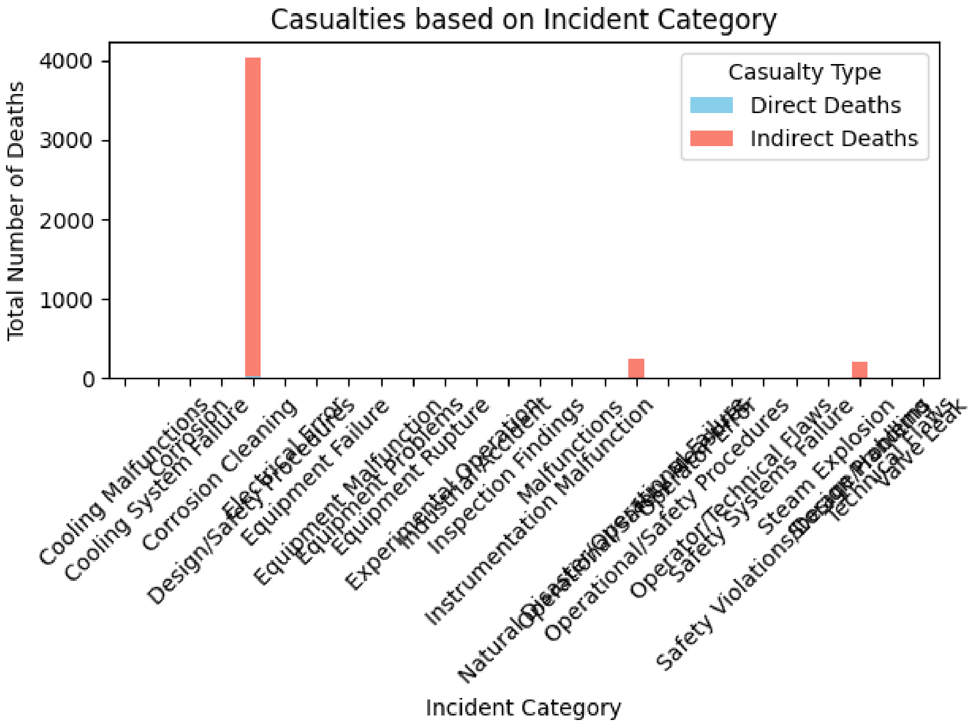
```
# Convert 'Numbers of Direct Deaths' and 'Numbers of InDirect Deaths' to numeric (some entries have commas)
df['Numbers of Direct Deaths'] = pd.to_numeric(df['Numbers of Direct Deaths'].replace(',', '', regex=True), errors='
df['Numbers of InDirect Deaths'] = pd.to_numeric(df['Numbers of InDirect Deaths'].replace(',', '', regex=True), error

# Group incidents by Category and sum the direct and indirect deaths
category_casualties = df.groupby('Category')[['Numbers of Direct Deaths', 'Numbers of InDirect Deaths']].sum()

# Display the total casualties per category
print("Casualties based on Incident Category:")
print(category_casualties)
```

```
Casualties based on Incident Category:
                                      Numbers of Direct Deaths  \
Category
Cooling Malfunctions                                       0.0
Cooling System Failure                                     0.0
Corrosion                                                  0.0
Corrosion Cleaning                                         0.0
Design/Safety Procedures                                  28.0
Electrical Error                                           0.0
Equipment Failure                                          4.0
Equipment Malfunction                                      0.0
Equipment Problems                                         0.0
Equipment Rupture                                          0.0
Experimental Operation                                     0.0
Industrial Accident                                        1.0
Inspection Findings                                        0.0
Instrumentation Malfunction                                0.0
Malfunctions                                               0.0
Natural Disaster/Operational Failure                       4.0
Operational/Safety Measures                                0.0
Operational/Safety Procedures                              2.0
Operator Error                                             5.0
Operator/Technical Flaws                                   0.0
Safety Systems Failure                                     0.0
Safety Violations/Design Problems                          0.0
Steam Explosion                                            4.0
Storage/Handling                                           0.0
Technical Flaws                                            0.0
Valve Leak                                                 0.0

                                      Numbers of InDirect Deaths
Category
Cooling Malfunctions                                         0.0
Cooling System Failure                                       0.0
Corrosion                                                    0.0
Corrosion Cleaning                                           0.0
Design/Safety Procedures                                  4000.0
Electrical Error                                             0.0
Equipment Failure                                            0.0
Equipment Malfunction                                        0.0
```

```
    Equipment Problems                                     0.0
    Equipment Rupture                                      0.0
    Experimental Operation                                 0.0
    Industrial Accident                                    0.0
    Inspection Findings                                    0.0
    Instrumentation Malfunction                            0.0
    Malfunctions                                           0.0
    Natural Disaster/Operational Failure                   0.0
    Operational/Safety Measures                          240.0
    Operational/Safety Procedures                          0.0
    Operator Error                                         0.0
    Operator/Technical Flaws                               0.0
    Safety Systems Failure                                 0.0
    Safety Violations/Design Problems                      0.0
    Steam Explosion                                        0.0
    Storage/Handling                                     200.0
    Technical Flaws                                        0.0
    Valve Leak                                             0.0
```

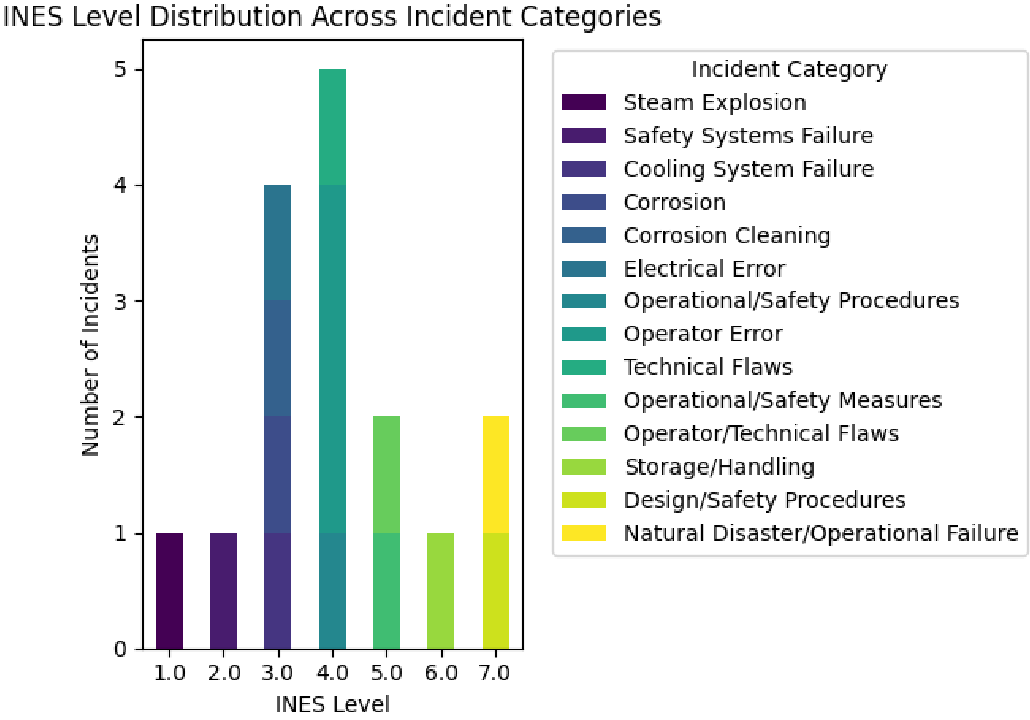
```
# Convert 'Numbers of Direct Deaths' and 'Numbers of InDirect Deaths' to numeric (some entries have commas)
df['Numbers of Direct Deaths'] = pd.to_numeric(df['Numbers of Direct Deaths'].replace(',', '', regex=True), errors='
df['Numbers of InDirect Deaths'] = pd.to_numeric(df['Numbers of InDirect Deaths'].replace(',', '', regex=True), error

# Group incidents by Category and sum the direct and indirect deaths
category_casualties = df.groupby('Category')[['Numbers of Direct Deaths', 'Numbers of InDirect Deaths']].sum()

# Plotting the casualties based on incident category
plt.figure(figsize=(14, 8))
category_casualties.plot(kind='bar', stacked=True, color=['skyblue', 'salmon'])

plt.title('Casualties based on Incident Category')
plt.xlabel('Incident Category')
plt.ylabel('Total Number of Deaths')
plt.legend(title='Casualty Type', labels=['Direct Deaths', 'Indirect Deaths'])
plt.xticks(rotation=45)
plt.tight_layout()

plt.show()
```

```
    <Figure size 1400x800 with 0 Axes>
```



```
print(df.columns)

     Index(['Location ', 'Incident', 'Category', 'Numbers of Direct Deaths',
            'Numbers of InDirect Deaths', 'INES LEVEL', 'Latitude', 'Longitude'],
           dtype='object')
```

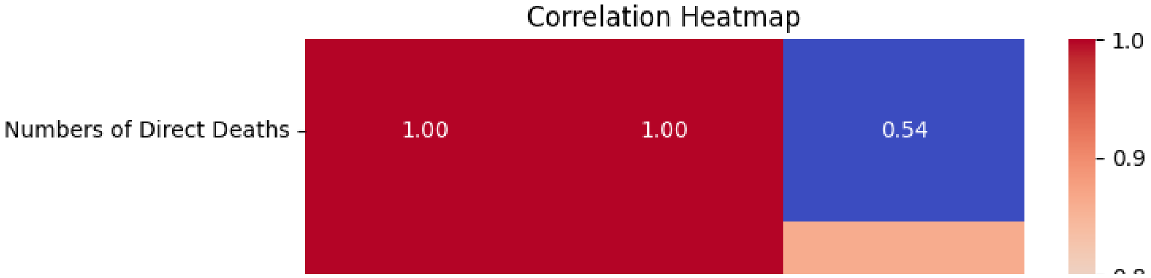
```
# Group incidents by INES level and Category, then count occurrences
ines_category_counts = df.groupby(['INES LEVEL', 'Category']).size().unstack(fill_value=0)

# Plotting INES level analysis
plt.figure(figsize=(10, 6))
ines_category_counts.plot(kind='bar', stacked=True, cmap='viridis')

plt.title('INES Level Distribution Across Incident Categories')
plt.xlabel('INES Level')
plt.ylabel('Number of Incidents')
plt.legend(title='Incident Category', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.xticks(rotation=0)
plt.tight_layout()

plt.show()

     <Figure size 1000x600 with 0 Axes>
```



```
# Convert casualty columns to numeric (some entries have commas)
df['Numbers of Direct Deaths'] = pd.to_numeric(df['Numbers of Direct Deaths'].replace(',', '', regex=True), errors='
df['Numbers of InDirect Deaths'] = pd.to_numeric(df['Numbers of InDirect Deaths'].replace(',', '', regex=True), error

# Select numerical columns for correlation analysis
numerical_columns = ['Numbers of Direct Deaths', 'Numbers of InDirect Deaths', 'INES LEVEL']

# Filter the dataframe for numerical columns
numerical_df = df[numerical_columns]

# Create a correlation matrix
correlation_matrix = numerical_df.corr()

# Display correlation matrix
print("Correlation Matrix:")
print(correlation_matrix)
```

```
Correlation Matrix:
                            Numbers of Direct Deaths  \
Numbers of Direct Deaths                    1.000000   
Numbers of InDirect Deaths                  1.000000   
INES LEVEL                                  0.536533   

                            Numbers of InDirect Deaths  INES LEVEL  
Numbers of Direct Deaths                      1.000000    0.536533  
Numbers of InDirect Deaths                    1.000000    0.861407  
INES LEVEL                                    0.861407    1.000000  
```

```
import seaborn as sns
# Plotting the correlation heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")

plt.title('Correlation Heatmap')
plt.tight_layout()

plt.show()
```


Correlation Heatmap
Numbers of Direct Deaths
1.00
1.00
0.54
1.0
0.9