In [230]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [231]:
```
df = pd.read_csv('AviationData.csv',encoding='latin1',low_memory=False)
```

In [232]:
```
df.head()
```

Out[232]:

	Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Country
0	20001218X45444	Accident	SEA87LA080	1948-10-24	MOOSE CREEK, ID	United States
1	20001218X45447	Accident	LAX94LA336	1962-07-19	BRIDGEPORT, CA	United States
2	20061025X01555	Accident	NYC07LA005	1974-08-30	Saltville, VA	United States
3	20001218X45448	Accident	LAX96LA321	1977-06-19	EUREKA, CA	United States
4	20041105X01764	Accident	CHI79FA064	1979-08-02	Canton, OH	United States

5 rows × 31 columns

In [233]:
```
df.tail()
```

Out[233]:

	Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Country
88884	20221227106491	Accident	ERA23LA093	2022-12-26	Annapolis, MD	United States
88885	20221227106494	Accident	ERA23LA095	2022-12-26	Hampton, NH	United States
88886	20221227106497	Accident	WPR23LA075	2022-12-26	Payson, AZ	United States
88887	20221227106498	Accident	WPR23LA076	2022-12-26	Morgan, UT	United States
88888	20221230106513	Accident	ERA23LA097	2022-12-29	Athens, GA	United States

5 rows × 31 columns

In [234]:
```
df.shape
```

Out[234]:
```
(88889, 31)
```

In [235]:
```
##Data information
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 88889 entries, 0 to 88888
Data columns (total 31 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Event.Id                88889 non-null  object 
 1   Investigation.Type      88889 non-null  object 
 2   Accident.Number         88889 non-null  object 
 3   Event.Date              88889 non-null  object 
 4   Location                88837 non-null  object 
 5   Country                 88663 non-null  object 
 6   Latitude                34382 non-null  object 
 7   Longitude               34373 non-null  object 
 8   Airport.Code            50249 non-null  object 
 9   Airport.Name            52790 non-null  object 
 10  Injury.Severity         87889 non-null  object 
 11  Aircraft.damage         85695 non-null  object 
 12  Aircraft.Category       32287 non-null  object 
 13  Registration.Number     87572 non-null  object 
 14  Make                    88826 non-null  object 
 15  Model                   88797 non-null  object 
 16  Amateur.Built           88787 non-null  object 
 17  Number.of.Engines       82805 non-null  float64
 18  Engine.Type             81812 non-null  object 
 19  FAR.Description         32023 non-null  object 
 20  Schedule                12582 non-null  object 
 21  Purpose.of.flight       82697 non-null  object 
 22  Air.carrier             16648 non-null  object 
 23  Total.Fatal.Injuries    77488 non-null  float64
 24  Total.Serious.Injuries  76379 non-null  float64
 25  Total.Minor.Injuries    76956 non-null  float64
 26  Total.Uninjured         82977 non-null  float64
 27  Weather.Condition       84397 non-null  object 
 28  Broad.phase.of.flight   61724 non-null  object 
 29  Report.Status           82508 non-null  object 
 30  Publication.Date        75118 non-null  object 
dtypes: float64(5), object(26)
memory usage: 21.0+ MB
```

In [118]:
```
#DATA CLEANING
```

In [236]:
```
##Checking for the missing data
df.isnull().sum()
```

Out[236]:
```
Event.Id                      0
Investigation.Type            0
Accident.Number               0
Event.Date                    0
Location                     52
Country                     226
Latitude                  54507
Longitude                 54516
Airport.Code              38640
Airport.Name              36099
Injury.Severity            1000
Aircraft.damage            3194
Aircraft.Category         56602
Registration.Number        1317
Make                         63
Model                        92
Amateur.Built               102
Number.of.Engines          6084
Engine.Type                7077
FAR.Description           56866
Schedule                  76307
Purpose.of.flight          6192
Air.carrier               72241
Total.Fatal.Injuries      11401
Total.Serious.Injuries    12510
Total.Minor.Injuries      11933
Total.Uninjured            5912
Weather.Condition          4492
Broad.phase.of.flight     27165
Report.Status              6381
Publication.Date          13771
dtype: int64
```

In [237]:
```
##Handling Missing Values
##Columns such as Injury.Severity, Aircraft.damage and Weather.condition a
##Drop columns that have excess missing values (>50%)
##Drop rows for the critical columns that have sparse missing data
df_cleaned = df.drop(columns=['Aircraft.Category', 'Schedule', 'Air.carrie
```

In [238]:
```
## Use mode for the categorical variables
df['Weather.Condition'].fillna(df['Weather.Condition'].mode()[0], inplace=
df['Aircraft.damage'].fillna('Unknown', inplace=True)
```

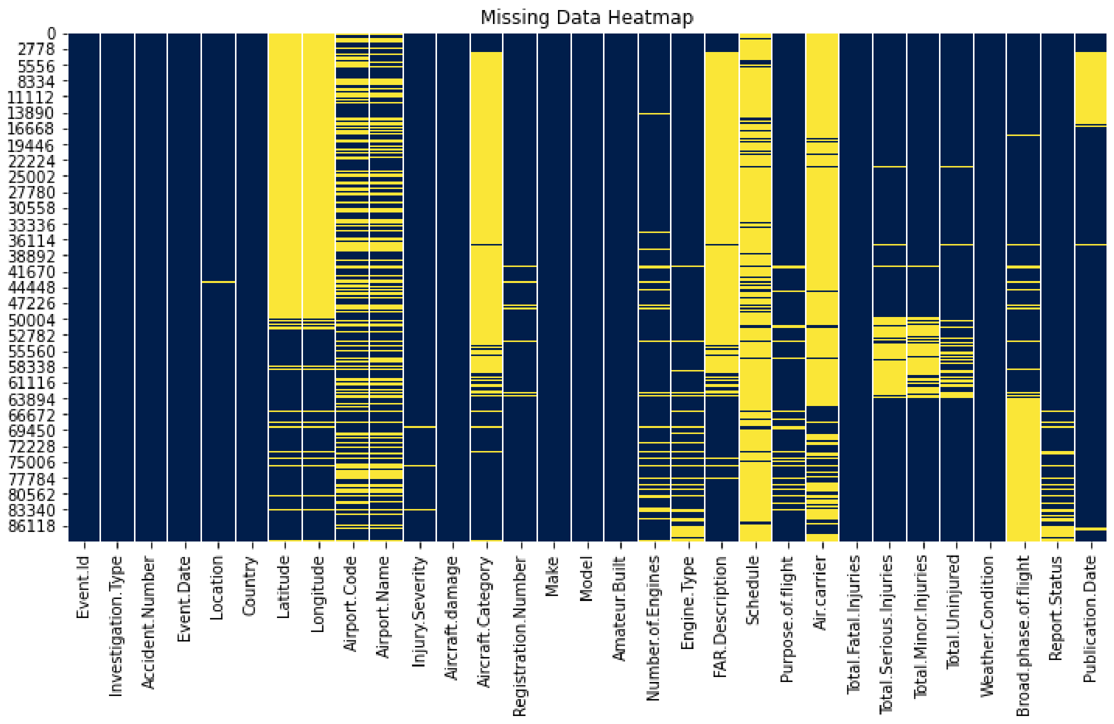
In [239]:
```
##Use Mean for the Numerical Variables
df['Total.Fatal.Injuries'].fillna(df['Total.Fatal.Injuries'].mean(), inpla
## Grouped Computation of the Numerical Variables
df['Total.Fatal.Injuries'] = df.groupby('Aircraft.damage')['Total.Fatal.Ir
    lambda x: x.fillna(x.mean()))
```

In [240]:
```
##Correlation Check
df.corr()
```

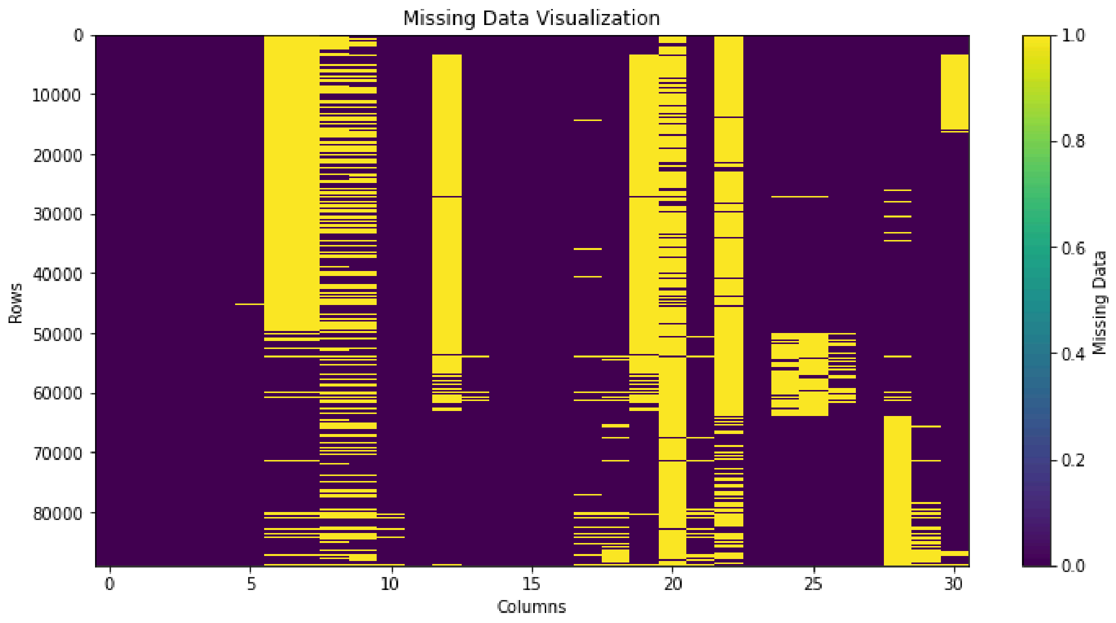
Out[240]:

	Number.of.Engines	Total.Fatal.Injuries	Total.Serious.Injuries	Total.Minc
Number.of.Engines	1.000000	0.091553	0.046157	
Total.Fatal.Injuries	0.091553	1.000000	0.135099	
Total.Serious.Injuries	0.046157	0.135099	1.000000	
Total.Minor.Injuries	0.098162	0.051716	0.326849	
Total.Uninjured	0.406058	-0.012921	0.052869	

In [242]:
```
##Missing Value Visualization
plt.figure(figsize=(12, 6))
sns.heatmap(df.isnull(), cbar=False, cmap="cividis")  # Alternative to vir
plt.title("Missing Data Heatmap")
plt.show()
```



In [243]:
```
#A heatmap to check for the colors
missing_data = df.isnull()
plt.figure(figsize=(12, 6))
plt.imshow(missing_data, aspect="auto", cmap="viridis", interpolation="nor
plt.colorbar(label="Missing Data")
plt.title("Missing Data Visualization")
plt.xlabel("Columns")
plt.ylabel("Rows")
plt.show()
```



In [248]:
```
##Check the state of the dataset after cleaning
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 88889 entries, 0 to 88888
Data columns (total 31 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Event.Id                88889 non-null  object 
 1   Investigation.Type      88889 non-null  object 
 2   Accident.Number         88889 non-null  object 
 3   Event.Date              88889 non-null  object 
 4   Location                88837 non-null  object 
 5   Country                 88663 non-null  object 
 6   Latitude                34382 non-null  object 
 7   Longitude               34373 non-null  object 
 8   Airport.Code            50249 non-null  object 
 9   Airport.Name            52790 non-null  object 
 10  Injury.Severity         87889 non-null  object 
 11  Aircraft.damage         88889 non-null  object 
 12  Aircraft.Category       32287 non-null  object 
 13  Registration.Number     87572 non-null  object 
 14  Make                    88826 non-null  object 
 15  Model                   88797 non-null  object 
 16  Amateur.Built           88787 non-null  object 
 17  Number.of.Engines       82805 non-null  float64
 18  Engine.Type             81812 non-null  object 
 19  FAR.Description         32023 non-null  object 
 20  Schedule                12582 non-null  object 
 21  Purpose.of.flight       82697 non-null  object 
 22  Air.carrier             16648 non-null  object 
 23  Total.Fatal.Injuries    88889 non-null  float64
 24  Total.Serious.Injuries  76379 non-null  float64
 25  Total.Minor.Injuries    76956 non-null  float64
 26  Total.Uninjured         82977 non-null  float64
 27  Weather.Condition       88889 non-null  object 
 28  Broad.phase.of.flight   61724 non-null  object 
 29  Report.Status           82508 non-null  object 
 30  Publication.Date        75118 non-null  object 
dtypes: float64(5), object(26)
memory usage: 21.0+ MB
```

In [245]:
```
df.describe()
```

Out[245]:

	Number.of.Engines	Total.Fatal.Injuries	Total.Serious.Injuries	Total.Minor.Injuries	Tota
count	82805.000000	88889.000000	76379.000000	76956.000000	82
mean	1.146585	0.647855	0.279881	0.357061	
std	0.446510	5.122070	1.544084	2.235625	
min	0.000000	0.000000	0.000000	0.000000	
25%	1.000000	0.000000	0.000000	0.000000	
50%	1.000000	0.000000	0.000000	0.000000	
75%	1.000000	0.647855	0.000000	0.000000	
max	8.000000	349.000000	161.000000	380.000000	

In [246]:
```
df.shape
```

Out[246]:
```
(88889, 31)
```

In [249]:
```
df.columns
```

Out[249]:
```
Index(['Event.Id', 'Investigation.Type', 'Accident.Number', 'Event.Dat
e',
       'Location', 'Country', 'Latitude', 'Longitude', 'Airport.Code',
       'Airport.Name', 'Injury.Severity', 'Aircraft.damage',
       'Aircraft.Category', 'Registration.Number', 'Make', 'Model',
       'Amateur.Built', 'Number.of.Engines', 'Engine.Type', 'FAR.Descrip
tion',
       'Schedule', 'Purpose.of.flight', 'Air.carrier', 'Total.Fatal.Inju
ries',
       'Total.Serious.Injuries', 'Total.Minor.Injuries', 'Total.Uninjure
d',
       'Weather.Condition', 'Broad.phase.of.flight', 'Report.Status',
       'Publication.Date'],
      dtype='object')
```

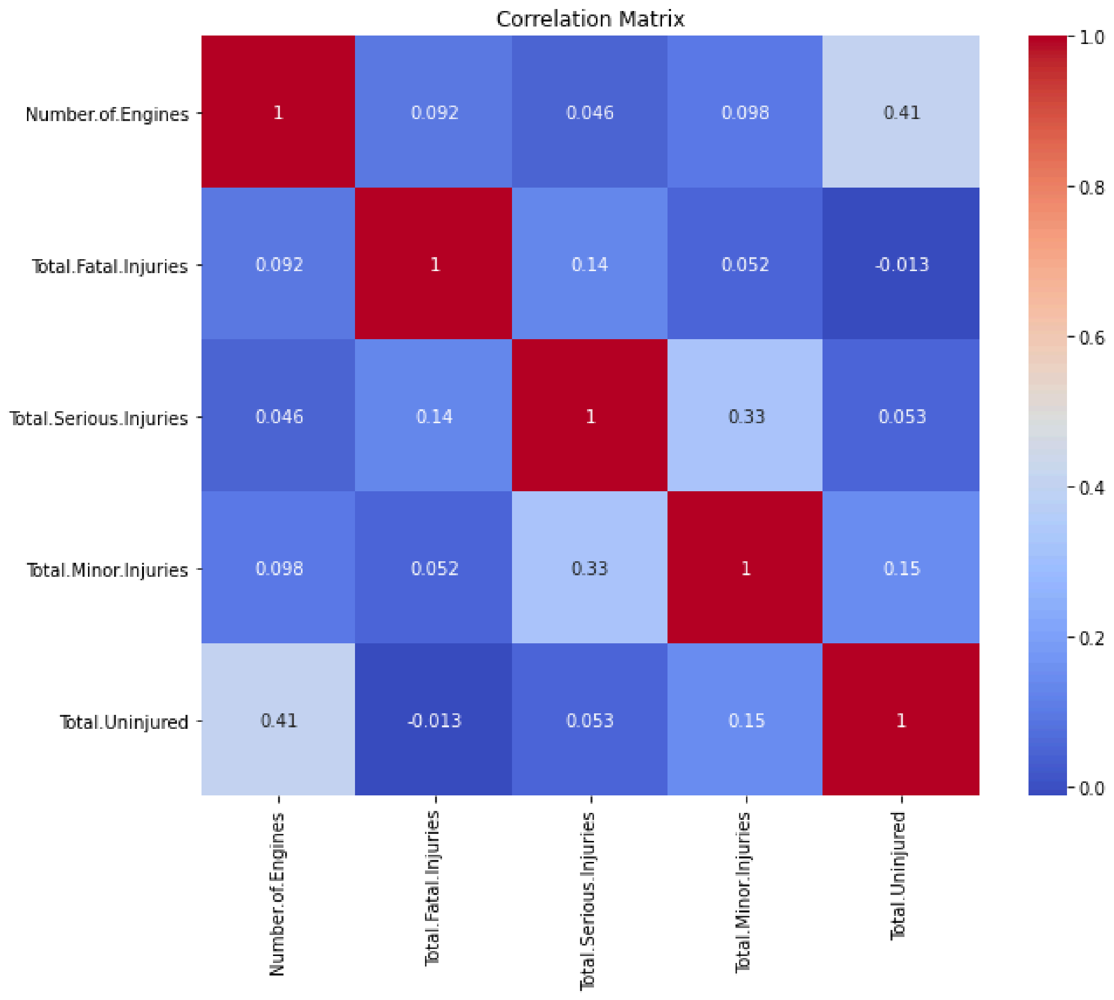
In [ ]:
```
```

In [250]:
```
#Explore the Data (EDA)
df.describe()
```

Out[250]:

	Number.of.Engines	Total.Fatal.Injuries	Total.Serious.Injuries	Total.Minor.Injuries	Tota
count	82805.000000	88889.000000	76379.000000	76956.000000	82
mean	1.146585	0.647855	0.279881	0.357061	
std	0.446510	5.122070	1.544084	2.235625	
min	0.000000	0.000000	0.000000	0.000000	
25%	1.000000	0.000000	0.000000	0.000000	
50%	1.000000	0.000000	0.000000	0.000000	
75%	1.000000	0.647855	0.000000	0.000000	
max	8.000000	349.000000	161.000000	380.000000	

In [251]:
```
#Visualize Relationships using Correlation
plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(), annot=True, cmap="coolwarm")
plt.title("Correlation Matrix")
plt.show()
```



In [253]:
```
df.columns
```

Out[253]:
```
Index(['Event.Id', 'Investigation.Type', 'Accident.Number', 'Event.Dat
e',
       'Location', 'Country', 'Latitude', 'Longitude', 'Airport.Code',
       'Airport.Name', 'Injury.Severity', 'Aircraft.damage',
       'Aircraft.Category', 'Registration.Number', 'Make', 'Model',
       'Amateur.Built', 'Number.of.Engines', 'Engine.Type', 'FAR.Descrip
tion',
       'Schedule', 'Purpose.of.flight', 'Air.carrier', 'Total.Fatal.Inju
ries',
       'Total.Serious.Injuries', 'Total.Minor.Injuries', 'Total.Uninjure
d',
       'Weather.Condition', 'Broad.phase.of.flight', 'Report.Status',
       'Publication.Date'],
      dtype='object')
```

In [252]:
```
#Dropping columns that have more than 25% of the total data missing. This
columns_to_drop = ['Latitude', 'Longitude', 'Airport.Code', 'Airport.Name'
df_clean = df.drop(columns=columns_to_drop)
```

In [255]:
```
#Checking the columns after cleaning
df_clean.columns
```

Out[255]:
```
Index(['Event.Id', 'Investigation.Type', 'Event.Date', 'Location', 'Coun
try',
       'Injury.Severity', 'Aircraft.damage', 'Make', 'Model',
       'Number.of.Engines', 'Purpose.of.flight', 'Total.Fatal.Injuries',
       'Total.Serious.Injuries', 'Total.Minor.Injuries', 'Total.Uninjure
d',
       'Weather.Condition'],
      dtype='object')
```

In [256]:
```
#Check for duplicates
df.columns.duplicated()
```

Out[256]:
```
array([False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False])
```

In [257]:
```
#Further cleaning
columns_to_drop = ['Latitude', 'Longitude', 'Airport.Code', 'Airport.Name'
                   'FAR.Description', 'Schedule', 'Air.carrier', 'Broad.ph
                   'Report.Status', 'Amateur.Built', 'Accident.Number', 'R
                   'Publication.Date', 'Engine.Type']
df_clean = df.drop(columns=columns_to_drop)
print(df_clean.columns)
```

```
Index(['Event.Id', 'Investigation.Type', 'Event.Date', 'Location', 'Coun
try',
       'Injury.Severity', 'Aircraft.damage', 'Make', 'Model',
       'Number.of.Engines', 'Purpose.of.flight', 'Total.Fatal.Injuries',
       'Total.Serious.Injuries', 'Total.Minor.Injuries', 'Total.Uninjure
d',
       'Weather.Condition'],
      dtype='object')
```

In [258]:
```
df.isnull().sum()
```

Out[258]:
```
Event.Id                      0
Investigation.Type            0
Accident.Number               0
Event.Date                    0
Location                     52
Country                     226
Latitude                  54507
Longitude                 54516
Airport.Code              38640
Airport.Name              36099
Injury.Severity            1000
Aircraft.damage               0
Aircraft.Category         56602
Registration.Number        1317
Make                         63
Model                        92
Amateur.Built               102
Number.of.Engines          6084
Engine.Type                7077
FAR.Description           56866
Schedule                  76307
Purpose.of.flight          6192
Air.carrier               72241
Total.Fatal.Injuries          0
Total.Serious.Injuries    12510
Total.Minor.Injuries      11933
Total.Uninjured            5912
Weather.Condition             0
Broad.phase.of.flight     27165
Report.Status              6381
Publication.Date          13771
dtype: int64
```

In [259]:
```
#Check the shape of the data
df_clean.shape
```

Out[259]:
```
(88889, 16)
```

In [260]:
```
df.head
```

Out[260]:
```
<bound method NDFrame.head of              Event.Id Investigation.Type A
ccident.Number  Event.Date  \
0      20001218X45444           Accident      SEA87LA080  1948-10-24
1      20001218X45447           Accident      LAX94LA336  1962-07-19
2      20061025X01555           Accident      NYC07LA005  1974-08-30
3      20001218X45448           Accident      LAX96LA321  1977-06-19
4      20041105X01764           Accident      CHI79FA064  1979-08-02
...               ...                ...             ...         ...
88884  20221227106491           Accident      ERA23LA093  2022-12-26
88885  20221227106494           Accident      ERA23LA095  2022-12-26
88886  20221227106497           Accident      WPR23LA075  2022-12-26
88887  20221227106498           Accident      WPR23LA076  2022-12-26
88888  20221230106513           Accident      ERA23LA097  2022-12-29

              Location        Country   Latitude   Longitude Airport.Cod
e  \
0      MOOSE CREEK, ID  United States        NaN         NaN           Na
N
1       BRIDGEPORT, CA  United States        NaN         NaN           Na
N
2        Saltville, VA  United States  36.922223  -81.878056           Na
N
3           EUREKA, CA  United States        NaN         NaN           Na
N
4           Canton, OH  United States        NaN         NaN           Na
N
...                ...            ...        ...         ...
...
88884    Annapolis, MD  United States        NaN         NaN           Na
N
88885      Hampton, NH  United States        NaN         NaN           Na
N
88886       Payson, AZ  United States    341525N    1112021W           PA
N
88887       Morgan, UT  United States        NaN         NaN           Na
N
88888       Athens, GA  United States        NaN         NaN           Na
N

      Airport.Name  ... Purpose.of.flight        Air.carrier  \
0              NaN  ...          Personal                NaN
1              NaN  ...          Personal                NaN
2              NaN  ...          Personal                NaN
3              NaN  ...          Personal                NaN
4              NaN  ...          Personal                NaN
...            ...  ...               ...                ...
88884          NaN  ...          Personal                NaN
88885          NaN  ...               NaN                NaN
88886       PAYSON  ...          Personal                NaN
88887          NaN  ...          Personal  MC CESSNA 210N LLC
88888          NaN  ...          Personal                NaN

      Total.Fatal.Injuries Total.Serious.Injuries Total.Minor.Injuries
\
0                      2.0                    0.0                  0.0
1                      4.0                    0.0                  0.0
2                      3.0                    NaN                  NaN
```

```
3                      2.0                    0.0                   0.0
4                      1.0                    2.0                   NaN
...                    ...                    ...                   ...
88884                  0.0                    1.0                   0.0
88885                  0.0                    0.0                   0.0
88886                  0.0                    0.0                   0.0
88887                  0.0                    0.0                   0.0
88888                  0.0                    1.0                   0.0

      Total.Uninjured Weather.Condition  Broad.phase.of.flight  \
0                 0.0               UNK                 Cruise
1                 0.0               UNK                Unknown
2                 NaN               IMC                 Cruise
3                 0.0               IMC                 Cruise
4                 0.0               VMC               Approach
...               ...               ...                    ...
88884             0.0               VMC                    NaN
88885             0.0               VMC                    NaN
88886             1.0               VMC                    NaN
88887             0.0               VMC                    NaN
88888             1.0               VMC                    NaN

        Report.Status Publication.Date
0      Probable Cause              NaN
1      Probable Cause       19-09-1996
2      Probable Cause       26-02-2007
3      Probable Cause       12-09-2000
4      Probable Cause       16-04-1980
...               ...              ...
88884             NaN       29-12-2022
88885             NaN              NaN
88886             NaN       27-12-2022
88887             NaN              NaN
88888             NaN       30-12-2022

[88889 rows x 31 columns]>
```

In [265]:

```
print(df_clean.isnull().sum())
```

```
Event.Id                      0
Investigation.Type            0
Event.Date                    0
Location                     52
Country                     226
Injury.Severity            1000
Aircraft.damage               0
Make                         63
Model                        92
Number.of.Engines          6084
Purpose.of.flight          6192
Total.Fatal.Injuries          0
Total.Serious.Injuries    12510
Total.Minor.Injuries      11933
Total.Uninjured            5912
Weather.Condition             0
dtype: int64
```

In [266]:
```
df_clean.shape
```
Out[266]:
```
(88889, 16)
```

In [267]:
```
#Replace Missing Values
categorical_columns = ['Location', 'Country', 'Aircraft.Category']
for col in categorical_columns:
    df[col].fillna(df[col].mode()[0], inplace=True)
```

In [276]:
```
numerical_columns = ['Latitude', 'Longitude', 'Total.Uninjured']
for col in numerical_columns:
    df[col] = pd.to_numeric(df[col], errors='coerce')
    if df[col].isnull().any():
        df[col].fillna(df[col].mean(), inplace=True)
```

In [270]:
```
columns_to_drop = ['FAR.Description', 'Schedule', 'Air.carrier']
df.drop(columns=columns_to_drop, inplace=True, errors='ignore')
```

In [271]:
```
columns_to_drop = ['Airport.Code', 'Airport.Name', 'Broad.phase.of.flight'
df.drop(columns=columns_to_drop, inplace=True, errors='ignore')
```

In [272]:
```
categorical_columns = ['Registration.Number', 'Make', 'Model', 'Amateur.Bu
for col in categorical_columns:
    if col in df.columns:
        df[col].fillna(df[col].mode()[0], inplace=True)
```

In [273]:
```
numerical_columns = ['Number.of.Engines', 'Engine.Type']
for col in numerical_columns:
    if col in df.columns:
        # Convert to numeric, coercing errors to NaN
        df[col] = pd.to_numeric(df[col], errors='coerce')
        # Fill missing values with the column mean
        df[col].fillna(df[col].mean(), inplace=True)
```

In [278]:
```
#Handling missing values based on column Type
df['Engine.Type'].fillna('Unknown', inplace=True)
df['Purpose.of.flight'].fillna(df['Purpose.of.flight'].mode()[0], inplace=
df['Report.Status'].fillna('Incomplete', inplace=True)
df['Injury.Severity'].fillna('Unknown', inplace=True)
df['Total.Serious.Injuries'].fillna(df['Total.Serious.Injuries'].mean(), i
df['Total.Minor.Injuries'].fillna(df['Total.Minor.Injuries'].mean(), inpla
```

In [290]:
```
df.isnull().sum()
```

Out[290]:
```
Event.Id                  0
Investigation.Type        0
Accident.Number           0
Event.Date                0
Location                  0
Country                   0
Latitude                  0
Longitude                 0
Injury.Severity           0
Aircraft.damage           0
Aircraft.Category         0
Registration.Number       0
Make                      0
Model                     0
Amateur.Built             0
Number.of.Engines         0
Engine.Type               0
Purpose.of.flight         0
Total.Fatal.Injuries      0
Total.Serious.Injuries    0
Total.Minor.Injuries      0
Total.Uninjured           0
Weather.Condition         0
Report.Status             0
dtype: int64
```

In [291]:
```
df.to_csv('cleaned_data.csv', index=False)
```

In [293]:
```
df.shape
```

Out[293]:
```
(88889, 24)
```

In [189]:
```
#Resolving Mixed Type Columns
print(df.dtypes)
```

```
Event.Id                  object
Investigation.Type        object
Accident.Number           object
Event.Date                object
Location                  object
Country                   object
Latitude                  object
Longitude                 object
Airport.Code              object
Airport.Name              object
Injury.Severity           object
Aircraft.damage           object
Aircraft.Category         object
Registration.Number       object
Make                      object
Model                     object
Amateur.Built             object
Number.of.Engines        float64
Engine.Type               object
FAR.Description           object
Schedule                  object
Purpose.of.flight         object
Air.carrier               object
Total.Fatal.Injuries     float64
Total.Serious.Injuries   float64
Total.Minor.Injuries     float64
Total.Uninjured          float64
Weather.Condition         object
Broad.phase.of.flight     object
Report.Status             object
Publication.Date          object
dtype: object
```

In [191]:
```
#Convert to a single type
df['Latitude'] = pd.to_numeric(df['Latitude'], errors='coerce')  # Convert
df['Longitude'] = pd.to_numeric(df['Longitude'], errors='coerce')
```

In [80]:
```
#The correlation between the five columns
df.corr()
```

Out[80]:

	Number.of.Engines	Total.Fatal.Injuries	Total.Serious.Injuries	Total.Minc
Number.of.Engines	1.000000	0.091553	0.046157	
Total.Fatal.Injuries	0.091553	1.000000	0.135099	
Total.Serious.Injuries	0.046157	0.135099	1.000000	
Total.Minor.Injuries	0.098162	0.051716	0.326849	
Total.Uninjured	0.406058	-0.012921	0.052869	
Event.Year	-0.018393	0.017132	0.033246	

In [284]:
```
#Summary and Exploration of Data
df.nunique()
```

Out[284]:
```
Event.Id                  87951
Investigation.Type            2
Accident.Number           88863
Event.Date                14782
Location                  27758
Country                     219
Latitude                   8879
Longitude                  9272
Injury.Severity             110
Aircraft.damage               4
Aircraft.Category            15
Registration.Number       79105
Make                       8237
Model                     12318
Amateur.Built                 2
Number.of.Engines             8
Engine.Type                   1
Purpose.of.flight            26
Total.Fatal.Injuries        126
Total.Serious.Injuries       51
Total.Minor.Injuries         58
Total.Uninjured             380
Weather.Condition             4
Report.Status             17076
dtype: int64
```

In [97]:
```
#Summary statistics for numerical columns
df.describe
```

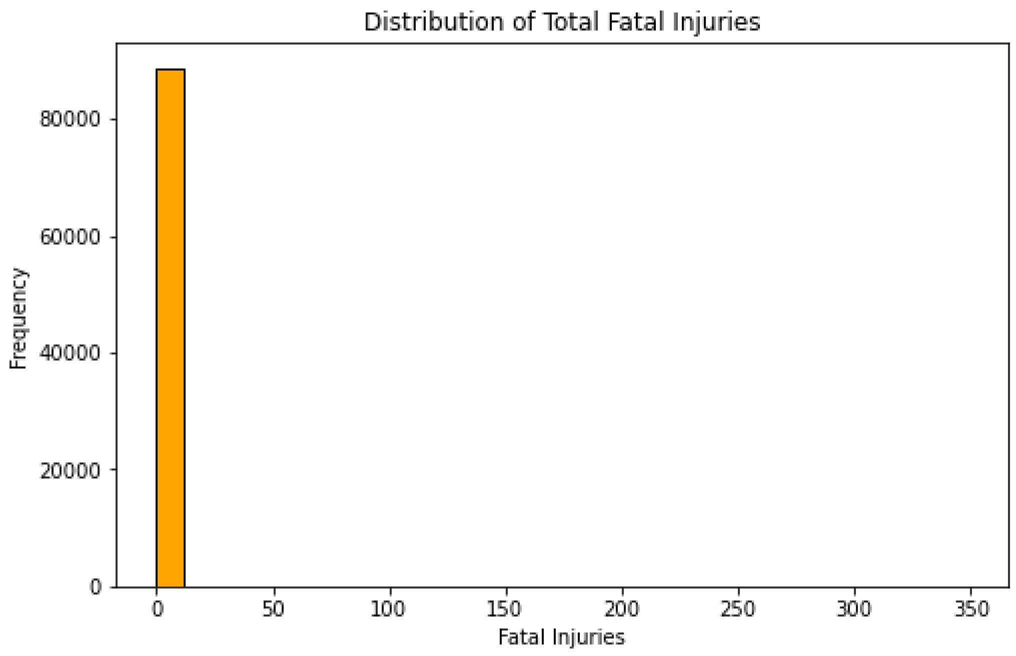
Out[97]:
```
<bound method NDFrame.describe of              Event.Id Investigation.Ty
pe Accident.Number  Event.Date  \
0      20001218X45444           Accident      SEA87LA080  1948-10-24
1      20001218X45447           Accident      LAX94LA336  1962-07-19
2      20061025X01555           Accident      NYC07LA005  1974-08-30
3      20001218X45448           Accident      LAX96LA321  1977-06-19
4      20041105X01764           Accident      CHI79FA064  1979-08-02
...               ...                ...             ...         ...
88884  20221227106491           Accident      ERA23LA093  2022-12-26
88885  20221227106494           Accident      ERA23LA095  2022-12-26
88886  20221227106497           Accident      WPR23LA075  2022-12-26
88887  20221227106498           Accident      WPR23LA076  2022-12-26
88888  20221230106513           Accident      ERA23LA097  2022-12-29

              Location        Country   Latitude   Longitude Airport.Cod
e  \
0      MOOSE CREEK, ID  United States        NaN         NaN          Na
N
1       BRIDGEPORT, CA  United States        NaN         NaN          Na
N
2        Saltville, VA  United States  36.922223  -81.878056          Na
N
3           EUREKA, CA  United States        NaN         NaN          Na
N
4           Canton, OH  United States        NaN         NaN          Na
N
...                ...            ...        ...         ...         
...
88884    Annapolis, MD  United States        NaN         NaN          Na
N
88885      Hampton, NH  United States        NaN         NaN          Na
N
88886       Payson, AZ  United States    341525N    1112021W          PA
N
88887       Morgan, UT  United States        NaN         NaN          Na
N
88888       Athens, GA  United States        NaN         NaN          Na
N

      Airport.Name  ...         Air.carrier Total.Fatal.Injuries  \
0              NaN  ...                 NaN                  2.0
1              NaN  ...                 NaN                  4.0
2              NaN  ...                 NaN                  3.0
3              NaN  ...                 NaN                  2.0
4              NaN  ...                 NaN                  1.0
...            ...  ...                 ...                  ...
88884          NaN  ...                 NaN                  0.0
88885          NaN  ...                 NaN                  0.0
88886       PAYSON  ...                 NaN                  0.0
88887          NaN  ...  MC CESSNA 210N LLC                  0.0
88888          NaN  ...                 NaN                  0.0

      Total.Serious.Injuries Total.Minor.Injuries Total.Uninjured  \
0                        0.0                  0.0             0.0
1                        0.0                  0.0             0.0
2                        NaN                  NaN             NaN
3                        0.0                  0.0             0.0
```

```
4                        2.0                    NaN              0.0
...                      ...                    ...              ...
88884                    1.0                    0.0              0.0
88885                    0.0                    0.0              0.0
88886                    0.0                    0.0              1.0
88887                    0.0                    0.0              0.0
88888                    1.0                    0.0              1.0

      Weather.Condition Broad.phase.of.flight   Report.Status  \
0                   UNK                Cruise  Probable Cause   
1                   UNK               Unknown  Probable Cause   
2                   IMC                Cruise  Probable Cause   
3                   IMC                Cruise  Probable Cause   
4                   VMC              Approach  Probable Cause   
...                 ...                   ...             ...   
88884               VMC                   NaN             NaN   
88885               VMC                   NaN             NaN   
88886               VMC                   NaN             NaN   
88887               VMC                   NaN             NaN   
88888               VMC                   NaN             NaN   

      Publication.Date Event.Year  
0                  NaN       1948  
1           19-09-1996       1962  
2           26-02-2007       1974  
3           12-09-2000       1977  
4           16-04-1980       1979  
...                ...        ...  
88884       29-12-2022       2022  
88885              NaN       2022  
88886       27-12-2022       2022  
88887              NaN       2022  
88888       30-12-2022       2022  

[88889 rows x 32 columns]>
```

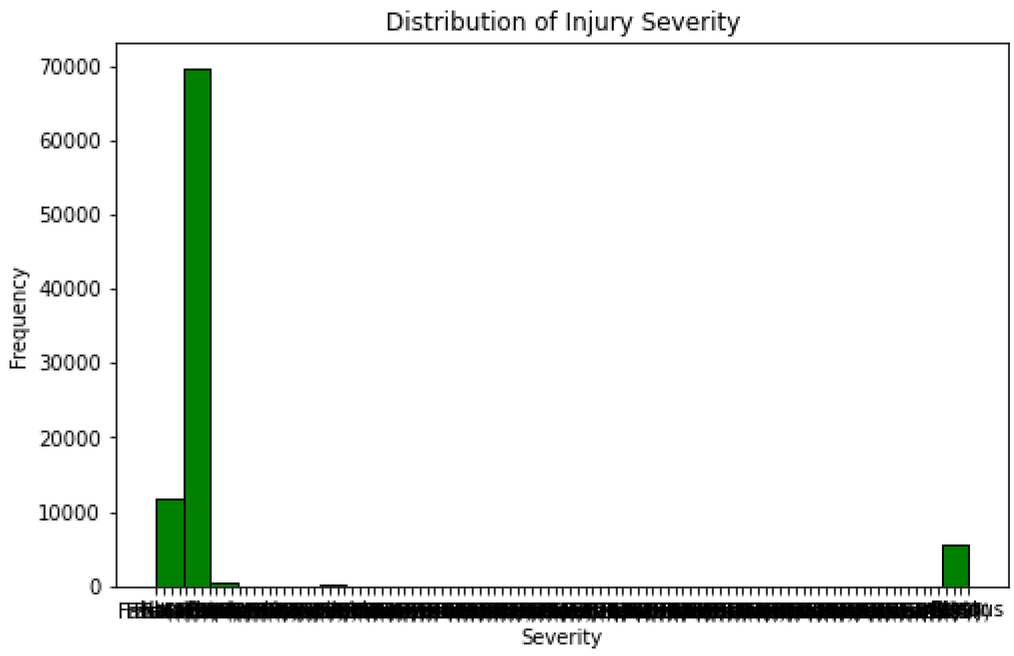
In [288]:
```
#Frequency distribution for Categorical variables
df['Weather.Condition'].value_counts()
df['Purpose.of.flight'].value_counts()
```

Out[288]:
```
Personal                     55640
Instructional                10601
Unknown                       6802
Aerial Application            4712
Business                      4018
Positioning                   1646
Other Work Use                1264
Ferry                          812
Aerial Observation             794
Public Aircraft                720
Executive/corporate            553
Flight Test                    405
Skydiving                      182
External Load                  123
Public Aircraft - Federal      105
Banner Tow                     101
Air Race show                   99
Public Aircraft - Local         74
Public Aircraft - State         64
Air Race/show                   59
Glider Tow                      53
Firefighting                    40
Air Drop                        11
ASHO                             6
PUBS                             4
PUBL                             1
Name: Purpose.of.flight, dtype: int64
```

In [58]:
```
#Histogram for Total Fatal Injuries
plt.figure(figsize=(8, 5))
df['Total.Fatal.Injuries'].hist(bins=30, color='orange', edgecolor='black'
plt.title("Distribution of Total Fatal Injuries")
plt.xlabel("Fatal Injuries")
plt.ylabel("Frequency")
plt.grid(False)
plt.show()
```

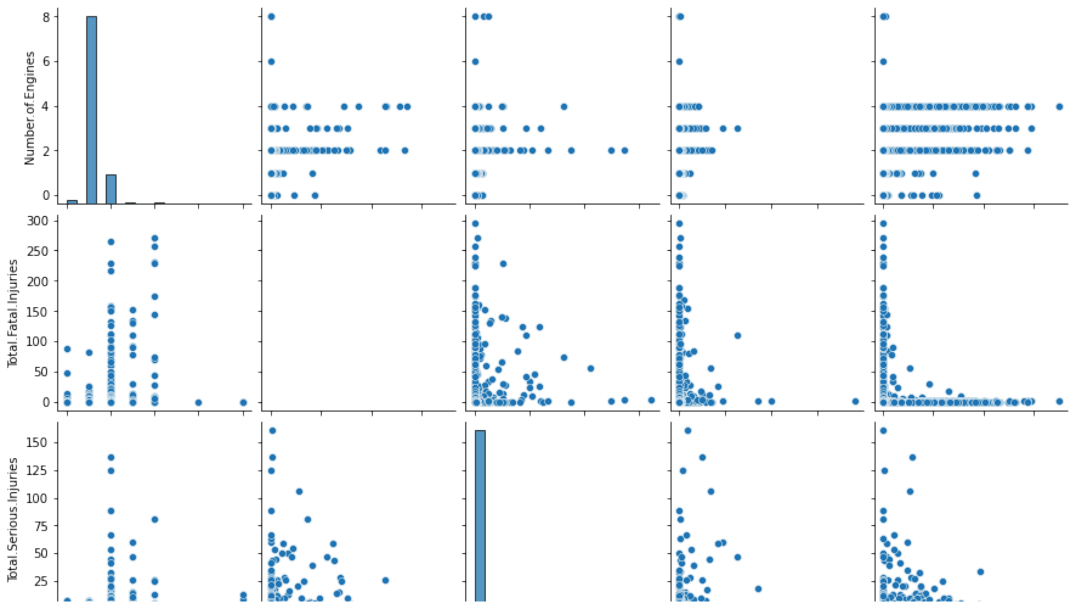


In [59]:
```
#Histogram for Injury Severity
plt.figure(figsize=(8, 5))
df['Injury.Severity'].hist(bins=30, color='green', edgecolor='black')
plt.title("Distribution of Injury Severity")
plt.xlabel("Severity")
plt.ylabel("Frequency")
plt.grid(False)
plt.show()
```

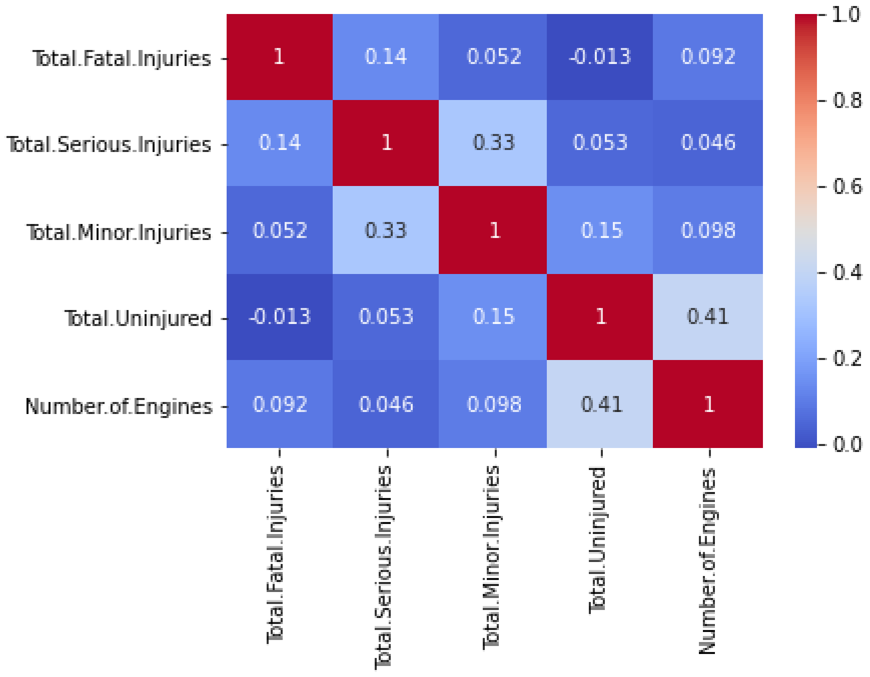


In [75]:
```
sns.pairplot(df)
```

Out[75]:
```
<seaborn.axisgrid.PairGrid at 0x22b190f35b0>
```



In [101]:
```
#Relationships between injury counts and other numerical columns
correlation_matrix = df[['Total.Fatal.Injuries', 'Total.Serious.Injuries',
                         'Total.Minor.Injuries', 'Total.Uninjured',
                         'Number.of.Engines']].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.show()
```



In [102]:
```
#Analysis of Injury Severity by Weather Conditions
df.groupby('Weather.Condition')[['Total.Fatal.Injuries', 'Total.Serious.Ir
```

Out[102]:

	Total.Fatal.Injuries	Total.Serious.Injuries
Weather.Condition		
IMC	2.020644	0.419777
UNK	2.886843	0.255405
Unk	1.244275	0.500000
VMC	0.522216	0.269129

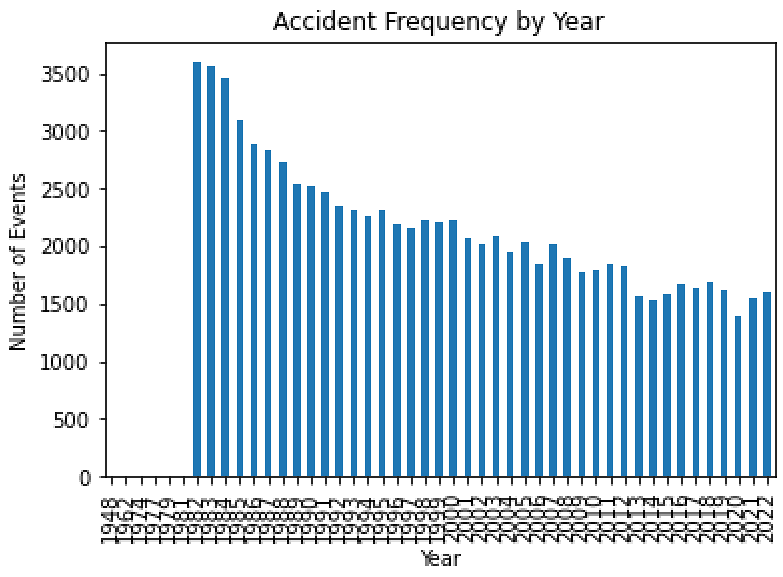
In [103]:
```
#Check if certain makes or models have higher fatality rates
df.groupby('Make')[['Total.Fatal.Injuries', 'Total.Serious.Injuries']].mea
```

Out[103]:

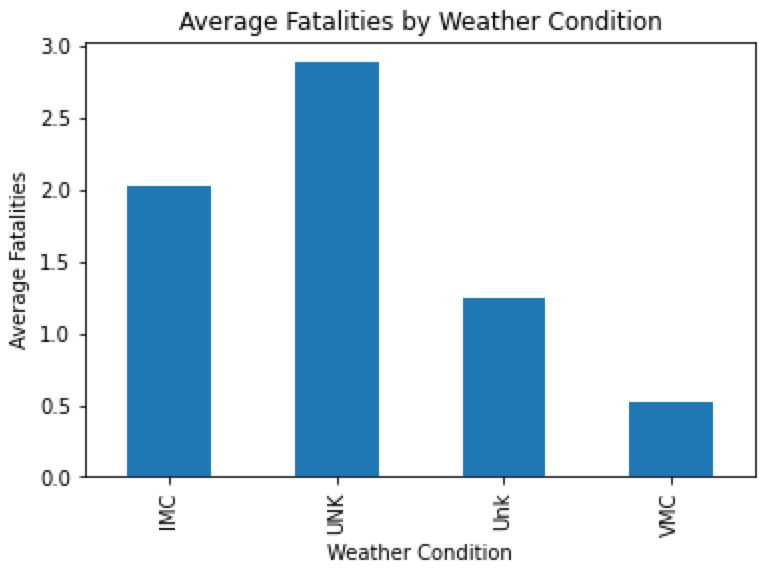
Make	Total.Fatal.Injuries	Total.Serious.Injuries
Tupolev	210.000000	NaN
TUPOLEV	44.500000	0.0
VIKING AIR LIMITED	23.000000	0.0
Aviocar CASA	18.000000	0.0
SUKHOI	14.333333	0.0
...	...	...
Honda Jet	0.000000	0.0
Honda Aircraft	0.000000	0.0
Honda	0.000000	0.0
Homer Davis	0.000000	0.0
unknown	0.000000	0.0

8237 rows × 2 columns

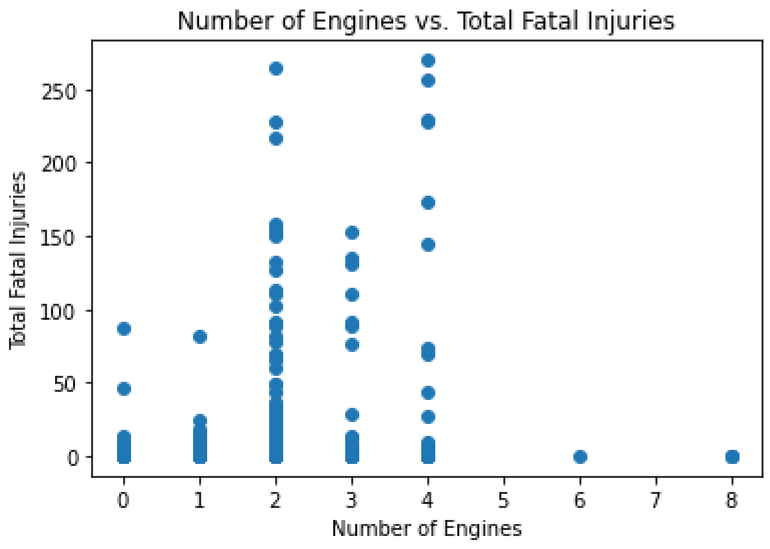
In [105]:
```
#Visualize Insights
#Histogram for Yearly Trends
df['Event.Year'].value_counts().sort_index().plot(kind='bar', title='Accic
plt.xlabel('Year')
plt.ylabel('Number of Events')
plt.show()
```



In [106]:
```
#Bar Chart for Weather Impact on Severity
df.groupby('Weather.Condition')['Total.Fatal.Injuries'].mean().plot(kind='
plt.xlabel('Weather Condition')
plt.ylabel('Average Fatalities')
plt.show()
```



In [107]:
```
#Scatter Plot for Engines vs. Fatalities
plt.scatter(df['Number.of.Engines'], df['Total.Fatal.Injuries'])
plt.title('Number of Engines vs. Total Fatal Injuries')
plt.xlabel('Number of Engines')
plt.ylabel('Total Fatal Injuries')
plt.show()
```


Number of Engines vs. Total Fatal Injuries
Total Fatal Injuries
Number of Engines