In [1]:

```
#modules to be used while manipulating the dataset
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [ ]:

```
#Load the AviationData.csv file using pandas
Aviation_data = pd.read_csv('AviationData.csv', encoding='cp1252')
```

In [ ]:

```
#An over view of the structure of the AviationData
Aviation_data.info(), Aviation_data.head(), Aviation_data.describe()
```

The company is interested in purcahsing and operating airplanes for commercial and private enterprises. We need to filter the Aircraft category to specifically Airplanes.

In [ ]:

```
#Check to see the different Aircrafts in the Aircraft Category column
Aviation_data['Aircraft.Category'].unique()
```

GOAL: To find the Lowest risk aircraft models that would minimize the potential for severe accidents, injuries and damage

To determine risk associated with Airplanes, columns such as Injury.Severity, Aircraft.Damage, Make, Model, Weather.Condition, and Broad.phase.of.flight, Total.Injuries are crucial for determining risks associated with aircraft.

In [5]:

```
#Filter the dataset to only include rows where Aircraft category is Airplane and Investigation Type is Accidents
Airplane_Data = Aviation_data[(Aviation_data['Aircraft.Category']=='Airplane') & (Aviation_data['Investigation.Ty
```

Airplane Data Cleanup

In [ ]:

```
#Missing values in injury columns with 0
Airplane_Data['Total.Fatal.Injuries'] = Airplane_Data['Total.Fatal.Injuries'].fillna(0)
Airplane_Data['Total.Serious.Injuries'] = Airplane_Data['Total.Serious.Injuries'].fillna(0)
Airplane_Data['Total.Minor.Injuries'] = Airplane_Data['Total.Minor.Injuries'].fillna(0)
Airplane_Data['Total.Uninjured'] = Airplane_Data['Total.Uninjured'].fillna(0)
```

In [ ]:

```
#Missing values in Aircraft.damage column with Unknown
Airplane_Data['Aircraft.damage'] = Airplane_Data['Aircraft.damage'].fillna('Unknown')
Airplane_Data['Purpose.of.flight'] = Airplane_Data['Purpose.of.flight'].fillna('Unknown')
Airplane_Data['Amateur.Built'] = Airplane_Data['Amateur.Built'].fillna('Unknown')

Airplane_Data['Weather.Condition'] = Airplane_Data['Weather.Condition'].str.upper()

Airplane_Data['Weather.Condition'] = Airplane_Data["Weather.Condition"].fillna('Uknown')
```

In [ ]:

```
#Standardize naming format of the Airplane Make column
Airplane_Data['Make'] = Airplane_Data['Make'].str.title()
Airplane_Data[['Make']]
```

EDA - Uninjured Analysis by Airplane Model and Make. (Make means Manufacturer and Model is the Airplane Model or version)

In [9]:

```
#Group the data by Aircraft "Model" and summarize total number of fatal, serious, minor injuries and Uninjured wh

Airplane_Damage_Summary = Airplane_Data.groupby('Model').agg(
    total_fatal_injuries=('Total.Fatal.Injuries', 'sum'),
    total_serious_injuries=('Total.Serious.Injuries', 'sum'),
    total_minor_injuries=('Total.Minor.Injuries', 'sum'),
    total_uninjured=('Total.Uninjured', 'sum'),
    total_events=('Model', 'count'),  # Total number of events per model
    aircraft_destroyed=('Aircraft.damage', lambda x: (x == 'Destroyed').sum())
).reset_index()
```

Compute the Rate of Injuries to assess risk for each Aircraft model. This will enable guide to understand the % of accidents that result in injuries or Uninjured

In [10]:

```
Airplane_Damage_Summary['Rate_fatal_injury'] = Airplane_Damage_Summary['total_fatal_injuries'] / Airplane_Damage_
Airplane_Damage_Summary['Rate_serious_injury'] = Airplane_Damage_Summary['total_serious_injuries'] / Airplane_Dam
Airplane_Damage_Summary['Rate_minor_injury'] = Airplane_Damage_Summary['total_minor_injuries'] / Airplane_Damage_
Airplane_Damage_Summary['Rate_Uninjured'] = Airplane_Damage_Summary['total_uninjured'] / Airplane_Damage_Summary[
```

In [11]:

```
#Save Airplane_Damage_Summary to excel for visualization in Tableau
Airplane_Damage_Summary.to_excel("Airplane_Injuries_Model.xlsx", index=False)
```

In [12]:

```
# Sort by the highest Rate Uninjury
```

In [12]:
```
# Sort by the highest Rate_Uninjured
Airplane_Damage_Summary_sorted = Airplane_Damage_Summary.sort_values(by='Rate_Uninjured', ascending=False)
```

In [ ]:
```
#Plot Models trends with Uninjured Passengers
top_10_models_Uninjured = Airplane_Damage_Summary_sorted.head(10)
sns.barplot(x='Rate_Uninjured', y='Model', data=top_10_models_Uninjured)
plt.title('Rate of Uninjured Passengers by Model')
plt.show()
```

Based on theh observations above, The MD 83 is the safest Airplane considering that it has the highest Uninjured rate. However, this view does not tell the complete story as it only appeared in 1 Event and 512 passengers were uninjured.

A better way would be to add a column called Decade and consider the number of uninjured passengers grouped by Airplane Make and Model. This way it becomes easier to tell the uninjured passengers over the decades for all accident events.

In [ ]:
```
#Creating a frequency sort of range for years in decades, a new column for the decade range, e.g "1974-1983"

Airplane_Data['Event.Date'] = pd.to_datetime(Airplane_Data['Event.Date'])


#Function to create the Decade range
def decade_range(year):
    lower = (year // 10) * 10 + 4  # The start of the range (e.g., 1974, 1984)
    upper = lower + 9  # The end of the range (e.g., 1983, 1993)
    return f"{lower}-{upper}"

#Creating new column named Decade Range
Airplane_Data['Decade Range'] = Airplane_Data['Event.Date'].dt.year.apply(decade_range)
```

In [ ]:
```
#Generate a bar graph to show the Top 10 Aircraft Models with Highest Uninjured and damage
Uninjured_by_model = Airplane_Data.groupby(['Model', 'Decade Range'])['Total.Uninjured'].sum().reset_index()


top_15_Uninjured_models = Uninjured_by_model.groupby('Model')['Total.Uninjured'].sum().nlargest(15).reset_index()

top_15_Uninjured_models_sorted = top_15_Uninjured_models.sort_values(by='Total.Uninjured', ascending=False)

top_15_Uninjured_data = Uninjured_by_model[Uninjured_by_model['Model'].isin(top_15_Uninjured_models_sorted['Model


plt.figure(figsize=(12, 6))
sns.barplot(x='Total.Uninjured', y='Model', hue='Decade Range', data=top_15_Uninjured_data, palette='Reds_d')
plt.title('Top 10 Aircraft Models with Highest Uninjured Over the Years')
plt.xlabel('Total Uninjured')
plt.ylabel('Aircraft Model')
plt.legend(title='Decade Range', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```

In [ ]:
```
#Determin the number of times each Airplane model appeared in a Airplane accident and the total of uninjured

Airplane_UninjuredSummary_Model = Airplane_Data.groupby('Model').agg(
    total_uninjured_model = ('Total.Uninjured', 'sum'),
    total_Events_model = ('Total.Uninjured', 'count')).reset_index()


Airplane_UninjuredSummary_Model_sorted = Airplane_UninjuredSummary_Model.sort_values(by='total_uninjured_model',
Airplane_UninjuredSummary_Model_sorted.head(15)
```

The above details shows the top 15 safest Airplane models. The Airplane 737 can be considered the safest based on the number of Uninjured passengers in all the Accident Events it was involved in.

In [ ]:
```
#Generate a bar graph to show the Top 10 Aircraft Makes with Highest Uninjured and damage

Uninjured_by_Make = Airplane_Data.groupby(['Make', 'Decade Range'])['Total.Uninjured'].sum().reset_index()


top_10_Uninjured_Make = Uninjured_by_Make.groupby('Make')['Total.Uninjured'].sum().nlargest(10).reset_index()


top_10_Uninjured_Make_data = Uninjured_by_Make[Uninjured_by_Make['Make'].isin(top_10_Uninjured_Make['Make'])]


plt.figure(figsize=(12, 6))
sns.barplot(x='Total.Uninjured', y='Make', hue='Decade Range', data=top_10_Uninjured_Make_data, palette='Blues_d'
plt.title('Top 10 Aircraft Models with Highest Uninjured Over the Years')
plt.xlabel('Total Uninjured')
plt.ylabel('Aircraft Make')
plt.legend(title='Decade Range', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```

Boeing is the manufacturer of of airplanes with that have the largest number of uninjured passengers.

In [ ]:

```
#Add a column called Event.Year in our dataframe
Airplane_Data['Event.Year'] = Airplane_Data['Event.Date'].dt.year

#Calculate Total Accidents, Fatal Accidents and Aircraft.damage = Destroyed
Accident_Rates_Year = Airplane_Data.groupby('Event.Year').agg(
    Total_Accidents = ('Accident.Number', 'count'),
    Fatal_Accidents = ('Total.Fatal.Injuries', lambda x: (x > 0).sum()),
    Airplane_Loss_Accidents = ('Total.Fatal.Injuries', lambda x: (x == 'Destroyed').sum())
).reset_index()

#Add the column for the rates
Accident_Rates_Year['Fatal_Accident_Rate'] = (Accident_Rates_Year['Fatal_Accidents']/Accident_Rates_Year['Total_A
Accident_Rates_Year['Airplane_Loss_Accidents_Rate'] = (Accident_Rates_Year['Airplane_Loss_Accidents']/Accident_Ra

Accident_Rates_Year
```

In [19]:

```
#Save the Rate of Injuries by Year into a CSV named Accident_Rates_Year for visualization in Tableau
Accident_Rates_Year.to_excel('Accident_Rates_Year.xlsx', index=False)
```

In [ ]:

```
Airplane_Data
```

In [21]:

```
#Clean Airplane_Clean_Data
Airplane_CleanData = Airplane_Data.drop(columns=['Location', 'Country','Latitude','Longitude','Airport.Code','Ai
                                        'FAR.Description','Schedule','Air.carrier','Report.Status','Publication.
```

In [22]:

```
Airplane_CleanData.to_excel("Aviation_CleanData.xlsx", index=False)
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js