PROJECT STUDY

1. OBJECTIVES

-Get to under the data inself by having a general overview on excel sheet. -Understanding of dataset -Data cleaning -stastical overview to understand the insights -visualization based on the problem statement 2. Conclusions and recommendations NB. All guided by the problem statement.

DATA UNDERSTANDING.

In [3]:
```
#Importing libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
```

In [4]:
```
#Loading data
df = pd.read_csv('AviationData.csv', encoding='latin1', low_memory= False)
```

In [5]:
```
#Checking rows and columns
df.shape
```

Out[5]: (88889, 31)

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

Out[6]:

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

5 rows × 31 columns

In [7]:

```
#checking the tail
df.tail()
```

Out[7]:

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

5 rows × 31 columns

In [8]:

```
#checking on the datatypes
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 [9]:
```
#list of columns
df.columns
```

Out[9]:
```
Index(['Event.Id', 'Investigation.Type', 'Accident.Number', 'Event.Dat
e',
       'Location', 'Country', 'Latitude', 'Longitude', 'Airport.Code',
       'Airport.Name', 'Injury.Severity', 'Aircraft.damage',
       'Aircraft.Category', 'Registration.Number', 'Make', 'Model',
       'Amateur.Built', 'Number.of.Engines', 'Engine.Type', 'FAR.Descrip
tion',
       'Schedule', 'Purpose.of.flight', 'Air.carrier', 'Total.Fatal.Inju
ries',
       'Total.Serious.Injuries', 'Total.Minor.Injuries', 'Total.Uninjure
d',
       'Weather.Condition', 'Broad.phase.of.flight', 'Report.Status',
       'Publication.Date'],
      dtype='object')
```

In [10]:
```
#stastistical summaries
df.describe()
```

Out[10]:

	Number.of.Engines	Total.Fatal.Injuries	Total.Serious.Injuries	Total.Minor.Injuries	Tota
count	82805.000000	77488.000000	76379.000000	76956.000000	82
mean	1.146585	0.647855	0.279881	0.357061	
std	0.446510	5.485960	1.544084	2.235625	
min	0.000000	0.000000	0.000000	0.000000	
25%	1.000000	0.000000	0.000000	0.000000	
50%	1.000000	0.000000	0.000000	0.000000	
75%	1.000000	0.000000	0.000000	0.000000	
max	8.000000	349.000000	161.000000	380.000000	

DATA PREPARATION

In [11]:
```
df['Event.Date'] = pd.to_datetime(df['Event.Date'])
```

In [12]:
```
df['Year'] = df['Event.Date'].dt.year
df['Month.Abbr'] = df['Event.Date'].dt.month_name().str[:3]
df['Day.Name.Abbr'] = df['Event.Date'].dt.day_name().str[:3]
df.head()
```

Out[12]:

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

5 rows × 34 columns

In [13]:
```
#checking if there are duplicates
df.duplicated().sum()
```

Out[13]: 0

In [14]:

```
#Checking for missing values
df.isna().sum()
```

Out[14]:

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

Realized we have columns having more that 50% of data missing. Cleaning this I am going to drop columns that do not meet 50% mark

In [15]:

```
#Dropping columns that do not hit that 50% mark
df = df.dropna(axis=1, thresh=0.5 * df.shape[0])
print(df.columns)
```

```
Index(['Event.Id', 'Investigation.Type', 'Accident.Number', 'Event.Dat
e',
       'Location', 'Country', 'Airport.Code', 'Airport.Name',
       'Injury.Severity', 'Aircraft.damage', 'Registration.Number', 'Mak
e',
       'Model', 'Amateur.Built', 'Number.of.Engines', 'Engine.Type',
       'Purpose.of.flight', 'Total.Fatal.Injuries', 'Total.Serious.Injur
ies',
       'Total.Minor.Injuries', 'Total.Uninjured', 'Weather.Condition',
       'Broad.phase.of.flight', 'Report.Status', 'Publication.Date', 'Ye
ar',
       'Month.Abbr', 'Day.Name.Abbr'],
      dtype='object')
```

In [16]:

```
#checking the effect of the code above
#successfully dropped 6 columns
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 88889 entries, 0 to 88888
Data columns (total 28 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  datetime64[ns]
 4   Location                88837 non-null  object        
 5   Country                 88663 non-null  object        
 6   Airport.Code            50249 non-null  object        
 7   Airport.Name            52790 non-null  object        
 8   Injury.Severity         87889 non-null  object        
 9   Aircraft.damage         85695 non-null  object        
 10  Registration.Number     87572 non-null  object        
 11  Make                    88826 non-null  object        
 12  Model                   88797 non-null  object        
 13  Amateur.Built           88787 non-null  object        
 14  Number.of.Engines       82805 non-null  float64       
 15  Engine.Type             81812 non-null  object        
 16  Purpose.of.flight       82697 non-null  object        
 17  Total.Fatal.Injuries    77488 non-null  float64       
 18  Total.Serious.Injuries  76379 non-null  float64       
 19  Total.Minor.Injuries    76956 non-null  float64       
 20  Total.Uninjured         82977 non-null  float64       
 21  Weather.Condition       84397 non-null  object        
 22  Broad.phase.of.flight   61724 non-null  object        
 23  Report.Status           82508 non-null  object        
 24  Publication.Date        75118 non-null  object        
 25  Year                    88889 non-null  int64         
 26  Month.Abbr              88889 non-null  object        
 27  Day.Name.Abbr           88889 non-null  object        
dtypes: datetime64[ns](1), float64(5), int64(1), object(21)
memory usage: 19.0+ MB
```

In [17]:
```
df.isnull().sum()
```

Out[17]:
```
Event.Id                      0
Investigation.Type            0
Accident.Number               0
Event.Date                    0
Location                     52
Country                     226
Airport.Code              38640
Airport.Name              36099
Injury.Severity            1000
Aircraft.damage            3194
Registration.Number        1317
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
Report.Status              6381
Publication.Date          13771
Year                          0
Month.Abbr                    0
Day.Name.Abbr                 0
dtype: int64
```

In [18]:

```
#I went a head and dropped more columns with missing values
df.drop(['Airport.Code', 'Airport.Name'], axis=1, inplace=True)
df.isnull().sum()
```

Out[18]:

```
Event.Id                      0
Investigation.Type            0
Accident.Number               0
Event.Date                    0
Location                     52
Country                     226
Injury.Severity            1000
Aircraft.damage            3194
Registration.Number        1317
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
Report.Status              6381
Publication.Date          13771
Year                          0
Month.Abbr                    0
Day.Name.Abbr                 0
dtype: int64
```

In [19]:

```
#dropped a number of rows with more than 10 missing values
df = df.dropna(thresh=10)
df.shape
```

Out[19]:

```
(88889, 26)
```

In [20]:

```
#I needed to work on US data only
df = df[(df['Investigation.Type'] == 'Accident') & (df['Country'] == 'Unit
```

In [21]:

```
#i did this to check on the data in order to replace the missing values
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 79906 entries, 0 to 88888
Data columns (total 26 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Event.Id                79906 non-null  object
 1   Investigation.Type      79906 non-null  object
 2   Accident.Number         79906 non-null  object
 3   Event.Date              79906 non-null  datetime64[ns]
 4   Location                79895 non-null  object
 5   Country                 79906 non-null  object
 6   Injury.Severity         79854 non-null  object
 7   Aircraft.damage         78782 non-null  object
 8   Registration.Number     79903 non-null  object
 9   Make                    79894 non-null  object
 10  Model                   79877 non-null  object
 11  Amateur.Built           79891 non-null  object
 12  Number.of.Engines       78147 non-null  float64
 13  Engine.Type             77007 non-null  object
 14  Purpose.of.flight       78025 non-null  object
 15  Total.Fatal.Injuries    69641 non-null  float64
 16  Total.Serious.Injuries  68921 non-null  float64
 17  Total.Minor.Injuries    69551 non-null  float64
 18  Total.Uninjured         74911 non-null  float64
 19  Weather.Condition       79345 non-null  object
 20  Broad.phase.of.flight   59297 non-null  object
 21  Report.Status           77341 non-null  object
 22  Publication.Date        67649 non-null  object
 23  Year                    79906 non-null  int64
 24  Month.Abbr              79906 non-null  object
 25  Day.Name.Abbr           79906 non-null  object
dtypes: datetime64[ns](1), float64(5), int64(1), object(19)
memory usage: 16.5+ MB
```

In [22]:
```
#I replaced the columns with float values with mean
columns = ['Number.of.Engines', 'Total.Fatal.Injuries', 'Total.Minor.Injur
for col in columns:
    df[col].fillna(df[col].mean(), inplace=True)
print(df.isna().sum())
```

```
Event.Id                     0
Investigation.Type           0
Accident.Number              0
Event.Date                   0
Location                    11
Country                      0
Injury.Severity             52
Aircraft.damage           1124
Registration.Number          3
Make                        12
Model                       29
Amateur.Built               15
Number.of.Engines            0
Engine.Type               2899
Purpose.of.flight         1881
Total.Fatal.Injuries         0
Total.Serious.Injuries   10985
Total.Minor.Injuries         0
Total.Uninjured              0
Weather.Condition          561
Broad.phase.of.flight    20609
Report.Status             2565
Publication.Date         12257
Year                         0
Month.Abbr                   0
Day.Name.Abbr                0
dtype: int64
```

In [23]:
```
#Replace all the remaining columns with object entries witht the mode valu
columns = ['Location', 'Aircraft.damage', 'Registration.Number', 'Make',
           'Engine.Type', 'Purpose.of.flight', 'Total.Serious.Injuries',
           'Report.Status', 'Publication.Date','Broad.phase.of.flight', 'I
for col in columns:
    df[col].fillna(df[col].mode()[0], inplace=True)
print(df.isna().sum())
```

```
Event.Id                  0
Investigation.Type        0
Accident.Number           0
Event.Date                0
Location                  0
Country                   0
Injury.Severity           0
Aircraft.damage           0
Registration.Number       0
Make                      0
Model                     0
Amateur.Built             0
Number.of.Engines         0
Engine.Type               0
Purpose.of.flight         0
Total.Fatal.Injuries      0
Total.Serious.Injuries    0
Total.Minor.Injuries      0
Total.Uninjured           0
Weather.Condition         0
Broad.phase.of.flight     0
Report.Status             0
Publication.Date          0
Year                      0
Month.Abbr                0
Day.Name.Abbr             0
dtype: int64
```

In [24]:
```
#output cleaned data (saving to my folder)
df.to_csv('Cleaned_AviationData.csv', index=False)
```

In [25]:
```
df.head()
```

Out[25]:

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

5 rows × 26 columns

DATA VISUALIZATION

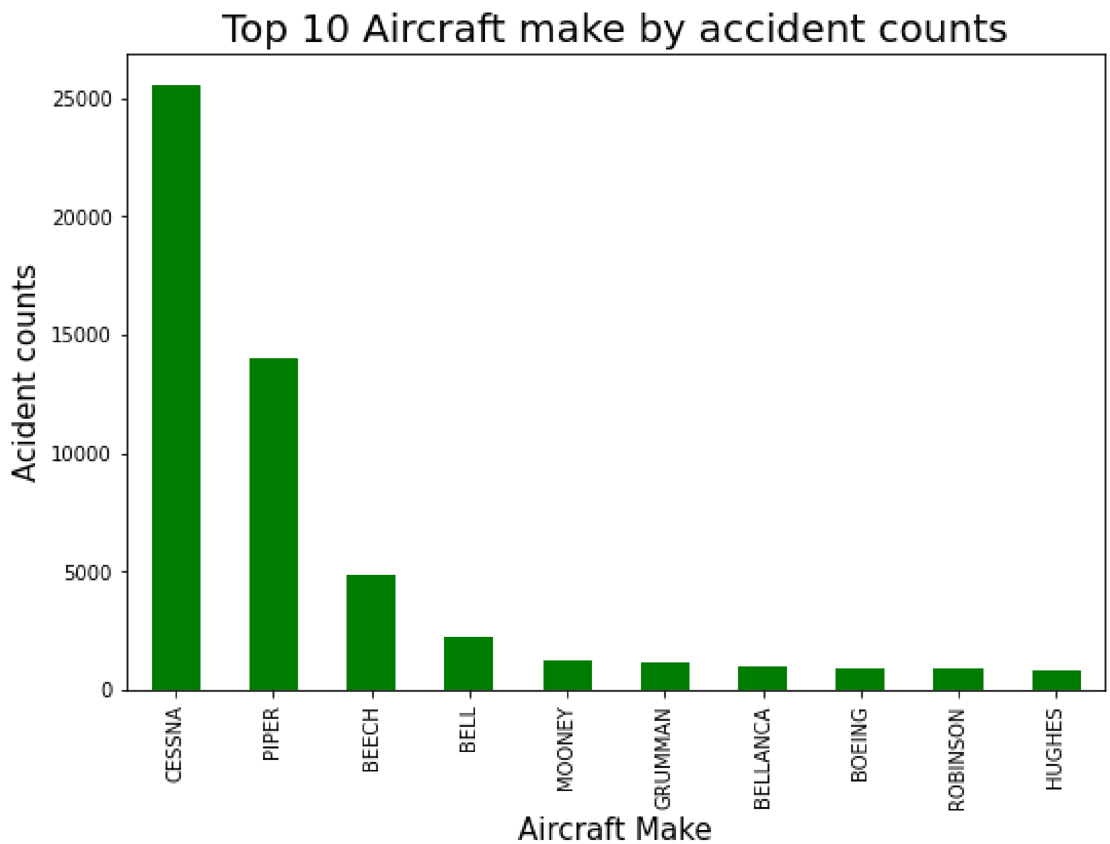
In [26]:
```
#checking on the correlation of our values
df.select_dtypes(include=['float64', 'int64']).corr()
```

Out[26]:

	Number.of.Engines	Total.Fatal.Injuries	Total.Serious.Injuries	Total.Mino
Number.of.Engines	1.000000	0.103510	0.055313	
Total.Fatal.Injuries	0.103510	1.000000	0.121479	
Total.Serious.Injuries	0.055313	0.121479	1.000000	
Total.Minor.Injuries	0.093307	0.087622	0.325106	
Total.Uninjured	0.295745	-0.014330	0.067038	
Year	-0.020822	-0.009538	0.024708	

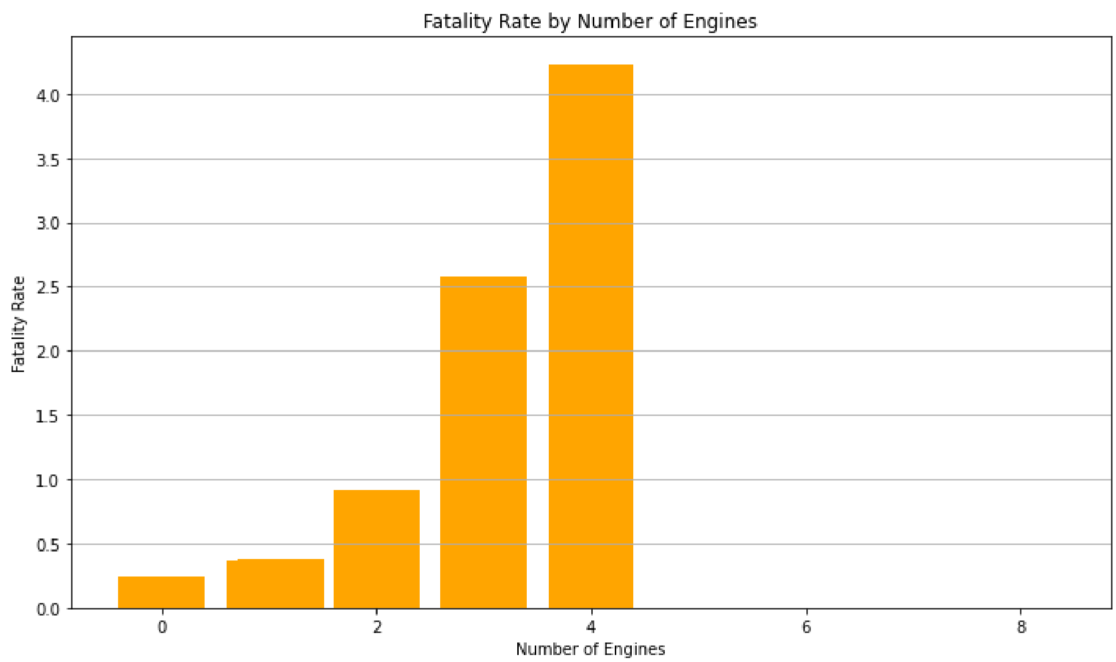
In [27]:
```
#visual plot on correlation
plt.figure(figsize=(10, 8))
sns.heatmap(df.select_dtypes(include=['float64', 'int64']).corr(), annot=
plt.title('Correlation Heatmap')
plt.show()
```

In [28]:
```
#Top 10 Aircraft make involved in the accidents
plt.figure(figsize=(9, 6))
df['Make'].str.upper().value_counts().sort_values(ascending=False)[:10].pl
plt.xlabel("Aircraft Make", size=15)
plt.ylabel("Acident counts", size=15)
plt.title("Top 10 Aircraft make by accident counts", size=20)
plt.show()
```



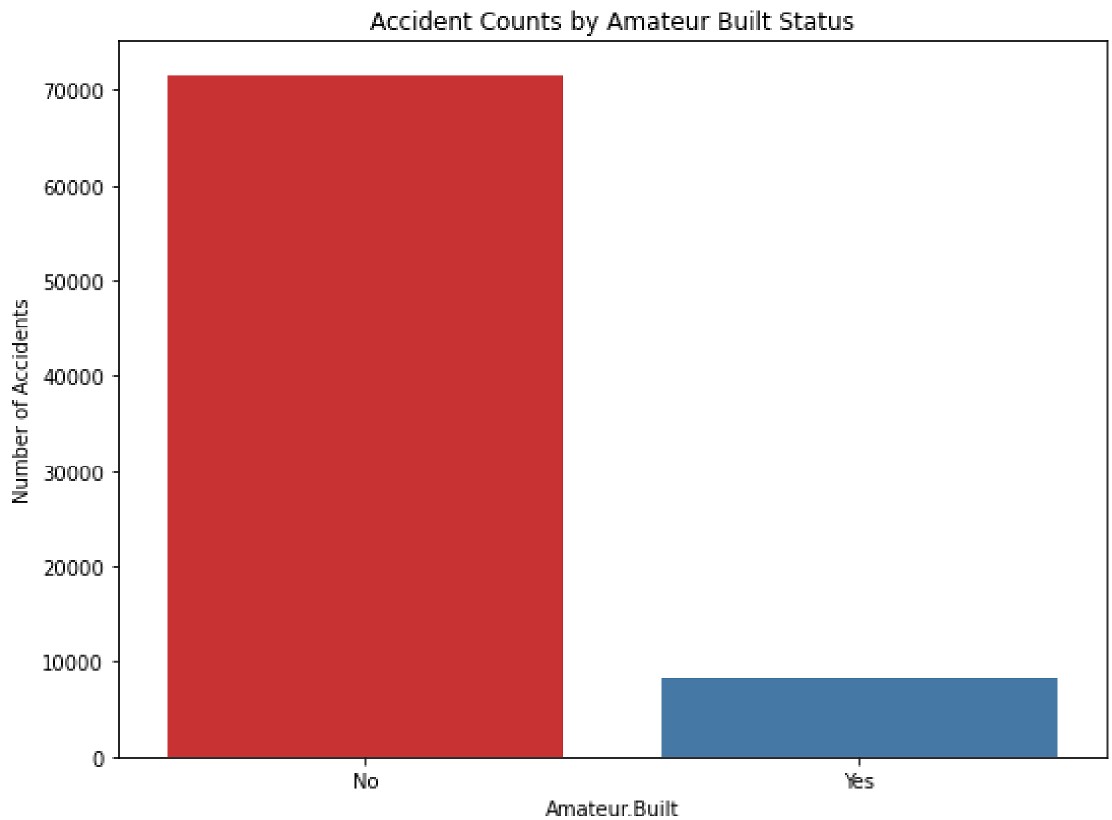
In [29]:

```
# getting a numerical group
df['Total.Fatal.Injuries'] = pd.to_numeric(df['Total.Fatal.Injuries'], err
df['Number.of.Engines'] = pd.to_numeric(df['Number.of.Engines'], errors='c

# 2. Group data by number of engines and calculate totals
engines_grouped = df.groupby('Number.of.Engines').agg(
    Total_Fatalities=('Total.Fatal.Injuries', 'sum'),
    Total_Incidents=('Event.Id', 'count')  # Assuming 'Event.Id' is unique
).reset_index()
# 3. Calculate fatality rate
engines_grouped['Fatality_Rate'] = engines_grouped['Total_Fatalities'] / e

# 4. Plot the fatality rate
plt.figure(figsize=(10, 6))
plt.bar(engines_grouped['Number.of.Engines'], engines_grouped['Fatality_Ra
plt.title('Fatality Rate by Number of Engines')
plt.xlabel('Number of Engines')
plt.ylabel('Fatality Rate')
plt.grid(axis='y')
plt.tight_layout()
```



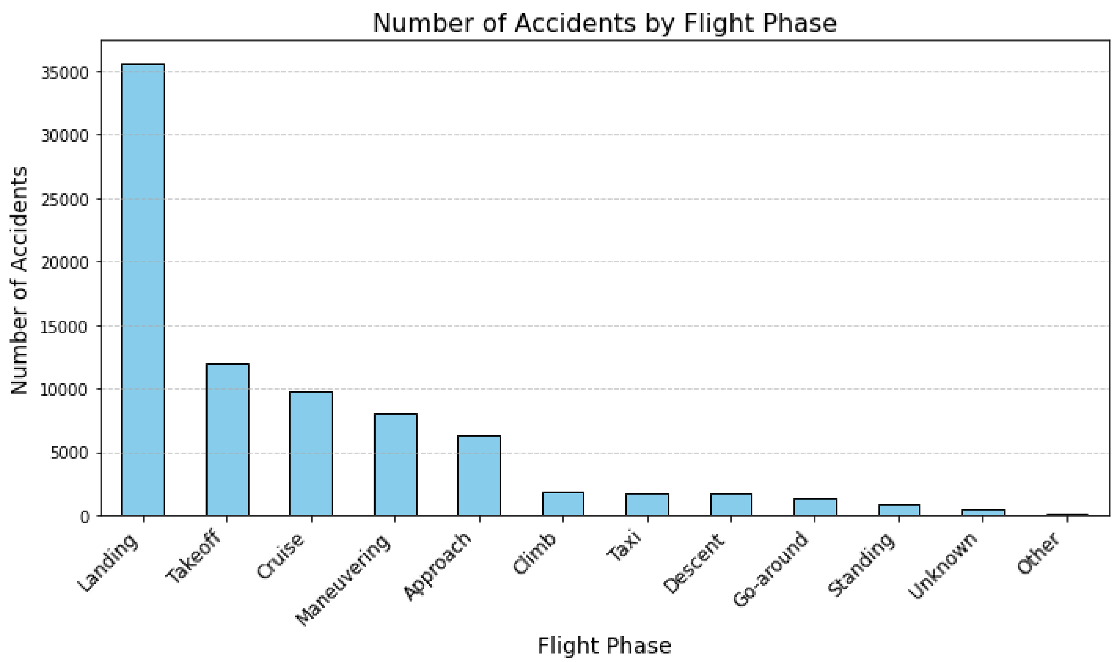
In [30]:

```
plt.figure(figsize=(8, 6))
amateur_built_counts=df['Amateur.Built'].value_counts()
sns.barplot(x=amateur_built_counts.index, y=amateur_built_counts.values, p
plt.xlabel('Amateur.Built')
plt.ylabel('Number of Accidents')
plt.title('Accident Counts by Amateur Built Status')
plt.tight_layout()
plt.show()
```

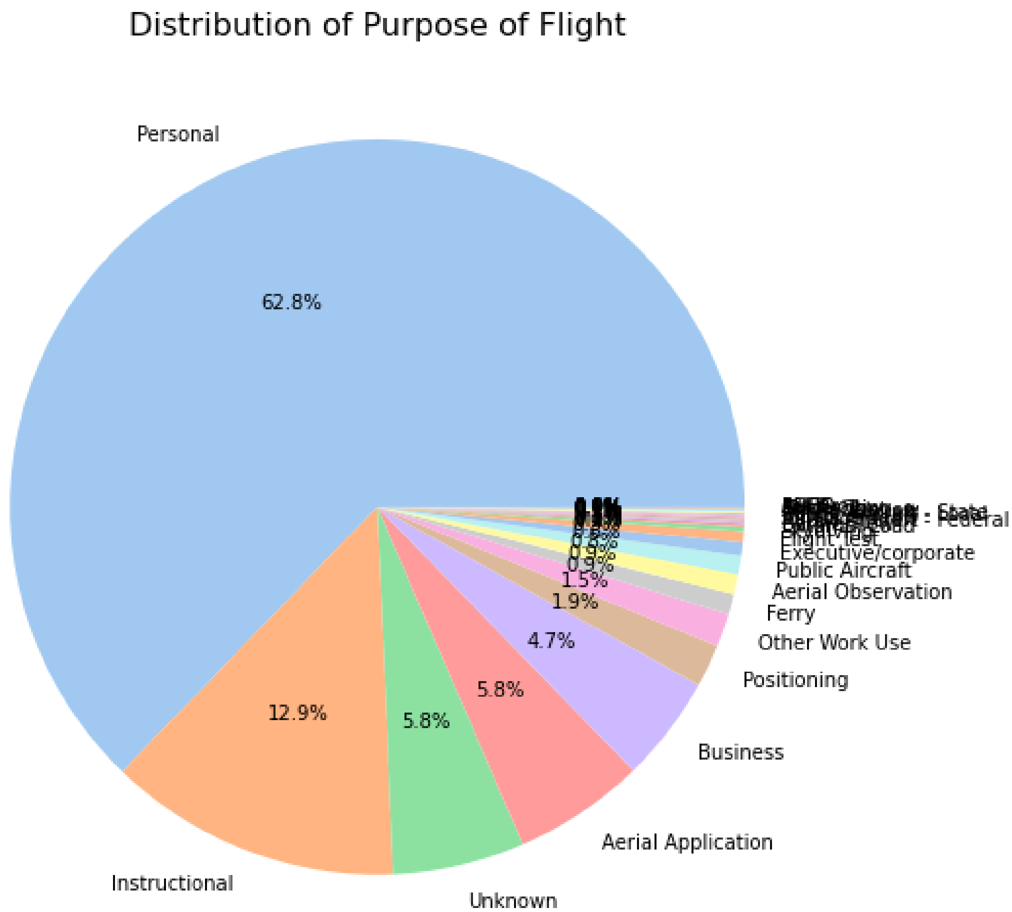


In [31]:

```
#Accidents By Flight Phase
# Count the number of accidents by flight phase
flight_phase_counts = df['Broad.phase.of.flight'].value_counts()

# Plot a bar chart of accidents by flight phase
plt.figure(figsize=(10, 6))
flight_phase_counts.plot(kind='bar', color='skyblue', edgecolor='black')
plt.title('Number of Accidents by Flight Phase', fontsize=16)
plt.xlabel('Flight Phase', fontsize=14)
plt.ylabel('Number of Accidents', fontsize=14)
plt.xticks(rotation=45, ha='right', fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



In [32]:
```
plt.figure(figsize=(8, 8))
df['Purpose.of.flight'].value_counts().plot.pie(
    autopct='%1.1f%%', colors=sns.color_palette('pastel')
)
plt.title('Distribution of Purpose of Flight', fontsize=16)
plt.ylabel('')
plt.tight_layout()
plt.show()
```



Conclusion The analysis of aviation event data reveals key trends that are essential for guiding the sales and marketing strategies of aircraft manufacturers. The following conclusions can be drawn:

1. Safety is a Top Priority: Manufactures should prioritize on aircraft with few accidents.
2. Technological Advancements Drive Sales: Modern aircraft equipped with advanced weather navigation and safety systems are better positioned in the market, especially under challenging conditions.
3. Customization for Purpose: Aircraft sales are heavily influenced by their purpose
4. Impact of Historical Trends: Certain manufacturers consistently exhibit strong performance and safety records, making them market leaders in specific segments.

Recommendations

1. Safety Enhancements:

Manufacturers should prioritize safety features for aircraft models associated with higher injury severity rates or damage. Enhanced weather navigation systems could reduce incidents under Instrument Meteorological Conditions (IMC), which are linked to a higher number of accidents. 2. Data-Driven Design:

Aircraft manufacturers should use accident data to refine designs, focusing on reducing damage during critical phases of flight (e.g., cruise and approach). Invest in technology that addresses recurring issues found in older models or designs with higher accident rates. 3. Targeted Marketing:

Focus on promoting models with excellent safety records for specific purposes of flight, such as commercial, recreational, or training purposes. Emphasize models with proven durability and resilience in diverse weather conditions. 4. Regulatory Collaboration:

Collaborate with aviation authorities to establish stricter safety regulations for amateur-built aircraft, as these may exhibit a higher accident rate. 5. Customer Education:

Provide comprehensive training and support for operators of high-performance models to ensure optimal handling and maintenance.