AVIATION ACCIDENTS

Business Understanding

Business Problem: My company is expanding into the aviation industry but lacks insights into aircraft safety risks. This analysis identifies the lowest-risk aircraft models for commercial and private use.

Problem Statement

The aviation industry may be able to improve safety measures by analyzing accident data to identify patterns in aircraft damage and fatality rates. Doing so will allow airlines, manufacturers, and regulatory agencies to better understand risk factors and implement strategies to enhance aviation safety. Using aircraft accident data, I will examine key trends to determine which aircraft models demonstrate strong safety records and provide actionable insights for improving aviation safety standards.

Business Objectives

1.Which aircraft models have the highest fatality rates?

2.Try and find the leading cause of aviation accidents?

3.Trends in aviation fatalities over the years?

4.Recomend actions to help mitigate/prevent aviation accidents and incedents?

Data Understanding

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
df = pd.read_csv("aviation-accident-data-2023-05-16.csv")
df.head()
```

Out[2]:

	date	type	registration	operator	fatalities	location	country	cat	year
0	date unk.	Antonov An-12B	T-1206	Indonesian AF	NaN	NaN	Unknown country	U1	unknown
1	date unk.	Antonov An-12B	T-1204	Indonesian AF	NaN	NaN	Unknown country	U1	unknown
2	date unk.	Antonov An-12B	T-1201	Indonesian AF	NaN	NaN	Unknown country	U1	unknown
3	date unk.	Antonov An-12BK	NaN	Soviet AF	NaN	Tiksi Airport (IKS)	Russia	A1	unknown
4	date unk.	Antonov An-12BP	CCCP-11815	Soviet AF	0	Massawa Airport ...	Eritrea	A1	unknown

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23967 entries, 0 to 23966
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   date          23967 non-null  object
 1   type          23967 non-null  object
 2   registration  22419 non-null  object
 3   operator      23963 non-null  object
 4   fatalities    20029 non-null  object
 5   location      23019 non-null  object
 6   country       23967 non-null  object
 7   cat           23967 non-null  object
 8   year          23967 non-null  object
dtypes: object(9)
memory usage: 1.6+ MB
```

In [4]:
```
df.describe()
```

Out[4]:

	date	type	registration	operator	fatalities	location	country	cat	year
count	23967	23967	22419	23963	20029	23019	23967	23967	23967
unique	15079	3201	21962	6017	369	14608	232	11	106
top	10-MAY-1940	Douglas C-47A (DC-3)	LZ-...	USAAF	0	unknown	USA	A1	1944
freq	171	1916	13	2604	10713	272	4377	17424	1505

In [5]:
```
df.shape
```

Out[5]: (23967, 9)

Data Preparation

Data Cleaning

In [6]:
```
# check for duplicated values
df.duplicated().value_counts()
```

Out[6]:
```
False    23852
True       115
dtype: int64
```

In [7]:
```
 #View duplicates
df[df.duplicated()].head(15)
```

Out[7]:

	date	type	registration	operator	fatalities	location	country	cat	year
542	13-APR-1940	Junkers Ju-52/3m	NaN	German AF	NaN	Gangsoya, Sogn o...	Norway	A1	1940
560	29-APR-1940	Junkers Ju-52/3m	NaN	German AF	0	Oslo-Fornebu Air...	Norway	A1	1940
568	10-MAY-1940	Junkers Ju-52/3m	NaN	German AF	NaN	Waalhaven	Netherlands	A1	1940
577	10-MAY-1940	Junkers Ju-52/3m	NaN	German AF	NaN	near Den Haag	Netherlands	A1	1940
579	10-MAY-1940	Junkers Ju-52/3m	NaN	German AF	NaN	Waalhaven	Netherlands	A1	1940
580	10-MAY-1940	Junkers Ju-52/3m	NaN	German AF	NaN	near Den Haag	Netherlands	A1	1940
581	10-MAY-1940	Junkers Ju-52/3m	NaN	German AF	NaN	near Den Haag	Netherlands	A1	1940
582	10-MAY-1940	Junkers Ju-52/3m	NaN	German AF	NaN	near Den Haag	Netherlands	A1	1940
584	10-MAY-1940	Junkers Ju-52/3m	NaN	German AF	NaN	Waalhaven	Netherlands	A1	1940
585	10-MAY-1940	Junkers Ju-52/3m	NaN	German AF	NaN	near Den Haag	Netherlands	A1	1940
587	10-MAY-1940	Junkers Ju-52/3m	NaN	German AF	NaN	near Den Haag	Netherlands	A1	1940
588	10-MAY-1940	Junkers Ju-52/3m	NaN	German AF	NaN	near Den Haag	Netherlands	A1	1940
589	10-MAY-1940	Junkers Ju-52/3m	NaN	German AF	NaN	Waalhaven	Netherlands	A1	1940
590	10-MAY-1940	Junkers Ju-52/3m	NaN	German AF	NaN	Waalhaven	Netherlands	A1	1940

	date	type	registration	operator	fatalities	location	country	cat	year
605	10-MAY-1940	Junkers Ju-52/3m	NaN	German AF	NaN	Waalhaven	Netherlands	A1	1940

In [8]:
```
# removing  duplicated values
df = df.drop_duplicates()
```

In [9]:
```
# verify that they have been removed

df.duplicated().value_counts()
```

Out[9]:
```
False    23852
dtype: int64
```

In [10]:
```
# gives the columns names in the dataset
df.columns
```

Out[10]:
```
Index(['date', 'type', 'registration', 'operator', 'fatalities', 'location',
       'country', 'cat', 'year'],
      dtype='object')
```

Dealing with missing values

Detecting Null Values

In [11]:
```
# detecting null  values
#isnull

df.isna().sum()
```

Out[11]:
```
date              0
type              0
registration    1434
operator           4
fatalities      3833
location         932
country            0
cat                0
year               0
dtype: int64
```

Handling missing values

In [12]:
```
df["operator"] = df["operator"].fillna("Unknown")
df.isna().sum()
```

Out[12]:
```
date              0
type              0
registration    1434
operator          0
fatalities      3833
location         932
country            0
cat                0
year               0
dtype: int64
```

In [13]:
```
df["fatalities"] = df["fatalities"].fillna("no fatalities")
df.isna().sum()
```

Out[13]:
```
date              0
type              0
registration    1434
operator          0
fatalities        0
location         932
country            0
cat                0
year               0
dtype: int64
```

In [14]:
```
df["location"] = df["operator"].fillna("unknown location")
df.isna().sum()
```

Out[14]:
```
date              0
type              0
registration    1434
operator          0
fatalities        0
location          0
country           0
cat               0
year              0
dtype: int64
```

In [15]:
```
df["registration"] = df["registration"].fillna("no registeration")
df.isna().sum()
```

Out[15]:
```
date            0
type            0
registration    0
operator        0
fatalities      0
location        0
country         0
cat             0
year            0
dtype: int64
```

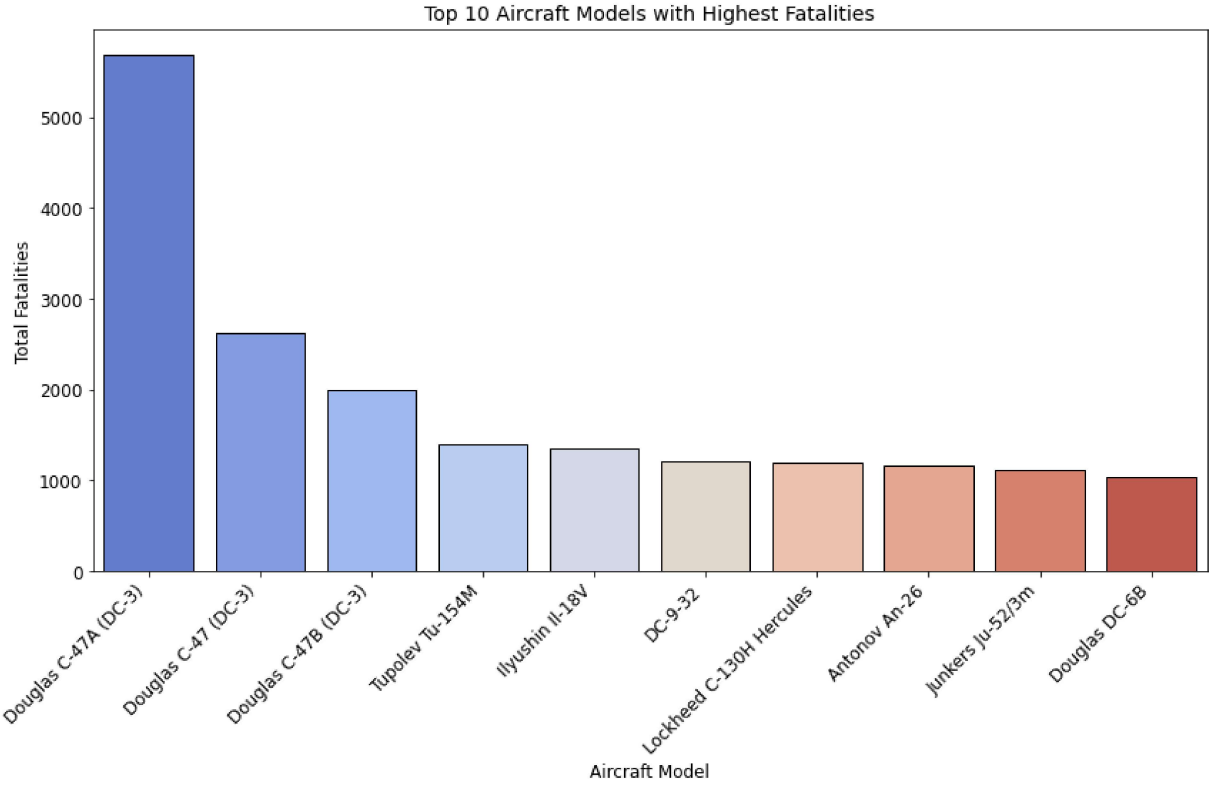
Data Analysis

Objective 1 : Which aircraft models have the highest fatality rates?

In [16]:
```
df["fatalities"] = pd.to_numeric(df["fatalities"], errors="coerce")
```

In [17]:
```
# Using groupby
top_fatal_models = df.groupby("type")["fatalities"].sum().sort_values(ascending=Fal
```

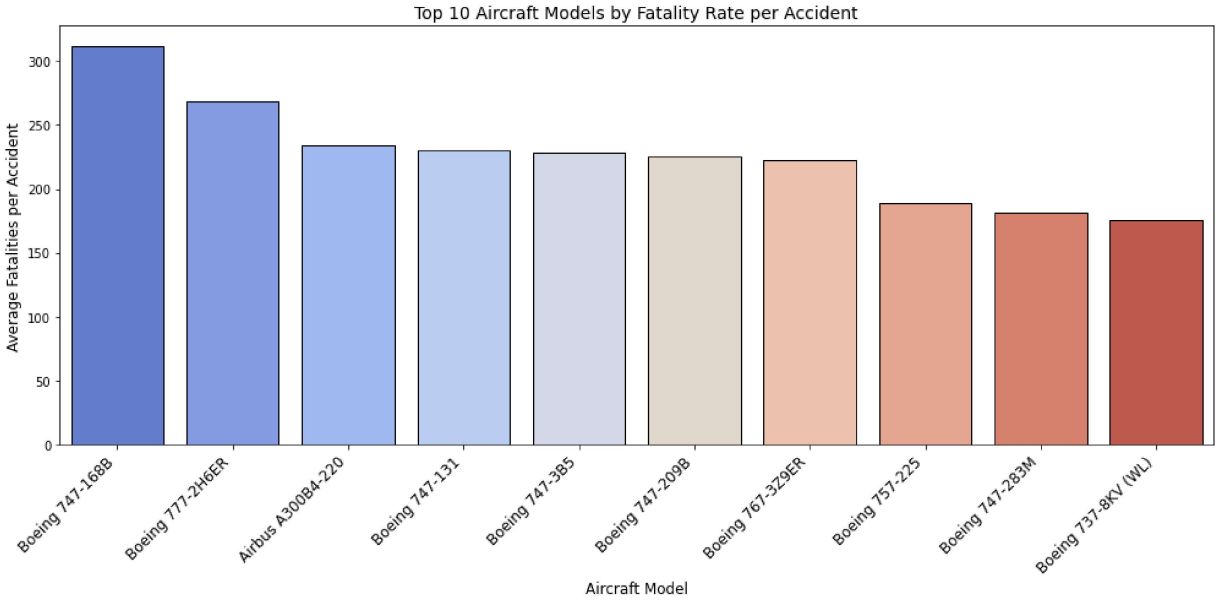
In [18]:
```
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(14, 7))
ax = sns.barplot(x=top_fatal_models.index, y=top_fatal_models.values, palette="cool
plt.xticks(rotation=45, ha="right", fontsize=12)
plt.yticks(fontsize=12)
plt.title("Top 10 Aircraft Models with Highest Fatalities", fontsize=14)
plt.xlabel("Aircraft Model", fontsize=12)
plt.ylabel("Total Fatalities", fontsize=12)
plt.show()
```



In [19]:
```
# Calcuate fatality rate per model
df["fatalities"] = pd.to_numeric(df["fatalities"], errors="coerce")
accident_counts = df.groupby("type")["fatalities"].count()
# Sum total fatalities per model
fatality_sums = df.groupby("type")["fatalities"].sum()
fatality_rate = (fatality_sums / accident_counts).sort_values(ascending=False).head
```

In [20]:
```
plt.figure(figsize=(14, 7))
ax = sns.barplot(x=fatality_rate.index, y=fatality_rate.values, palette="coolwarm",
plt.title("Top 10 Aircraft Models by Fatality Rate per Accident", fontsize=14)
plt.xticks(rotation=45, ha="right", fontsize=12)
```

```
plt.xlabel("Aircraft Model", fontsize=12)
plt.ylabel("Average Fatalities per Accident", fontsize=12)
plt.tight_layout()
plt.show()
```

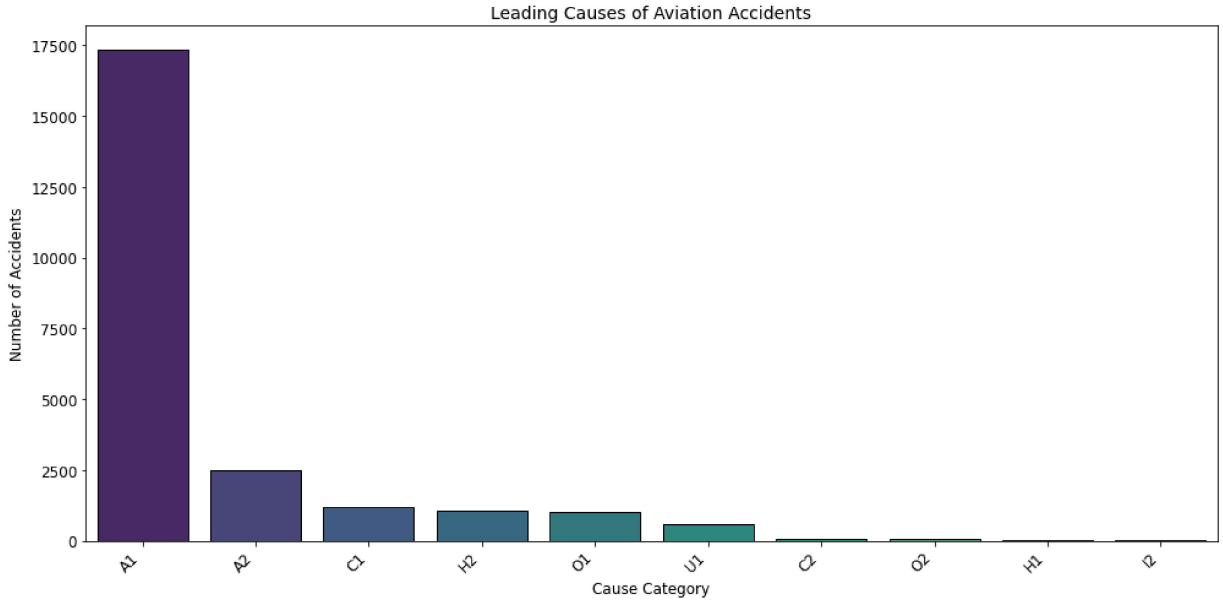


Boeing 747-168B has the highest fatalities (approximately 320 cases) -has the highest fatality per accident due to it's large size.Boeing 747-168B has one largest seating capacity amoung aircraft models.

Douglas C47-A also has significant fatalities (is an older generation model tending to have higher accident rates due to outdated systems)

Objective 2 :Try and find the leading cause of aviation accidents?

In [21]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Count number of accidents by cause
leading_causes = df["cat"].value_counts().head(10)

# Plotting
plt.figure(figsize=(14, 7))
sns.barplot(x=leading_causes.index, y=leading_causes.values, palette="viridis", edg

# Customize labels
plt.xticks(rotation=45, ha="right", fontsize=12)
plt.yticks(fontsize=12)
plt.title("Leading Causes of Aviation Accidents", fontsize=14)
plt.xlabel("Cause Category", fontsize=12)
plt.ylabel("Number of Accidents", fontsize=12)
plt.tight_layout()

plt.show()
```

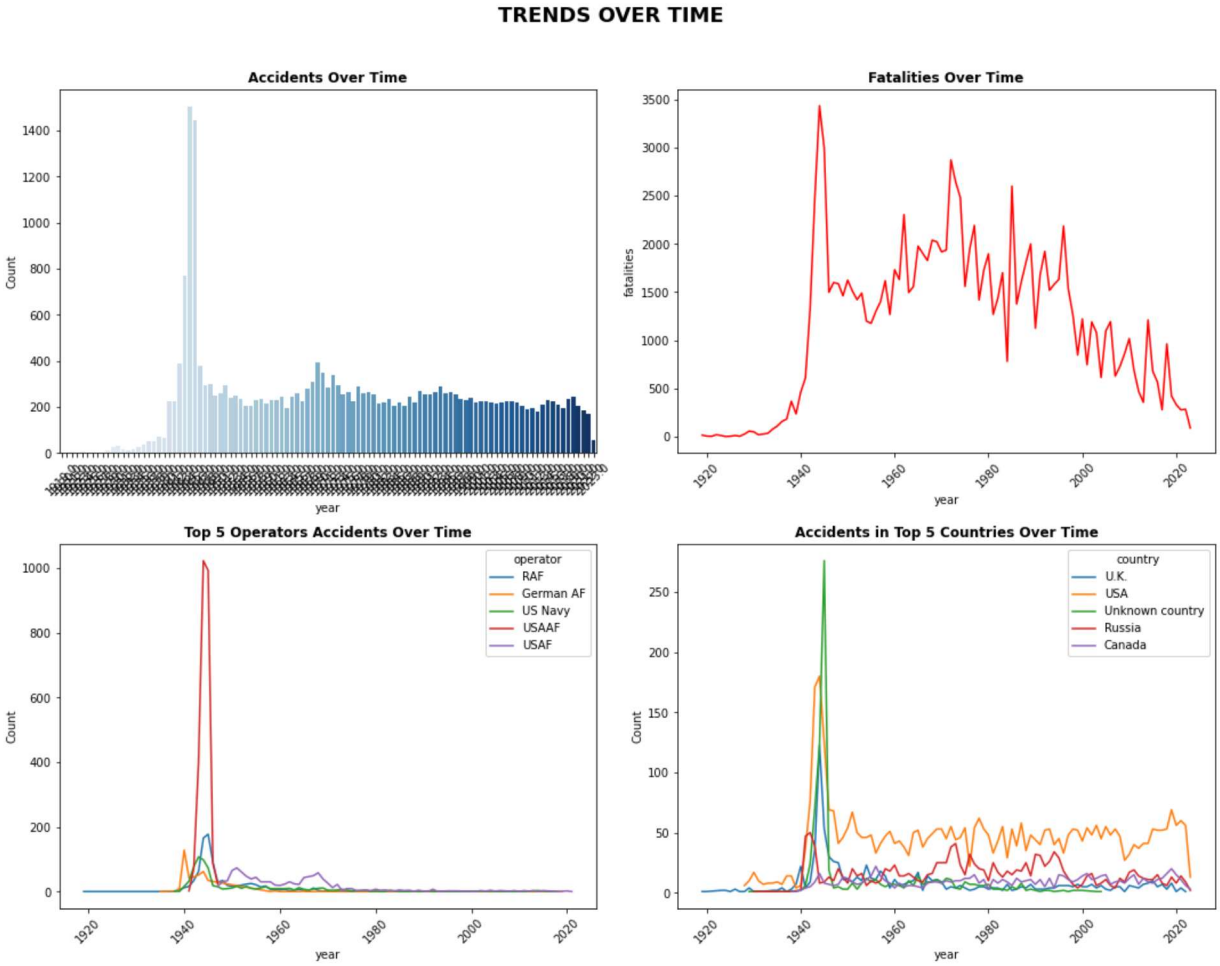


In [22]:
```
# See all unique values in the 'cat' column
print(df["cat"].unique())
```

```
['U1' 'A1' 'A2' 'O1' 'H2' 'C1' 'C2' 'H1' 'O2' 'I2' 'I1']
```

In [23]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Make sure year is numeric
df['year'] = pd.to_numeric(df['year'], errors='coerce')

# Create figure and axes
fig, axes = plt.subplots(2, 2, figsize=(15, 12))
fig.suptitle('TRENDS OVER TIME', fontsize=18, fontweight='bold')

# 1st Plot: Number of accidents per year
accidents_per_year = df.groupby('year').size().reset_index(name='Count')
sns.barplot(x='year', y='Count', data=accidents_per_year, ax=axes[0, 0], palette='B
axes[0, 0].set_title("Accidents Over Time", fontweight='bold')
axes[0, 0].tick_params(axis='x', rotation=45)

# 2nd Plot: Fatalities per year
fatalities_per_year = df.groupby('year')['fatalities'].sum().reset_index()
sns.lineplot(x='year', y='fatalities', data=fatalities_per_year, ax=axes[0, 1], col
axes[0, 1].set_title("Fatalities Over Time", fontweight='bold')
axes[0, 1].tick_params(axis='x', rotation=45)

# 3rd Plot: Top 5 operators with most accidents over time
top_5_operators = df['operator'].value_counts().head(5).index
df_top_ops = df[df['operator'].isin(top_5_operators)]
ops_trends = df_top_ops.groupby(['year', 'operator']).size().reset_index(name='Coun
sns.lineplot(x='year', y='Count', hue='operator', data=ops_trends, ax=axes[1, 0])
axes[1, 0].set_title("Top 5 Operators Accidents Over Time", fontweight='bold')
axes[1, 0].tick_params(axis='x', rotation=45)

# 4th Plot: Accidents by country over time (top 5)
```

```
top_5_countries = df['country'].value_counts().head(5).index
df_top_countries = df[df['country'].isin(top_5_countries)]
country_trends = df_top_countries.groupby(['year', 'country']).size().reset_index(n
sns.lineplot(x='year', y='Count', hue='country', data=country_trends, ax=axes[1, 1]
axes[1, 1].set_title("Accidents in Top 5 Countries Over Time", fontweight='bold')
axes[1, 1].tick_params(axis='x', rotation=45)

# Adjust layout and show
plt.tight_layout(rect=[0, 0, 1, 0.96])
plt.show()
```



The cat column in aviation accident datasets typically stands for category or cause category of the accident.

Depending on the dataset source, it might contain labels like:

Pilot Error → Mistakes by the flight crew

Mechanical Failure → Engine or system failures

Weather → Storms, low visibility, turbulence

Sabotage → Hijackings or terrorism

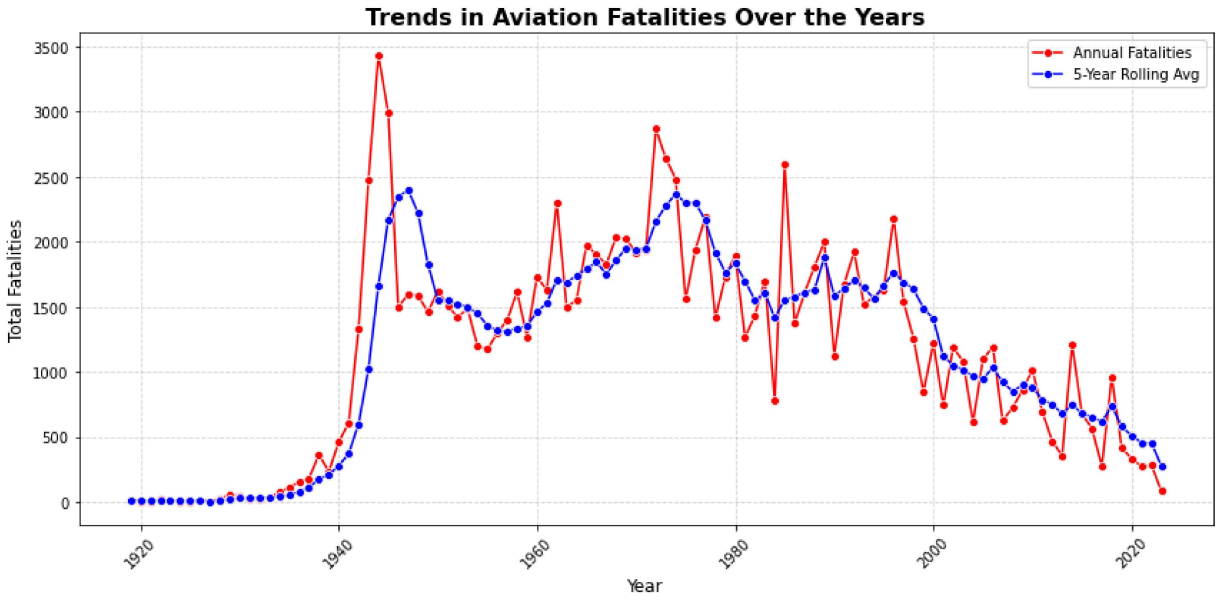
Other → Anything else not covered above

In the ASN (Aviation Safety Network) database or similar aviation-accident data sources, A1 usually denotes a hull loss (the aircraft is destroyed or damaged beyond safe repair)

In conclusion the data shows most aviation accidence happen as a result of damaged air crafts.Most of the other graphs depicts the accidence taking place in the 1940s when repairing technology was poor.A1 accidences mostly affected early aviation adoptors such as the USA and UK.

Objective :3 Trends in aviation fatalities over the years?

In [24]:
```
# Make sure 'year' and 'fatalities' are numeric
df["year"] = pd.to_numeric(df["year"], errors="coerce")
df["fatalities"] = pd.to_numeric(df["fatalities"], errors="coerce")

# Group by year and sum fatalities
fatalities_per_year = df.groupby("year")["fatalities"].sum().reset_index()

# Add 5-year rolling average
fatalities_per_year["Rolling_Avg"] = fatalities_per_year["fatalities"].rolling(wind

# Plot
plt.figure(figsize=(12, 6))
sns.lineplot(x="year", y="fatalities", data=fatalities_per_year, marker="o", label=
sns.lineplot(x="year", y="Rolling_Avg", data=fatalities_per_year, marker="o", label

# Formatting
plt.title("Trends in Aviation Fatalities Over the Years", fontsize=16, fontweight="
plt.xlabel("Year", fontsize=12)
plt.ylabel("Total Fatalities", fontsize=12)
plt.xticks(rotation=45)
plt.legend()
plt.grid(True, linestyle="--", alpha=0.5)
plt.tight_layout()

plt.show()
```

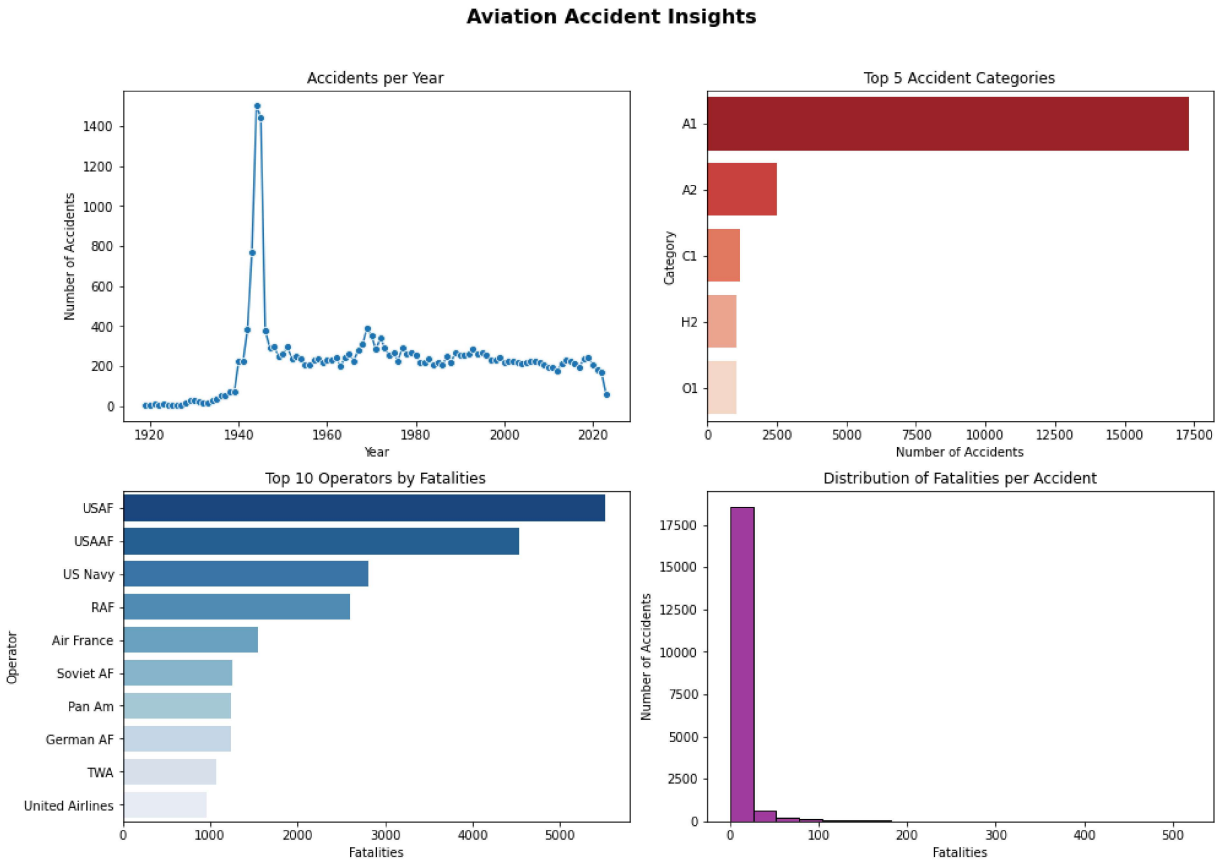


The roling avarage is there to remove outliering years.This illuminates the average trend fetal aeroplane accidences occurred in the early adoption years.As technology got better over the years accidence happened less often.

Objective:4 Recomend actions to help mitigate/prevent aviation accidents and incedents?

In [25]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Create figure with subplots (2 rows, 2 columns)
fig, axes = plt.subplots(2, 2, figsize=(14, 10))
fig.suptitle("Aviation Accident Insights", fontsize=16, fontweight='bold')

#  1 Accidents per Year (Trend)
accidents_per_year = df.groupby("year").size()
sns.lineplot(x=accidents_per_year.index, y=accidents_per_year.values, marker="o", a
axes[0, 0].set_title("Accidents per Year")
axes[0, 0].set_xlabel("Year")
axes[0, 0].set_ylabel("Number of Accidents")

#  2 Top 5 Accident Categories
top_cats = df["cat"].value_counts().head(5)
sns.barplot(x=top_cats.values, y=top_cats.index, ax=axes[0, 1], palette="Reds_r")
axes[0, 1].set_title("Top 5 Accident Categories")
axes[0, 1].set_xlabel("Number of Accidents")
axes[0, 1].set_ylabel("Category")

#  3 Fatalities per Operator (Top 10)
fatalities_by_operator = df.groupby("operator")['fatalities'].sum().sort_values(asc
sns.barplot(x=fatalities_by_operator.values, y=fatalities_by_operator.index, ax=axe
axes[1, 0].set_title("Top 10 Operators by Fatalities")
axes[1, 0].set_xlabel("Fatalities")
axes[1, 0].set_ylabel("Operator")

# 4 Fatalities Distribution
sns.histplot(df["fatalities"], bins=20, kde=False, ax=axes[1, 1], color="purple")
```

```
axes[1, 1].set_title("Distribution of Fatalities per Accident")
axes[1, 1].set_xlabel("Fatalities")
axes[1, 1].set_ylabel("Number of Accidents")

# Adjust layout and save
plt.tight_layout(rect=[0, 0, 1, 0.96])  # prevent overlap with suptitle

plt.show()
```



As the above plots indicate.Aviation accidents and incidents usually result from a chain of factors rather than a single cause. Effective prevention and mitigation require actions across people, technology, procedures, and culture.

SAVING DATA TO CSV FILE

We will need this cleaned data to plot visualizations therefore we need to a csv file so we can easily retrive it to make the different plots.

We will save this file as cleaned_aviation_data.csv and store it in the data set folder inside our projects folder

In [26]:
```
#Saving our cleaned data to csv file

df.to_csv("phase_1_project.csv", index=False)
```

Recomendations and conclusion

Our data illuminates the average trend fetal aero plane accidences occurred in the early adoption years. As technology got better over the years accidence happened less often. Aviation accidents and incidents usually result from a chain of factors rather than a single cause. Effective prevention and mitigation require actions across people, technology, procedures, and culture. An improved aircraft safety measure, Enhanced pilot training and increased awareness can contribute to aviation safety.

This analysis will ensure a successful expansion into the aviation industry.