Final Project Submission

Please fill out:

- Student name: Trevor Obonyo
- Student pace: full time
- Scheduled project review date/time: 30th June
- Instructor name: Faith Rotich
- Blog post URL:

In [1]:
```
# Your code here - remember to use markdown cells for comments as well!
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import plotly.express as px
```

In [2]:
```
aviation_data = pd.read_csv(r'D:\flatiron\dsc-phase-1-project-v3\data\Aviation_Data.
aviation_data.head()
```

Out[2]:

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

5 rows × 31 columns

In [3]:
```
relevant_columns = [
    'Event.Date',
    'Model',
    'Make',
    'Total.Fatal.Injuries',
    'Total.Serious.Injuries',
    'Aircraft.Category'
]
```

In [4]:
```
df = aviation_data[relevant_columns]
df
```

Out[4]:

	Event.Date	Model	Make	Total.Fatal.Injuries	Total.Serious.Injuries	Aircraft.Category
0	1948-10-24	108-3	Stinson	2.0	0.0	NaN
1	1962-07-19	PA24-180	Piper	4.0	0.0	NaN

	Event.Date	Model	Make	Total.Fatal.Injuries	Total.Serious.Injuries	Aircraft.Category
2	1974-08-30	172M	Cessna	3.0	NaN	NaN
3	1977-06-19	112	Rockwell	2.0	0.0	NaN
4	1979-08-02	501	Cessna	1.0	2.0	NaN
...	...	...	...	...	...	...
90343	2022-12-26	PA-28-151	PIPER	0.0	1.0	NaN
90344	2022-12-26	7ECA	BELLANCA	0.0	0.0	NaN
90345	2022-12-26	8GCBC	AMERICAN CHAMPION AIRCRAFT	0.0	0.0	Airplane
90346	2022-12-26	210N	CESSNA	0.0	0.0	NaN
90347	2022-12-29	PA-24-260	PIPER	0.0	1.0	NaN

90348 rows × 6 columns

Handle missing data

Drop rows where 'Aircraft.Model' or 'Fatal.Injuries' are missing, as they are critical for risk assessment

In [5]:
```
df = df.dropna(subset=['Model', 'Total.Fatal.Injuries'])
df
```

Out[5]:

	Event.Date	Model	Make	Total.Fatal.Injuries	Total.Serious.Injuries	Aircraft.Category
0	1948-10-24	108-3	Stinson	2.0	0.0	NaN
1	1962-07-19	PA24-180	Piper	4.0	0.0	NaN
2	1974-08-30	172M	Cessna	3.0	NaN	NaN
3	1977-06-19	112	Rockwell	2.0	0.0	NaN
4	1979-08-02	501	Cessna	1.0	2.0	NaN
...	...	...	...	...	...	...
90343	2022-12-26	PA-28-151	PIPER	0.0	1.0	NaN
90344	2022-12-26	7ECA	BELLANCA	0.0	0.0	NaN

	Event.Date	Model	Make	Total.Fatal.Injuries	Total.Serious.Injuries	Aircraft.Category
90345	2022-12-26	8GCBC	AMERICAN CHAMPION AIRCRAFT	0.0	0.0	Airplane
90346	2022-12-26	210N	CESSNA	0.0	0.0	NaN
90347	2022-12-29	PA-24-260	PIPER	0.0	1.0	NaN

77408 rows × 6 columns

Standardize aircraft make and model names

Convert to uppercase and remove whitespace to ensure consistency

In [6]:
```
#df.loc['Make'] = df['Make'].str.upper().str.strip()
#df.loc['Model'] = df['Model'].str.upper().str.strip()
df.loc[:, ['Make', 'Model']] = df.loc[:, ['Make', 'Model']].astype(str).apply(lambda
df
```

Out[6]:

	Event.Date	Model	Make	Total.Fatal.Injuries	Total.Serious.Injuries	Aircraft.Category
0	1948-10-24	108-3	STINSON	2.0	0.0	NaN
1	1962-07-19	PA24-180	PIPER	4.0	0.0	NaN
2	1974-08-30	172M	CESSNA	3.0	NaN	NaN
3	1977-06-19	112	ROCKWELL	2.0	0.0	NaN
4	1979-08-02	501	CESSNA	1.0	2.0	NaN
...	...	...	...	...	...	...
90343	2022-12-26	PA-28-151	PIPER	0.0	1.0	NaN
90344	2022-12-26	7ECA	BELLANCA	0.0	0.0	NaN
90345	2022-12-26	8GCBC	AMERICAN CHAMPION AIRCRAFT	0.0	0.0	Airplane
90346	2022-12-26	210N	CESSNA	0.0	0.0	NaN
90347	2022-12-29	PA-24-260	PIPER	0.0	1.0	NaN

77408 rows × 6 columns

Create a combined 'Aircraft' column for easier grouping

In [7]:
```
df.loc[:, 'Aircraft'] = df['Make'] + ' ' + df['Model']
#df['Aircraft'] = df[['Make', 'Model']].apply(lambda x: ''.join(x), axis = 1)
df
```

```
C:\Users\trevor\AppData\Local\Temp\ipykernel_15992\4175533184.py:1: SettingWithCopyWa
rning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  df.loc[:, 'Aircraft'] = df['Make'] + ' ' + df['Model']
```

Out[7]:

	Event.Date	Model	Make	Total.Fatal.Injuries	Total.Serious.Injuries	Aircraft.Category	
0	1948-10-24	108-3	STINSON	2.0	0.0	NaN	
1	1962-07-19	PA24-180	PIPER	4.0	0.0	NaN	
2	1974-08-30	172M	CESSNA	3.0	NaN	NaN	
3	1977-06-19	112	ROCKWELL	2.0	0.0	NaN	R
4	1979-08-02	501	CESSNA	1.0	2.0	NaN	
...	...	...	...	...	...	...	
90343	2022-12-26	PA-28-151	PIPER	0.0	1.0	NaN	
90344	2022-12-26	7ECA	BELLANCA	0.0	0.0	NaN	B
90345	2022-12-26	8GCBC	AMERICAN CHAMPION AIRCRAFT	0.0	0.0	Airplane	A CH
90346	2022-12-26	210N	CESSNA	0.0	0.0	NaN	
90347	2022-12-29	PA-24-260	PIPER	0.0	1.0	NaN	

77408 rows × 7 columns

Convert 'Event.Date' to datetime format

In [8]:
```
df['Event.Date'] = pd.to_datetime(df['Event.Date'])
df
```

```
C:\Users\trevor\AppData\Local\Temp\ipykernel_15992\4196615550.py:1: SettingWithCopyWa
rning:
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Event.Date'] = pd.to_datetime(df['Event.Date'])
```

Out[8]:

	Event.Date	Model	Make	Total.Fatal.Injuries	Total.Serious.Injuries	Aircraft.Category	
0	1948-10-24	108-3	STINSON	2.0	0.0	NaN	
1	1962-07-19	PA24-180	PIPER	4.0	0.0	NaN	
2	1974-08-30	172M	CESSNA	3.0	NaN	NaN	
3	1977-06-19	112	ROCKWELL	2.0	0.0	NaN	R
4	1979-08-02	501	CESSNA	1.0	2.0	NaN	
...	...	...	...	...	...	...	
90343	2022-12-26	PA-28-151	PIPER	0.0	1.0	NaN	
90344	2022-12-26	7ECA	BELLANCA	0.0	0.0	NaN	B
90345	2022-12-26	8GCBC	AMERICAN CHAMPION AIRCRAFT	0.0	0.0	Airplane	A CH
90346	2022-12-26	210N	CESSNA	0.0	0.0	NaN	
90347	2022-12-29	PA-24-260	PIPER	0.0	1.0	NaN	

77408 rows × 7 columns

Filter data for modern relevance and specific category

Include only accidents from 2000 onwards and for the 'Airplane' category

This focuses on recent data and fixed-wing aircraft suitable for commercial and private enterprises

In [9]:
```
df = df[(df['Event.Date'].dt.year >= 2000) & (df['Aircraft.Category'] == 'Airplane')]
df
```

Out[9]:

	Event.Date	Model	Make	Total.Fatal.Injuries	Total.Serious.Injuries	Aircraft.Category
47779	2000-01-30	A 310	AIRBUS INDUSTRIE	169.0	NaN	Airplane

	Event.Date	Model	Make	Total.Fatal.Injuries	Total.Serious.Injuries	Aircraft.Category
47864	2000-02-16	208B	CESSNA	1.0	NaN	Airplane
47869	2000-02-16	182M	CESSNA	1.0	NaN	Airplane
47870	2000-02-16	DC-8-71F	DOUGLAS	3.0	NaN	Airplane
48128	2000-04-05	35A	LEARJET	3.0	NaN	Airplane
...	...	...	...	...	...	...
90328	2022-12-13	PA42	PIPER	0.0	0.0	Airplane
90332	2022-12-14	SR22	CIRRUS DESIGN CORP	0.0	0.0	Airplane
90335	2022-12-15	SA226TC	SWEARINGEN	0.0	0.0	Airplane
90336	2022-12-16	R172K	CESSNA	0.0	1.0	Airplane
90345	2022-12-26	8GCBC	AMERICAN CHAMPION AIRCRAFT	0.0	0.0	Airplane

21171 rows × 7 columns

Calculate safety metrics by aircraft model

Group by 'Aircraft' and compute:

- Total_Accidents: Number of accidents
- Fatal_Accidents: Number of accidents with at least one fatality
- Total_Fatalities: Sum of fatal injuries
- Avg_Fatalities_Per_Accident: Average fatalities per accident

In [10]:
```
metrics = df.groupby('Aircraft').agg(
    Total_Accidents=('Aircraft', 'count'),
    Fatal_Accidents=('Total.Fatal.Injuries', lambda x: (x > 0).sum()),
    Total_Fatalities=('Total.Fatal.Injuries', 'sum'),
    Avg_Fatalities_Per_Accident=('Total.Fatal.Injuries', 'mean')
).reset_index()
metrics
```

Out[10]:

	Aircraft	Total_Accidents	Fatal_Accidents	Total_Fatalities	Avg_Fatalities_Per_Accident
0	177MF LLC PITTS MODEL 12	1	0	0.0	0.0
1	2007 SAVAGE AIR LLC EPIC LT	1	0	0.0	0.0

	Aircraft	Total_Accidents	Fatal_Accidents	Total_Fatalities	Avg_Fatalities_Per_Accident
2	2021FX3 LLC CCX-2000	2	0	0.0	0.0
3	3XTRIM 450 ULTRA	1	1	1.0	1.0
4	5 RIVERS LLC SQ-2	1	0	0.0	0.0
...	...	...	...	...	...
6224	ZLIN Z50	1	1	1.0	1.0
6225	ZODIAC 601XL	1	1	1.0	1.0
6226	ZUBAIR S KHAN RAVEN	1	1	1.0	1.0
6227	ZUBER THOMAS P ZUBER SUPER DRIFTER	1	0	0.0	0.0
6228	ZWICKER MURRAY R GLASTAR	1	0	0.0	0.0

6229 rows × 5 columns

Add mock data for units produced to normalize accident rates

In a real scenario, this data would be sourced from aviation databases or manufacturers

Here, we use the top 5 aircraft by accident count and assign hypothetical production numbers

In [11]:
```
top_aircraft = metrics.nlargest(5, 'Total_Accidents')['Aircraft']
units_produced = pd.DataFrame({
    'Aircraft': top_aircraft,
    'Units_Produced': [6000, 6000, 6000, 6000, 6000]  #examples
})
top_aircraft
```

Out[11]:
```
1461    CESSNA 172
950     BOEING 737
1515    CESSNA 182
4745    PIPER PA28
1452    CESSNA 152
Name: Aircraft, dtype: object
```

Merge the units produced data with the metrics

In [12]:
```
metrics = metrics.merge(units_produced, on='Aircraft', how='left')
metrics
```

Out[12]:

	Aircraft	Total_Accidents	Fatal_Accidents	Total_Fatalities	Avg_Fatalities_Per_Accident	Units_Pr
0	177MF LLC PITTS MODEL 12	1	0	0.0	0.0	
1	2007 SAVAGE AIR LLC EPIC LT	1	0	0.0	0.0	
2	2021FX3 LLC CCX-2000	2	0	0.0	0.0	
3	3XTRIM 450 ULTRA	1	1	1.0	1.0	
4	5 RIVERS LLC SQ-2	1	0	0.0	0.0	
...	...	...	...	...	...	
6224	ZLIN Z50	1	1	1.0	1.0	
6225	ZODIAC 601XL	1	1	1.0	1.0	
6226	ZUBAIR S KHAN RAVEN	1	1	1.0	1.0	
6227	ZUBER THOMAS P ZUBER SUPER DRIFTER	1	0	0.0	0.0	
6228	ZWICKER MURRAY R GLASTAR	1	0	0.0	0.0	

6229 rows × 6 columns

Calculate normalized metrics

Accidents_Per_Unit and Fatal_Accidents_Per_Unit normalize accident counts by units produced

In [13]:
```
metrics['Accidents_Per_Unit'] = metrics['Total_Accidents'] / metrics['Units_Produced
metrics['Fatal_Accidents_Per_Unit'] = metrics['Fatal_Accidents'] / metrics['Units_Pr
metrics
```

Out[13]:

	Aircraft	Total_Accidents	Fatal_Accidents	Total_Fatalities	Avg_Fatalities_Per_Accident	Units_Pr
0	177MF LLC PITTS MODEL 12	1	0	0.0	0.0	
1	2007 SAVAGE AIR LLC EPIC LT	1	0	0.0	0.0	
2	2021FX3 LLC CCX-2000	2	0	0.0	0.0	
3	3XTRIM 450 ULTRA	1	1	1.0	1.0	
4	5 RIVERS LLC SQ-2	1	0	0.0	0.0	
...	...	...	...	...	...	
6224	ZLIN Z50	1	1	1.0	1.0	
6225	ZODIAC 601XL	1	1	1.0	1.0	
6226	ZUBAIR S KHAN RAVEN	1	1	1.0	1.0	
6227	ZUBER THOMAS P ZUBER SUPER DRIFTER	1	0	0.0	0.0	
6228	ZWICKER MURRAY R GLASTAR	1	0	0.0	0.0	

6229 rows × 8 columns

Identify lowest-risk aircraft

Filter for aircraft with known units produced and sort by Accidents_Per_Unit

Select the top 10 safest aircraft based on this metric

In [14]:
```
normalized_metrics = metrics.dropna(subset=['Units_Produced'])
lowest_risk = metrics.sort_values(by='Accidents_Per_Unit').head(10)
lowest_risk
```

Out[14]:

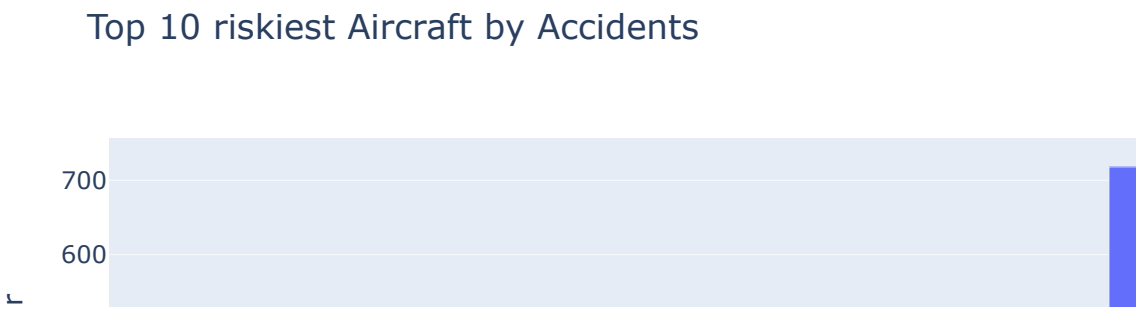
	Aircraft	Total_Accidents	Fatal_Accidents	Total_Fatalities	Avg_Fatalities_Per_Accident	Units_Prc
1452	CESSNA 152	238	31	45.0	0.189076	

	Aircraft	Total_Accidents	Fatal_Accidents	Total_Fatalities	Avg_Fatalities_Per_Accident	Units_Prc
4745	PIPER PA28	273	63	116.0	0.424908	
1515	CESSNA 182	278	73	144.0	0.517986	
950	BOEING 737	402	15	1341.0	3.335821	
1461	CESSNA 172	718	122	213.0	0.296657	
0	177MF LLC PITTS MODEL 12	1	0	0.0	0.000000	
1	2007 SAVAGE AIR LLC EPIC LT	1	0	0.0	0.000000	
2	2021FX3 LLC CCX-2000	2	0	0.0	0.000000	
3	3XTRIM 450 ULTRA	1	1	1.0	1.000000	
4	5 RIVERS LLC SQ-2	1	0	0.0	0.000000	

Visualize the top 10 safest aircraft

Bar chart showing Accidents_Per_Unit for the lowest-risk aircraft

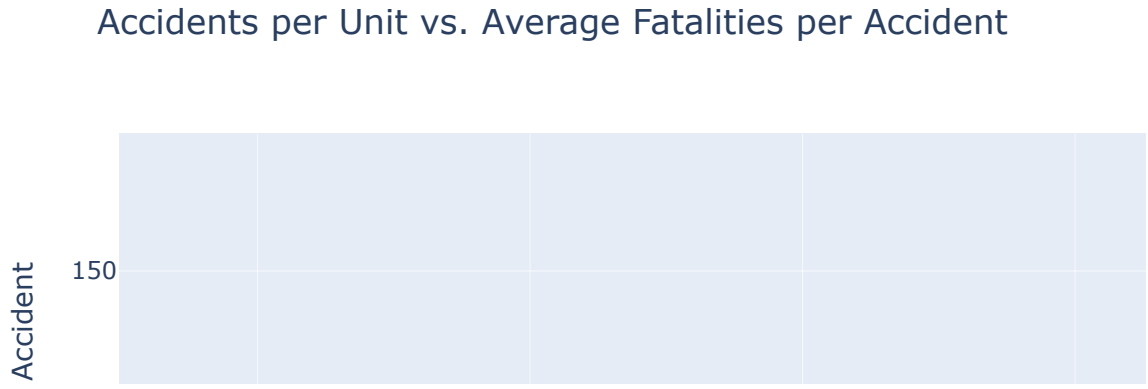
In [15]:
```
fig1 = px.bar(lowest_risk,
              x='Aircraft',
              y='Total_Accidents',
              title='Top 10 riskiest Aircraft by Accidents',
              labels={'Accidents_Per_Unit': 'Accidents per Unit'})
fig1.update_layout(xaxis_title="Aircraft Model", yaxis_title="Accidents number")
fig1.show()
```



Visualize the trade-off between accident frequency and severity

Scatter plot of Accidents_Per_Unit vs. Avg_Fatalities_Per_Accident

In [16]:
```
fig2 = px.scatter(metrics,
                  x='Accidents_Per_Unit',
                  y='Avg_Fatalities_Per_Accident',
                  text='Aircraft',
                  title='Accidents per Unit vs. Average Fatalities per Accident',
                  labels={'Accidents_Per_Unit': 'Accidents per Unit',
                          'Avg_Fatalities_Per_Accident': 'Avg Fatalities per Acciden
fig2.update_traces(textposition='top center')
fig2.update_layout(xaxis_title="Accidents per Unit", yaxis_title="Average Fatalities
fig2.show()
```



Saving the file

saving the metriccs to a csv file for further use

In [17]:
```
metrics.to_csv('aircraft_risk_metrics.csv', index = False)
```