Final Project Submission Please fill out:

Student name: Martin Murimi Student pace: full time Scheduled project review date/time:
Instructor name: Antonny Muiko Blog post URL:

In [3]: ▶|
```python
# Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [4]: ▶|
```python
pwd
```

Out[4]: `'C:\\Users\\USER\\Documents\\Flatiron\\project phase1\\phase-1-project-1'`

In [5]: ▶|
```python
#Loading the Data
aviation_df=pd.read_csv('AviationData.csv',encoding='windows-1252',low_
aviation_df.head()
```

Out[5]:

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

5 rows × 31 columns

◀ | ▬▬▬▬▬▬▬ | | ▶

In [6]:   ▶| ```python
           # Analyzing the data set
           aviation_df.isnull().sum()
           ```

Out[6]:    ```
           Event.Id                    0
           Investigation.Type          0
           Accident.Number             0
           Event.Date                  0
           Location                   52
           Country                   226
           Latitude                54507
           Longitude               54516
           Airport.Code            38640
           Airport.Name            36099
           Injury.Severity          1000
           Aircraft.damage          3194
           Aircraft.Category       56602
           Registration.Number      1317
           Make                       63
           Model                      92
           Amateur.Built             102
           Number.of.Engines        6084
           Engine.Type              7077
           FAR.Description         56866
           Schedule                76307
           Purpose.of.flight        6192
           Air.carrier             72241
           Total.Fatal.Injuries    11401
           Total.Serious.Injuries  12510
           Total.Minor.Injuries    11933
           Total.Uninjured          5912
           Weather.Condition        4492
           Broad.phase.of.flight   27165
           Report.Status            6381
           Publication.Date        13771
           dtype: int64
           ```

In [7]:   ▶| ```python
           #Dropping columns that have more than 50% values missing
           aviation_df.drop(columns=['Latitude', 'Longitude', 'Airport.Name', 'Air
           ```

In [8]: ▶| 
```python
#Analyzing the cleaned data
aviation_df.isnull().sum()
```

Out[8]:
```
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
dtype: int64
```

In [9]: ▶|
```python
# Dropping duplicate values using our primary keys(event ID and Accider
aviation_df= aviation_df.drop_duplicates( subset = ['Event.Id', 'Accide
aviation_df.shape
```

Out[9]: (88863, 23)

In [10]: ▶|
```python
# Replacing missing data(numerical) with median
aviation_df['Total.Fatal.Injuries'].fillna(aviation_df['Total.Fatal.Inj
aviation_df['Total.Serious.Injuries'].fillna(aviation_df['Total.Serious
aviation_df['Total.Minor.Injuries'].fillna(aviation_df['Total.Minor.Inj
aviation_df['Total.Uninjured'].fillna(aviation_df['Total.Uninjured'].me
aviation_df['Number.of.Engines'].fillna(aviation_df['Number.of.Engines'
```

In [11]: ▶|
```python
# Replacing missing data(categorical) with mode
aviation_df['Make'].fillna(aviation_df['Make'].mode()[0], inplace=True)
aviation_df['Model'].fillna(aviation_df['Model'].mode()[0], inplace=Tru
aviation_df['Investigation.Type'].fillna(aviation_df['Investigation.Typ
aviation_df['Injury.Severity'].fillna(aviation_df['Injury.Severity'].mo
aviation_df['Aircraft.damage'].fillna(aviation_df['Aircraft.damage'].mo
aviation_df['Amateur.Built'].fillna(aviation_df['Amateur.Built'].mode()
aviation_df['Engine.Type'].fillna(aviation_df['Engine.Type'].mode()[0],
aviation_df['Purpose.of.flight'].fillna(aviation_df['Purpose.of.flight'
aviation_df['Weather.Condition'].fillna(aviation_df['Weather.Condition'
aviation_df['Broad.phase.of.flight'].fillna(aviation_df['Broad.phase.of
aviation_df['Report.Status'].fillna(aviation_df['Report.Status'].mode()
```

In [12]: ▶|
```python
#analyzed data after changes
aviation_df.isnull().sum()
```

Out[12]:
```
Event.Id                    0
Investigation.Type          0
Accident.Number             0
Event.Date                  0
Location                   52
Country                   226
Injury.Severity             0
Aircraft.damage             0
Registration.Number      1317
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        13761
dtype: int64
```

In [13]: ▶| ```
# Drop more unneccesary columns based on my data and business problem
aviation_df.drop(columns=['Publication.Date'])
```

Out[13]:

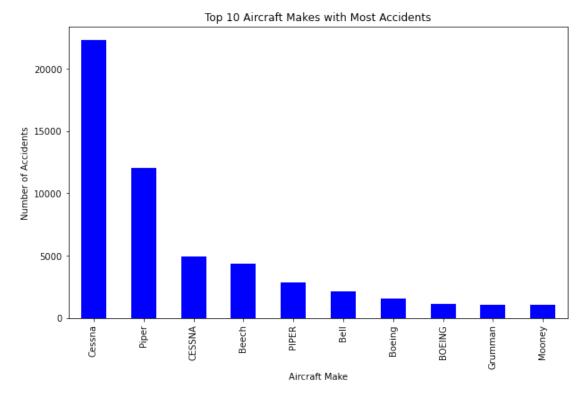|	Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Co
0	20001218X45444	Accident	SEA87LA080	1948-10-24	MOOSE CREEK, ID	U S
1	20001218X45447	Accident	LAX94LA336	1962-07-19	BRIDGEPORT, CA	U S
2	20061025X01555	Accident	NYC07LA005	1974-08-30	Saltville, VA	U S
3	20001218X45448	Accident	LAX96LA321	1977-06-19	EUREKA, CA	U S
4	20041105X01764	Accident	CHI79FA064	1979-08-02	Canton, OH	U S
...	...	...	...	...	...	
88858	20221227106491	Accident	ERA23LA093	2022-12-26	Annapolis, MD	U S
88859	20221227106494	Accident	ERA23LA095	2022-12-26	Hampton, NH	U S
88860	20221227106497	Accident	WPR23LA075	2022-12-26	Payson, AZ	U S
88861	20221227106498	Accident	WPR23LA076	2022-12-26	Morgan, UT	U S
88862	20221230106513	Accident	ERA23LA097	2022-12-29	Athens, GA	U S

88863 rows × 22 columns

In [14]: ▶| ```
aviation_df.shape
```

Out[14]:  (88863, 23)

In [15]:  ▶|  ```
aviation_df['Make'].value_counts()
```

Out[15]:
```
Cessna               22290
Piper                12029
CESSNA                4918
Beech                 4330
PIPER                 2840
                     ...
Harkness, Richard W.     1
GIBSON MARK              1
Maupin                   1
Fred Williams            1
Foster/hilsman           1
Name: Make, Length: 8236, dtype: int64
```

In [16]: ▶|
```python
# lets see which aircraft make has been in most accidents

#Grouping by 'Make' and counting accidents
accidents_by_make = aviation_df['Make'].value_counts()

#Selecting the top 10 makes with the most accidents
top_10_makes = accidents_by_make.head(10)

#Plotting a bar chart
plt.figure(figsize=(10, 6))
top_10_makes.plot(kind='bar', color='blue')
plt.title('Top 10 Aircraft Makes with Most Accidents')
plt.xlabel('Aircraft Make')
plt.ylabel('Number of Accidents')
plt.show()
```
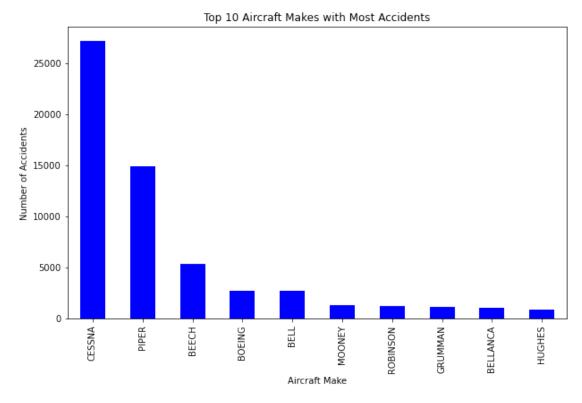
Top 10 Aircraft Makes with Most Accidents

In [17]: ▶|
```python
# some makes are duplicated, (Lower & Uppercase), we therefore combine
aviation_df['Make']= aviation_df['Make'].str.upper()
```

In [18]:
```python
accidents_by_make = aviation_df['Make'].value_counts()
top_10_makes = accidents_by_make.head(10)

plt.figure(figsize=(10, 6))
top_10_makes.plot(kind='bar', color='blue')
plt.title('Top 10 Aircraft Makes with Most Accidents')
plt.xlabel('Aircraft Make')
plt.ylabel('Number of Accidents')
plt.show()
```


Top 10 Aircraft Makes with Most Accidents

In [19]:
```python
# checking number of accident for CESSNA model
CESSNA_data = aviation_df[aviation_df['Make'] == 'CESSNA']

top_10_models = aviation_df['Model'].value_counts().head(10)

top_10_models.plot(kind='bar', color='blue')

plt.title('Top 10 Models for Cessna Aircraft in Accidents')
plt.xlabel('Model')
plt.ylabel('Number of Accidents')
plt.show()
```


In [20]:
```python
#the data is now ready for vizualization
aviation_df.to_csv("aviation_final.csv",index=False)
```

In [ ]:

In [ ]: