In [1]:
```
import pandas as pd
import matplotlib.pylab as plt
import seaborn as sns

import numpy as np
```

1.Data Analysis

In [2]:
```
#Opening my CSV files
df = pd.read_csv("AviationData.csv",encoding='latin1')
df1 = pd.read_csv("USState_Codes.csv")
```

```
C:\Users\mulwa\AppData\Local\Temp\ipykernel_17976\92795887.py:2: DtypeWarnin
g: Columns (6,7,28) have mixed types. Specify dtype option on import or set l
ow_memory=False.
  df = pd.read_csv("AviationData.csv",encoding='latin1')
```

In [3]:
```
df.head()
```

Out[3]:

	Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Country	L
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	36.9
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 [4]:
```
#Checking the shape of my data
df.shape
```

Out[4]: (88889, 31)

In [5]:
```
df1.shape
```

Out[5]: (62, 2)

In [6]:
```
#Checking my  df 1 data
df1.head()
```

Out[6]:

	US_State	Abbreviation
0	Alabama	AL
1	Alaska	AK
2	Arizona	AZ
3	Arkansas	AR
4	California	CA

In [7]:
```
#Checking the tail of my data i.e my df data
df.tail()
```

Out[7]:

	Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Country	Lat
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	341
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 [8]:
```
#Getting Just random samples in my df data
df.sample(5)
```

Out[8]:

	Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Country
37855	20001207X03852	Accident	ANC95LA125	1995-07-23	SHISHMAREF, AK	United States
62932	20070828X01252	Accident	DFW07CA161	2007-07-14	LEVELLAND, TX	United States
85831	20210204102598	Accident	CEN21LA123	2021-02-03	Weidman, MI	United States
35894	20001206X02023	Accident	CHI94LA302	1994-08-26	FAIRFIELD, IA	United States
37961	20001207X04185	Accident	CHI95LA260	1995-08-05	GRAND FORKS, ND	United States

5 rows × 31 columns

In [9]:
```
#Knowing the data types
df.dtypes
```

Out[9]:
```
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 [10]:
```
df.describe()
```

Out[10]:

	Number.of.Engines	Total.Fatal.Injuries	Total.Serious.Injuries	Total.Minor.Injuries	Total.Uninj
count	82805.000000	77488.000000	76379.000000	76956.000000	82977.00
mean	1.146585	0.647855	0.279881	0.357061	5.32
std	0.446510	5.485960	1.544084	2.235625	27.91
min	0.000000	0.000000	0.000000	0.000000	0.00
25%	1.000000	0.000000	0.000000	0.000000	0.00
50%	1.000000	0.000000	0.000000	0.000000	1.00
75%	1.000000	0.000000	0.000000	0.000000	2.00
max	8.000000	349.000000	161.000000	380.000000	699.00

In [11]:
```
#Getting Columns in my Data
df.columns
```

Out[11]:
```
Index(['Event.Id', 'Investigation.Type', 'Accident.Number', 'Event.Date',
       '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.Descriptio
n',
       'Schedule', 'Purpose.of.flight', 'Air.carrier', 'Total.Fatal.Injurie
s',
       'Total.Serious.Injuries', 'Total.Minor.Injuries', 'Total.Uninjured',
       'Weather.Condition', 'Broad.phase.of.flight', 'Report.Status',
       'Publication.Date'],
      dtype='object')
```

In [12]:
```
#Getting Summary of my Data Frame
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            50132 non-null  object 
 9   Airport.Name            52704 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     87507 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             81793 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           82505 non-null  object 
 30  Publication.Date        75118 non-null  object 
dtypes: float64(5), object(26)
memory usage: 21.0+ MB
```

In [13]:
```
#Getting a more concrete summary
df.info(verbose=False)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 88889 entries, 0 to 88888
Columns: 31 entries, Event.Id to Publication.Date
dtypes: float64(5), object(26)
memory usage: 21.0+ MB
```

In [14]:
```
#Getting the statics value of number
df.describe()
```

Out[14]:

	Number.of.Engines	Total.Fatal.Injuries	Total.Serious.Injuries	Total.Minor.Injuries	Total.Uninj
count	82805.000000	77488.000000	76379.000000	76956.000000	82977.00
mean	1.146585	0.647855	0.279881	0.357061	5.32
std	0.446510	5.485960	1.544084	2.235625	27.91
min	0.000000	0.000000	0.000000	0.000000	0.00
25%	1.000000	0.000000	0.000000	0.000000	0.00
50%	1.000000	0.000000	0.000000	0.000000	1.00
75%	1.000000	0.000000	0.000000	0.000000	2.00
max	8.000000	349.000000	161.000000	380.000000	699.00

In [15]:
```
#Checking Null Numbers
df.isna().sum()
```

Out[15]:
```
Event.Id                    0
Investigation.Type          0
Accident.Number             0
Event.Date                  0
Location                   52
Country                   226
Latitude                54507
Longitude               54516
Airport.Code            38757
Airport.Name            36185
Injury.Severity          1000
Aircraft.damage          3194
Aircraft.Category       56602
Registration.Number      1382
Make                       63
Model                      92
Amateur.Built             102
Number.of.Engines        6084
Engine.Type              7096
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            6384
Publication.Date        13771
dtype: int64
```

In [53]:
```
##Columns with numeric Data numerics
df.select_dtypes(include="number")
```

Out[53]:

	No_of_Engines	Major_Injuries	Minor_Injuries	Uninjured	Total_Injuries	Year	Month
0	1.0	0.0	0.0	0.0	0.0	1948	10
1	1.0	0.0	0.0	0.0	0.0	1962	7
2	1.0	NaN	NaN	NaN	0.0	1974	8
3	1.0	0.0	0.0	0.0	0.0	1977	6
4	NaN	2.0	NaN	0.0	0.0	1979	8
...	...	...	...	...	...	...	...
88884	NaN	1.0	0.0	0.0	1.0	2022	12
88885	NaN	0.0	0.0	0.0	0.0	2022	12
88886	1.0	0.0	0.0	1.0	0.0	2022	12
88887	NaN	0.0	0.0	0.0	0.0	2022	12
88888	NaN	1.0	0.0	1.0	1.0	2022	12

88889 rows × 7 columns

In [17]:
```
#Columns with object Data types
df.select_dtypes(include="object")
```

Out[17]:

	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
...	...	...	...	...	...	...
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

88889 rows × 26 columns

In [18]:
```
data = pd.read_csv("USData_cleaned.csv")
```

In [19]:
```
data.head()
```

Out[19]:

	Unnamed: 0	ID	Investigation_Type	Accident_NO	Date	Country	Injury_Severity
0	0	20061025X01555	Accident	NYC07LA005	1974-08-30	United States	Fatal(3)
1	1	20001218X45446	Accident	CHI81LA106	1981-08-01	United States	Fatal(4)
2	2	20020917X01656	Accident	ANC82FAG14	1982-01-02	United States	Fatal(3)
3	3	20020917X02481	Accident	NYC82DA016	1982-01-02	United States	Non-Fatal
4	4	20020917X01894	Accident	CHI82FEC08	1982-01-02	United States	Non-Fatal

5 rows × 29 columns

2.Data Cleaning

In [20]:
```
df.isna().sum()
```

Out[20]:
```
Event.Id                    0
Investigation.Type          0
Accident.Number             0
Event.Date                  0
Location                   52
Country                   226
Latitude                54507
Longitude               54516
Airport.Code            38757
Airport.Name            36185
Injury.Severity          1000
Aircraft.damage          3194
Aircraft.Category       56602
Registration.Number      1382
Make                       63
Model                      92
Amateur.Built             102
Number.of.Engines        6084
Engine.Type              7096
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            6384
Publication.Date        13771
dtype: int64
```

In [21]:
```
#Dropping Columns some columns and assighning them to a new dataframe
df2 = df[['Event.Id', 'Investigation.Type', 'Accident.Number', 'Event.Date',
       '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.Description',
       'Schedule',
       'Purpose.of.flight',
       # 'Air.carrier',
       # 'Total.Fatal.Injuries',
       'Total.Serious.Injuries', 'Total.Minor.Injuries', 'Total.Uninjured',
       'Weather.Condition',
       # 'Broad.phase.of.flight',
       'Report.Status',
       'Publication.Date']].copy()
```

In [22]:
```
df2.head(2)
```

Out[22]:

	Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Country	Inju
0	20001218X45444	Accident	SEA87LA080	1948-10-24	MOOSE CREEK, ID	United States	
1	20001218X45447	Accident	LAX94LA336	1962-07-19	BRIDGEPORT, CA	United States	

2 rows × 22 columns

In [23]:
```
#Checking the unique values in Investigation type
df['Injury.Severity'].value_counts().head(10)
```

Out[23]:
```
Injury.Severity
Non-Fatal    67357
Fatal(1)      6167
Fatal         5262
Fatal(2)      3711
Incident      2219
Fatal(3)      1147
Fatal(4)       812
Fatal(5)       235
Minor          218
Serious        173
Name: count, dtype: int64
```

In [24]:
```
df2.columns
```

Out[24]:
```
Index(['Event.Id', 'Investigation.Type', 'Accident.Number', 'Event.Date',
       'Location', 'Country', 'Injury.Severity', 'Aircraft.damage',
       'Registration.Number', 'Make', 'Model', 'Amateur.Built',
       'Number.of.Engines', 'Engine.Type', 'Schedule', 'Purpose.of.flight',
       'Total.Serious.Injuries', 'Total.Minor.Injuries', 'Total.Uninjured',
       'Weather.Condition', 'Report.Status', 'Publication.Date'],
      dtype='object')
```

In [25]:
```
#Renaming the columns
df = df2.rename(columns = {
    'Event.Id': 'ID',
    'Investigation.Type':'Investigation_Type',
    'Accident.Number':'Accident_NO',
    'Event.Date':'Date',
    'Injury.Severity':'Injury_Severity',
    'Aircraft.damage':'Damage',
    'Registration.Number':'Reg_Number',
    'Amateur.Built':'Amateur_Built',
    'Number.of.Engines':'No_of_Engines',
    'Engine.Type':'Engine_Type',
    'Purpose.of.flight':'Purpose_of_flight',
    'Total.Serious.Injuries':'Major_Injuries',
    'Total.Minor.Injuries':'Minor_Injuries',
    'Total.Uninjured':'Uninjured',
    'Report.Status':'Report_Status',
    'Weather.Condition':'Weather_Condition',
    'Publication.Date':'Publication_Date'
}).copy()
```

In [26]:
```
df.columns
```

Out[26]:
```
Index(['ID', 'Investigation_Type', 'Accident_NO', 'Date', 'Location',
       'Country', 'Injury_Severity', 'Damage', 'Reg_Number', 'Make', 'Model',
       'Amateur_Built', 'No_of_Engines', 'Engine_Type', 'Schedule',
       'Purpose_of_flight', 'Major_Injuries', 'Minor_Injuries', 'Uninjured',
       'Weather_Condition', 'Report_Status', 'Publication_Date'],
      dtype='object')
```

In [27]:
```
df.head()
```

Out[27]:

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

5 rows × 22 columns

In [28]:
```
df.isna().sum()
```

Out[28]:
```
ID                      0
Investigation_Type      0
Accident_NO             0
Date                    0
Location               52
Country               226
Injury_Severity      1000
Damage               3194
Reg_Number           1382
Make                   63
Model                  92
Amateur_Built         102
No_of_Engines        6084
Engine_Type          7096
Schedule            76307
Purpose_of_flight    6192
Major_Injuries      12510
Minor_Injuries      11933
Uninjured            5912
Weather_Condition    4492
Report_Status        6384
Publication_Date    13771
dtype: int64
```

In [29]:
```
#Get Total Injuries Column
df['Total_Injuries'] = df['Minor_Injuries'] + df['Major_Injuries']
```

In [30]:
```
#Checking Duplicates
duplicates = df[df.duplicated()]
print(len(duplicates))
```

```
0
```

In [31]:
```
df.dtypes
```

Out[31]:
```
ID                    object
Investigation_Type    object
Accident_NO           object
Date                  object
Location              object
Country               object
Injury_Severity       object
Damage                object
Reg_Number            object
Make                  object
Model                 object
Amateur_Built         object
No_of_Engines        float64
Engine_Type           object
Schedule              object
Purpose_of_flight     object
Major_Injuries       float64
Minor_Injuries       float64
Uninjured            float64
Weather_Condition     object
Report_Status         object
Publication_Date      object
Total_Injuries       float64
dtype: object
```

In [32]:
```
#Change date to date
df['Date']= pd.to_datetime(df['Date'])
```

In [33]:
```
#Extract In Date the yaers and month
df['Year'] = df['Date'].dt.year
df['Month'] = df['Date'].dt.month
```

In [34]:
```
df.dtypes
```

Out[34]:
```
ID                          object
Investigation_Type          object
Accident_NO                 object
Date                datetime64[ns]
Location                    object
Country                     object
Injury_Severity             object
Damage                      object
Reg_Number                  object
Make                        object
Model                       object
Amateur_Built               object
No_of_Engines              float64
Engine_Type                 object
Schedule                    object
Purpose_of_flight           object
Major_Injuries             float64
Minor_Injuries             float64
Uninjured                  float64
Weather_Condition           object
Report_Status               object
Publication_Date            object
Total_Injuries             float64
Year                         int32
Month                        int32
dtype: object
```

In [35]:
```
#To check for missing values in my date column
missing = df.isnull().sum()
missing
```

Out[35]:
```
ID                      0
Investigation_Type      0
Accident_NO             0
Date                    0
Location               52
Country               226
Injury_Severity      1000
Damage               3194
Reg_Number           1382
Make                   63
Model                  92
Amateur_Built         102
No_of_Engines        6084
Engine_Type          7096
Schedule            76307
Purpose_of_flight    6192
Major_Injuries      12510
Minor_Injuries      11933
Uninjured            5912
Weather_Condition    4492
Report_Status        6384
Publication_Date    13771
Total_Injuries      14053
Year                    0
Month                   0
dtype: int64
```

In [36]:
```
#In the missing values in the columns add some values to make sense
#With location write un known in the null locations

df['Location'] = df['Location'].fillna('Unknown')
df['Country'] = df['Country'].fillna('Unknown')
df['Injury_Severity'] = df['Injury_Severity'].fillna('None')

#Drop Schedule Column
# df = df.drop(columns=['Schedule'])
```

In [37]:
```
#Replace NAN with 0.0 in Total Injuries
df['Total_Injuries'] =df['Total_Injuries'].fillna(0.0)
```

In [38]:
```
df["Report_Status"] .unique()
```

Out[38]:
```
array(['Probable Cause', 'Factual', 'Foreign', ...,
       'The pilot did not ensure adequate clearance from construction vehicle
s during taxi.',
       'The pilot\x92s failure to secure the magneto switch before attempting
to hand rotate the engine which resulted in an inadvertent engine start, a ru
naway airplane, and subsequent impact with parked airplanes. Contributing to
the accident was the failure to properly secure the airplane with chocks.',
       'The pilot\x92s loss of control due to a wind gust during landing.'],
      dtype=object)
```

In [39]:
```
missing = df.isnull().sum()
missing
```

Out[39]:
```
ID                      0
Investigation_Type      0
Accident_NO             0
Date                    0
Location                0
Country                 0
Injury_Severity         0
Damage               3194
Reg_Number           1382
Make                   63
Model                  92
Amateur_Built         102
No_of_Engines        6084
Engine_Type          7096
Schedule            76307
Purpose_of_flight     6192
Major_Injuries       12510
Minor_Injuries       11933
Uninjured             5912
Weather_Condition     4492
Report_Status         6384
Publication_Date     13771
Total_Injuries          0
Year                    0
Month                   0
dtype: int64
```

In [40]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 88889 entries, 0 to 88888
Data columns (total 25 columns):
 #   Column              Non-Null Count  Dtype         
---  ------              --------------  -----         
 0   ID                  88889 non-null  object        
 1   Investigation_Type  88889 non-null  object        
 2   Accident_NO         88889 non-null  object        
 3   Date                88889 non-null  datetime64[ns]
 4   Location            88889 non-null  object        
 5   Country             88889 non-null  object        
 6   Injury_Severity     88889 non-null  object        
 7   Damage              85695 non-null  object        
 8   Reg_Number          87507 non-null  object        
 9   Make                88826 non-null  object        
 10  Model               88797 non-null  object        
 11  Amateur_Built       88787 non-null  object        
 12  No_of_Engines       82805 non-null  float64       
 13  Engine_Type         81793 non-null  object        
 14  Schedule            12582 non-null  object        
 15  Purpose_of_flight   82697 non-null  object        
 16  Major_Injuries      76379 non-null  float64       
 17  Minor_Injuries      76956 non-null  float64       
 18  Uninjured           82977 non-null  float64       
 19  Weather_Condition   84397 non-null  object        
 20  Report_Status       82505 non-null  object        
 21  Publication_Date    75118 non-null  object        
 22  Total_Injuries      88889 non-null  float64       
 23  Year                88889 non-null  int32         
 24  Month               88889 non-null  int32         
dtypes: datetime64[ns](1), float64(5), int32(2), object(17)
memory usage: 16.3+ MB
```

In [41]:
```
df['Country'].value_counts()
```

Out[41]:
```
Country
United States                        82248
Brazil                                 374
Canada                                 359
Mexico                                 358
United Kingdom                         344
                                     ...  
Saint Vincent and the Grenadines         1
Cambodia                                 1
Malampa                                  1
AY                                       1
Turks and Caicos Islands                 1
Name: count, Length: 219, dtype: int64
```

In [42]:
```
df.head()
```

Out[42]:

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

5 rows × 25 columns

In [43]:
```
#Cleaning my USStates.csv
df1.head(20)
```

Out[43]:

	US_State	Abbreviation
0	Alabama	AL
1	Alaska	AK
2	Arizona	AZ
3	Arkansas	AR
4	California	CA
5	Colorado	CO
6	Connecticut	CT
7	Delaware	DE
8	Florida	FL
9	Georgia	GA
10	Hawaii	HI
11	Idaho	ID
12	Illinois	IL
13	Indiana	IN
14	Iowa	IA
15	Kansas	KS
16	Kentucky	KY
17	Louisiana	LA
18	Maine	ME
19	Maryland	MD

In [44]:
```
#Have the only US-States country in its own variable
USData = df[df["Country"] == 'United States']
USData.head(5)
```

Out[44]:

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

5 rows × 25 columns

In [45]:
```
USData.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 82248 entries, 0 to 88888
Data columns (total 25 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   ID                  82248 non-null  object
 1   Investigation_Type  82248 non-null  object
 2   Accident_NO         82248 non-null  object
 3   Date                82248 non-null  datetime64[ns]
 4   Location            82248 non-null  object
 5   Country             82248 non-null  object
 6   Injury_Severity     82248 non-null  object
 7   Damage              80269 non-null  object
 8   Reg_Number          82132 non-null  object
 9   Make                82227 non-null  object
 10  Model               82210 non-null  object
 11  Amateur_Built       82227 non-null  object
 12  No_of_Engines       80373 non-null  float64
 13  Engine_Type         79206 non-null  object
 14  Schedule            10297 non-null  object
 15  Purpose_of_flight   79819 non-null  object
 16  Major_Injuries      70873 non-null  float64
 17  Minor_Injuries      71519 non-null  float64
 18  Uninjured           77243 non-null  float64
 19  Weather_Condition   81603 non-null  object
 20  Report_Status       79637 non-null  object
 21  Publication_Date    69567 non-null  object
 22  Total_Injuries      82248 non-null  float64
 23  Year                82248 non-null  int32
 24  Month               82248 non-null  int32
dtypes: datetime64[ns](1), float64(5), int32(2), object(17)
memory usage: 15.7+ MB
```

In [46]:
```
#Cleaning Number of Engines
print(USData['No_of_Engines'].unique())
```

```
[ 1. nan  2.  0.  3.  4.  8.  6.]
```

In [47]:
```
data.columns
```

Out[47]:
```
Index(['Unnamed: 0', 'ID', 'Investigation_Type', 'Accident_NO', 'Date',
       'Country', 'Injury_Severity', 'Damage', 'Reg_Number', 'Make', 'Model',
       'Amateur_Built', 'No_of_Engines', 'Engine_Type', 'Schedule',
       'Purpose_of_flight', 'Major_Injuries', 'Minor_Injuries', 'Uninjured',
       'Weather_Condition', 'Report_Status', 'Publication_Date',
       'Total_Injuries', 'Year', 'Month', 'State_Abbr', 'Location_Name',
       'US_State', 'Abbreviation'],
      dtype='object')
```

In [48]:
```
def filtertop(data,column_name):
    topvalues =data[column_name].value_counts().nlargest(10).index
    return data[data[column_name].isin(topvalues)]

Columns = ['Make', 'Model', 'Injury_Severity', 'Report_Status']

for column in Columns:
    data = filtertop(data, column)

data.to_csv('USData_cleaned.csv', index=False)
```

In [49]:
```
data.columns
```

Out[49]:
```
Index(['Unnamed: 0', 'ID', 'Investigation_Type', 'Accident_NO', 'Date',
       'Country', 'Injury_Severity', 'Damage', 'Reg_Number', 'Make', 'Model',
       'Amateur_Built', 'No_of_Engines', 'Engine_Type', 'Schedule',
       'Purpose_of_flight', 'Major_Injuries', 'Minor_Injuries', 'Uninjured',
       'Weather_Condition', 'Report_Status', 'Publication_Date',
       'Total_Injuries', 'Year', 'Month', 'State_Abbr', 'Location_Name',
       'US_State', 'Abbreviation'],
      dtype='object')
```

In [50]:
```
data['Location_Name'].head(10)
```

Out[50]:
```
0          Saltville
1             COTTON
2            SKWENTA
3            GALETON
4           YPSILANTI
5         FORT WORTH
6              PAXTON
7              ODESSA
8    NEW PHILADELPHI
9           CUTCHOGUE
Name: Location_Name, dtype: object
```

In [51]:
```
#Merge my state code and country data
location = pd.read_csv('USState_Codes.csv')
location.head(40)
```

Out[51]:

	US_State	Abbreviation
0	Alabama	AL
1	Alaska	AK
2	Arizona	AZ
3	Arkansas	AR
4	California	CA
5	Colorado	CO
6	Connecticut	CT
7	Delaware	DE
8	Florida	FL
9	Georgia	GA
10	Hawaii	HI
11	Idaho	ID
12	Illinois	IL
13	Indiana	IN
14	Iowa	IA
15	Kansas	KS
16	Kentucky	KY
17	Louisiana	LA
18	Maine	ME
19	Maryland	MD
20	Massachusetts	MA
21	Michigan	MI
22	Minnesota	MN
23	Mississippi	MS
24	Missouri	MO
25	Montana	MT
26	Nebraska	NE
27	Nevada	NV
28	New Hampshire	NH
29	New Jersey	NJ
30	New Mexico	NM
31	New York	NY
32	North Carolina	NC
33	North Dakota	ND
34	Ohio	OH
35	Oklahoma	OK

	US_State	Abbreviation
36	Oregon	OR
37	Pennsylvania	PA
38	Rhode Island	RI
39	South Carolina	SC

In [52]:
```
data['State_Abbr'] =data['Location'].str.split(',').str[-1].str.strip()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File c:\Users\mulwa\.conda\envs\myenv\Lib\site-packages\pandas\core\indexes\base.py:3805, in Index.get_loc(self, key)
   3804 try:
-> 3805     return self._engine.get_loc(casted_key)
   3806 except KeyError as err:

File index.pyx:167, in pandas._libs.index.IndexEngine.get_loc()

File index.pyx:196, in pandas._libs.index.IndexEngine.get_loc()

File pandas\\_libs\\hashtable_class_helper.pxi:7081, in pandas._libs.hashtable.PyObjectHashTable.get_item()

File pandas\\_libs\\hashtable_class_helper.pxi:7089, in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'Location'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Cell In[52], line 1
----> 1 data['State_Abbr'] =data['Location'].str.split(',').str[-1].str.strip()

File c:\Users\mulwa\.conda\envs\myenv\Lib\site-packages\pandas\core\frame.py:4102, in DataFrame.__getitem__(self, key)
   4100 if self.columns.nlevels > 1:
   4101     return self._getitem_multilevel(key)
-> 4102 indexer = self.columns.get_loc(key)
   4103 if is_integer(indexer):
   4104     indexer = [indexer]

File c:\Users\mulwa\.conda\envs\myenv\Lib\site-packages\pandas\core\indexes\base.py:3812, in Index.get_loc(self, key)
   3807     if isinstance(casted_key, slice) or (
   3808         isinstance(casted_key, abc.Iterable)
   3809         and any(isinstance(x, slice) for x in casted_key)
   3810     ):
   3811         raise InvalidIndexError(key)
-> 3812     raise KeyError(key) from err
   3813 except TypeError:
   3814     # If we have a listlike key, _check_indexing_error will raise
   3815     #  InvalidIndexError. Otherwise we fall through and re-raise
   3816     #  the TypeError.
   3817     self._check_indexing_error(key)

KeyError: 'Location'
```

In [194]:
```
data['Location_Name'] = data['Location'].str.split(',').str[0].str.strip()
```

In [198]:
```
data.drop(columns=['Location'], inplace=True)
```

In [200]:
```
merged_data = pd.merge(data, location, how='left', left_on='State_Abbr', right
```

In [209]:
```
merged_data.to_csv('Clean_AviationData.csv')
```

In [211]:
```
D2 = pd.read_csv('Clean_AviationData.csv')
D2.head(10)
```

Out[211]:

	Unnamed: 0	ID	Investigation_Type	Accident_NO	Date	Country	Injury_Severity
0	0	20061025X01555	Accident	NYC07LA005	1974-08-30	United States	Fatal(3)
1	1	20001218X45446	Accident	CHI81LA106	1981-08-01	United States	Fatal(4)
2	2	20020917X01656	Accident	ANC82FAG14	1982-01-02	United States	Fatal(3)
3	3	20020917X02481	Accident	NYC82DA016	1982-01-02	United States	Non-Fatal
4	4	20020917X01894	Accident	CHI82FEC08	1982-01-02	United States	Non-Fatal
5	5	20020917X01992	Accident	FTW82DA036	1982-01-03	United States	Non-Fatal
6	6	20020917X01777	Accident	CHI82DA021	1982-01-06	United States	Non-Fatal
7	7	20020917X02414	Accident	MIA82FLD14	1982-01-08	United States	Fatal(1)
8	8	20020917X01881	Accident	CHI82FA024	1982-01-08	United States	Fatal(1)
9	9	20020917X02484	Accident	NYC82DA019	1982-01-08	United States	Non-Fatal

10 rows × 29 columns

In [124]:
```
USData['No_of_Engines'] = USData['No_of_Engines'].fillna(0).head(10)
```

```
C:\Users\mulwa\AppData\Local\Temp\ipykernel_9128\1452430218.py:1: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  USData['No_of_Engines'] = USData['No_of_Engines'].fillna(0).head(10)
```

In [125]:
```
print(USData['No_of_Engines'].unique())
```

```
[ 1.  0.  2. nan]
```

In [126]:
```
USData['No_of_Engines'] = pd.to_numeric(USData['No_of_Engines'])
```

```
C:\Users\mulwa\AppData\Local\Temp\ipykernel_9128\3574668169.py:1: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  USData['No_of_Engines'] = pd.to_numeric(USData['No_of_Engines'])
```

In [127]:
```
USData.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 82248 entries, 0 to 88888
Data columns (total 25 columns):
 #   Column              Non-Null Count  Dtype         
---  ------              --------------  -----         
 0   ID                  82248 non-null  object        
 1   Investigation_Type  82248 non-null  object        
 2   Accident_NO         82248 non-null  object        
 3   Date                82248 non-null  datetime64[ns]
 4   Location            82248 non-null  object        
 5   Country             82248 non-null  object        
 6   Injury_Severity     82248 non-null  object        
 7   Damage              80269 non-null  object        
 8   Reg_Number          82132 non-null  object        
 9   Make                82227 non-null  object        
 10  Model               82210 non-null  object        
 11  Amateur_Built       82227 non-null  object        
 12  No_of_Engines       10 non-null     float64       
 13  Engine_Type         79206 non-null  object        
 14  Schedule            10297 non-null  object        
 15  Purpose_of_flight   79819 non-null  object        
 16  Major_Injuries      70873 non-null  float64       
 17  Minor_Injuries      71519 non-null  float64       
 18  Uninjured           77243 non-null  float64       
 19  Weather_Condition   81603 non-null  object        
 20  Report_Status       79637 non-null  object        
 21  Publication_Date    69567 non-null  object        
 22  Total_Injuries      82248 non-null  float64       
 23  Year                82248 non-null  int32         
 24  Month               82248 non-null  int32         
dtypes: datetime64[ns](1), float64(5), int32(2), object(17)
memory usage: 15.7+ MB
```

In [128]:
```
USData.isnull().sum()
```

Out[128]:
```
ID                      0
Investigation_Type      0
Accident_NO             0
Date                    0
Location                0
Country                 0
Injury_Severity         0
Damage               1979
Reg_Number            116
Make                   21
Model                  38
Amateur_Built          21
No_of_Engines       82238
Engine_Type          3042
Schedule            71951
Purpose_of_flight    2429
Major_Injuries      11375
Minor_Injuries      10729
Uninjured            5005
Weather_Condition     645
Report_Status        2611
Publication_Date    12681
Total_Injuries          0
Year                    0
Month                   0
dtype: int64
```

In [129]:
```
numerics=USData.select_dtypes(include=['number'])
numerics
print(numerics.isnull().sum())
```

```
No_of_Engines     82238
Major_Injuries    11375
Minor_Injuries    10729
Uninjured          5005
Total_Injuries        0
Year                  0
Month                 0
dtype: int64
```

In [130]:
```
USData['Make'] = USData['Make'].head(10)
USData['Injury_Severity'] = USData['Injury_Severity'].head(10)
USData['Model'] = USData['Model'].head(10)
```

```
C:\Users\mulwa\AppData\Local\Temp\ipykernel_9128\2119715679.py:1: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  USData['Make'] = USData['Make'].head(10)
C:\Users\mulwa\AppData\Local\Temp\ipykernel_9128\2119715679.py:2: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  USData['Injury_Severity'] = USData['Injury_Severity'].head(10)
C:\Users\mulwa\AppData\Local\Temp\ipykernel_9128\2119715679.py:3: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  USData['Model'] = USData['Model'].head(10)
```

In [131]:
```
USData['Major_Injuries'] = USData['Major_Injuries'].fillna('0')
USData['Minor_Injuries'] = USData['Minor_Injuries'].fillna('0')
USData['Uninjured'] = USData['Uninjured'].fillna('0')
USData['Year'] = USData['Year'].fillna('0')
```

```
C:\Users\mulwa\AppData\Local\Temp\ipykernel_9128\1614647534.py:1: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  USData['Major_Injuries'] = USData['Major_Injuries'].fillna('0')
C:\Users\mulwa\AppData\Local\Temp\ipykernel_9128\1614647534.py:2: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  USData['Minor_Injuries'] = USData['Minor_Injuries'].fillna('0')
C:\Users\mulwa\AppData\Local\Temp\ipykernel_9128\1614647534.py:3: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  USData['Uninjured'] = USData['Uninjured'].fillna('0')
C:\Users\mulwa\AppData\Local\Temp\ipykernel_9128\1614647534.py:4: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  USData['Year'] = USData['Year'].fillna('0')
```

In [132]:
```
USData['Year'] = pd.to_numeric(USData['Year'], errors='coerce').astype('Int32'
```

```
C:\Users\mulwa\AppData\Local\Temp\ipykernel_9128\2597975752.py:1: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  USData['Year'] = pd.to_numeric(USData['Year'], errors='coerce').astype('Int
32')
```

In [133]:
```
USData['Uninjured'] = pd.to_numeric(USData['Uninjured'], errors='coerce')
```

```
C:\Users\mulwa\AppData\Local\Temp\ipykernel_9128\2813533755.py:1: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  USData['Uninjured'] = pd.to_numeric(USData['Uninjured'], errors='coerce')
```

In [134]:
```
USData['Year'].unique()
```

Out[134]:
```
<IntegerArray>
[1948, 1962, 1974, 1977, 1979, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 198
8,
 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 200
1,
 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 201
4,
 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022]
Length: 47, dtype: Int32
```

In [135]:
```
numerics=USData.select_dtypes(include=['number'])
numerics
print(numerics.isnull().sum())
```

```
No_of_Engines     82238
Uninjured             0
Total_Injuries        0
Year                  0
Month                 0
dtype: int64
```

In [136]:
```
USData['No_of_Engines'] = USData['No_of_Engines'].fillna('None')
```

```
C:\Users\mulwa\AppData\Local\Temp\ipykernel_9128\3087232067.py:1: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  USData['No_of_Engines'] = USData['No_of_Engines'].fillna('None')
```

In [137]:
```
USData.columns
```

Out[137]:
```
Index(['ID', 'Investigation_Type', 'Accident_NO', 'Date', 'Location',
       'Country', 'Injury_Severity', 'Damage', 'Reg_Number', 'Make', 'Model',
       'Amateur_Built', 'No_of_Engines', 'Engine_Type', 'Schedule',
       'Purpose_of_flight', 'Major_Injuries', 'Minor_Injuries', 'Uninjured',
       'Weather_Condition', 'Report_Status', 'Publication_Date',
       'Total_Injuries', 'Year', 'Month'],
      dtype='object')
```

In [138]:
```
USData['Location'].isnull().sum()
```

Out[138]:
```
0
```

In [139]:
```
# USData[['City','State']] = USData['Location'].str.split(', ', expand=True)
# USData.head(5)
```

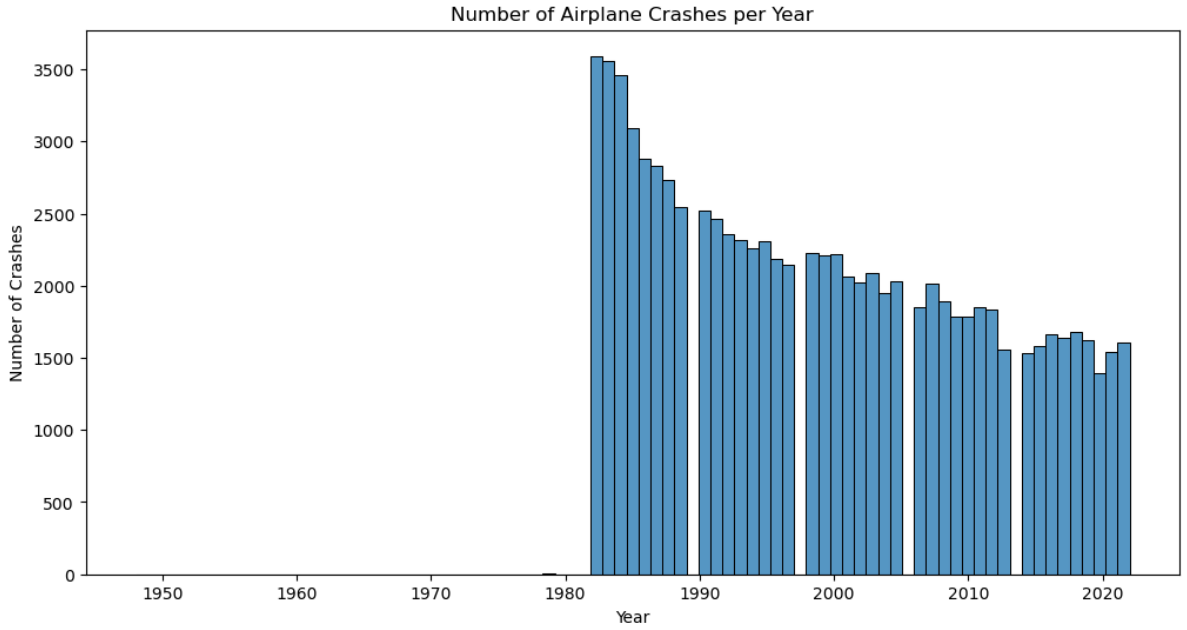
Visualization

In [140]:
```
df["Year"] .unique()
```

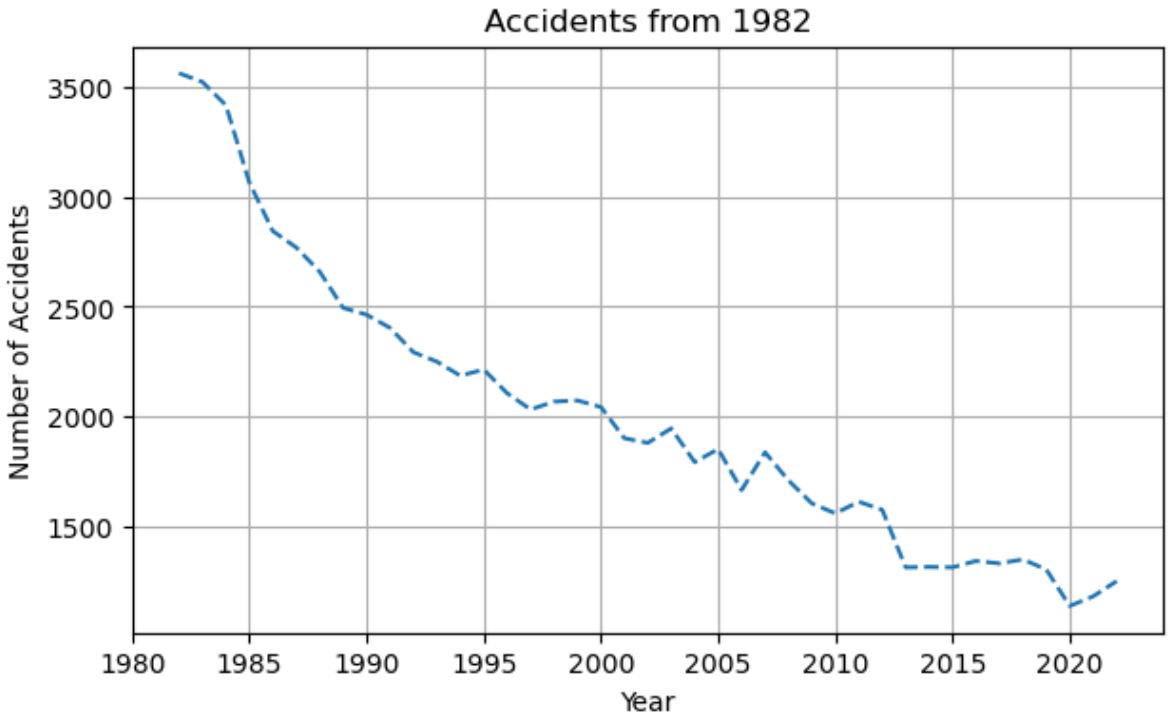
Out[140]:
```
array([1948, 1962, 1974, 1977, 1979, 1981, 1982, 1983, 1984, 1985, 1986,
       1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997,
       1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008,
       2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019,
       2020, 2021, 2022])
```

In [141]:
```
#States with most accidents
```

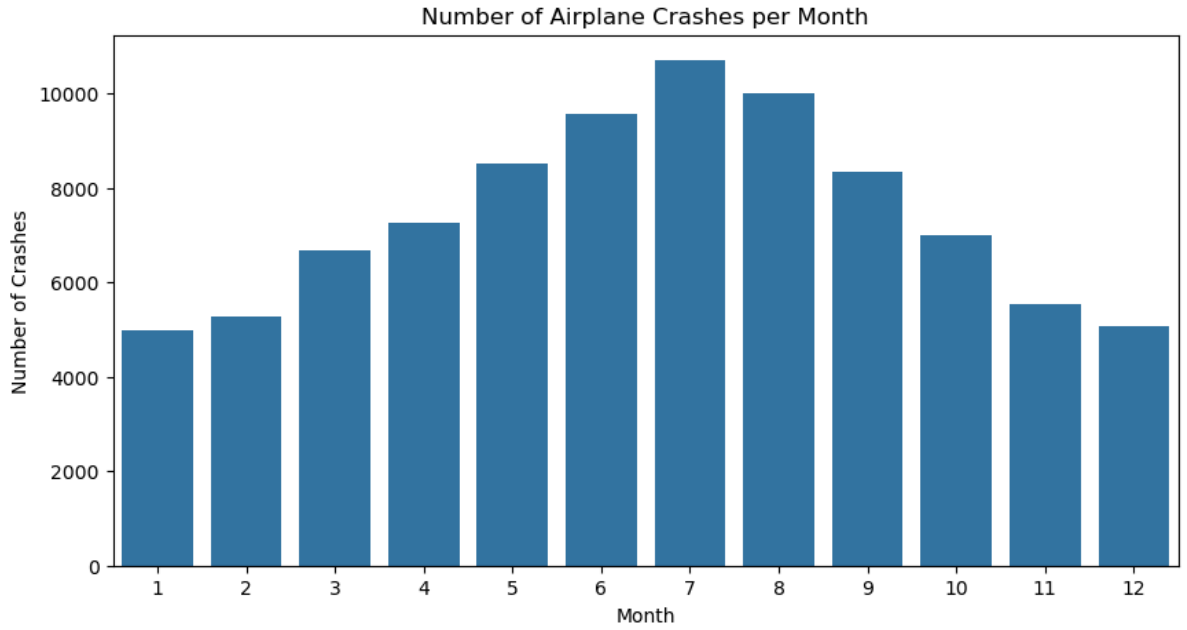
In [142]:
```
#Crashes per year
plt.figure(figsize=(12, 6))
sns.histplot(df['Year'],)
plt.title('Number of Airplane Crashes per Year')
plt.xlabel('Year')
plt.ylabel('Number of Crashes')
plt.show()
```



In [143]:
```
#To get the Number of aviation accidents from 1981 because it is 0 from 1950

from1982 =USData[USData['Year'] >= 1982]

allaccidents = from1982['Year'].value_counts().sort_index()
# allaccidents

plt.figure(figsize=(7,4))
plt.plot(allaccidents.index,allaccidents.values,linestyle='--',)
plt.title('Accidents from 1982')
plt.xlabel('Year')
plt.ylabel('Number of Accidents')
plt.grid(True)
plt.show()
```



In [144]:
```
#Getting number of crashes in each month
plt.figure(figsize=(10, 5))
sns.countplot(x='Month', data=df)
plt.title('Number of Airplane Crashes per Month')
plt.xlabel('Month')
plt.ylabel('Number of Crashes')
plt.show()
```

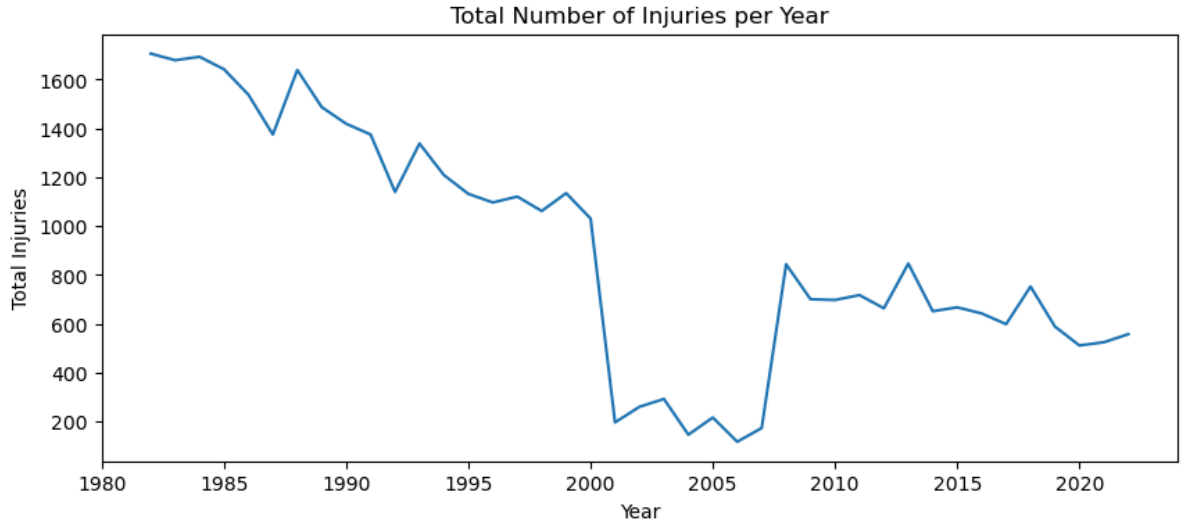


In [145]:
```
df1.head(5)
```

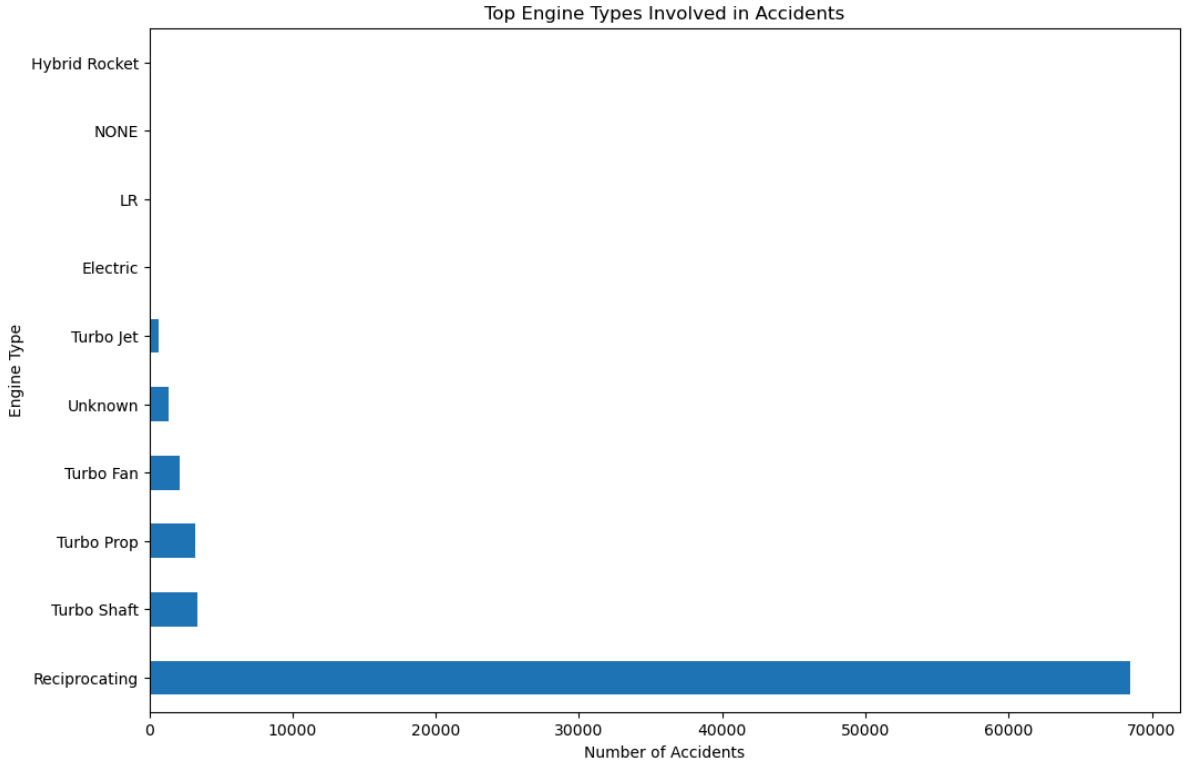
Out[145]:

	US_State	Abbreviation
0	Alabama	AL
1	Alaska	AK
2	Arizona	AZ
3	Arkansas	AR
4	California	CA

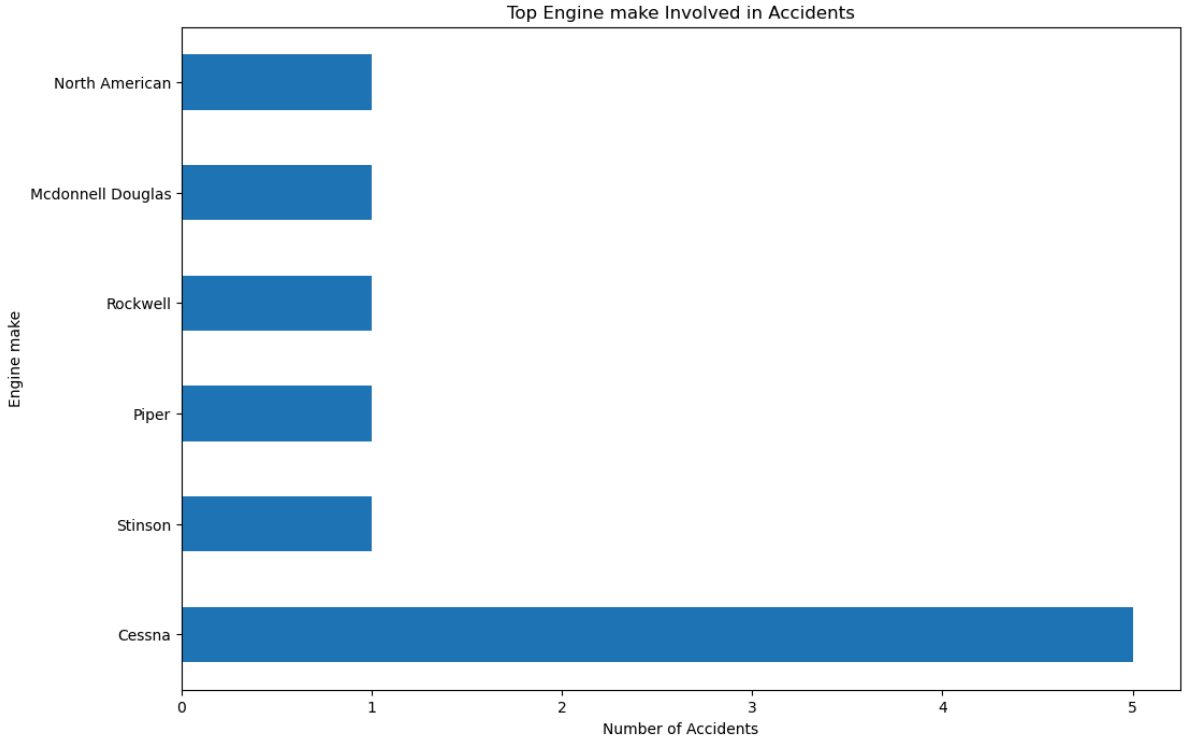
In [146]:
```
#Total Injuries in year

YearlyInjuries = from1982.groupby("Year")['Total_Injuries'].sum().reset_index(

plt.figure(figsize=(10,4))
plt.plot(YearlyInjuries['Year'],YearlyInjuries['Total_Injuries'])
plt.title('Total Number of Injuries per Year')
plt.xlabel('Year')
plt.ylabel('Total Injuries')
plt.show()
```



In [147]:
```
#Engine types Involved in accident
enginetypecounts = USData['Engine_Type'].value_counts().head(10)

plt.figure(figsize=(12, 8))
enginetypecounts.plot(kind='barh')
plt.title('Top Engine Types Involved in Accidents')
plt.xlabel('Number of Accidents')
plt.ylabel('Engine Type')
plt.show()
```



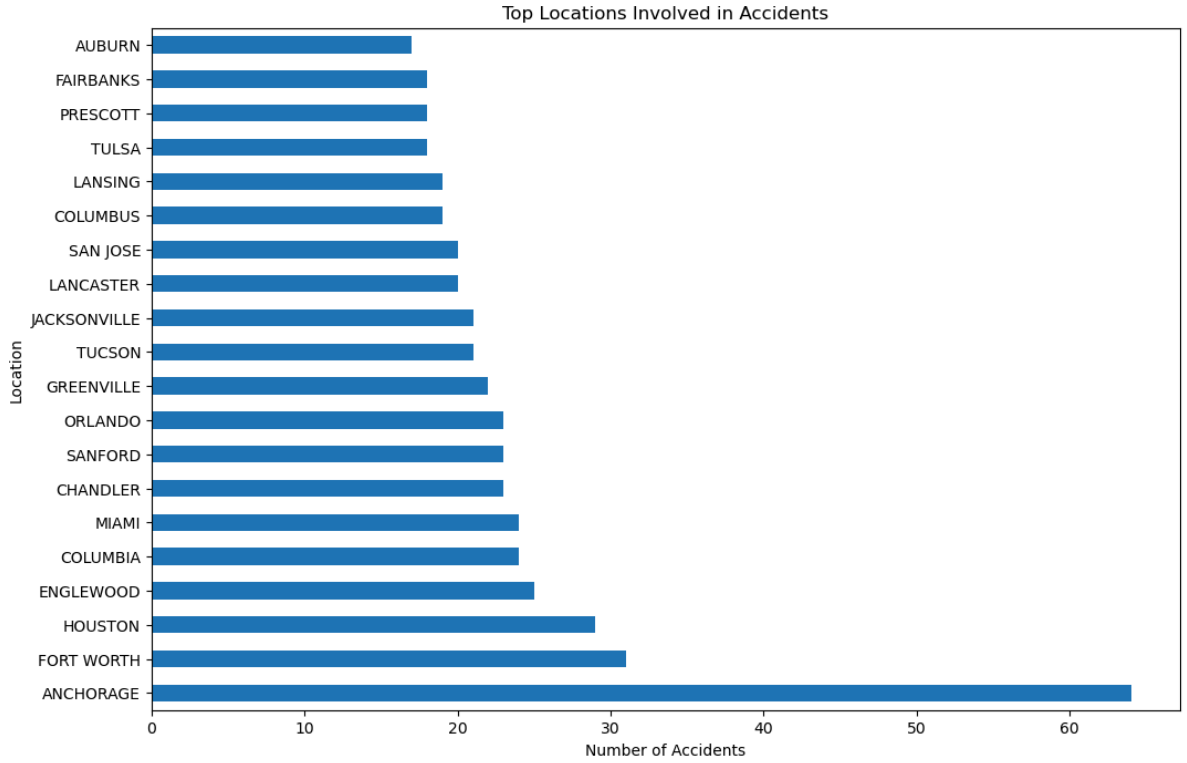
In [148]:
```
#Make types Involved in accident
enginemakecounts = USData['Make'].value_counts().head(10)

plt.figure(figsize=(12, 8))
enginemakecounts.plot(kind='barh')
plt.title('Top Engine make Involved in Accidents')
plt.xlabel('Number of Accidents')
plt.ylabel('Engine make')
plt.show()
```



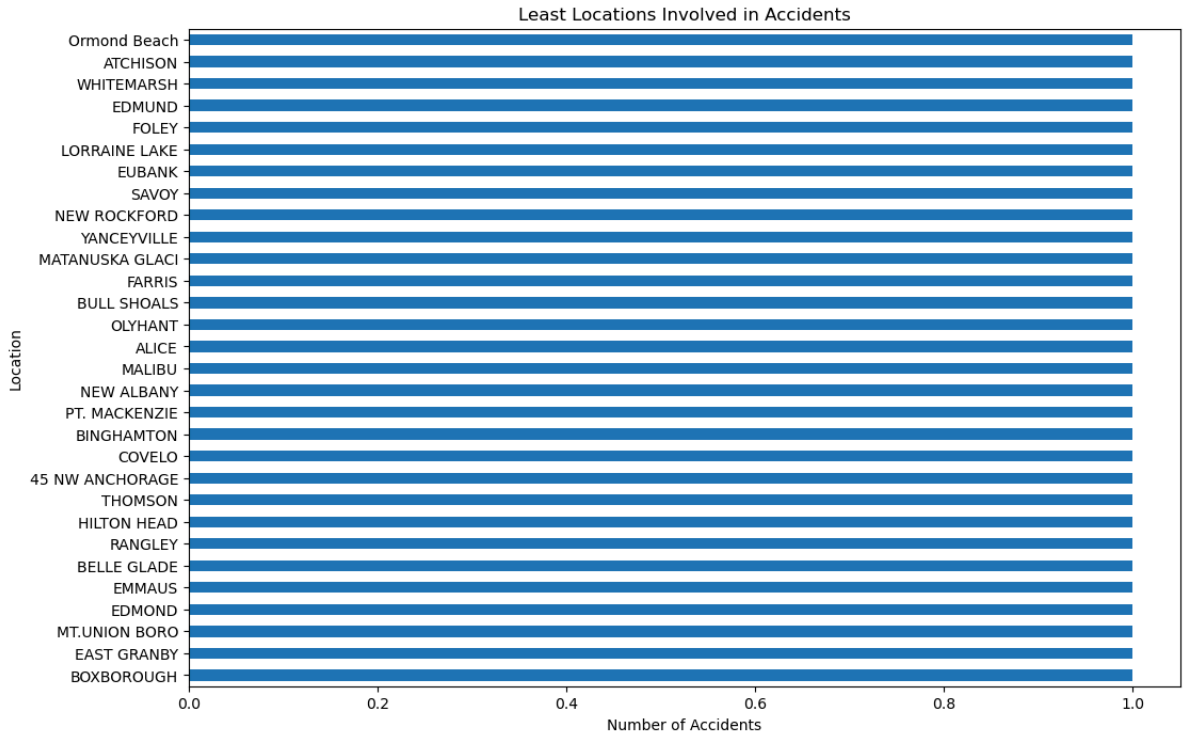
In [104]:
```
data.columns
```

Out[104]:
```
Index(['Unnamed: 0', 'ID', 'Investigation_Type', 'Accident_NO', 'Date',
       'Country', 'Injury_Severity', 'Damage', 'Reg_Number', 'Make', 'Model',
       'Amateur_Built', 'No_of_Engines', 'Engine_Type', 'Schedule',
       'Purpose_of_flight', 'Major_Injuries', 'Minor_Injuries', 'Uninjured',
       'Weather_Condition', 'Report_Status', 'Publication_Date',
       'Total_Injuries', 'Year', 'Month', 'State_Abbr', 'Location_Name',
       'US_State', 'Abbreviation'],
      dtype='object')
```

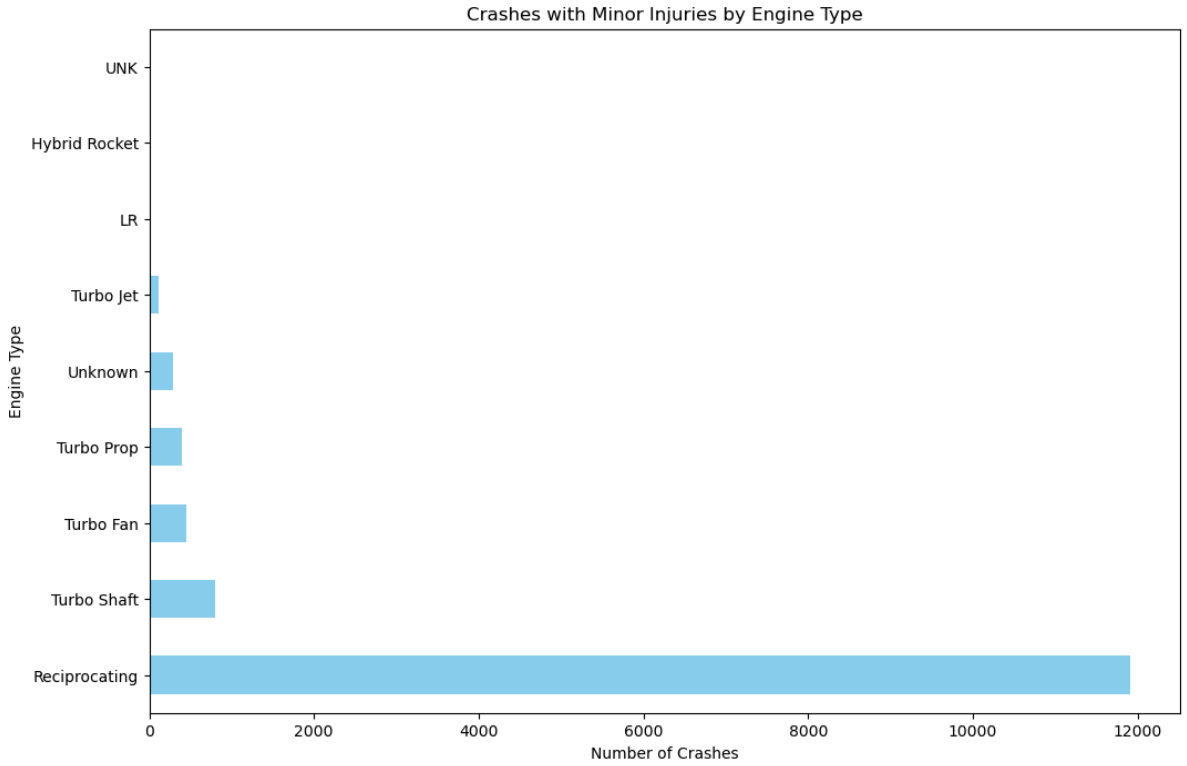
In [105]:
```
#Location vs Accidents
locations = data['Location_Name'].value_counts().head(20)
plt.figure(figsize=(12, 8))
locations.plot(kind='barh')
plt.title('Top Locations Involved in Accidents')
plt.xlabel('Number of Accidents')
plt.ylabel('Location')
plt.show()
```



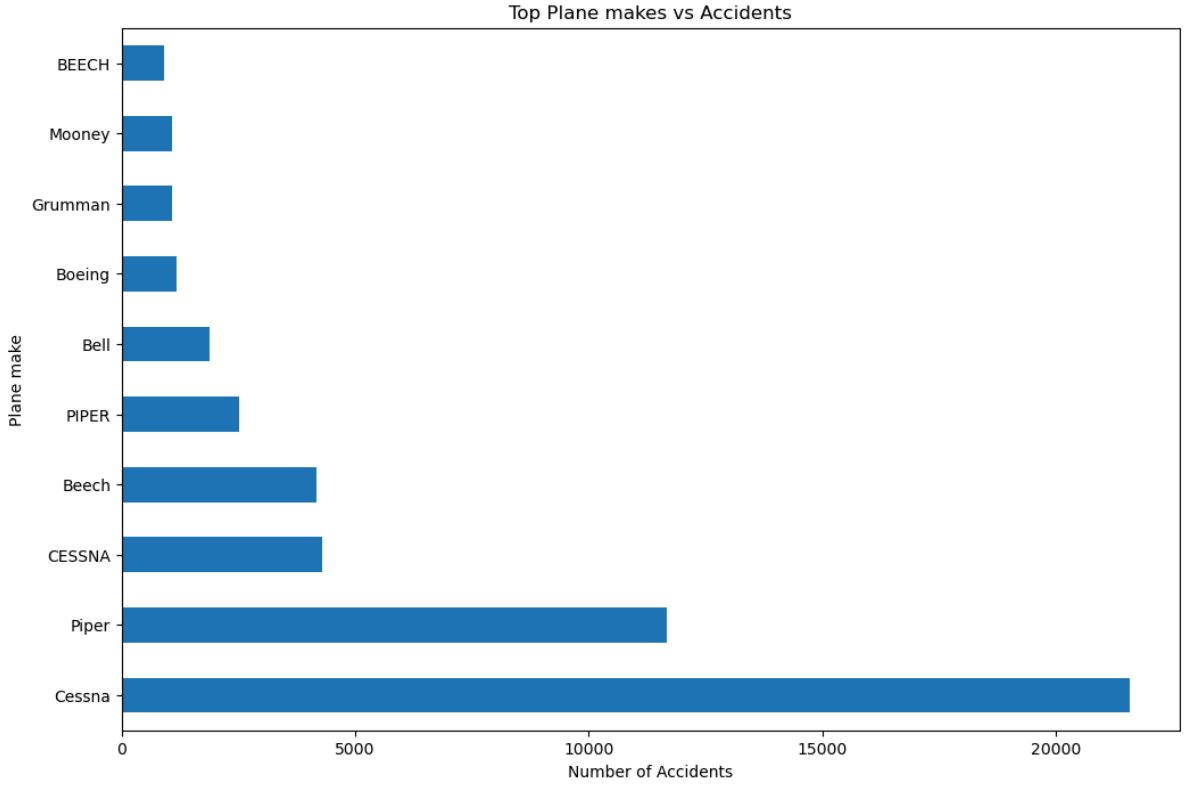
In [106]:
```
#Location vs Accidents
leastlocations = data['Location_Name'].value_counts().tail(30)
plt.figure(figsize=(12, 8))
leastlocations.plot(kind='barh')
plt.title('Least Locations Involved in Accidents')
plt.xlabel('Number of Accidents')
plt.ylabel('Location')
plt.show()
```



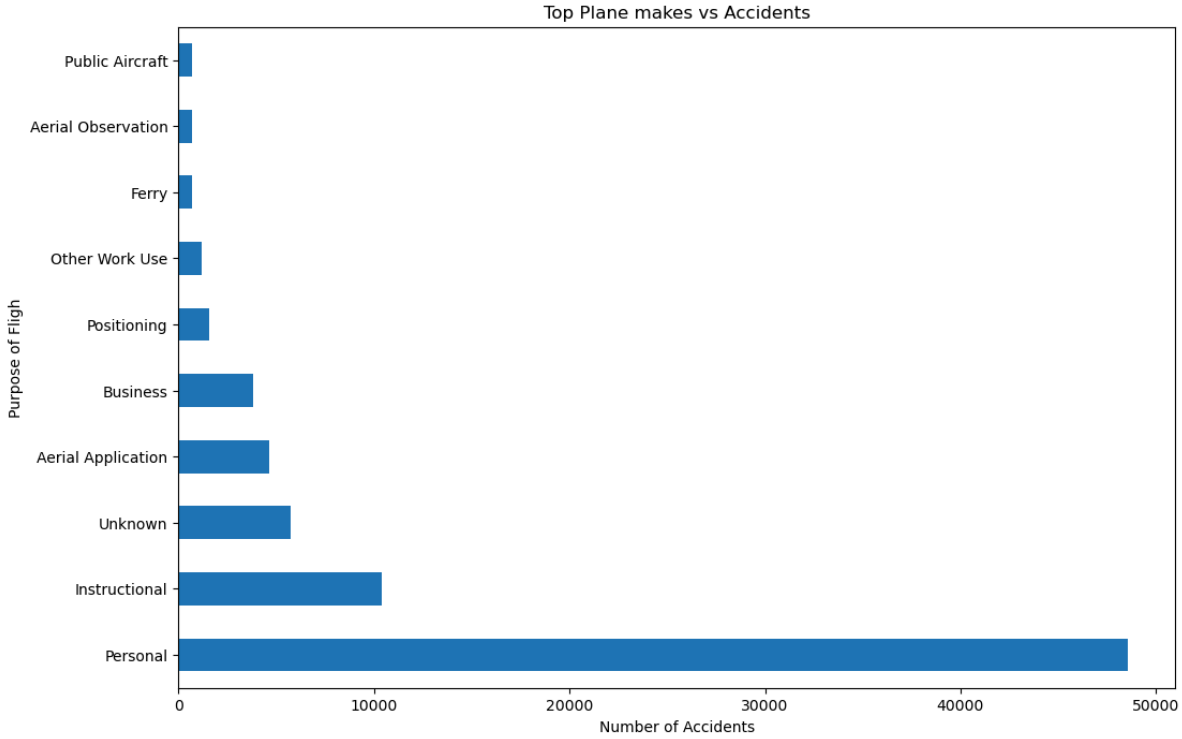
In [65]:
```
#Major vs Engine Types code
majorinjuries = USData[USData['Major_Injuries'] >0]
enginemajtypecounts = minorinjuries['Engine_Type'].value_counts()

plt.figure(figsize=(12, 8))
enginemajtypecounts.plot(kind='barh', color='skyblue')
plt.title('Crashes with Minor Injuries by Engine Type')
plt.xlabel('Number of Crashes')
plt.ylabel('Engine Type')
plt.show()
```



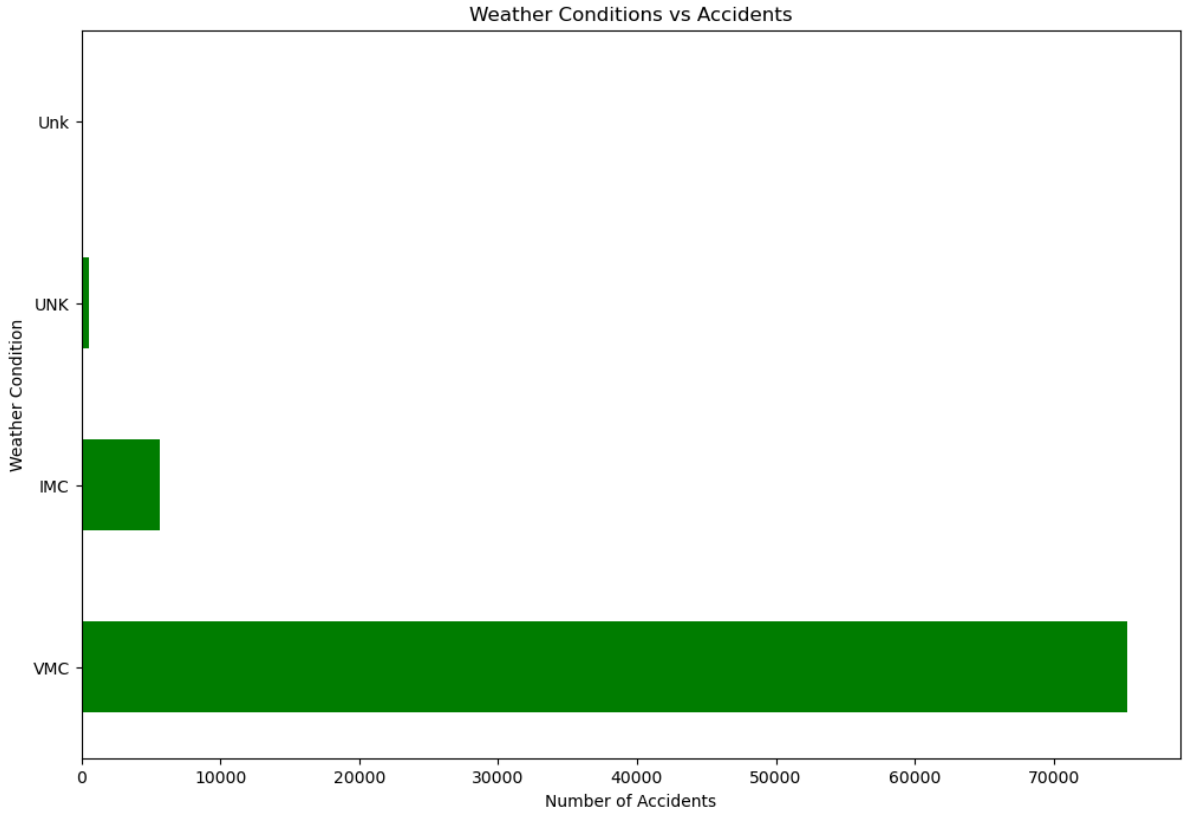
In [66]:
```
#Top Plane makes involved in accidents
enginetypecounts = USData['Make'].value_counts().head(10)

plt.figure(figsize=(12, 8))
enginetypecounts.plot(kind='barh')
plt.title('Top Plane makes vs Accidents')
plt.xlabel('Number of Accidents')
plt.ylabel('Plane make')
plt.show()
```



In [67]:
```
#Purpose of Flights vs Accident
purposecount = USData['Purpose_of_flight'].value_counts().head(10)

plt.figure(figsize=(12, 8))
purposecount.plot(kind='barh')
plt.title('Top Plane makes vs Accidents')
plt.xlabel('Number of Accidents')
plt.ylabel('Purpose of Fligh')
plt.show()
```

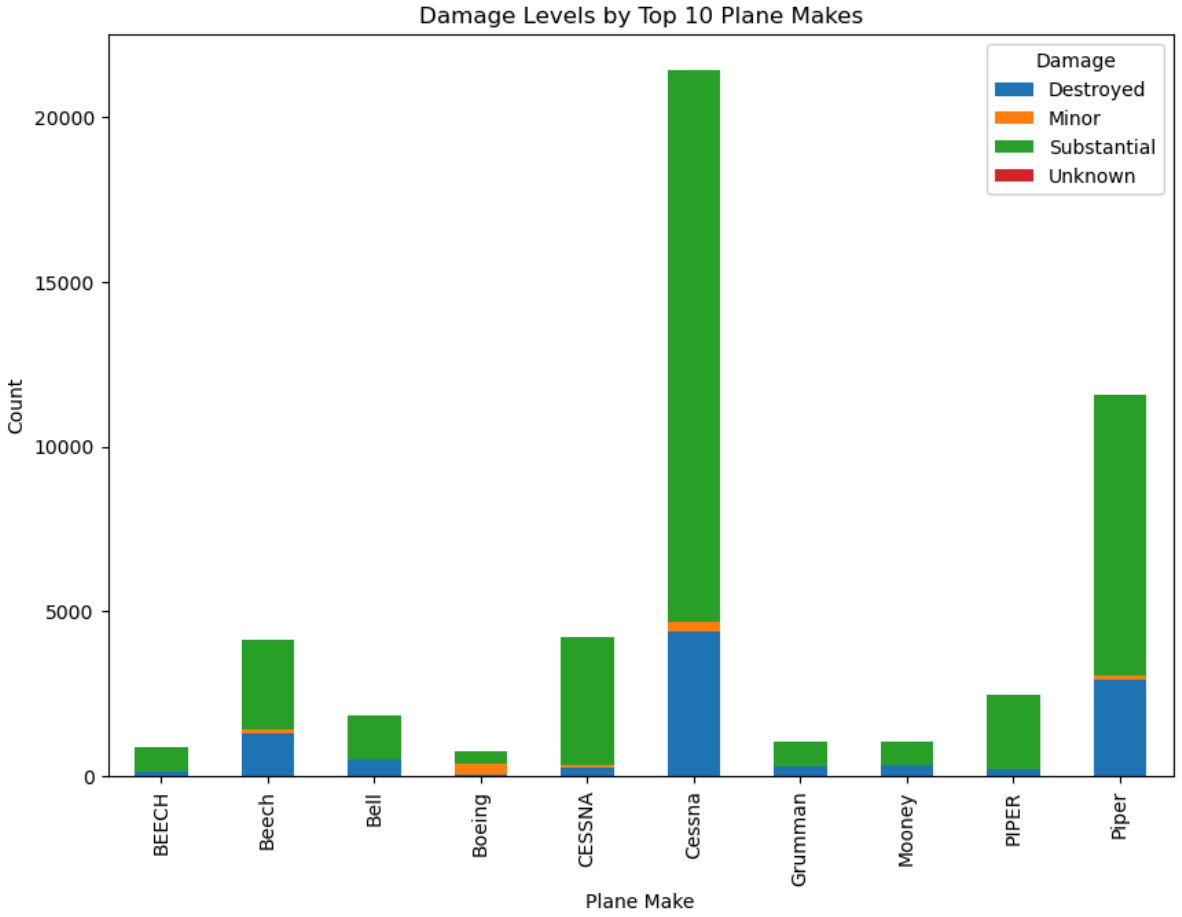


In [68]:
```
#Weather Conditions vs Accident
weather = USData['Weather_Condition'].value_counts()

plt.figure(figsize=(12, 8))
weather.plot(kind='barh',color='green')
plt.title('Weather Conditions vs Accidents')
plt.xlabel('Number of Accidents')
plt.ylabel('Weather Condition')
plt.show()
```



In [69]:

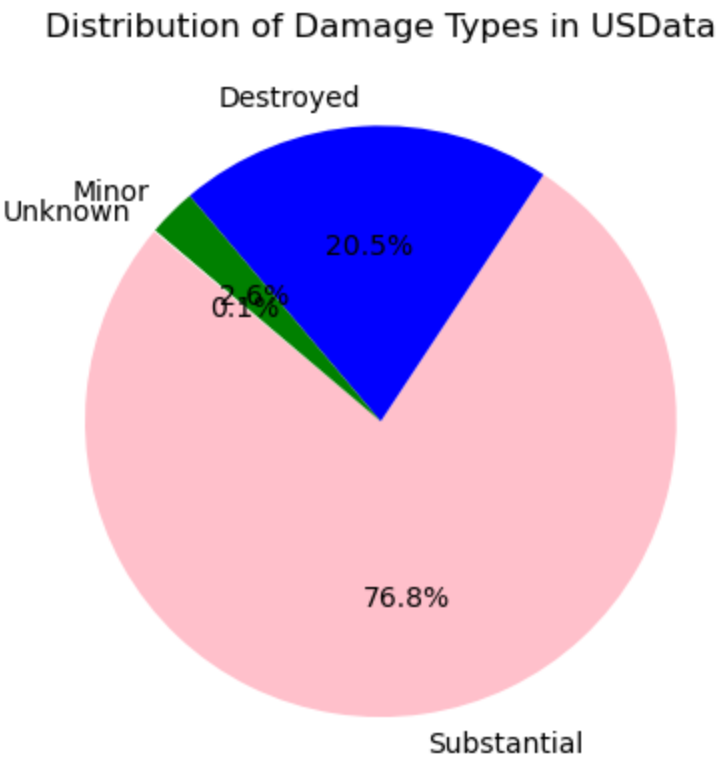
```
top_10_makes = USData['Make'].value_counts().nlargest(10).index
USData_top = USData[USData['Make'].isin(top_10_makes)]

makedamage_counts = USData_top.groupby(['Make', 'Damage']).size().unstack(fill

makedamage_counts.plot(kind='bar', stacked=True, figsize=(10, 7))
plt.title('Damage Levels by Top 10 Plane Makes')
plt.xlabel('Plane Make')
plt.ylabel('Count')
plt.legend(title='Damage')
plt.show()
```



In [70]:
```
USData.info()
```

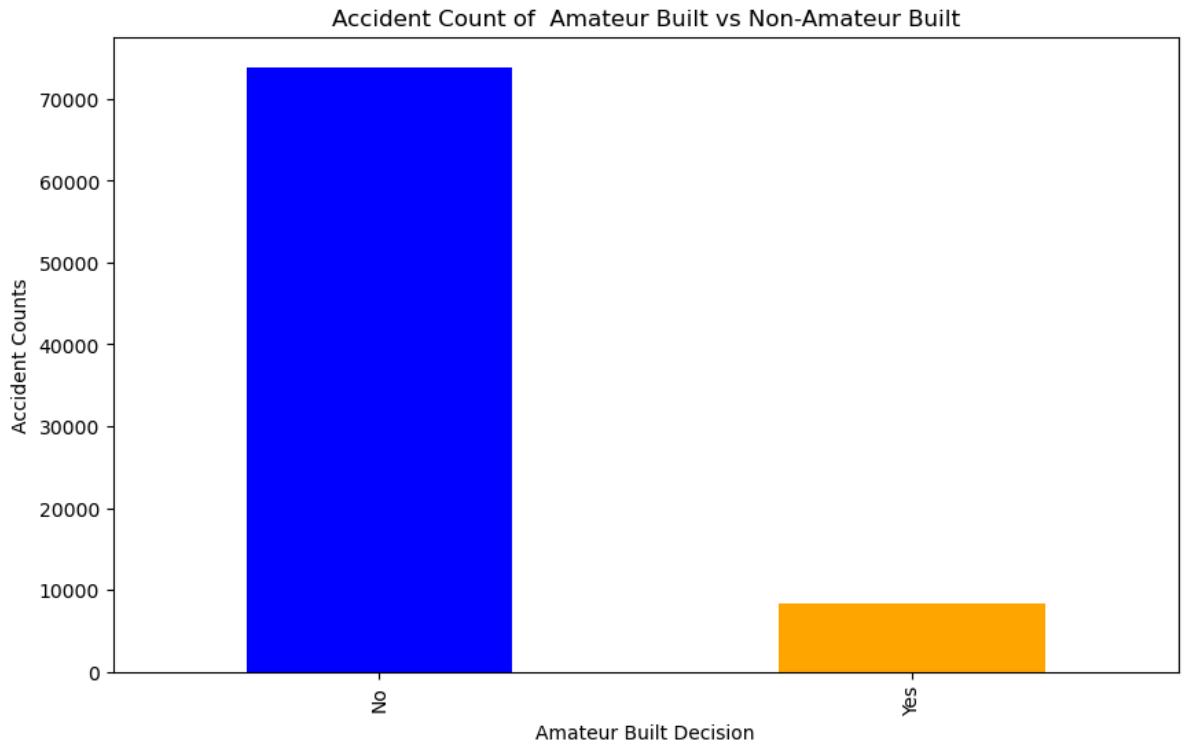
```
<class 'pandas.core.frame.DataFrame'>
Index: 82248 entries, 0 to 88888
Data columns (total 25 columns):
 #   Column              Non-Null Count  Dtype         
---  ------              --------------  -----         
 0   ID                  82248 non-null  object        
 1   Investigation_Type  82248 non-null  object        
 2   Accident_NO         82248 non-null  object        
 3   Date                82248 non-null  datetime64[ns]
 4   Location            82248 non-null  object        
 5   Country             82248 non-null  object        
 6   Injury_Severity     82248 non-null  object        
 7   Damage              80269 non-null  object        
 8   Reg_Number          82132 non-null  object        
 9   Make                82227 non-null  object        
 10  Model               82210 non-null  object        
 11  Amateur_Built       82227 non-null  object        
 12  No_of_Engines       82248 non-null  float64       
 13  Engine_Type         79206 non-null  object        
 14  Schedule            10297 non-null  object        
 15  Purpose_of_flight   79819 non-null  object        
 16  Major_Injuries      70873 non-null  float64       
 17  Minor_Injuries      71519 non-null  float64       
 18  Uninjured           77243 non-null  float64       
 19  Weather_Condition   81603 non-null  object        
 20  Report_Status       79637 non-null  object        
 21  Publication_Date    69567 non-null  object        
 22  Total_Injuries      82248 non-null  float64       
 23  Year                82248 non-null  int32         
 24  Month               82248 non-null  int32         
dtypes: datetime64[ns](1), float64(5), int32(2), object(17)
memory usage: 15.7+ MB
```

In [80]:
```
#Pie Chart Representation of Damage types in US

damage_counts = USData['Damage'].value_counts()

plt.pie(damage_counts, labels=damage_counts.index, autopct='%1.1f%%', startang
plt.title('Distribution of Damage Types in USData')

plt.show()
```



In [91]:

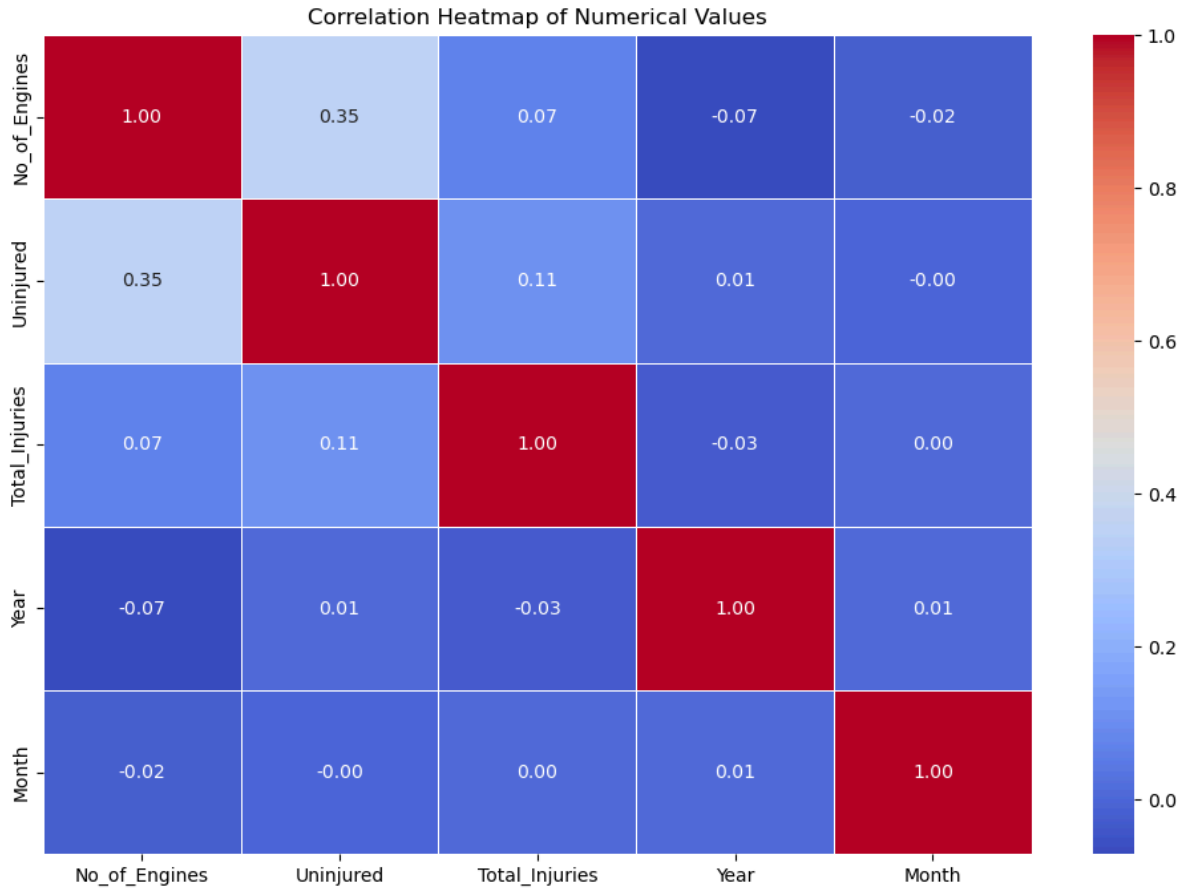
```
amateur_builddata = USData['Amateur_Built'].value_counts()

plt.figure(figsize=(10, 6))
amateur_builddata.plot(kind='bar', color=['blue', 'orange'])

plt.xlabel('Amateur Built Decision')
plt.ylabel('Accident Counts')
plt.title('Accident Count of  Amateur Built vs Non-Amateur Built')

plt.show()
```



In [ ]:

In [80]:
```
#Heat Map of all numeric Values
allnum =USData.select_dtypes(include=['number'])
print(allnum)
corrnum= allnum.corr()

plt.figure(figsize=(12, 8))
sns.heatmap(corrnum, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
plt.title('Correlation Heatmap of Numerical Values')
plt.show()
```

```
       No_of_Engines  Uninjured  Total_Injuries  Year  Month
0                1.0        0.0             0.0  1948     10
1                1.0        0.0             0.0  1962      7
2                1.0        0.0             0.0  1974      8
3                1.0        0.0             0.0  1977      6
4                0.0        0.0             0.0  1979      8
...              ...        ...             ...   ...    ...
88884            0.0        0.0             1.0  2022     12
88885            0.0        0.0             0.0  2022     12
88886            1.0        1.0             0.0  2022     12
88887            0.0        0.0             0.0  2022     12
88888            0.0        1.0             1.0  2022     12

[82248 rows x 5 columns]
```



In [78]:
```
print(USData['Year'].dtype)
print(USData['Uninjured'].dtype)
```

```
Int32
float64
```

In [124]:
```
USData.columns
```

Out[124]:
```
Index(['ID', 'Investigation_Type', 'Accident_NO', 'Date', 'Location',
       'Country', 'Injury_Severity', 'Damage', 'Reg_Number', 'Make', 'Model',
       'Amateur_Built', 'No_of_Engines', 'Engine_Type', 'Schedule',
       'Purpose_of_flight', 'Major_Injuries', 'Minor_Injuries', 'Uninjured',
       'Weather_Condition', 'Report_Status', 'Publication_Date',
       'Total_Injuries', 'Year', 'Month'],
      dtype='object')
```

In [113]:
```
numerics=USData.select_dtypes(include=['number'])
numerics
```

Out[113]:

	No_of_Engines	Total_Injuries	Month
0	1.0	0.0	10
1	1.0	0.0	7
2	1.0	0.0	8
3	1.0	0.0	6
4	0.0	0.0	8
...	...	...	...
88884	0.0	1.0	12
88885	0.0	0.0	12
88886	1.0	0.0	12
88887	0.0	0.0	12
88888	0.0	1.0	12

82248 rows × 3 columns

In [86]:
```
# USData['Amateur_Built'].unique()
```

Out[86]:
```
array(['No', 'Yes', nan], dtype=object)
```

In [92]:
```
#Number of Injured in each year
USData.head(10)
```

Out[92]:

	ID	Investigation_Type	Accident_NO	Date	Location	Country	Injury_Sev
0	20001218X45444	Accident	SEA87LA080	1948-10-24	MOOSE CREEK, ID	United States	Fa
1	20001218X45447	Accident	LAX94LA336	1962-07-19	BRIDGEPORT, CA	United States	Fa
2	20061025X01555	Accident	NYC07LA005	1974-08-30	Saltville, VA	United States	Fa
3	20001218X45448	Accident	LAX96LA321	1977-06-19	EUREKA, CA	United States	Fa
4	20041105X01764	Accident	CHI79FA064	1979-08-02	Canton, OH	United States	Fa
5	20170710X52551	Accident	NYC79AA106	1979-09-17	BOSTON, MA	United States	Non-
6	20001218X45446	Accident	CHI81LA106	1981-08-01	COTTON, MN	United States	Fa
7	20020909X01562	Accident	SEA82DA022	1982-01-01	PULLMAN, WA	United States	Non-
8	20020909X01561	Accident	NYC82DA015	1982-01-01	EAST HANOVER, NJ	United States	Non-
9	20020909X01560	Accident	MIA82DA029	1982-01-01	JACKSONVILLE, FL	United States	Non-

10 rows × 25 columns

In [53]:
```
USData.columns
```

Out[53]:
```
Index(['ID', 'Investigation_Type', 'Accident_NO', 'Date', 'Location',
       'Country', 'Injury_Severity', 'Damage', 'Reg_Number', 'Make', 'Model',
       'Amateur_Built', 'No_of_Engines', 'Engine_Type', 'Schedule',
       'Purpose_of_flight', 'Major_Injuries', 'Minor_Injuries', 'Uninjured',
       'Weather_Condition', 'Report_Status', 'Publication_Date',
       'Total_Injuries', 'Year', 'Month'],
      dtype='object')
```

In [107]:
```
data.columns
```

Out[107]:
```
Index(['Unnamed: 0', 'ID', 'Investigation_Type', 'Accident_NO', 'Date',
       'Country', 'Injury_Severity', 'Damage', 'Reg_Number', 'Make', 'Model',
       'Amateur_Built', 'No_of_Engines', 'Engine_Type', 'Schedule',
       'Purpose_of_flight', 'Major_Injuries', 'Minor_Injuries', 'Uninjured',
       'Weather_Condition', 'Report_Status', 'Publication_Date',
       'Total_Injuries', 'Year', 'Month', 'State_Abbr', 'Location_Name',
       'US_State', 'Abbreviation'],
      dtype='object')
```