Generate Simulated Data

In [ ]:

```
import numpy as np
import pandas as pd
np.random.seed(42)
```

In [ ]:

```
timeframe = pd.date_range(start="2023-01-01", end="2023-12-31", freq="D")
```

In [ ]:

```
typeofincident = np.random.choice(["Fall", "Slips", "Equipment Malfunction"], size=
```

In [ ]:

```
severity = np.random.choice(["Low", "Medium", "High"], size=len(timeframe))
```

In [ ]:

```
data = {
    "Date": timeframe,
    "Incident Type": typeofincident,
    "Severity": severity
}
```

In [ ]:

```
randomdataforworkplaceinjury = pd.DataFrame(data)
randomdataforworkplaceinjury.to_csv("accident.csv", index=False)
print("data has been save successfully")
```

Exploring the generated data

In [27]:

```
readingthedata = pd.read_csv("accident.csv")
print(readingthedata)
```

```
           Date          Incident Type Severity
0      1/1/2023  Equipment Malfunction   Medium
1      1/2/2023                   Fall   Medium
2      1/3/2023  Equipment Malfunction     High
3      1/4/2023  Equipment Malfunction   Medium
4      1/5/2023                   Fall     High
..          ...                    ...      ...
360  12/27/2023                  Slips      Low
361  12/28/2023  Equipment Malfunction      Low
362  12/29/2023                  Slips      Low
363  12/30/2023                  Slips      Low
364  12/31/2023  Equipment Malfunction   Medium

[365 rows x 3 columns]
```

In [26]:

```
print(readingthedata.head())
```

```
       Date          Incident Type Severity
0  1/1/2023  Equipment Malfunction   Medium
1  1/2/2023                   Fall   Medium
2  1/3/2023  Equipment Malfunction     High
3  1/4/2023  Equipment Malfunction   Medium
4  1/5/2023                   Fall     High
```

In [28]:

```
print(readingthedata.tail())
```

```
           Date          Incident Type Severity
360  12/27/2023                  Slips      Low
361  12/28/2023  Equipment Malfunction      Low
362  12/29/2023                  Slips      Low
363  12/30/2023                  Slips      Low
364  12/31/2023  Equipment Malfunction   Medium
```

In [29]:

```
print(readingthedata.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 3 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   Date           365 non-null    object
 1   Incident Type  365 non-null    object
 2   Severity       365 non-null    object
dtypes: object(3)
memory usage: 8.7+ KB
None
```

In [30]:

```
print(df.describe())
```

```
                      Date
count                  365
mean   2023-07-02 00:00:00
min    2023-01-01 00:00:00
25%    2023-04-02 00:00:00
50%    2023-07-02 00:00:00
75%    2023-10-01 00:00:00
max    2023-12-31 00:00:00
```

In [43]:

```
print(readingthedata[['Incident Type']]. value_counts())
```

```
Incident Type
Equipment Malfunction    127
Fall                     126
Slips                    112
Name: count, dtype: int64
```

In [47]:

```
print(readingthedata['Severity']. value_counts())
```

```
Severity
Low       139
Medium    120
High      106
Name: count, dtype: int64
```

In [49]:

```
print(readingthedata[readingthedata['Severity'] == "High"])
```

```
           Date          Incident Type Severity
2      1/3/2023  Equipment Malfunction     High
4      1/5/2023                   Fall     High
6      1/7/2023  Equipment Malfunction     High
7      1/8/2023                  Slips     High
14    1/15/2023                  Slips     High
..          ...                    ...      ...
341   12/8/2023                  Slips     High
342   12/9/2023  Equipment Malfunction     High
344  12/11/2023                   Fall     High
354  12/21/2023                   Fall     High
356  12/23/2023                   Fall     High

[106 rows x 3 columns]
```

In [50]:

```python
print(readingthedata.groupby('Incident Type').agg({'Severity': 'count'}))  # Count
```

```
                       Severity
Incident Type
Equipment Malfunction       127
Fall                        126
Slips                       112
```

Data Cleaning

In [53]:

```python
print(readingthedata.isnull().sum())
```

```
Date             0
Incident Type    0
Severity         0
dtype: int64
```

No missing value. Ready for performing data analysis

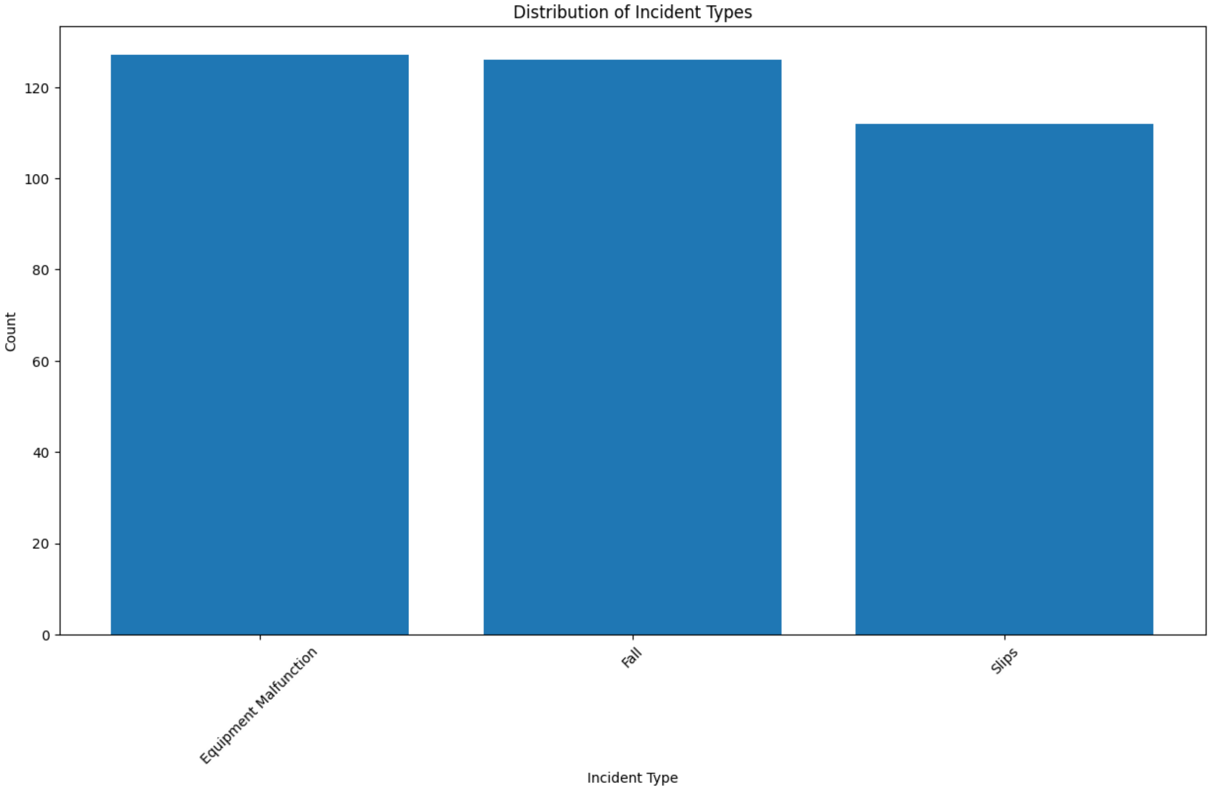
Data Visualization

In [7]:

```python
import matplotlib.pyplot as plt
import pandas as pd
readingthedata = pd.read_csv("accident.csv")
```

In [10]:

```python
incidentcounting = readingthedata['Incident Type'].value_counts()
plt.figure(figsize=(15, 8))
plt.bar(incidentcounting.index, incidentcounting.values)
plt.xlabel('Incident Type')
plt.ylabel('Count')
plt.title('Distribution of Incident Types')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.show()
```



Identifying patterns and trends

In [ ]:

```
import pandas as pd
readingthedata = pd.read_csv("accident.csv")
```

In [ ]:

```
readingthedata.dropna(inplace=True)
```

In [17]:

```
readingthedata['Date'] = pd.to_datetime(readingthedata['Date'])
print(readingthedata['Date'])
```

```
0     2023-01-01
1     2023-01-02
2     2023-01-03
3     2023-01-04
4     2023-01-05
         ...
360   2023-12-27
361   2023-12-28
362   2023-12-29
363   2023-12-30
364   2023-12-31
Name: Date, Length: 365, dtype: datetime64[ns]
```

The most common incident type

In [18]:

```
incidentcount = readingthedata.groupby('Incident Type')['Date'].count()
print(incidentcount)
```

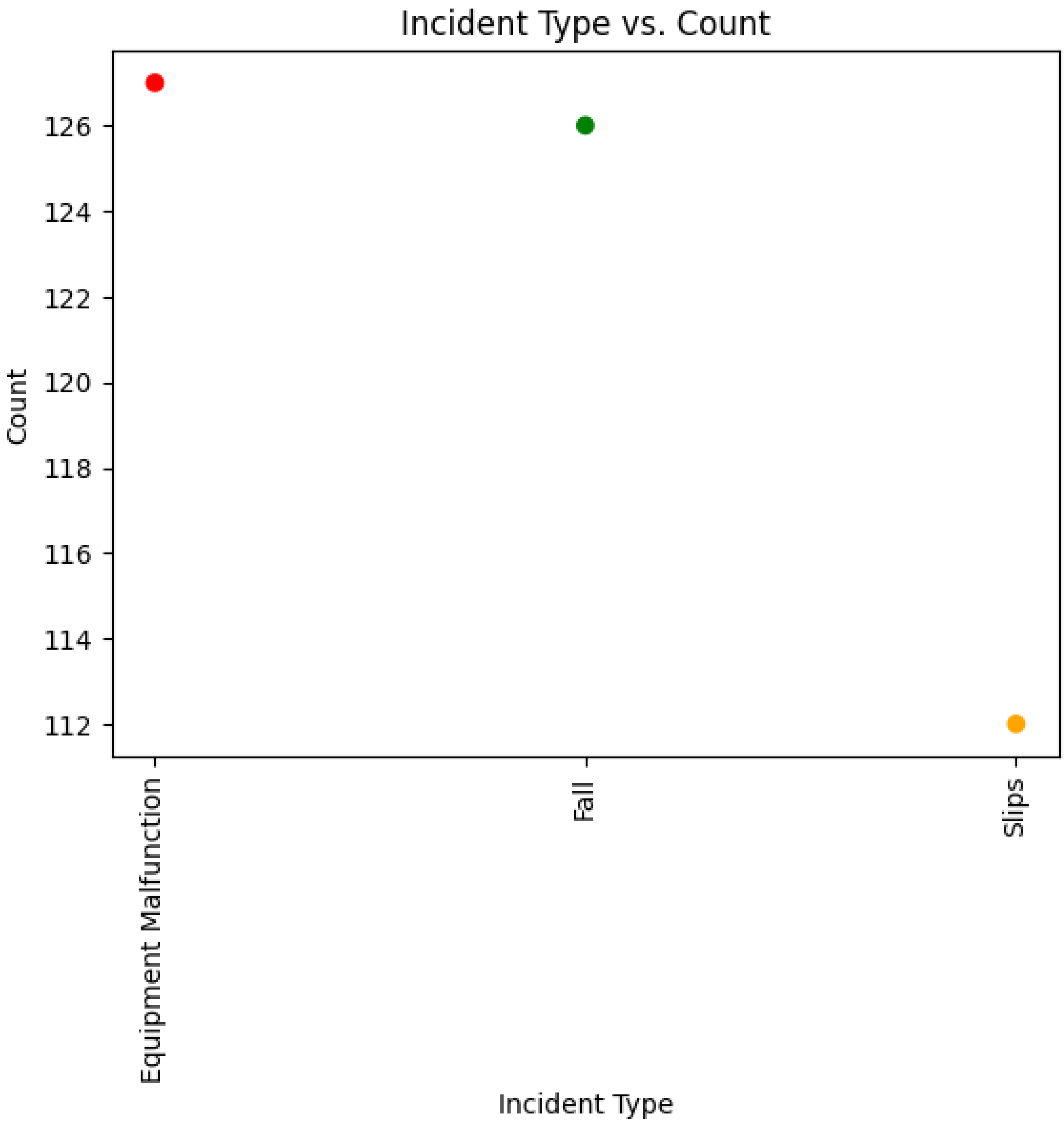
```
Incident Type
Equipment Malfunction    127
Fall                     126
Slips                    112
Name: Date, dtype: int64
```

In [54]:

```
import matplotlib.pyplot as plt
import numpy as np
incidentcount = readingthedata.groupby('Incident Type')['Date'].count()
colors = np.array(["red","green",'orange'])
plt.scatter(incidentcount.index, incidentcount.values, c = colors)
plt.xlabel('Incident Type')
plt.ylabel('Count')
plt.title('Incident Type vs. Count')


plt.xticks(rotation=90)

# Show the plot
plt.show()
```



Result: THE MOST COMMON INCIDENT TYPE

The most common incident type is "Equipment Malfunction," followed by "Fall" and "slips"

In [19]:

```
severitymean = readingthedata.groupby('Severity')['Severity'].count()
print(severitymean)
```

```
Severity
High      106
Low       139
Medium    120
Name: Severity, dtype: int64
```

In [27]:

```
highseverityincidents = readingthedata[readingthedata['Severity'] == 'High']
print(highseverityincidents)
highseverityincidents.to_csv("highseverity.csv", index=False)
print("Data has been saved successfully as highseverity.csv")
```

```
            Date          Incident Type Severity
2       1/3/2023  Equipment Malfunction     High
4       1/5/2023                   Fall     High
6       1/7/2023  Equipment Malfunction     High
7       1/8/2023                  Slips     High
14     1/15/2023                  Slips     High
..           ...                    ...      ...
341    12/8/2023                  Slips     High
342    12/9/2023  Equipment Malfunction     High
344   12/11/2023                   Fall     High
354   12/21/2023                   Fall     High
356   12/23/2023                   Fall     High

[106 rows x 3 columns]
Data has been saved successfully as highseverity.csv
```

Question: Finding out which incident type cause high severity

In [29]:

```python
import pandas as pd
readingthedata = pd.read_csv("accident.csv")
severitymapping = {"Low": 1, "Medium": 2, "High": 3}
readingthedata['Severity'] = readingthedata['Severity'].map(severitymapping)
meanseverity = readingthedata['Severity'].mean()
print("Mean Severity:", meanseverity)
```

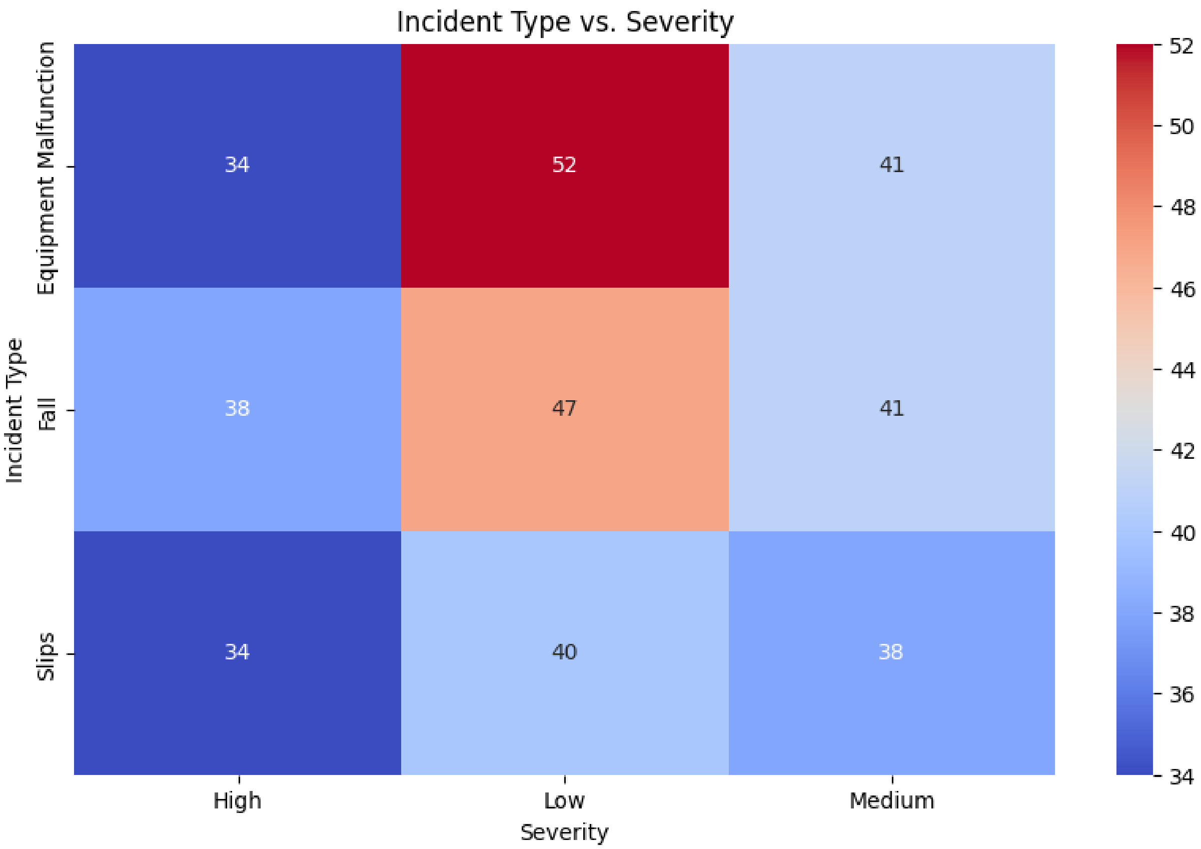
```
Mean Severity: 1.9095890410958904
```

In [ ]:

```python
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
readingthedata = pd.read_csv("accident.csv")
```

In [40]:

```python
incident = readingthedata.pivot_table(index='Incident Type', columns='Severity', va
plt.figure(figsize=(10, 6))
sns.heatmap(incident, annot=True, cmap='coolwarm')
plt.xlabel('Severity')
plt.ylabel('Incident Type')
plt.title('Incident Type vs. Severity')
plt.show()
```



Result: Incident Type "Fall" causing most of the high severity