Business Understanding

Problem Statement : Skylard Ltd. is interested in purchasing and operating airplanes for commercial and private enterprises. The aim is to analyze aviation accidents data to provide insights and recommendations on which aircrafts are the lowest risk for the company.

Objectives: 1.Identify trends and patterns in aviation accident data to determine which aircraft models have the highest accident rates. 2.Evaluate injury severity by aircraft models to assess overall risk levels per model. 3.Develop recommendations on the safest aircraft models for fleet acquisition.

Data Understanding ¶

Dataset Source: National Transportation Safety Board (1962 – 2023); Size: 97,404 rows, 31 columns; Focus: Civil aviation accidents and incidents (US and international waters)

Data Cleaning

In [1]:
```
#Import the required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
#Reading the file
df = pd.read_csv("AviationData.csv", encoding='latin1')
```

```
C:\Users\billm\anaconda3\envs\learn-env\lib\site-packages\IPython\core\interactiveshell.py:3145: DtypeWarning:
Columns (6,7,28) have mixed types.Specify dtype option on import or set low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

In [3]:
```
#Checking top 5 columns
df.head()
```

Out[3]:

	Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Country	Latitude	Longitude	Airport.Code	Airport.Nar
0	20001218X45444	Accident	SEA87LA080	1948-10-24	MOOSE CREEK, ID	United States	NaN	NaN	NaN	N
1	20001218X45447	Accident	LAX94LA336	1962-07-19	BRIDGEPORT, CA	United States	NaN	NaN	NaN	N
2	20061025X01555	Accident	NYC07LA005	1974-08-30	Saltville, VA	United States	36.9222	-81.8781	NaN	N
3	20001218X45448	Accident	LAX96LA321	1977-06-19	EUREKA, CA	United States	NaN	NaN	NaN	N
4	20041105X01764	Accident	CHI79FA064	1979-08-02	Canton, OH	United States	NaN	NaN	NaN	N

5 rows × 31 columns

In [4]:
```
#checking the last 5 columns
df.tail()
```

Out[4]:

	Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Country	Latitude	Longitude	Airport.Code	Airport.Na
88884	20221227106491	Accident	ERA23LA093	2022-12-26	Annapolis, MD	United States	NaN	NaN	NaN	N
88885	20221227106494	Accident	ERA23LA095	2022-12-26	Hampton, NH	United States	NaN	NaN	NaN	N
88886	20221227106497	Accident	WPR23LA075	2022-12-26	Payson, AZ	United States	341525N	1112021W	PAN	PAYS
88887	20221227106498	Accident	WPR23LA076	2022-12-26	Morgan, UT	United States	NaN	NaN	NaN	N
88888	20221230106513	Accident	ERA23LA097	2022-12-29	Athens, GA	United States	NaN	NaN	NaN	N

5 rows × 31 columns

In [5]:
```
#checking the dataset shape
df.shape
```

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

#Observation: The dataset has 88,889 rows and 31 columns.

In [6]:
```
#checking the dataset 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 [7]:
```
#checking the info 2
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 [8]:
```
#checking the info 2
df.describe()
```

Out[8]:

	Number.of.Engines	Total.Fatal.Injuries	Total.Serious.Injuries	Total.Minor.Injuries	Total.Uninjured
count	82805.000000	77488.000000	76379.000000	76956.000000	82977.000000
mean	1.146585	0.647855	0.279881	0.357061	5.325440
std	0.446510	5.485960	1.544084	2.235625	27.913634
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.000000	0.000000	0.000000	0.000000	0.000000
50%	1.000000	0.000000	0.000000	0.000000	1.000000
75%	1.000000	0.000000	0.000000	0.000000	2.000000
max	8.000000	349.000000	161.000000	380.000000	699.000000

In [9]:
```
#concise summary statistics
df.describe().T
```

Out[9]:

	count	mean	std	min	25%	50%	75%	max
Number.of.Engines	82805.0	1.146585	0.446510	0.0	1.0	1.0	1.0	8.0
Total.Fatal.Injuries	77488.0	0.647855	5.485960	0.0	0.0	0.0	0.0	349.0
Total.Serious.Injuries	76379.0	0.279881	1.544084	0.0	0.0	0.0	0.0	161.0
Total.Minor.Injuries	76956.0	0.357061	2.235625	0.0	0.0	0.0	0.0	380.0
Total.Uninjured	82977.0	5.325440	27.913634	0.0	0.0	1.0	2.0	699.0

#Observation: The highest investigation type is accident in this dataset is accident. #Observation: The most common aircraft make and model in this dataset Cessna 152.

In [10]:
```
#Create a dataframe copy to be used in data cleaning
df1 = df.copy(deep=True)
```

In [11]:
```
for column in df1:
  unique_values = df1[column].unique()
  print(f"Unique values in column '{column}','\n': {unique_values}","\n')
```

```
Unique values in column 'Event.Id','
': ['20001218X45444' '20001218X45447' '20061025X01555' ... '20221227106497'
 '20221227106498' '20221230106513']

Unique values in column 'Investigation.Type','
': ['Accident' 'Incident']

Unique values in column 'Accident.Number','
': ['SEA87LA080' 'LAX94LA336' 'NYC07LA005' ... 'WPR23LA075' 'WPR23LA076'
 'ERA23LA097']

Unique values in column 'Event.Date','
': ['1948-10-24' '1962-07-19' '1974-08-30' ... '2022-12-22' '2022-12-26'
 '2022-12-29']

Unique values in column 'Location','
': ['MOOSE CREEK, ID' 'BRIDGEPORT, CA' 'Saltville, VA' ... 'San Manual, AZ'
 'Auburn Hills, MI' 'Brasnorte, ']
```

Checking duplicates and dropping all unnecessary columns

In [12]:
```
#drop the duplicates
df1=df1.drop_duplicates()
df1
```

Out[12]:

	Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Country	Latitude	Longitude	Airport.Code	Airpo
0	20001218X45444	Accident	SEA87LA080	1948-10-24	MOOSE CREEK, ID	United States	NaN	NaN	NaN	
1	20001218X45447	Accident	LAX94LA336	1962-07-19	BRIDGEPORT, CA	United States	NaN	NaN	NaN	
2	20061025X01555	Accident	NYC07LA005	1974-08-30	Saltville, VA	United States	36.9222	-81.8781	NaN	
3	20001218X45448	Accident	LAX96LA321	1977-06-19	EUREKA, CA	United States	NaN	NaN	NaN	
4	20041105X01764	Accident	CHI79FA064	1979-08-02	Canton, OH	United States	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	...	...	
88884	20221227106491	Accident	ERA23LA093	2022-12-26	Annapolis, MD	United States	NaN	NaN	NaN	
88885	20221227106494	Accident	ERA23LA095	2022-12-26	Hampton, NH	United States	NaN	NaN	NaN	
88886	20221227106497	Accident	WPR23LA075	2022-12-26	Payson, AZ	United States	341525N	1112021W	PAN	F
88887	20221227106498	Accident	WPR23LA076	2022-12-26	Morgan, UT	United States	NaN	NaN	NaN	
88888	20221230106513	Accident	ERA23LA097	2022-12-29	Athens, GA	United States	NaN	NaN	NaN	

88889 rows × 31 columns

In [13]:
```
#check the colunms
df1.columns
```

Out[13]:
```
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.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'],
      dtype='object')
```

In [14]:
```
#Drop unnessary columns
df1.drop(['Latitude', 'Longitude','Airport.Code','Airport.Name', 'Schedule', 'Accident.Number', 'Registration.
```

In [15]:
```
df1.columns
```

Out[15]:
```
Index(['Event.Id', 'Investigation.Type', 'Event.Date', 'Location', 'Country',
       'Injury.Severity', 'Aircraft.damage', 'Aircraft.Category', 'Make',
       'Model', 'Amateur.Built', 'Number.of.Engines', 'Engine.Type',
       'Purpose.of.flight', 'Air.carrier', 'Total.Fatal.Injuries',
       'Total.Serious.Injuries', 'Total.Minor.Injuries', 'Total.Uninjured',
       'Weather.Condition', 'Broad.phase.of.flight', 'Publication.Date'],
      dtype='object')
```

#Drop additional columns and remain with those most relevant to the analysis.

In [16]:
```
#Drop unnessary columns
df1.drop(['Location', 'Air.carrier','Injury.Severity','Publication.Date'], axis = 1, inplace=True)
```

Filling in null values

In [17]:
```
#check the missing values
df1.isnull().sum()
```

Out[17]:
```
Event.Id                    0
Investigation.Type          0
Event.Date                  0
Country                   226
Aircraft.damage          3194
Aircraft.Category       56602
Make                       63
Model                      92
Amateur.Built             102
Number.of.Engines        6084
Engine.Type              7077
Purpose.of.flight        6192
Total.Fatal.Injuries    11401
Total.Serious.Injuries  12510
Total.Minor.Injuries    11933
Total.Uninjured          5912
Weather.Condition        4492
Broad.phase.of.flight   27165
dtype: int64
```

Filling in categorical variables using mode

In [18]:
```
# filling in categorical variables with the mode
country_mode =df1.Country.mode()
country_mode
```

Out[18]:
```
0    United States
dtype: object
```

In [19]:
```
df1["Country"].fillna(country_mode[0], inplace=True)
```

In [20]:
```
aircraftdamage_mode =df1["Aircraft.damage"].mode()
aircraftdamage_mode
```

Out[20]:
```
0    Substantial
dtype: object
```

In [21]:
```
df1["Aircraft.damage"].fillna(aircraftdamage_mode[0], inplace=True)
```

In [22]:
```
aircraftcategory_mode =df1["Aircraft.Category"].mode()
aircraftcategory_mode
```

Out[22]:
```
0    Airplane
dtype: object
```

In [23]:
```
df1["Aircraft.Category"].fillna(aircraftcategory_mode[0], inplace=True)
```

In [24]:
```
make_mode =df1.Make.mode()
make_mode
```

Out[24]:
```
0    Cessna
dtype: object
```

In [25]:
```
df1["Make"].fillna(make_mode[0], inplace=True)
```

In [26]:
```
model_mode =df1.Model.mode()
model_mode
```

Out[26]:
```
0    152
dtype: object
```

In [27]:
```
df1["Model"].fillna(model_mode[0], inplace=True)
```

In [28]:
```
amateurbuilt_mode =df1["Amateur.Built"].mode()
amateurbuilt_mode
```

Out[28]:
```
0    No
dtype: object
```

In [29]:
```
df1["Amateur.Built"].fillna(amateurbuilt_mode[0], inplace=True)
```

In [30]:
```
enginetype_mode =df1["Engine.Type"].mode()
enginetype_mode
```

Out[30]:
```
0    Reciprocating
dtype: object
```

In [31]:
```
df1["Engine.Type"].fillna(enginetype_mode[0], inplace=True)
```

In [32]:
```
purposeofflight_mode =df1["Purpose.of.flight"].mode()
purposeofflight_mode
```

Out[32]:
```
0    Personal
dtype: object
```

In [33]:
```
df1["Purpose.of.flight"].fillna(purposeofflight_mode[0], inplace=True)
```

In [34]:
```
weathercondition_mode =df1["Weather.Condition"].mode()
weathercondition_mode
```

Out[34]:
```
0    VMC
dtype: object
```

In [35]:
```
df1["Weather.Condition"].fillna(weathercondition_mode[0], inplace=True)
```

In [36]:
```
broadphaseofflight_mode =df1["Broad.phase.of.flight"].mode()
broadphaseofflight_mode
```

Out[36]:
```
0    Landing
dtype: object
```

In [37]:
```
df1["Broad.phase.of.flight"].fillna(broadphaseofflight_mode[0], inplace=True)
```

In [38]:
```
#check the if the missing values in the categorical columns have been filled
df1.isnull().sum()
```

Out[38]:
```
Event.Id                    0
Investigation.Type          0
Event.Date                  0
Country                     0
Aircraft.damage             0
Aircraft.Category           0
Make                        0
Model                       0
Amateur.Built               0
Number.of.Engines        6084
Engine.Type                 0
Purpose.of.flight           0
Total.Fatal.Injuries    11401
Total.Serious.Injuries  12510
Total.Minor.Injuries    11933
Total.Uninjured          5912
Weather.Condition           0
Broad.phase.of.flight       0
dtype: int64
```

In [39]:
```
df1['Make'].unique()
```

Out[39]:
```
array(['Stinson', 'Piper', 'Cessna', ..., 'JAMES R DERNOVSEK',
       'ORLICAN S R O', 'ROYSE RALPH L'], dtype=object)
```

In [40]:
```
#Replacing values in the model column
print (df ['Make'].value_counts())
```

```
Cessna                          22227
Piper                           12029
CESSNA                           4922
Beech                            4330
PIPER                            2841
                                ...
LEE PAUL                            1
Tubbs S/Performance Air Inc         1
WOGENSTAHL WALLACE L                1
HENRIE RAYMOND                      1
Drlik                               1
Name: Make, Length: 8237, dtype: int64
```

In [41]:
```
#replace uppercase make names with sentece case
df1['Make'] = df1['Make'].str.replace('CESSNA','Cessna')
df1['Make'] = df1['Make'].str.replace('PIPER','Piper')
df1['Make'] = df1['Make'].str.replace('BEECH','Beech')
df1['Make'] = df1['Make'].str.replace('BELL','Bell')
df1['Make'] = df1['Make'].str.replace('BOEING','Boeing')
df1['Make'] = df1['Make'].str.replace('AIRBUS','Airbus')

print (df1 ['Make'].value_counts())
```

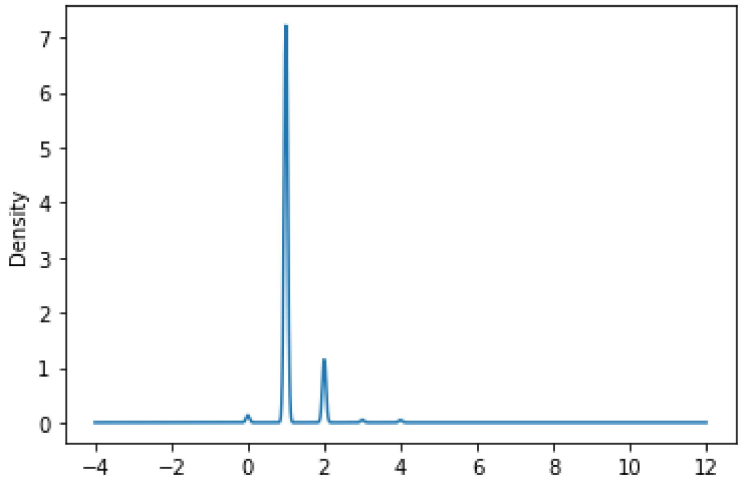
```
Cessna                            27212
Piper                             14870
Beech                              5372
Boeing                             2745
Bell                               2722
                                  ...
RICHARD LACOURSE                      1
AMS FLIGHT                            1
COBALT AIRCRAFT INDUSTRIES INC        1
John Thomason                         1
Drlik                                 1
Name: Make, Length: 8230, dtype: int64
```

Filling in numerical variables using mean/median

In [42]:
```
#check data types
df1.dtypes
```

Out[42]:
```
Event.Id                   object
Investigation.Type         object
Event.Date                 object
Country                    object
Aircraft.damage            object
Aircraft.Category          object
Make                       object
Model                      object
Amateur.Built              object
Number.of.Engines         float64
Engine.Type                object
Purpose.of.flight          object
Total.Fatal.Injuries      float64
Total.Serious.Injuries    float64
Total.Minor.Injuries      float64
Total.Uninjured           float64
Weather.Condition          object
Broad.phase.of.flight      object
dtype: object
```

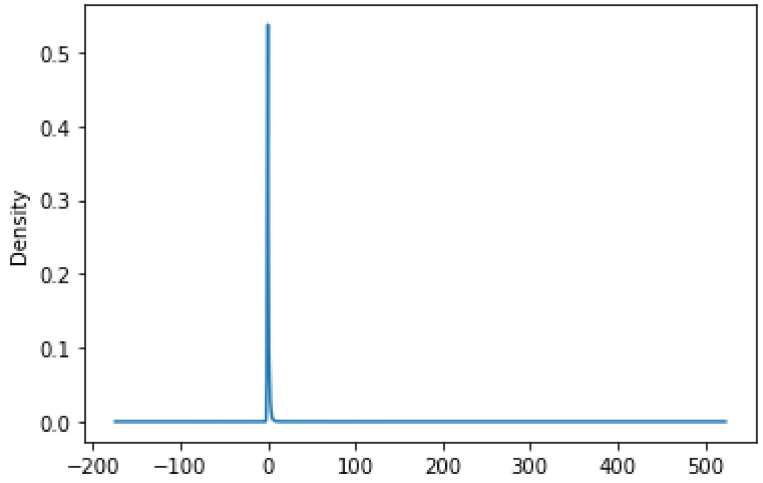
In [43]:
```
df1['Number.of.Engines'].plot.kde()
plt.show()
```



#Observation: Data is skewed, use median to fill in the null values

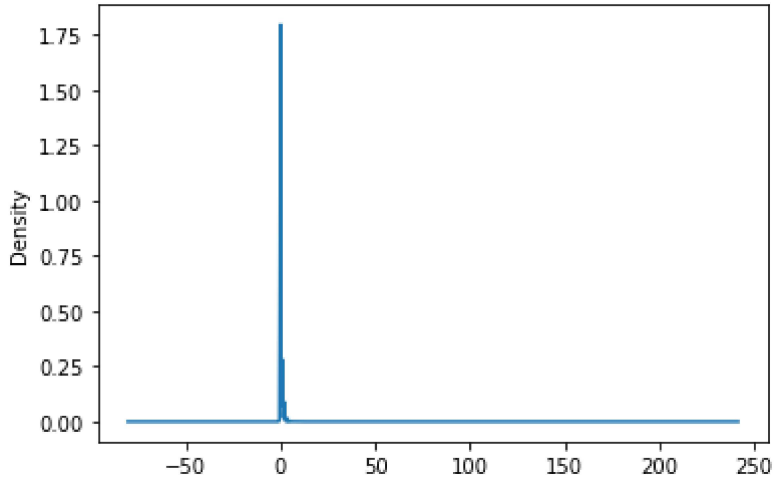
In [44]:
```
#Get the median
median_values = df1['Number.of.Engines'].median()

#fill up the missing value
df1['Number.of.Engines'].fillna(median_values, inplace=True)
```

In [45]:
```
df1['Total.Fatal.Injuries'].plot.kde()
plt.show()
```



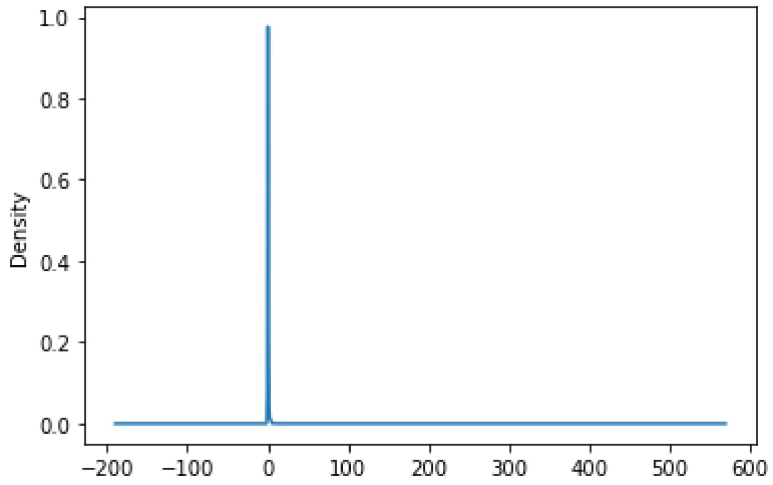
In [46]:
```
#Get the median
median_values = df1['Total.Fatal.Injuries'].median()

#fill up the missing value
df1['Total.Fatal.Injuries'].fillna(median_values, inplace=True)
```

In [47]:
```
df1['Total.Serious.Injuries'].plot.kde()
plt.show()
```



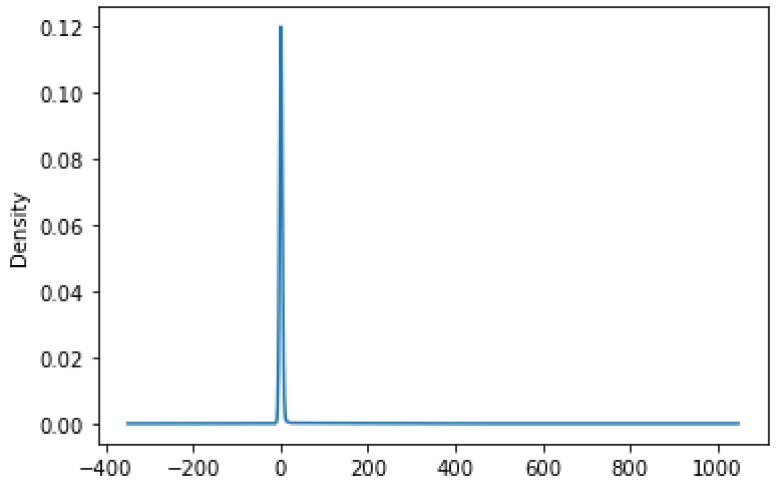
In [48]:
```
#Get the median
median_values = df1['Total.Serious.Injuries'].median()

#fill up the missing value
df1['Total.Serious.Injuries'].fillna(median_values, inplace=True)
```

In [49]:
```
df1['Total.Minor.Injuries'].plot.kde()
plt.show()
```



In [50]:
```
#Get the median
median_values = df1['Total.Minor.Injuries'].median()

#fill up the missing value
df1['Total.Minor.Injuries'].fillna(median_values, inplace=True)
```

In [51]:
```
df1['Total.Uninjured'].plot.kde()
plt.show()
```



In [52]:
```
#Get the median
median_values = df1['Total.Uninjured'].median()

#fill up the missing value
df1['Total.Uninjured'].fillna(median_values, inplace=True)
```

In [53]:
```
#check if there are any missing values
df1.isnull().sum()
```

Out[53]:
```
Event.Id                0
Investigation.Type      0
Event.Date              0
Country                 0
Aircraft.damage         0
Aircraft.Category       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
Broad.phase.of.flight   0
dtype: int64
```

Check for duplicates

In [54]:
```
#check for duplicates
df1.duplicated().sum()
```

Out[54]:
```
29
```

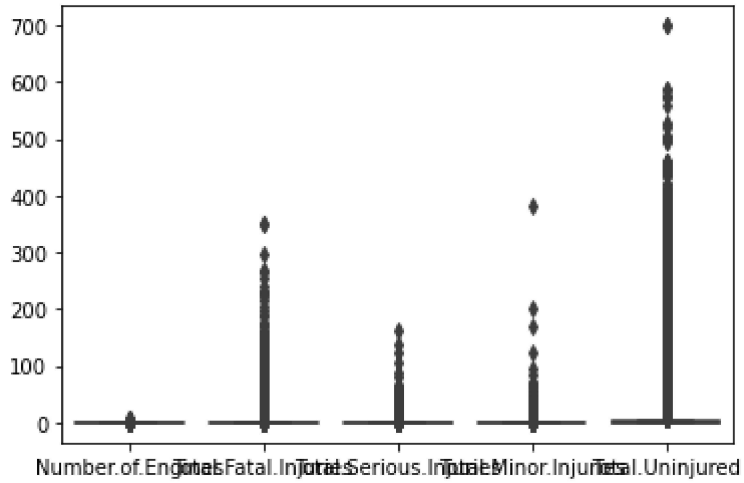
In [55]:
```
#drop the duplicates
df1.drop_duplicates(inplace=True)

#check
df1.duplicated().sum()
```

Out[55]:
```
0
```

Check for outliers

In [56]:
```
#checking for outliers
sns.boxplot(data=df1)
```

Out[56]: <AxesSubplot:>



In [ ]:
```
#Observation: Outlier in total minor injured column
```

In [57]:
```
max = df1['Total.Minor.Injuries'].quantile(0.995)
max
```

Out[57]: 5.0

In [58]:
```
df1[df1["Total.Minor.Injuries"] > max]
```

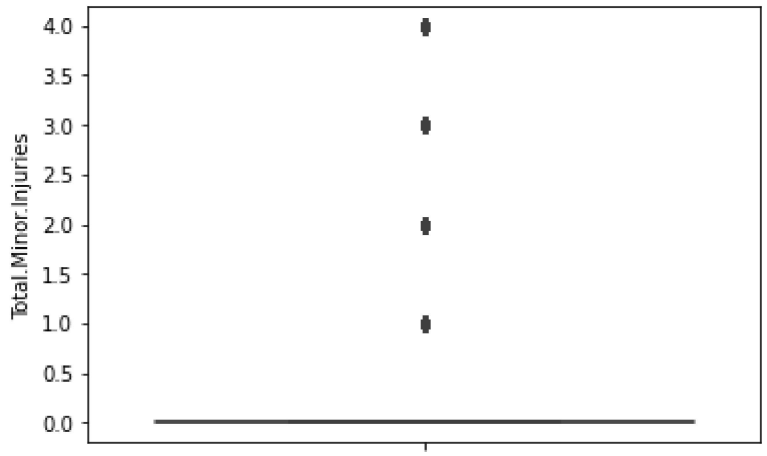
Out[58]:

	Event.Id	Investigation.Type	Event.Date	Country	Aircraft.damage	Aircraft.Category	Make	Model	Amateur.Built	Numb
155	20020917X01909	Accident	1982-01-23	United States	Destroyed	Airplane	Mcdonnell Douglas	DC-10-30	No	
229	20020917X01923	Accident	1982-02-03	United States	Destroyed	Airplane	Mitsubishi	MU2B-40	No	
1343	20020917X03653	Accident	1982-05-26	United States	Substantial	Airplane	Douglas	DC10-10	No	
1347	20020917X03237	Incident	1982-05-27	United States	Substantial	Airplane	Douglas	DC-8-61	No	
1969	20020917X03515	Accident	1982-07-16	United States	Minor	Airplane	Mcdonnell Douglas	DC-10-10	No	
...	...	...	...	...	...	...	...	...	...	
84456	20200205X32547	Accident	2020-02-05	Turkey	Substantial	Airplane	Boeing	737	No	
85473	20201008102116	Accident	2020-10-08	United States	Substantial	Airplane	Cessna	414	No	
86814	20210916103879	Accident	2021-08-28	Spain	Unknown	Airplane	BRITTEN NORMAN	BN2A	No	
86864	20210909103831	Accident	2021-09-09	United States	Substantial	Airplane	Cessna	402C	No	
87788	20220519105100	Accident	2022-05-11	Brazil	Substantial	Airplane	Cessna	208	No	

320 rows × 18 columns

In [59]:
```
df2 = df1[df1["Total.Minor.Injuries"] < max]
sns.boxplot(y='Total.Minor.Injuries', data=df2)
```

Out[59]: <AxesSubplot:ylabel='Total.Minor.Injuries'>



In [60]:
```
#save the clean dataset
df2.to_csv("Cleaned_Aircraft_Risk_Analysis.csv", index=False)
```

In [61]:
```
data = pd.read_csv("Cleaned_Aircraft_Risk_Analysis.csv")
data.head()
```

Out[61]:

	Event.Id	Investigation.Type	Event.Date	Country	Aircraft.damage	Aircraft.Category	Make	Model	Amateur.Built	Number.of.E
0	20001218X45444	Accident	1948-10-24	United States	Destroyed	Airplane	Stinson	108-3	No	
1	20001218X45447	Accident	1962-07-19	United States	Destroyed	Airplane	Piper	PA24-180	No	
2	20061025X01555	Accident	1974-08-30	United States	Destroyed	Airplane	Cessna	172M	No	
3	20001218X45448	Accident	1977-06-19	United States	Destroyed	Airplane	Rockwell	112	No	
4	20041105X01764	Accident	1979-08-02	United States	Destroyed	Airplane	Cessna	501	No	

In [62]:
```
data.shape
```

Out[62]:
```
(88411, 18)
```

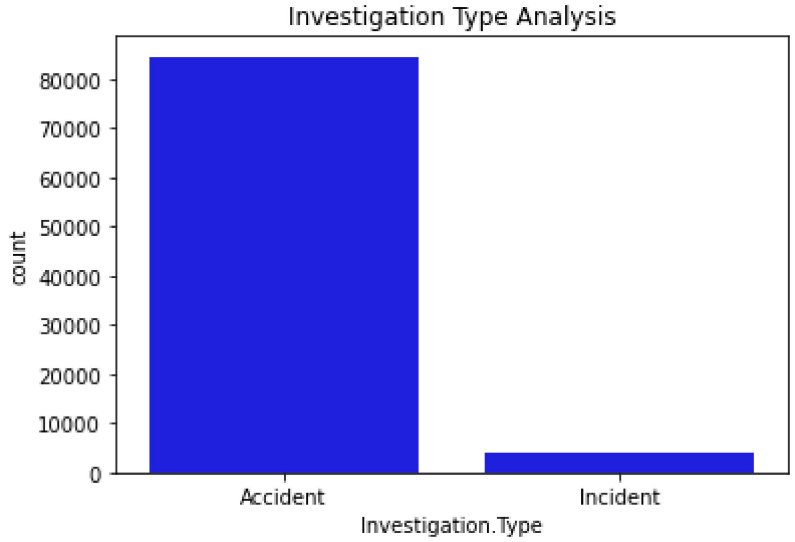
Exploratory Data Analysis

Univariate Analysis

In [63]:
```
#which investigation type has the highest records

investigation_count = data["Investigation.Type"].value_counts()
investigation_count
```

Out[63]:
```
Accident    84588
Incident     3823
Name: Investigation.Type, dtype: int64
```

In [64]:
```
sns.countplot(x= data["Investigation.Type"], color= "b")
plt.title("Investigation Type Analysis")
plt.show()
```

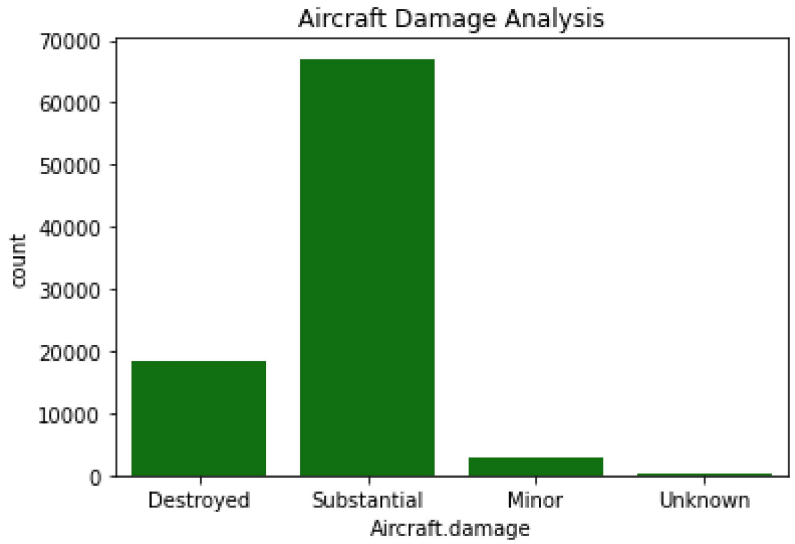


#Observation: Highest investigation type is accidents.

In [66]:
```
#which investigation type has the highest records

damage_count = data["Aircraft.damage"].value_counts()
damage_count
```

Out[66]:
```
Substantial    67051
Destroyed      18486
Minor           2756
Unknown          118
Name: Aircraft.damage, dtype: int64
```

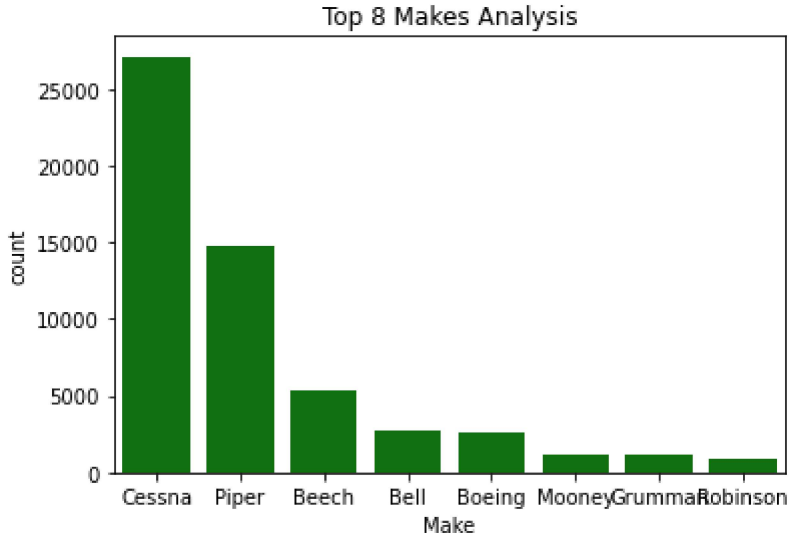
In [67]:
```
sns.countplot(x= data["Aircraft.damage"], color= "g")
plt.title("Aircraft Damage Analysis")
plt.show()
```



#Observation: Most aircrafts in the dataset had substantial damage.

In [86]:

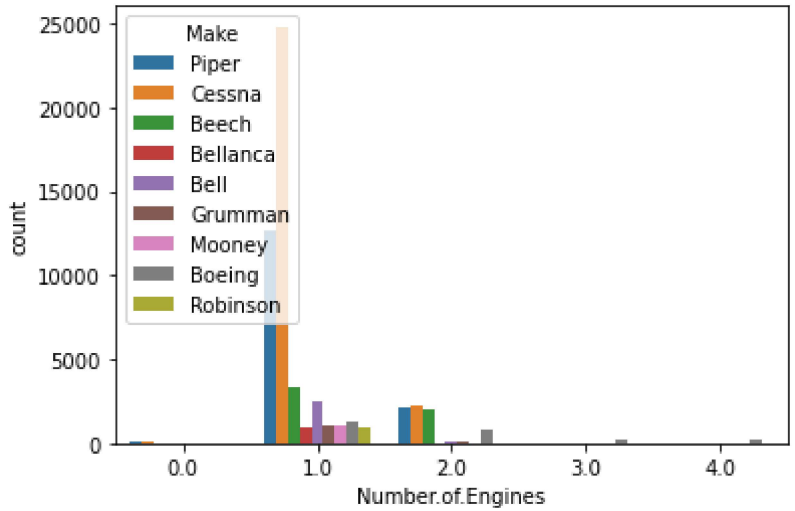
```
top_8_makes = data["Make"].value_counts().head(8).index
relevant_data = data[data["Make"].isin(top_8_makes)]

sns.countplot(x=relevant_data["Make"], order=top_8_makes, color="g")
plt.title("Top 8 Makes Analysis")
plt.show()
```



Bivariate Analysis

In [84]:
```
#bivariate analysis
#number of engines, make
sns.countplot(x = "Number.of.Engines",  hue="Make", data=relevant_data)
```

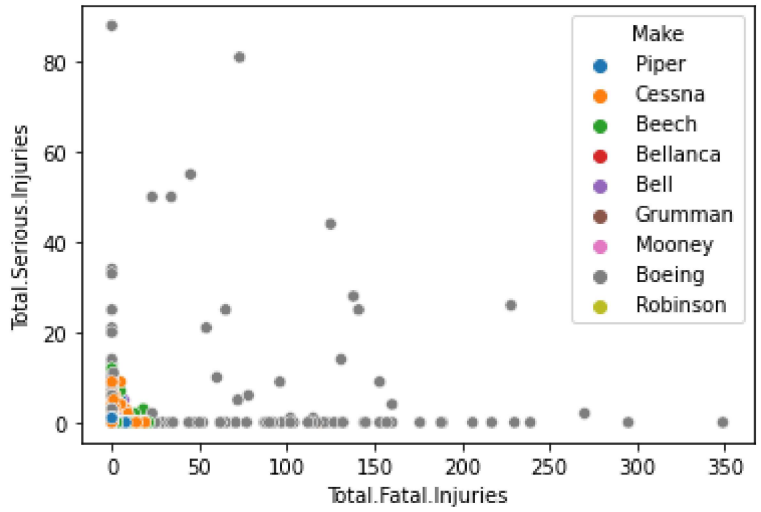
Out[84]: <AxesSubplot:xlabel='Number.of.Engines', ylabel='count'>



Multivariate Analysis

In [85]:
```
#multivariate analysis
#total fatal injuries, total serious injuries, make

sns.scatterplot(x="Total.Fatal.Injuries", y="Total.Serious.Injuries", hue="Make", data=relevant_data)
```

Out[85]: <AxesSubplot:xlabel='Total.Fatal.Injuries', ylabel='Total.Serious.Injuries'>



#Observation: The makes with the highest total fatal and serious injuries are Boeing,Cessna and Beech.