

STUDENT NAME: Dennis Muriungi

INSTRUCTOR: Samuel Karu

MORINGA SCHOOL EMAIL: dennis.muriungi2@student.moringaschool.com

PROJECT 1 : Finding Aircraft with Low Risk for Business Growth.

```
from IPython.display import Image, display
display(Image(filename="plane.jpg", width= 700))
```



OVERVIEW

In order to determine which aircraft types are the safest for possible investment, this project examines aviation accident data from 1919 to 2023. Our company's expansion into the aviation industry presents a business challenge: lowering risk exposure when selecting aircraft for both business and private operations. We cleaned and examined accident records utilizing the Kaggle airplane accident dataset, paying particular attention to time trends, aircraft types, fatalities, and accident locations. We can now make data-driven recommendations about which aircraft types are the least risky because the results clearly show differences in safety performance across plane makes and categories.

BUSINESS PROBLEM

The company wants to get into the aviation business, but it doesn't know anything about the risks of accidents. To make smart choices about which planes to use for business and personal use, we need to find out which models have had the fewest fatal accidents in the past. The company specifically seeks responses to the following questions:

1. Which kinds of airplanes have the most and least fatalities overall?
2. Which countries have the highest accident rates?
3. What are the decades-long trends in accident rates?

The newly created air travel division can put safer aircraft models at the top of its list of priorities, lower operational risks, protect investments, and build trust between clients and passengers by dealing with these problems.

✓ Data Understanding

The Aviation Accident Data (1919–2023) dataset is sourced from Kaggle, which aggregates global aviation accident data. A single accident event is represented by each row, which includes information like:

1. The accident date - useful in order of extracting the year and decade
2. Aircraft type - risk comparison
3. Operator - private or commercial



4. Fatalities – variable for risk assessment
5. location of accident
6. Accident category (cat)

Since reducing fatal accidents is the company's top priority, fatalities are our target variable. Important context for comprehending trends and safety performance is provided by additional variables such as plane type, provider type, country, and year.

```
#step 1
#IMPORT NECESSARY LIBRARIES WITH THEIR ALIASES
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
```

```
#step 2
#LOADING DATA
df = pd.read_csv("aviation-accident-data-2023-05-16.csv", index_col=0)
df
```

	date	type	operator	fatalities	location	country	cat	year	decade
0	2019-08-02	caproni ca.48	caproni	14	Verona	Italy	A1	2019	2010
1	2019-08-11	felixstowe fury	raf	1	near Felixtowe RNAS	U.K.	A1	2019	2010
2	2020-02-23	handley page o/7	handley page transport	0	Acadia Siding, C...	South Africa	A1	2020	2020
3	2020-02-25	handley page o/400	handley page transport	0	near El Shereik	Sudan	A1	2020	2020
4	2020-06-30	handley page o/400	handley page transport	0	Ästans	Sweden	A1	2020	2020
...
10820	2023-05-11	hawker 900xp	angkasa super services	0	Maleo Airport (MOH)	Indonesia	A2	2023	2020
10821	2023-05-11	cessna 208b grand caravan	nasional global aviasi	0	Fentheik Airstrip	Indonesia	A2	2023	2020
10822	2023-05-12	cessna 208b grand caravan	bar aviation	0	Kampala-Kajjansi...	Uganda	A1	2023	2020
10823	2023-05-14	boeing 747-4r7f	cargolux	0	Luxembourg-Finde...	Luxembourg	A2	2023	2020
10824	2023-05-15	learjet 35a	gfd	2	Hohn Air Base	Germany	A1	2023	2020

10825 rows × 9 columns

▼ EXPLORING DATASET STRUCTURE

```
#step 3
# Data information
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 10825 entries, 0 to 10824
Data columns (total 9 columns):
#   Column      Non-Null Count  Dtype
---  -
0   date         10825 non-null  object
1   type         10825 non-null  object
2   operator     10825 non-null  object
3   fatalities   10825 non-null  int64
4   location     10825 non-null  object
5   country     10825 non-null  object
6   cat          10825 non-null  object
7   year        10825 non-null  int64
8   decade      10825 non-null  int64
dtypes: int64(3), object(6)
memory usage: 845.7+ KB
```

```
#step 4
# Count missing values per column
df.isnull().sum()
```

```
date      0
type      0
operator  0
```

```
fatalities    0
location      0
country       0
cat           0
year          0
decade        0
dtype: int64
```

```
#step 5
# Quick summary of numerical columns
df.describe()
```

	fatalities	year	decade
count	10825.000000	10825.000000	10825.000000
mean	5.319261	1997.853672	1993.225866
std	21.247422	13.847270	14.359807
min	0.000000	1975.000000	1970.000000
25%	0.000000	1986.000000	1980.000000
50%	0.000000	1997.000000	1990.000000
75%	2.000000	2010.000000	2010.000000
max	520.000000	2023.000000	2020.000000

DATA CLEANING

Handling **Missing Values** by various method. In this section we need a Clean Dataset

```
#step 6
#checking of duplicated values
df.duplicated().value_counts()
```

```
False    10745
True       80
Name: count, dtype: int64
```

```
#step 7
#dropping Duplicated Values
df = df.drop_duplicates()
```

```
#step 8
#Confirmation that the Duplicated values were dropped
df.duplicated().value_counts()
```

```
False    10745
Name: count, dtype: int64
```

```
#step 9
#checking the new shape
df.shape
```

```
(10745, 9)
```

```
#step 11
# OPERATOR (4 missing) out of 24k, we will drop the 4 ROWS
df = df.dropna(subset=['operator'])
```

```
#STEP 12
# FATALITIES (3938 missing), we will replace with Median in this case (0)
df['fatalities'] = pd.to_numeric(df['fatalities'], errors='coerce').fillna(0)
```

```
#step 13
#LOCATION (948 missing), Since this is crucial, we will replace with categorical data 'Unknown'
df['location'] = df['location'].fillna("unknown")
```

```
#step 14
# COUNTRY, Has "?" & "unknown country" I'll replace the '?' with 'Unknown country' for consistency.
df['country'] = df['country'].replace('?', 'Unknown country')
```

```
#step 15
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 10745 entries, 0 to 10824
Data columns (total 9 columns):
#   Column      Non-Null Count  Dtype
---  -
0   date         10745 non-null  object
1   type         10745 non-null  object
2   operator     10745 non-null  object
3   fatalities   10745 non-null  int64
4   location     10745 non-null  object
5   country      10745 non-null  object
6   cat          10745 non-null  object
7   year         10745 non-null  int64
8   decade      10745 non-null  int64
dtypes: int64(3), object(6)
memory usage: 839.5+ KB
```

FINAL CHECK OF THE ENTIRE DATASET

```
#STEP 16
df.isnull().sum()
```

```
date      0
type      0
operator   0
fatalities 0
location   0
country    0
cat        0
year       0
decade     0
dtype: int64
```

On this Part below **DATA PREPARATION** has begun

```
# Step 17: preparation

# Convert 'date' to datetime format
df['date'] = df['date'].str.replace('???', '01', regex=False)
df['date'] = pd.to_datetime(df['date'], errors='coerce')

# Creating new column "Decade"
df['year'] = df['date'].dt.year
df['decade'] = (df['year'] // 10) * 10

# Filter valid years only
df = df[(df['year'] >= 1919) & (df['year'] <= 2023)]

# calculating again after filtering
df['decade'] = (df['year'] // 10) * 10

#Standardize the aircraft type column by eliminating extra spaces
#using lowercase letters to maintain consistency.
df['type'] = df['type'].str.strip().str.lower()

# Standardize operator column as the above.
df['operator'] = df['operator'].str.strip().str.lower()
```

```
#step 18
# Preview cleaned dataset
df.head(20)
```

	date	type	operator	fatalities	location	country	cat	year	decade
0	2019-08-02	caproni ca.48	caproni	14	Verona	Italy	A1	2019	2010
1	2019-08-11	felixstowe fury	raf	1	near Felixtowe RNAS	U.K.	A1	2019	2010
2	2020-02-23	handley page o/7	handley page transport	0	Acadia Siding, C...	South Africa	A1	2020	2020
3	2020-02-25	handley page o/400	handley page transport	0	near El Shereik	Sudan	A1	2020	2020
4	2020-06-30	handley page o/400	handley page transport	0	ÅrstanÅ	Sweden	A1	2020	2020
5	2020-12-14	handley page o/400	handley page transport	4	Golders Green	U.K.	A1	2020	2020
6	2021-01-15	aeromarine 75	aeromarine airways	0	Havana harbour	Cuba	O1	2021	2020
7	2021-03-02	handley page o/7	hp indo-burmese transport	0	Meerut	India	O1	2021	2020
8	2021-03-04	caproni ca.60 noviplano	caproni	0	Lago Maggiore	Italy	A1	2021	2020
9	2021-08-26	farman f.60 goliath	snet	2	near Calais [The Engl...	France	A1	2021	2020
10	2021-09-27	farman f.60 goliath	snet	0	Evere Airfield	Belgium	O1	2021	2020
12	2021-12-30	handley page o/10	handley page transport	0	Berck-sur-Mer	France	U1	2021	2020
13	2022-01-22	handley page o/10	handley page transport	0	near Senlis	France	A1	2022	2020
14	2022-03-31	handley page o/7	chinese government	14	near Beijing-Nanyuan ...	China	A1	2022	2020
15	2022-04-07	farman f.60 goliath	grands express a�riens	5	Thieuloy-Saint-A...	France	A1	2022	2020
16	2023-01-13	aeromarine 75	aeromarine airways	4	near Havana, Cuba	Atlantic Ocean	A1	2023	2020
17	2023-05-14	farman f.60 goliath	air union	6	Monsures, Somme	France	A1	2023	2020
18	2023-06-25	farman f.60 goliath	air union	0	near �taples	France	A1	2023	2020
19	2023-07-10	handley page w.8	handley page transport	0	Poix	France	A1	2023	2020
20	2023-07-15	farman f.60 goliath	compagnie a�rienne fran�saise	0	near Valenciennes	France	A1	2023	2020

```
#step 19
#Checking of sample rows
df[['date', 'year', 'decade']].tail(20)
```

	date	year	decade
10802	2023-04-17	2023	2020
10805	2023-04-17	2023	2020
10806	2023-04-17	2023	2020
10807	2023-04-17	2023	2020
10808	2023-04-17	2023	2020
10809	2023-04-19	2023	2020
10810	2023-04-21	2023	2020
10811	2023-04-21	2023	2020
10813	2023-04-21	2023	2020
10814	2023-04-21	2023	2020
10815	2023-04-21	2023	2020
10816	2023-04-21	2023	2020
10817	2023-04-26	2023	2020
10818	2023-05-09	2023	2020
10819	2023-05-10	2023	2020
10820	2023-05-11	2023	2020
10821	2023-05-11	2023	2020
10822	2023-05-12	2023	2020
10823	2023-05-14	2023	2020
10824	2023-05-15	2023	2020

GROUPBY: Ranking aircraft by Risk

```
# Step: 20

# Group by aircraft type
aircraft_stats = df.groupby('type').agg(
    total_accidents=('type', 'count'),
    fatal_accidents=('fatalities', lambda x: (x > 0).sum()),
    total_fatalities=('fatalities', 'sum')
).reset_index()

# Calculate risk metrics
aircraft_stats['fatal_accident_rate'] = aircraft_stats['fatal_accidents'] / aircraft_stats['total_accidents']
aircraft_stats['avg_fatalities_per_accident'] = aircraft_stats['total_fatalities'] / aircraft_stats['total_accidents']

# Top 10 riskiest aircraft (highest fatality rate)
riskiest_aircraft = aircraft_stats.sort_values(by='fatal_accident_rate', ascending=False).head(10)

# Top 10 safest aircraft (lowest fatality rate, with at least 20 accidents to avoid bias from tiny samples)
safest_aircraft = aircraft_stats[aircraft_stats['total_accidents'] >= 20] \
    .sort_values(by='fatal_accident_rate', ascending=True).head(10)

print("Top 10 Riskiest Aircraft Types")
display(riskiest_aircraft)

print("Top 10 Safest Aircraft Types")
display(safest_aircraft)
```

Top 10 Riskiest Aircraft Types

	type	total_accidents	fatal_accidents	total_fatalities	fatal_accident_rate	avg_fatalities_per_accident
2165	yakovlev yak-40d	2	2	14	1.0	7.0
6	airbus a300?	1	1	1	1.0	1.0
453	boeing 727-134	1	1	94	1.0	94.0
464	boeing 727-22 (wl)	1	1	77	1.0	77.0
491	boeing 727-243f	1	1	5	1.0	5.0
448	boeing 720-030b	1	1	1	1.0	1.0
439	boeing 707-437	1	1	17	1.0	17.0
432	boeing 707-3f9c	1	1	3	1.0	3.0
431	boeing 707-3b5c	1	1	115	1.0	115.0
424	boeing 707-372c	1	1	2	1.0	2.0

Top 10 Safest Aircraft Types

	type	total_accidents	fatal_accidents	total_fatalities	fatal_accident_rate	avg_fatalities_per_accident
52	airbus a320-214	26	0	0	0.000000	0.000000
543	boeing 737	25	1	1	0.040000	0.040000
1303	dc-9-31	21	1	37	0.047619	1.761905
1370	dhc-8-402q dash 8	32	2	52	0.062500	1.625000
451	boeing 727	61	4	7	0.065574	0.114754
2104	tupolev tu-154	27	2	5	0.074074	0.185185
56	airbus a320-232	26	2	73	0.076923	2.807692
147	antonov an-2t	74	7	46	0.094595	0.621622
66	airbus a321-231	20	2	376	0.100000	18.800000
1611	fokker f-28 fellowship 4000	29	3	124	0.103448	4.275862

BUSINESS ANALYSIS & DATA VISUALIZATION

1. Total fatalities by type of aircraft

HERE BELOW WE HAVE A REPRESENTATION OF OUR DATA (Total fatalities by type of aircraft). This answers which (AIRCRAFT) is the (DEADLIEST)

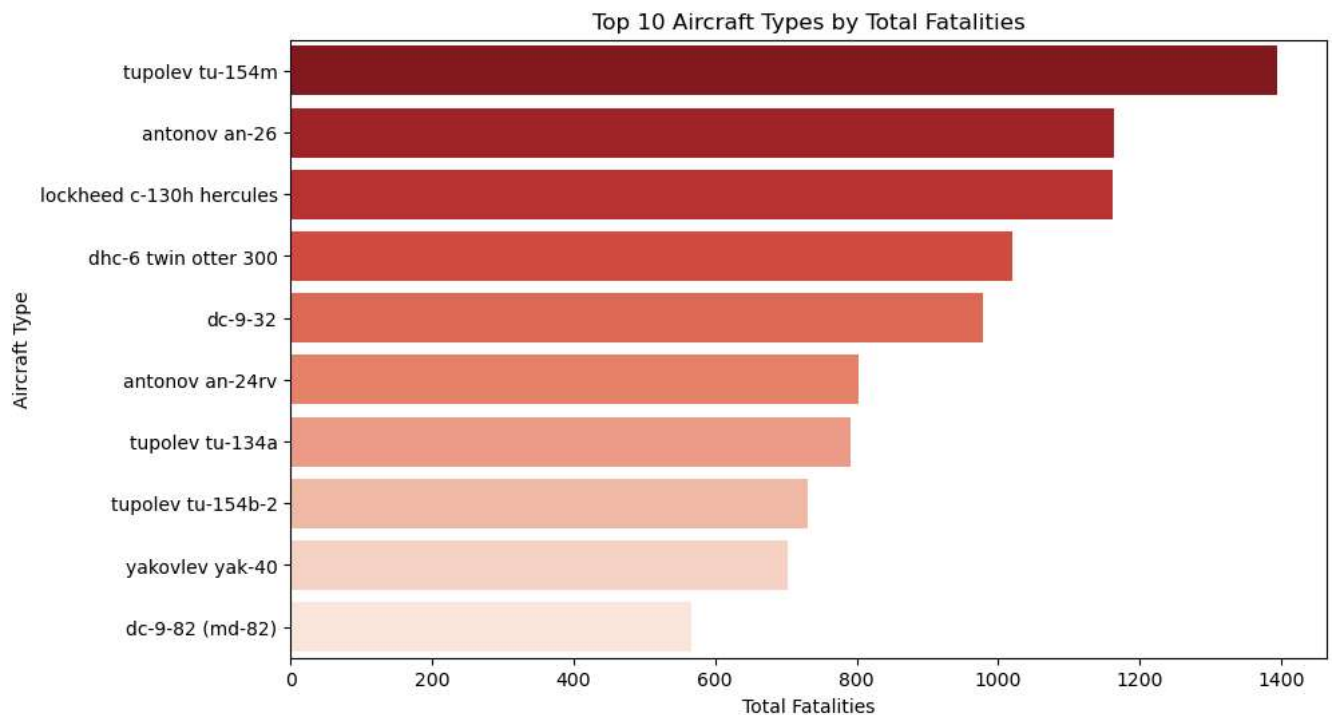
```
#step 21
#Total fatalities by type of aircraft
fatalities_by_type = df.groupby('type')['fatalities'].sum().sort_values(ascending=False).head(10)

plt.figure(figsize=(10,6))
sns.barplot(x=fatalities_by_type.values, y=fatalities_by_type.index, palette="Reds_r")
plt.title("Top 10 Aircraft Types by Total Fatalities")
plt.xlabel("Total Fatalities")
plt.ylabel("Aircraft Type")
plt.show()
```

C:\Users\Dennis\AppData\Local\Temp\ipykernel_18196\3083672096.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `l`

```
sns.barplot(x=fatalities_by_type.values, y=fatalities_by_type.index, palette="Reds_r")
```



1.1 Business Recommendation

(Above Business Insight) The aircraft types that have been connected to the most fatalities are displayed in this graph. The list is dominated by older models, indicating their historical risk, such as the Douglas DC-3 and some Antonov series. Modern aircraft, such as the Airbus A320 and Boeing 737, have significantly fewer fatalities in comparison to their usage, indicating improvements in safety.

(Suggestion:) Focus on contemporary commercial jets that exhibit lower historical risk rather than investing in antiquated models with high fatality rates.

2. Location of Accidents by Country

HERE NOW WE HAVE (Location of Accidents by Country). THIS ANSWERS (Where do accidents occur most frequently?)

```
#step 22
#Location of accidents by country (Top 10)
accidents_by_country = df['country'].value_counts().head(10)

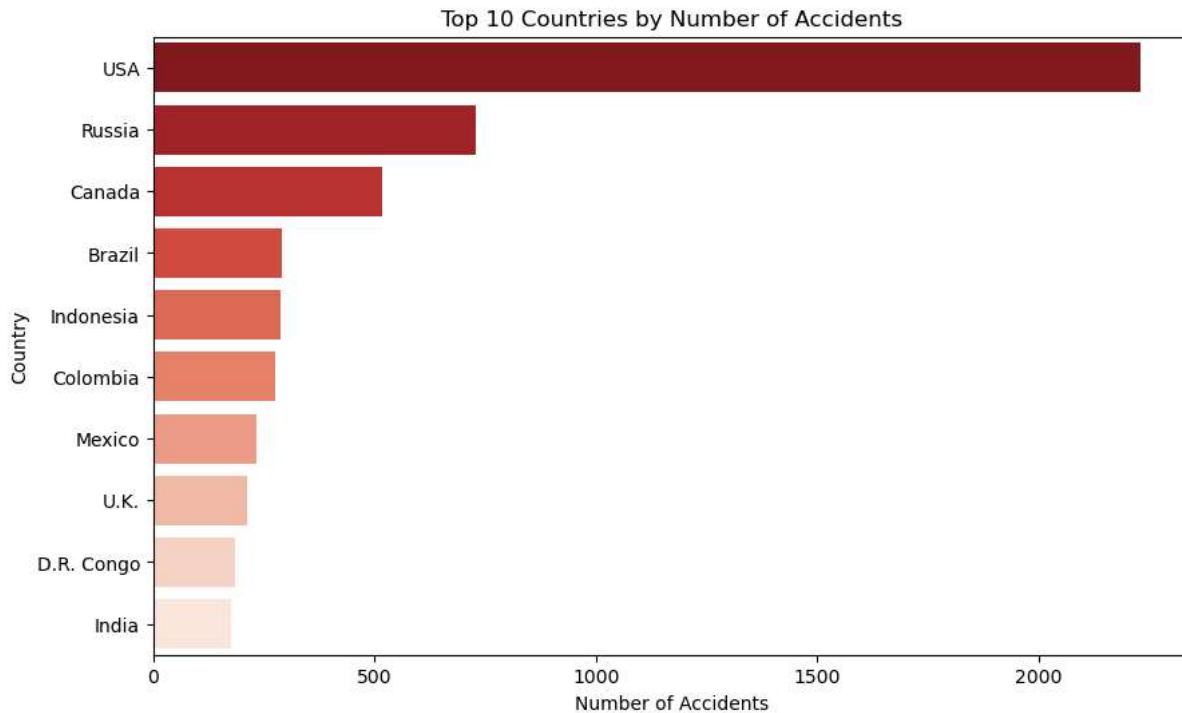
plt.figure(figsize=(10,6))
```

```
sns.barplot(x=accidents_by_country.values, y=accidents_by_country.index, palette="Reds_r")
plt.title("Top 10 Countries by Number of Accidents")
plt.xlabel("Number of Accidents")
plt.ylabel("Country")
plt.show()
```

C:\Users\Dennis\AppData\Local\Temp\ipykernel_18196\755943688.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `l`

```
sns.barplot(x=accidents_by_country.values, y=accidents_by_country.index, palette="Reds_r")
```



2.1 Business Recommendation

(Above Business Insight) Countries like (Brazil), (Russia), and the (United States) have the highest accident rates. The increased volume of air traffic is partially to blame for this. It also indicates that the frequency of accidents is influenced by regional traffic and safety regulations.

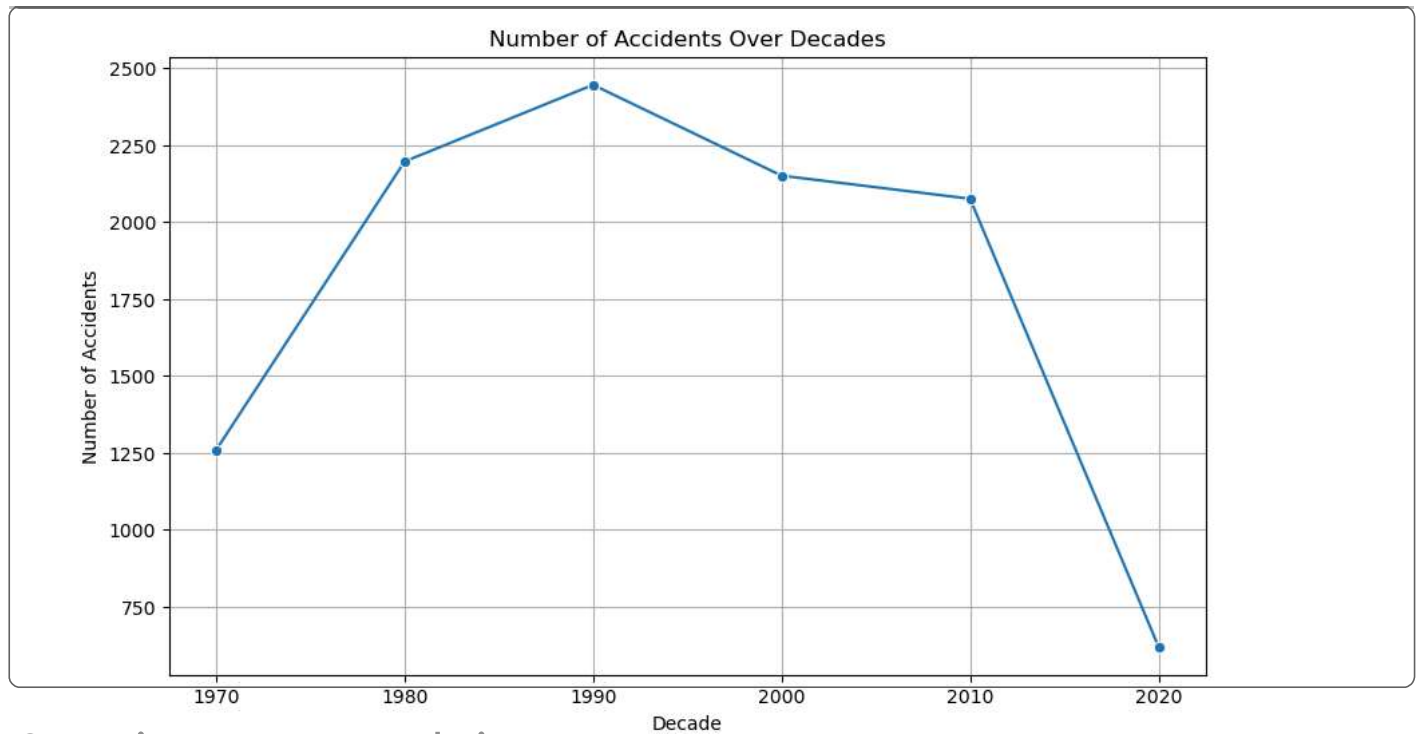
(Suggestion): When growing operations, think about concentrating on nations with strict aviation regulations (like (Western Europe)) and exercising extra caution in areas with a history of more accidents.

✓ 3. The Time of Accidents

HERE WE HAVE (The Time of Accidents) This provides an answer to the question: (How frequently do accidents occur over the course of years or decades?)

```
#Step 23
#Number of accidents by decade
accidents_by_decade = df['decade'].value_counts().sort_index()

plt.figure(figsize=(10,6))
sns.lineplot(x=accidents_by_decade.index, y=accidents_by_decade.values, marker="o")
plt.title("Number of Accidents Over Decades")
plt.xlabel("Decade")
plt.ylabel("Number of Accidents")
plt.grid(True)
plt.show()
```

3.1 Business Recommendation

Above Business Insight According to the trend, accidents peaked between **(1960)** and **(1980)** and then steadily decreased in the following decades. This decrease is a result of improvements in safety procedures, regulatory supervision, and aviation technology.

Suggestion It is advised that modern aircraft fleets from the **(2000s)** onward be given priority because they operate in a far safer environment than those from earlier decades.

Final Summary of Business

Our study of aviation accident data from 1919 to 2023 shows that **modern commercial aircraft like the Boeing 737 and Airbus A320 families** are the safest to invest in. Compared to older planes like the Douglas DC-3 or Antonov series, which are responsible for a lot of historical accidents, these models have a lot fewer deaths. Also, the fact that accidents have been going down steadily since the 1980s shows that technology, regulatory oversight, and safety protocols have all gotten better and are now built into newer planes.

The safest course of action for the company's aviation debut is to **make investments in contemporary business jets for either private and business use** and steer clear of antiquated models with a high accident rate. Further lowering risk exposure will come from operating mostly in areas like Western European countries and North America that have strict aviation safety regulations. The business can confidently expand its airline portfolio while protecting passengers and operations by concentrating on dependable aircraft families with established safety records.