In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import math

from warnings import filterwarnings
filterwarnings("ignore")
```

In [2]:
```
#df1= pd.read_csv('USState_Codes.csv')
```

In [3]:
```
df= pd.read_csv('AviationData.csv',encoding = "ISO-8859-1")
df
```

Out[3]:

	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 × 31 columns

In [4]:
```
## Understanding the data
```

In [5]:
```
# there  are 88889 rows and 31 columns
df.shape
```

Out[5]:
```
(88889, 31)
```

In [6]:
```
# understanding the data
df.head()
```

Out[6]:

	Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Country	Lati
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.
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 [7]:
```
#getting the metadata
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 [8]:
```
df.columns
```

Out[8]:
```
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')
```

Below columns contains floats(numbers with decimal points).

Total.Fatal.Injuries, Total.Serious.Injuries, Total.Minor.Injuries, Total.Uninjured, Number.of.Engines ,

All other columns has str (object)

Some Columns eg latitude and longitude has a lot of missing data

In [9]:
```
# Basic desciption of the data.
df.describe
```

Out[9]:
```
<bound method NDFrame.describe of             Event.Id Investigation.Type Ac
cident.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.Code  \
0      MOOSE CREEK, ID  United States      NaN       NaN          NaN
1       BRIDGEPORT, CA  United States      NaN       NaN          NaN
2        Saltville, VA  United States  36.9222  -81.8781          NaN
3           EUREKA, CA  United States      NaN       NaN          NaN
4           Canton, OH  United States      NaN       NaN          NaN
...                ...            ...      ...       ...          ...
88884    Annapolis, MD  United States      NaN       NaN          NaN
88885      Hampton, NH  United States      NaN       NaN          NaN
88886       Payson, AZ  United States  341525N  1112021W          PAN
88887       Morgan, UT  United States      NaN       NaN          NaN
88888       Athens, GA  United States      NaN       NaN          NaN

      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              NaN                    NaN
88885            0.0              NaN                    NaN
88886            1.0              VMC                    NaN
88887            0.0              NaN                    NaN
88888            1.0              NaN                    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 [10]:
```
#checking if there are rows with missing values
df.dropna(axis="index",how="all")
```

Out[10]:

	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 × 31 columns

In [11]:
```
#checking if there are columns with missing values
df.dropna(axis="columns",how="all")
```

Out[11]:

	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

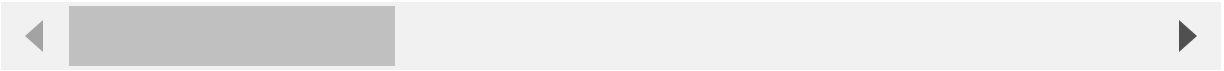
88889 rows × 31 columns

In [12]:
```
#checking if missing values are classified as missing values
df.isna()
df
```

Out[12]:

	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 × 31 columns



In [13]:
```
df.dtypes
```

Out[13]:
```
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 [14]:
```
#checking for duplicates .result , no duplicated rows
df.duplicated().value_counts()
```

Out[14]:
```
False    88889
dtype: int64
```

DATA CLEANNING

Picking relevants columns for analysis

In [15]:
```
relevant_columns=df[['Event.Id' ,'Investigation.Type','Location','Country', 'Latitude', 'Longitude','Make', 'Model','Aircraft.damage','Weather.Condition', 'Broad.phase.of.flight','Total.Fatal.Injuries','Total.Serious.Injuries','Total.Minor.Injuries','Total.Uninjured','Event.Date']]
df1=relevant_columns
df1
```

Out[15]:

	Event.Id	Investigation.Type	Location	Country	Latitude	Longitude	Ma
0	20001218X45444	Accident	MOOSE CREEK, ID	United States	NaN	NaN	Stins
1	20001218X45447	Accident	BRIDGEPORT, CA	United States	NaN	NaN	Pi
2	20061025X01555	Accident	Saltville, VA	United States	36.9222	-81.8781	Cess
3	20001218X45448	Accident	EUREKA, CA	United States	NaN	NaN	Rockv
4	20041105X01764	Accident	Canton, OH	United States	NaN	NaN	Cess
...	...	...	...	...	...	...	
88884	20221227106491	Accident	Annapolis, MD	United States	NaN	NaN	PIP
88885	20221227106494	Accident	Hampton, NH	United States	NaN	NaN	BELLAN
88886	20221227106497	Accident	Payson, AZ	United States	341525N	1112021W	AMERIC CHAMPI AIRCRA
88887	20221227106498	Accident	Morgan, UT	United States	NaN	NaN	CESS
88888	20221230106513	Accident	Athens, GA	United States	NaN	NaN	PIP

88889 rows × 16 columns

In [16]:
```
#metadata for df1
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 88889 entries, 0 to 88888
Data columns (total 16 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Event.Id                88889 non-null  object 
 1   Investigation.Type      88889 non-null  object 
 2   Location                88837 non-null  object 
 3   Country                 88663 non-null  object 
 4   Latitude                34382 non-null  object 
 5   Longitude               34373 non-null  object 
 6   Make                    88826 non-null  object 
 7   Model                   88797 non-null  object 
 8   Aircraft.damage         85695 non-null  object 
 9   Weather.Condition       84397 non-null  object 
 10  Broad.phase.of.flight   61724 non-null  object 
 11  Total.Fatal.Injuries    77488 non-null  float64
 12  Total.Serious.Injuries  76379 non-null  float64
 13  Total.Minor.Injuries    76956 non-null  float64
 14  Total.Uninjured         82977 non-null  float64
 15  Event.Date              88889 non-null  object 
dtypes: float64(4), object(12)
memory usage: 10.9+ MB
```

Handling Missing Data

In [17]:
```
df1.dtypes
```

Out[17]:
```
Event.Id                 object
Investigation.Type       object
Location                 object
Country                  object
Latitude                 object
Longitude                object
Make                     object
Model                    object
Aircraft.damage          object
Weather.Condition        object
Broad.phase.of.flight    object
Total.Fatal.Injuries    float64
Total.Serious.Injuries  float64
Total.Minor.Injuries    float64
Total.Uninjured         float64
Event.Date               object
dtype: object
```

In [18]:
```
df1['year'] = [int(i.split('-')[0]) for i in df1['Event.Date']]
df1['month'] = [int(i.split('-')[1]) for i in df1['Event.Date']]
df1['day'] = [int(i.split('-')[2]) for i in df1['Event.Date']]
```

In [19]:
```
#drop original columns"event.date" and latitude and longitudes because of high missing values
df2=df1.drop(['Event.Date','Latitude','Longitude'],axis=1)
df2
```

Out[19]:

	Event.Id	Investigation.Type	Location	Country	Make	Model	Aircraft.(
0	20001218X45444	Accident	MOOSE CREEK, ID	United States	Stinson	108-3	De
1	20001218X45447	Accident	BRIDGEPORT, CA	United States	Piper	PA24-180	De
2	20061025X01555	Accident	Saltville, VA	United States	Cessna	172M	De
3	20001218X45448	Accident	EUREKA, CA	United States	Rockwell	112	De
4	20041105X01764	Accident	Canton, OH	United States	Cessna	501	De
...	...	...	...	...	...	...	
88884	20221227106491	Accident	Annapolis, MD	United States	PIPER	PA-28-151	
88885	20221227106494	Accident	Hampton, NH	United States	BELLANCA	7ECA	
88886	20221227106497	Accident	Payson, AZ	United States	AMERICAN CHAMPION AIRCRAFT	8GCBC	Sul
88887	20221227106498	Accident	Morgan, UT	United States	CESSNA	210N	
88888	20221230106513	Accident	Athens, GA	United States	PIPER	PA-24-260	

88889 rows × 16 columns

In [20]:
```
#calculating  the basic summary statistics for each column

df2.describe()
```

Out[20]:

	Total.Fatal.Injuries	Total.Serious.Injuries	Total.Minor.Injuries	Total.Uninjured	year
count	77488.000000	76379.000000	76956.000000	82977.000000	88889.000000
mean	0.647855	0.279881	0.357061	5.325440	1999.206662
std	5.485960	1.544084	2.235625	27.913634	11.888226
min	0.000000	0.000000	0.000000	0.000000	1948.000000
25%	0.000000	0.000000	0.000000	0.000000	1989.000000
50%	0.000000	0.000000	0.000000	1.000000	1998.000000
75%	0.000000	0.000000	0.000000	2.000000	2009.000000
max	349.000000	161.000000	380.000000	699.000000	2022.000000

In [21]:
```
#for missing value in numerical categories , we replace with mean

df2["Total.Fatal.Injuries"].fillna(df2["Total.Fatal.Injuries"].mean(),inplace=
True)
df2["Total.Serious.Injuries"].fillna(df2["Total.Serious.Injuries"].mean(),inpl
ace=True)
df2["Total.Minor.Injuries"].fillna(df2["Total.Minor.Injuries"].mean(),inplace=
True)
df2["Total.Uninjured"].fillna(df2["Total.Uninjured"].mean(),inplace=True)

df2
```

Out[21]:

	Event.Id	Investigation.Type	Location	Country	Make	Model	Aircraft.(
0	20001218X45444	Accident	MOOSE CREEK, ID	United States	Stinson	108-3	De
1	20001218X45447	Accident	BRIDGEPORT, CA	United States	Piper	PA24-180	De
2	20061025X01555	Accident	Saltville, VA	United States	Cessna	172M	De
3	20001218X45448	Accident	EUREKA, CA	United States	Rockwell	112	De
4	20041105X01764	Accident	Canton, OH	United States	Cessna	501	De
...	...	...	...	...	...	...	
88884	20221227106491	Accident	Annapolis, MD	United States	PIPER	PA-28-151	
88885	20221227106494	Accident	Hampton, NH	United States	BELLANCA	7ECA	
88886	20221227106497	Accident	Payson, AZ	United States	AMERICAN CHAMPION AIRCRAFT	8GCBC	Sul
88887	20221227106498	Accident	Morgan, UT	United States	CESSNA	210N	
88888	20221230106513	Accident	Athens, GA	United States	PIPER	PA-24-260	

88889 rows × 16 columns

In [22]:
```
#sorting columns with missing data
missing_data=df2.isnull().sum()
missing_data.sort_values(ascending= False)
```

Out[22]:
```
Broad.phase.of.flight     27165
Weather.Condition          4492
Aircraft.damage            3194
Country                     226
Model                        92
Make                         63
Location                     52
day                           0
month                         0
year                          0
Total.Uninjured               0
Total.Minor.Injuries          0
Total.Serious.Injuries        0
Total.Fatal.Injuries          0
Investigation.Type            0
Event.Id                      0
dtype: int64
```

Getting unique values and their distribution of values in the categorical columns.

In [23]:
```
df2['Broad.phase.of.flight'].unique()
```

Out[23]:
```
array(['Cruise', 'Unknown', 'Approach', 'Climb', 'Takeoff', 'Landing',
       'Taxi', 'Descent', 'Maneuvering', 'Standing', 'Go-around', 'Other',
       nan], dtype=object)
```

In [24]:
```
df2['Broad.phase.of.flight'].value_counts()
```

Out[24]:
```
Landing        15428
Takeoff        12493
Cruise         10269
Maneuvering     8144
Approach        6546
Climb           2034
Taxi            1958
Descent         1887
Go-around       1353
Standing         945
Unknown          548
Other            119
Name: Broad.phase.of.flight, dtype: int64
```

In [25]:
```
categories=['Weather.Condition',]
for c in categories:
    print(c , df2[c].unique())
```

```
Weather.Condition ['UNK' 'IMC' 'VMC' nan 'Unk']
```

In [26]:
```
df2['Weather.Condition'].value_counts()
```

Out[26]:
```
VMC    77303
IMC     5976
UNK      856
Unk      262
Name: Weather.Condition, dtype: int64
```

In [27]:
```
df2['Aircraft.damage'].unique()
```

Out[27]:
```
array(['Destroyed', 'Substantial', 'Minor', nan, 'Unknown'], dtype=object)
```

In [28]:
```
df2['Aircraft.damage'].value_counts()
```

Out[28]:
```
Substantial    64148
Destroyed      18623
Minor           2805
Unknown          119
Name: Aircraft.damage, dtype: int64
```

In [29]:
```
#Replacing missing values in categorical data

df2['Weather.Condition'].fillna('UNK', inplace=True)
df2['Broad.phase.of.flight'].fillna('Unknown', inplace=True)
df2['Aircraft.damage'].fillna('Unknown', inplace=True)
df2
```

Out[29]:

	Event.Id	Investigation.Type	Location	Country	Make	Model	Aircraft.c
0	20001218X45444	Accident	MOOSE CREEK, ID	United States	Stinson	108-3	De
1	20001218X45447	Accident	BRIDGEPORT, CA	United States	Piper	PA24-180	De
2	20061025X01555	Accident	Saltville, VA	United States	Cessna	172M	De
3	20001218X45448	Accident	EUREKA, CA	United States	Rockwell	112	De
4	20041105X01764	Accident	Canton, OH	United States	Cessna	501	De
...	...	...	...	...	...	...	
88884	20221227106491	Accident	Annapolis, MD	United States	PIPER	PA-28-151	U
88885	20221227106494	Accident	Hampton, NH	United States	BELLANCA	7ECA	U
88886	20221227106497	Accident	Payson, AZ	United States	AMERICAN CHAMPION AIRCRAFT	8GCBC	Sul
88887	20221227106498	Accident	Morgan, UT	United States	CESSNA	210N	U
88888	20221230106513	Accident	Athens, GA	United States	PIPER	PA-24-260	U

88889 rows × 16 columns

In [30]:
```
#safe df to excel for analysis using Tableau
df2.to_csv('AviationData1.csv',index= False)
```

Data Analysis

Using 5 W Analysis to understand the accidents data

Question to asnswer

5W 1H

what = Material (What make of airplanes reported most accidents)

why = why do accidents happen? root cause

when = The time when accidents occurred

where = where did the accidents mostly occured ?location, country

who = who was involved ,fatity levels

how = how does it happen?when do most accidents occur

In [31]:
```
#plane make that get damaged most = "What"
make_counts=df2['Make'].value_counts()
make_counts
```

Out[31]:
```
Cessna             22227
Piper              12029
CESSNA              4922
Beech               4330
PIPER               2841
                   ...
BOYD BRUCE             1
Casten                 1
SEACE DAVID A          1
POWERCHUTE             1
Gray Jim Robert        1
Name: Make, Length: 8237, dtype: int64
```

In [40]:
```
Model_values = df2['Model'].value_counts()

Model_values
```

Out[40]:
```
152                2367
172                1756
172N               1164
PA-28-140           932
150                 829
                   ...
DUNCAN/VARIEZE        1
B 206 SERIES 1        1
Airborne Edge-X       1
Renegade II           1
28                    1
Name: Model, Length: 12318, dtype: int64
```

In [32]:
```
# when do most accidents occur
whend= df2['Broad.phase.of.flight'].value_counts()
whend
```

Out[32]:
```
Unknown        27713
Landing        15428
Takeoff        12493
Cruise         10269
Maneuvering     8144
Approach        6546
Climb           2034
Taxi            1958
Descent         1887
Go-around       1353
Standing         945
Other            119
Name: Broad.phase.of.flight, dtype: int64
```

In [33]:
```
#where did we get most accidents?...USA
top_10_countries = df1["Country"].value_counts()
top_10_countries
```

Out[33]:
```
United States               82248
Brazil                        374
Canada                        359
Mexico                        358
United Kingdom                344
                             ...
Turks and Caicos Islands        1
Bosnia And Herzegovina          1
Anguilla                        1
Scotland                        1
Corsica                         1
Name: Country, Length: 219, dtype: int64
```

In [34]:
```
#seasons with hightest accidents="when"
seasons=df2['Weather.Condition'].value_counts()
seasons
```

Out[34]:
```
VMC    77303
IMC     5976
UNK     5348
Unk      262
Name: Weather.Condition, dtype: int64
```

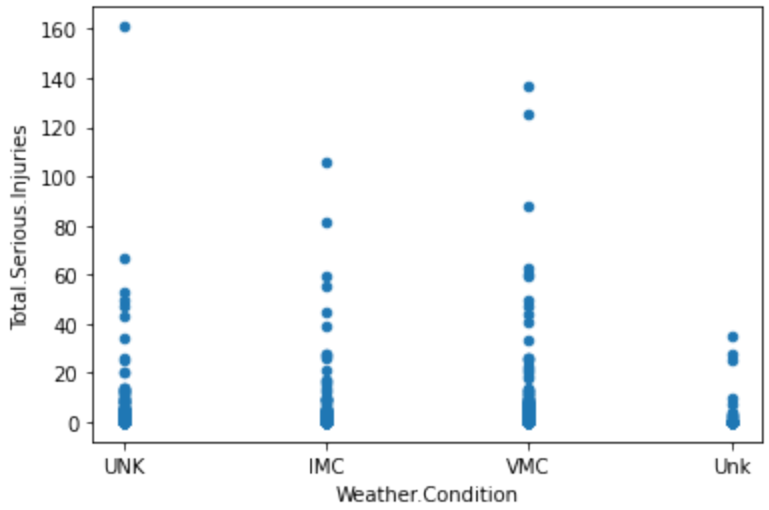
In [35]:
```
#setting "year" column as our indec
df2 = df2.set_index('year')
df2
```

Out[35]:

year	Event.Id	Investigation.Type	Location	Country	Make	Model	Aircraft.da
1948	20001218X45444	Accident	MOOSE CREEK, ID	United States	Stinson	108-3	Des
1962	20001218X45447	Accident	BRIDGEPORT, CA	United States	Piper	PA24-180	Des
1974	20061025X01555	Accident	Saltville, VA	United States	Cessna	172M	Des
1977	20001218X45448	Accident	EUREKA, CA	United States	Rockwell	112	Des
1979	20041105X01764	Accident	Canton, OH	United States	Cessna	501	Des
...	...	...	...	...	...	...	
2022	20221227106491	Accident	Annapolis, MD	United States	PIPER	PA-28-151	Un
2022	20221227106494	Accident	Hampton, NH	United States	BELLANCA	7ECA	Un
2022	20221227106497	Accident	Payson, AZ	United States	AMERICAN CHAMPION AIRCRAFT	8GCBC	Subs
2022	20221227106498	Accident	Morgan, UT	United States	CESSNA	210N	Un
2022	20221230106513	Accident	Athens, GA	United States	PIPER	PA-24-260	Un

88889 rows × 15 columns

In [36]:
```
df2.plot('Weather.Condition', 'Total.Serious.Injuries', kind='scatter');
```

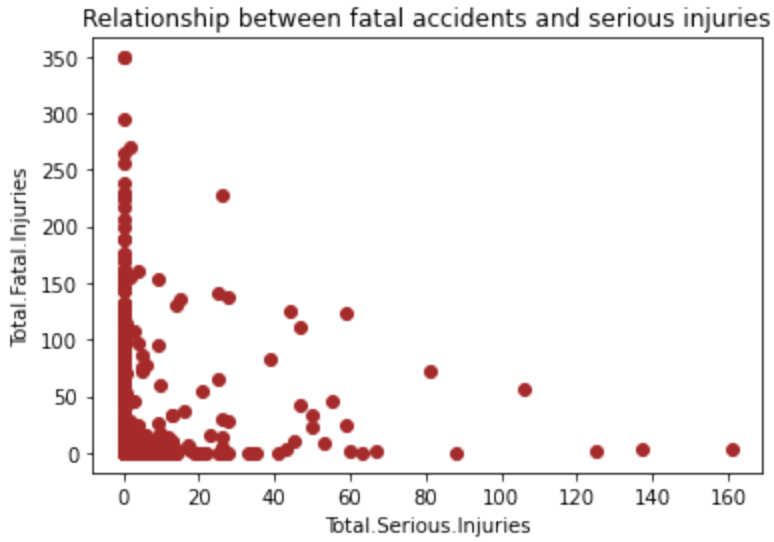


In [37]:
```
# Creating a scatter plot with specified marker size, color, and transparency
plt.scatter(df2['Total.Serious.Injuries'], df2['Total.Fatal.Injuries'],c='brow
n')

# Adding labels to the axes
plt.xlabel('Total.Serious.Injuries')
plt.ylabel('Total.Fatal.Injuries')

# Setting the title of the plot
plt.title('Relationship between fatal accidents and serious injuries')

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



In [ ]:

In [ ]:

In [ ]: