Phase-One-Project

July 25, 2025

1 AVIATION EXPANSION PROJECT

1.1 Description

Mawingu Airlines has recently enjoyed a run of success, and now we feel that now is the right time to expand our ventures, specifically in the aviation industry. This project seeks to establish whether that path is viable at this point in time by conducting a thorough analysis of data related to various aircraft models. We will determine which aircraft are the lowest risk for the company to start this new business endeavor and then translate your findings into actionable insights that the head of the new aviation division can use to help decide which aircraft to purchase.

1.2 Methodology

The data in focus is contained in the *Aviation dataset* from the National Transportation Safety Board that includes aviation accident data from 1962 to 2023 about civil aviation accidents and selected incidents in the United States and international waters. Our key areas of focus will be three areas: - **Data Cleaning and Imputation**: We will perform key cleaning techniques on our data and filling in missing values - **Data Analysis**: Analysis of key metrics needed for insights into the business - **Data Vizualizations**: Graphical vizualization of the key metrics

1.3 1. Data Cleaning and Imputation

As explained earlier, our data is contained in the *Aviation Dataset* which we will load using the *Pandas* library. This will make viewing our data easier and enable cleaning of the data to be fast and efficient. As always, we will import pandas using the standard alias and read it into our notebook.

[1]:
```
import pandas as pd
Aviation_data = pd.read_csv('Aviation_Data.csv')
Aviation_data.head(20)
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_8244\1316027294.py:2: DtypeWarning:
Columns (6,7,28) have mixed types. Specify dtype option on import or set
low_memory=False.
  Aviation_data = pd.read_csv('Aviation_Data.csv')
```

[1]:

	Event.Id	Investigation.Type	Accident.Number	Event.Date	\
0	20001218X45444	Accident	SEA87LA080	1948-10-24	
1	20001218X45447	Accident	LAX94LA336	1962-07-19	
2	20061025X01555	Accident	NYC07LA005	1974-08-30	

```
3   20001218X45448           Accident      LAX96LA321  1977-06-19
4   20041105X01764           Accident      CHI79FA064  1979-08-02
5   20170710X52551           Accident      NYC79AA106  1979-09-17
6   20001218X45446           Accident      CHI81LA106  1981-08-01
7   20020909X01562           Accident      SEA82DA022  1982-01-01
8   20020909X01561           Accident      NYC82DA015  1982-01-01
9   20020909X01560           Accident      MIA82DA029  1982-01-01
10  20020909X01559           Accident      FTW82DA034  1982-01-01
11  20020909X01558           Accident      ATL82DKJ10  1982-01-01
12  20020917X02148           Accident      FTW82FRJ07  1982-01-02
13  20020917X02134           Accident      FTW82FRA14  1982-01-02
14  20020917X02119           Accident      FTW82FPJ10  1982-01-02
15  20020917X02117           Accident      FTW82FPG08  1982-01-02
16  20020917X01962           Accident      DEN82DTM08  1982-01-02
17  20020917X01656           Accident      ANC82FAG14  1982-01-02
18  20020917X02481           Accident      NYC82DA016  1982-01-02
19  20020917X02339           Accident      MIA82DA028  1982-01-02

            Location        Country   Latitude  Longitude Airport.Code  \
0    MOOSE CREEK, ID  United States        NaN        NaN          NaN
1     BRIDGEPORT, CA  United States        NaN        NaN          NaN
2      Saltville, VA  United States  36.922223 -81.878056          NaN
3         EUREKA, CA  United States        NaN        NaN          NaN
4         Canton, OH  United States        NaN        NaN          NaN
5         BOSTON, MA  United States  42.445277 -70.758333          NaN
6         COTTON, MN  United States        NaN        NaN          NaN
7       PULLMAN, WA  United States        NaN        NaN          NaN
8   EAST HANOVER, NJ  United States        NaN        NaN          N58
9   JACKSONVILLE, FL  United States        NaN        NaN          JAX
10         HOBBS, NM  United States        NaN        NaN          NaN
11     TUSKEGEE, AL  United States        NaN        NaN          NaN
12        HOMER, LA  United States        NaN        NaN          NaN
13       HEARNE, TX  United States        NaN        NaN          T72
14    CHICKASHA, OK  United States        NaN        NaN          NaN
15   LITTLE ROCK, AR  United States        NaN        NaN          NaN
16        MIDWAY, UT  United States        NaN        NaN          NaN
17      SKWENTA, AK  United States        NaN        NaN          NaN
18      GALETON, PA  United States        NaN        NaN          5G6
19        MIAMI, FL  United States        NaN        NaN          NaN

          Airport.Name  … Purpose.of.flight Air.carrier  \
0                  NaN  …          Personal         NaN
1                  NaN  …          Personal         NaN
2                  NaN  …          Personal         NaN
3                  NaN  …          Personal         NaN
4                  NaN  …          Personal         NaN
5                  NaN  …               NaN  Air Canada
```

```
6                  NaN  …          Personal         NaN
7   BLACKBURN AG STRIP  …          Personal         NaN
8              HANOVER  …          Business         NaN
9    JACKSONVILLE INTL  …          Personal         NaN
10                 NaN  …          Personal         NaN
11           TUSKEGEE  …          Personal         NaN
12                 NaN  …          Personal         NaN
13    HEARNE MUNICIPAL  …          Personal         NaN
14                 NaN  …          Personal         NaN
15                 NaN  …          Personal         NaN
16         FIELD RANCH  …          Personal         NaN
17                 NaN  …          Personal         NaN
18      CHERRY SPRINGS  …          Personal         NaN
19                 NaN  …          Personal         NaN

   Total.Fatal.Injuries Total.Serious.Injuries Total.Minor.Injuries  \
0                   2.0                   0.0                   0.0
1                   4.0                   0.0                   0.0
2                   3.0                   NaN                   NaN
3                   2.0                   0.0                   0.0
4                   1.0                   2.0                   NaN
5                   NaN                   NaN                   1.0
6                   4.0                   0.0                   0.0
7                   0.0                   0.0                   0.0
8                   0.0                   0.0                   0.0
9                   0.0                   0.0                   3.0
10                  0.0                   0.0                   0.0
11                  0.0                   0.0                   0.0
12                  0.0                   0.0                   1.0
13                  1.0                   0.0                   0.0
14                  1.0                   0.0                   0.0
15                  2.0                   0.0                   0.0
16                  0.0                   0.0                   0.0
17                  3.0                   0.0                   0.0
18                  0.0                   0.0                   0.0
19                  0.0                   0.0                   0.0

   Total.Uninjured Weather.Condition  Broad.phase.of.flight   Report.Status  \
0              0.0               UNK                 Cruise  Probable Cause
1              0.0               UNK               Unknown  Probable Cause
2              NaN               IMC                 Cruise  Probable Cause
3              0.0               IMC                 Cruise  Probable Cause
4              0.0               VMC              Approach  Probable Cause
5             44.0               VMC                 Climb  Probable Cause
6              0.0               IMC               Unknown  Probable Cause
7              2.0               VMC                Takeoff  Probable Cause
8              2.0               IMC                Landing  Probable Cause
```

```
9              0.0              IMC               Cruise  Probable Cause
10             1.0              VMC             Approach  Probable Cause
11             1.0              VMC              Landing  Probable Cause
12             0.0              IMC               Cruise  Probable Cause
13             0.0              IMC              Takeoff  Probable Cause
14             0.0              IMC               Cruise  Probable Cause
15             0.0              IMC               Cruise  Probable Cause
16             1.0              IMC                 Taxi  Probable Cause
17             0.0              VMC              Unknown  Probable Cause
18             1.0              VMC                 Taxi  Probable Cause
19             2.0              VMC               Cruise  Probable Cause

   Publication.Date
0               NaN
1        19-09-1996
2        26-02-2007
3        12-09-2000
4        16-04-1980
5        19-09-2017
6        06-11-2001
7        01-01-1982
8        01-01-1982
9        01-01-1982
10       01-01-1982
11       01-01-1982
12       02-01-1983
13       02-01-1983
14       02-01-1983
15       02-01-1983
16       02-01-1983
17       02-01-1983
18       02-01-1983
19       02-01-1983

[20 rows x 31 columns]
```

As you can see, pandas has indicated that some columns have mixed data types therefore making it difficult for pandas to inerpret them. We can set the parameter ***low_memory=False*** in our ***pd.read_csv*** function.

[2]:
```
import pandas as pd
Aviation_data = pd.read_csv('Aviation_Data.csv', low_memory=False)
Aviation_data.head(20)
```

[2]:
```
          Event.Id Investigation.Type Accident.Number  Event.Date  \
0   20001218X45444           Accident      SEA87LA080  1948-10-24
1   20001218X45447           Accident      LAX94LA336  1962-07-19
2   20061025X01555           Accident      NYC07LA005  1974-08-30
```

```
3   20001218X45448           Accident      LAX96LA321  1977-06-19
4   20041105X01764           Accident      CHI79FA064  1979-08-02
5   20170710X52551           Accident      NYC79AA106  1979-09-17
6   20001218X45446           Accident      CHI81LA106  1981-08-01
7   20020909X01562           Accident      SEA82DA022  1982-01-01
8   20020909X01561           Accident      NYC82DA015  1982-01-01
9   20020909X01560           Accident      MIA82DA029  1982-01-01
10  20020909X01559           Accident      FTW82DA034  1982-01-01
11  20020909X01558           Accident      ATL82DKJ10  1982-01-01
12  20020917X02148           Accident      FTW82FRJ07  1982-01-02
13  20020917X02134           Accident      FTW82FRA14  1982-01-02
14  20020917X02119           Accident      FTW82FPJ10  1982-01-02
15  20020917X02117           Accident      FTW82FPG08  1982-01-02
16  20020917X01962           Accident      DEN82DTM08  1982-01-02
17  20020917X01656           Accident      ANC82FAG14  1982-01-02
18  20020917X02481           Accident      NYC82DA016  1982-01-02
19  20020917X02339           Accident      MIA82DA028  1982-01-02

            Location        Country   Latitude   Longitude Airport.Code  \
0    MOOSE CREEK, ID  United States        NaN         NaN          NaN
1     BRIDGEPORT, CA  United States        NaN         NaN          NaN
2      Saltville, VA  United States  36.922223  -81.878056          NaN
3         EUREKA, CA  United States        NaN         NaN          NaN
4         Canton, OH  United States        NaN         NaN          NaN
5         BOSTON, MA  United States  42.445277  -70.758333          NaN
6         COTTON, MN  United States        NaN         NaN          NaN
7       PULLMAN, WA  United States        NaN         NaN          NaN
8   EAST HANOVER, NJ  United States        NaN         NaN          N58
9   JACKSONVILLE, FL  United States        NaN         NaN          JAX
10         HOBBS, NM  United States        NaN         NaN          NaN
11      TUSKEGEE, AL  United States        NaN         NaN          NaN
12         HOMER, LA  United States        NaN         NaN          NaN
13        HEARNE, TX  United States        NaN         NaN          T72
14     CHICKASHA, OK  United States        NaN         NaN          NaN
15   LITTLE ROCK, AR  United States        NaN         NaN          NaN
16        MIDWAY, UT  United States        NaN         NaN          NaN
17       SKWENTA, AK  United States        NaN         NaN          NaN
18       GALETON, PA  United States        NaN         NaN          5G6
19         MIAMI, FL  United States        NaN         NaN          NaN

          Airport.Name  … Purpose.of.flight Air.carrier  \
0                  NaN  …          Personal         NaN
1                  NaN  …          Personal         NaN
2                  NaN  …          Personal         NaN
3                  NaN  …          Personal         NaN
4                  NaN  …          Personal         NaN
5                  NaN  …               NaN  Air Canada
```

```
6                 NaN  …          Personal         NaN
7   BLACKBURN AG STRIP  …          Personal         NaN
8              HANOVER  …          Business         NaN
9    JACKSONVILLE INTL  …          Personal         NaN
10                NaN  …          Personal         NaN
11          TUSKEGEE  …          Personal         NaN
12                NaN  …          Personal         NaN
13   HEARNE MUNICIPAL  …          Personal         NaN
14                NaN  …          Personal         NaN
15                NaN  …          Personal         NaN
16        FIELD RANCH  …          Personal         NaN
17                NaN  …          Personal         NaN
18     CHERRY SPRINGS  …          Personal         NaN
19                NaN  …          Personal         NaN

   Total.Fatal.Injuries Total.Serious.Injuries Total.Minor.Injuries  \
0                   2.0                   0.0                   0.0
1                   4.0                   0.0                   0.0
2                   3.0                   NaN                   NaN
3                   2.0                   0.0                   0.0
4                   1.0                   2.0                   NaN
5                   NaN                   NaN                   1.0
6                   4.0                   0.0                   0.0
7                   0.0                   0.0                   0.0
8                   0.0                   0.0                   0.0
9                   0.0                   0.0                   3.0
10                  0.0                   0.0                   0.0
11                  0.0                   0.0                   0.0
12                  0.0                   0.0                   1.0
13                  1.0                   0.0                   0.0
14                  1.0                   0.0                   0.0
15                  2.0                   0.0                   0.0
16                  0.0                   0.0                   0.0
17                  3.0                   0.0                   0.0
18                  0.0                   0.0                   0.0
19                  0.0                   0.0                   0.0

   Total.Uninjured Weather.Condition  Broad.phase.of.flight   Report.Status  \
0              0.0               UNK                 Cruise  Probable Cause
1              0.0               UNK                Unknown  Probable Cause
2              NaN               IMC                 Cruise  Probable Cause
3              0.0               IMC                 Cruise  Probable Cause
4              0.0               VMC               Approach  Probable Cause
5             44.0               VMC                  Climb  Probable Cause
6              0.0               IMC                Unknown  Probable Cause
7              2.0               VMC                Takeoff  Probable Cause
8              2.0               IMC                Landing  Probable Cause
```

```
9              0.0              IMC                Cruise  Probable Cause
10             1.0              VMC              Approach  Probable Cause
11             1.0              VMC               Landing  Probable Cause
12             0.0              IMC                Cruise  Probable Cause
13             0.0              IMC               Takeoff  Probable Cause
14             0.0              IMC                Cruise  Probable Cause
15             0.0              IMC                Cruise  Probable Cause
16             1.0              IMC                  Taxi  Probable Cause
17             0.0              VMC               Unknown  Probable Cause
18             1.0              VMC                  Taxi  Probable Cause
19             2.0              VMC                Cruise  Probable Cause

   Publication.Date
0               NaN
1        19-09-1996
2        26-02-2007
3        12-09-2000
4        16-04-1980
5        19-09-2017
6        06-11-2001
7        01-01-1982
8        01-01-1982
9        01-01-1982
10       01-01-1982
11       01-01-1982
12       02-01-1983
13       02-01-1983
14       02-01-1983
15       02-01-1983
16       02-01-1983
17       02-01-1983
18       02-01-1983
19       02-01-1983

[20 rows x 31 columns]
```

Great! We have managed to get a small overview of our data. One can see that some columns contain missing values. We will have to see the extent of our missing valies in the data.

[3]:
```
Aviation_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 90348 entries, 0 to 90347
Data columns (total 31 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Event.Id                88889 non-null  object
 1   Investigation.Type      90348 non-null  object
```

```
 2   Accident.Number         88889 non-null  object
 3   Event.Date              88889 non-null  object
 4   Location                88837 non-null  object
 5   Country                 88663 non-null  object
 6   Latitude                34382 non-null  object
 7   Longitude               34373 non-null  object
 8   Airport.Code            50132 non-null  object
 9   Airport.Name            52704 non-null  object
 10  Injury.Severity         87889 non-null  object
 11  Aircraft.damage         85695 non-null  object
 12  Aircraft.Category       32287 non-null  object
 13  Registration.Number     87507 non-null  object
 14  Make                    88826 non-null  object
 15  Model                   88797 non-null  object
 16  Amateur.Built           88787 non-null  object
 17  Number.of.Engines       82805 non-null  float64
 18  Engine.Type             81793 non-null  object
 19  FAR.Description         32023 non-null  object
 20  Schedule                12582 non-null  object
 21  Purpose.of.flight       82697 non-null  object
 22  Air.carrier             16648 non-null  object
 23  Total.Fatal.Injuries    77488 non-null  float64
 24  Total.Serious.Injuries  76379 non-null  float64
 25  Total.Minor.Injuries    76956 non-null  float64
 26  Total.Uninjured         82977 non-null  float64
 27  Weather.Condition       84397 non-null  object
 28  Broad.phase.of.flight   61724 non-null  object
 29  Report.Status           82505 non-null  object
 30  Publication.Date        73659 non-null  object
dtypes: float64(5), object(26)
memory usage: 21.4+ MB
```

[4]:
```python
# Get total number of missing values per column
missing_values = Aviation_data.isnull().sum()

# Get percentage of missing values per column
missing_percentage = (Aviation_data.isnull().sum() / len(Aviation_data)) * 100

# Create a summary DataFrame
missing_data = pd.DataFrame({
    'Missing Values': missing_values,
    'Percentage (%)': missing_percentage
})

# Display columns with missing values (filter out zeros)
missing_data[missing_data['Missing Values'] > 0].sort_values('Percentage (%)',␣
 ↪ascending=False)
```

[4]:

	Missing Values	Percentage (%)
Schedule	77766	86.073848
Air.carrier	73700	81.573471
FAR.Description	58325	64.555939
Aircraft.Category	58061	64.263736
Longitude	55975	61.954886
Latitude	55966	61.944924
Airport.Code	40216	44.512330
Airport.Name	37644	41.665560
Broad.phase.of.flight	28624	31.681941
Publication.Date	16689	18.471909
Total.Serious.Injuries	13969	15.461327
Total.Minor.Injuries	13392	14.822686
Total.Fatal.Injuries	12860	14.233851
Engine.Type	8555	9.468942
Report.Status	7843	8.680878
Purpose.of.flight	7651	8.468367
Number.of.Engines	7543	8.348829
Total.Uninjured	7371	8.158454
Weather.Condition	5951	6.586753
Aircraft.damage	4653	5.150086
Registration.Number	2841	3.144508
Injury.Severity	2459	2.721698
Country	1685	1.865011
Amateur.Built	1561	1.727764
Model	1551	1.716695
Make	1522	1.684597
Location	1511	1.672422
Accident.Number	1459	1.614867
Event.Date	1459	1.614867
Event.Id	1459	1.614867

Wow! The table above us shows that there are a significant number of columns with missing values. This will be problematic for us since it will hamper with our analysis and our vizualiations later in the project. Let us now categorize our columns in terms of missing values.

1.3.1 Critical Missing (>50% missing):

- Schedule (86.07%)
- Air.carrier (81.57%)
- FAR.Description (64.56%)
- Aircraft.Category (64.26%)
- Longitude (61.95%)
- Latitude (61.94%)

1.3.2 Moderate Missing (10-50% missing):

- Airport.Code (44.51%)
- Airport.Name (41.67%)
- Broad.phase.of.flight (31.68%)

1.3.3 Low Missing (<10% missing):

- All other columns

For our data, we can see that we have six columns that contain a significant percentage of missing values within the columns. For the three topmost columns, since they are missing more than 80% of data and likely won't provide meaningful analysis even with imputation, we can consider dropping them.

1.3.4 1. Critical missing columns

[5]:
```
columns_to_drop = ['Schedule', 'Air.carrier', 'FAR.Description']
Aviation_data = Aviation_data.drop(columns=columns_to_drop)
```

For the columns with location data and moderate percentage of missing data, we can conduct data imputation and fill them with the relevant summary statistics, i.e., mean, median, e.t.c. , or we can decide to drop them since we have the country columns.

1.3.5 2(a). Aircraft Category:

[6]:
```
Aviation_data['Aircraft.Category'] = Aviation_data['Aircraft.Category'].
↪fillna('Unknown')
```

1.3.6 2(b). Airport.code, Airport.name and Broad.phase.of.flight:

We can perform mode imputation on these columns since dropping them will prove harmful to our analysis:

[7]:
```
Aviation_data['Airport.Code'] = Aviation_data['Airport.Code'].
↪fillna(Aviation_data['Airport.Code'].mode()[0])
Aviation_data['Airport.Name'] = Aviation_data['Airport.Name'].
↪fillna(Aviation_data['Airport.Name'].mode()[0])
Aviation_data['Broad.phase.of.flight'] = Aviation_data['Broad.phase.of.flight'].
↪fillna(Aviation_data['Broad.phase.of.flight'].mode()[0])
```

1.3.7 2(c). Latitude and longitude columns:

[8]:
```
Aviation_data['Latitude'] = pd.to_numeric(Aviation_data['Latitude'],␣
↪errors='coerce')  # 'coerce' turns invalid values to NaN
Aviation_data['Longitude'] = pd.to_numeric(Aviation_data['Longitude'],␣
↪errors='coerce')
```

Since we can see that the location columns contain a high percentage of missing values, we can just decide to drop them since they will hamper us with our analysis:

[9]:
```
Aviation_data.drop(columns=['Latitude', 'Longitude'], inplace=True)
```

1.3.8 3. Low missing columns:

Lets have a look at how our dataset looks now.

[10]:
```
Aviation_data.head(20)
```

[10]:

	Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Country	Airport.Code	Airport.Name
0	20001218X45444	Accident	SEA87LA080	1948-10-24	MOOSE CREEK, ID	United States	NONE	Private
1	20001218X45447	Accident	LAX94LA336	1962-07-19	BRIDGEPORT, CA	United States	NONE	Private
2	20061025X01555	Accident	NYC07LA005	1974-08-30	Saltville, VA	United States	NONE	Private
3	20001218X45448	Accident	LAX96LA321	1977-06-19	EUREKA, CA	United States	NONE	Private
4	20041105X01764	Accident	CHI79FA064	1979-08-02	Canton, OH	United States	NONE	Private
5	20170710X52551	Accident	NYC79AA106	1979-09-17	BOSTON, MA	United States	NONE	Private
6	20001218X45446	Accident	CHI81LA106	1981-08-01	COTTON, MN	United States	NONE	Private
7	20020909X01562	Accident	SEA82DA022	1982-01-01	PULLMAN, WA	United States	NONE	BLACKBURN AG STRIP
8	20020909X01561	Accident	NYC82DA015	1982-01-01	EAST HANOVER, NJ	United States	N58	HANOVER
9	20020909X01560	Accident	MIA82DA029	1982-01-01	JACKSONVILLE, FL	United States	JAX	JACKSONVILLE INTL
10	20020909X01559	Accident	FTW82DA034	1982-01-01	HOBBS, NM	United States	NONE	Private
11	20020909X01558	Accident	ATL82DKJ10	1982-01-01	TUSKEGEE, AL	United States	NONE	TUSKEGEE
12	20020917X02148	Accident	FTW82FRJ07	1982-01-02	HOMER, LA	United States	NONE	Private
13	20020917X02134	Accident	FTW82FRA14	1982-01-02	HEARNE, TX	United States	T72	HEARNE MUNICIPAL
14	20020917X02119	Accident	FTW82FPJ10	1982-01-02				
15	20020917X02117	Accident	FTW82FPG08	1982-01-02				
16	20020917X01962	Accident	DEN82DTM08	1982-01-02				
17	20020917X01656	Accident	ANC82FAG14	1982-01-02				
18	20020917X02481	Accident	NYC82DA016	1982-01-02				
19	20020917X02339	Accident	MIA82DA028	1982-01-02				

```
14     CHICKASHA, OK  United States         NONE             Private
15   LITTLE ROCK, AR  United States         NONE             Private
16        MIDWAY, UT  United States         NONE         FIELD RANCH
17       SKWENTA, AK  United States         NONE             Private
18       GALETON, PA  United States          5G6      CHERRY SPRINGS
19         MIAMI, FL  United States         NONE             Private

   Injury.Severity Aircraft.damage  …    Engine.Type Purpose.of.flight  \
0         Fatal(2)       Destroyed  …  Reciprocating          Personal
1         Fatal(4)       Destroyed  …  Reciprocating          Personal
2         Fatal(3)       Destroyed  …  Reciprocating          Personal
3         Fatal(2)       Destroyed  …  Reciprocating          Personal
4         Fatal(1)       Destroyed  …            NaN          Personal
5        Non-Fatal     Substantial  …      Turbo Fan               NaN
6         Fatal(4)       Destroyed  …  Reciprocating          Personal
7        Non-Fatal     Substantial  …  Reciprocating          Personal
8        Non-Fatal     Substantial  …  Reciprocating          Business
9        Non-Fatal     Substantial  …  Reciprocating          Personal
10       Non-Fatal     Substantial  …  Reciprocating          Personal
11       Non-Fatal     Substantial  …  Reciprocating          Personal
12       Non-Fatal       Destroyed  …  Reciprocating          Personal
13        Fatal(1)       Destroyed  …  Reciprocating          Personal
14        Fatal(1)       Destroyed  …  Reciprocating          Personal
15        Fatal(2)       Destroyed  …  Reciprocating          Personal
16       Non-Fatal       Destroyed  …  Reciprocating          Personal
17        Fatal(3)       Destroyed  …  Reciprocating          Personal
18       Non-Fatal     Substantial  …  Reciprocating          Personal
19       Non-Fatal     Substantial  …  Reciprocating          Personal

   Total.Fatal.Injuries Total.Serious.Injuries Total.Minor.Injuries  \
0                   2.0                    0.0                   0.0
1                   4.0                    0.0                   0.0
2                   3.0                    NaN                   NaN
3                   2.0                    0.0                   0.0
4                   1.0                    2.0                   NaN
5                   NaN                    NaN                   1.0
6                   4.0                    0.0                   0.0
7                   0.0                    0.0                   0.0
8                   0.0                    0.0                   0.0
9                   0.0                    0.0                   3.0
10                  0.0                    0.0                   0.0
11                  0.0                    0.0                   0.0
12                  0.0                    0.0                   1.0
13                  1.0                    0.0                   0.0
14                  1.0                    0.0                   0.0
15                  2.0                    0.0                   0.0
16                  0.0                    0.0                   0.0
```

```
17                 3.0                  0.0                 0.0
18                 0.0                  0.0                 0.0
19                 0.0                  0.0                 0.0

    Total.Uninjured Weather.Condition Broad.phase.of.flight   Report.Status  \
0               0.0               UNK                Cruise  Probable Cause
1               0.0               UNK               Unknown  Probable Cause
2               NaN               IMC                Cruise  Probable Cause
3               0.0               IMC                Cruise  Probable Cause
4               0.0               VMC              Approach  Probable Cause
5              44.0               VMC                 Climb  Probable Cause
6               0.0               IMC               Unknown  Probable Cause
7               2.0               VMC              Takeoff   Probable Cause
8               2.0               IMC               Landing  Probable Cause
9               0.0               IMC                Cruise  Probable Cause
10              1.0               VMC              Approach  Probable Cause
11              1.0               VMC               Landing  Probable Cause
12              0.0               IMC                Cruise  Probable Cause
13              0.0               IMC              Takeoff   Probable Cause
14              0.0               IMC                Cruise  Probable Cause
15              0.0               IMC                Cruise  Probable Cause
16              1.0               IMC                  Taxi  Probable Cause
17              0.0               VMC               Unknown  Probable Cause
18              1.0               VMC                  Taxi  Probable Cause
19              2.0               VMC                Cruise  Probable Cause

    Publication.Date
0                NaN
1         19-09-1996
2         26-02-2007
3         12-09-2000
4         16-04-1980
5         19-09-2017
6         06-11-2001
7         01-01-1982
8         01-01-1982
9         01-01-1982
10        01-01-1982
11        01-01-1982
12        02-01-1983
13        02-01-1983
14        02-01-1983
15        02-01-1983
16        02-01-1983
17        02-01-1983
18        02-01-1983
19        02-01-1983
```

```
[20 rows x 26 columns]
```

We have managed to drop 5 columns. Lets go ahead and work on the remaining columns.

1.3.9 3(a). Injury columns

For the injury columns, you notice that the severity column indicates the extent of the injuries recorded in a particular crash. One can also notice thatsome rows have for example *fatal(2)* and it corresponds to the total fatal injuries column. However some have *Fatal* but you find that both total fatal and total serious columns having inputs leaving one to question whether the severity and the various injury columns influence each other. We will need to perform some cleaning and create a new column to sort the mix-up

[11]:
```
# Check unique values in Injury.Severity to define rules
print(Aviation_data['Injury.Severity'].unique())
```

```
['Fatal(2)' 'Fatal(4)' 'Fatal(3)' 'Fatal(1)' 'Non-Fatal' 'Incident'
 'Fatal(8)' 'Fatal(78)' 'Fatal(7)' 'Fatal(6)' 'Fatal(5)' 'Fatal(153)'
 'Fatal(12)' 'Fatal(14)' 'Fatal(23)' 'Fatal(10)' 'Fatal(11)' 'Fatal(9)'
 'Fatal(17)' 'Fatal(13)' 'Fatal(29)' 'Fatal(70)' 'Unavailable'
 'Fatal(135)' 'Fatal(31)' 'Fatal(256)' 'Fatal(25)' 'Fatal(82)'
 'Fatal(156)' 'Fatal(28)' 'Fatal(18)' 'Fatal(43)' 'Fatal(15)' 'Fatal(270)'
 'Fatal(144)' 'Fatal(174)' 'Fatal(111)' 'Fatal(131)' 'Fatal(20)'
 'Fatal(73)' 'Fatal(27)' 'Fatal(34)' 'Fatal(87)' 'Fatal(30)' 'Fatal(16)'
 'Fatal(47)' 'Fatal(56)' 'Fatal(37)' 'Fatal(132)' 'Fatal(68)' 'Fatal(54)'
 'Fatal(52)' 'Fatal(65)' 'Fatal(72)' 'Fatal(160)' 'Fatal(189)'
 'Fatal(123)' 'Fatal(33)' 'Fatal(110)' 'Fatal(230)' 'Fatal(97)'
 'Fatal(349)' 'Fatal(125)' 'Fatal(35)' 'Fatal(228)' 'Fatal(75)'
 'Fatal(104)' 'Fatal(229)' 'Fatal(80)' 'Fatal(217)' 'Fatal(169)'
 'Fatal(88)' 'Fatal(19)' 'Fatal(60)' 'Fatal(113)' 'Fatal(143)' 'Fatal(83)'
 'Fatal(24)' 'Fatal(44)' 'Fatal(64)' 'Fatal(92)' 'Fatal(118)' 'Fatal(265)'
 'Fatal(26)' 'Fatal(138)' 'Fatal(206)' 'Fatal(71)' 'Fatal(21)' 'Fatal(46)'
 'Fatal(102)' 'Fatal(115)' 'Fatal(141)' 'Fatal(55)' 'Fatal(121)'
 'Fatal(45)' 'Fatal(145)' 'Fatal(117)' 'Fatal(107)' 'Fatal(124)'
 'Fatal(49)' 'Fatal(154)' 'Fatal(96)' 'Fatal(114)' 'Fatal(199)'
 'Fatal(89)' 'Fatal(57)' 'Fatal' nan 'Minor' 'Serious']
```

[12]:
```
# Rule 1: If "Fatal(1)", set Total.Fatal.Injuries=1, others=0
fatal_mask = Aviation_data['Injury.Severity'].str.contains('Fatal', na=False)
Aviation_data.loc[fatal_mask, 'Total.Fatal.Injuries'] = 1.0
Aviation_data.loc[fatal_mask, ['Total.Serious.Injuries', 'Total.Minor.
 ↪Injuries']] = 0.0

# Rule 2: If "Non-Fatal", set Total.Minor.Injuries=1, others=0
non_fatal_mask = Aviation_data['Injury.Severity'].str.contains('Non-Fatal',␣
 ↪na=False)
Aviation_data.loc[non_fatal_mask, 'Total.Minor.Injuries'] = 1.0
```

```
Aviation_data.loc[non_fatal_mask, ['Total.Fatal.Injuries', 'Total.Serious.
 ↪Injuries']] = 0.0
```

[13]:
```
missing_severity = Aviation_data['Injury.Severity'].isna() #checking for␣
 ↪missing values
has_minor_injuries = (Aviation_data['Total.Minor.Injuries'] > 0)
Aviation_data.loc[missing_severity & has_minor_injuries, 'Injury.Severity'] =␣
 ↪'Non-Fatal(1)'
```

[14]:
```
# Check Fatal cases
print(Aviation_data[fatal_mask][['Injury.Severity', 'Total.Fatal.Injuries',␣
 ↪'Total.Serious.Injuries']].head())

# Check Non-Fatal cases
print(Aviation_data[non_fatal_mask][['Injury.Severity', 'Total.Minor.Injuries',␣
 ↪'Total.Serious.Injuries']].head())
```

```
  Injury.Severity  Total.Fatal.Injuries  Total.Serious.Injuries
0        Fatal(2)                   1.0                     0.0
1        Fatal(4)                   1.0                     0.0
2        Fatal(3)                   1.0                     0.0
3        Fatal(2)                   1.0                     0.0
4        Fatal(1)                   1.0                     0.0
   Injury.Severity  Total.Minor.Injuries  Total.Serious.Injuries
5        Non-Fatal                   1.0                     0.0
7        Non-Fatal                   1.0                     0.0
8        Non-Fatal                   1.0                     0.0
9        Non-Fatal                   1.0                     0.0
10       Non-Fatal                   1.0                     0.0
```

We have now resolved the conflict between the severity and the various injury count columns, but we still haven't dealt with the missing values within the columns. Let's do so.

[15]:
```
# Rule: If "Fatal(1)", fill NaN in Total.Fatal.Injuries with 1, others with 0
Aviation_data.loc[Aviation_data['Injury.Severity'].str.contains('Fatal',␣
 ↪na=False), 'Total.Fatal.Injuries'] = (
    Aviation_data.loc[Aviation_data['Injury.Severity'].str.contains('Fatal',␣
 ↪na=False), 'Total.Fatal.Injuries'].fillna(1)
)
Aviation_data.loc[Aviation_data['Injury.Severity'].str.contains('Fatal',␣
 ↪na=False), ['Total.Serious.Injuries', 'Total.Minor.Injuries']] = (
    Aviation_data.loc[Aviation_data['Injury.Severity'].str.contains('Fatal',␣
 ↪na=False), ['Total.Serious.Injuries', 'Total.Minor.Injuries']].fillna(0)
)

# Rule: If "Non-Fatal(1)", fill NaN in Total.Minor.Injuries with 1, others with␣
 ↪0
```

```
Aviation_data.loc[Aviation_data['Injury.Severity'].str.contains('Non-Fatal',␣
 ↪na=False), 'Total.Minor.Injuries'] = (
    Aviation_data.loc[Aviation_data['Injury.Severity'].str.
 ↪contains('Non-Fatal', na=False), 'Total.Minor.Injuries'].fillna(1)
)
Aviation_data.loc[Aviation_data['Injury.Severity'].str.contains('Non-Fatal',␣
 ↪na=False), ['Total.Fatal.Injuries', 'Total.Serious.Injuries']] = (
    Aviation_data.loc[Aviation_data['Injury.Severity'].str.
 ↪contains('Non-Fatal', na=False), ['Total.Fatal.Injuries', 'Total.Serious.
 ↪Injuries']].fillna(0)
)

# For rows with missing Injury.Severity, fill all injury NaN with 0 (or median␣
 ↪if preferred)
injury_cols = ['Total.Fatal.Injuries', 'Total.Serious.Injuries', 'Total.Minor.
 ↪Injuries']
no_severity_mask = Aviation_data['Injury.Severity'].isna()
Aviation_data.loc[no_severity_mask, injury_cols] = Aviation_data.
 ↪loc[no_severity_mask, injury_cols].fillna(0)
```

[16]:
```
# Fill NaN with 0 (assuming no uninjured people if data is missing)
Aviation_data['Total.Uninjured'] = Aviation_data['Total.Uninjured'].fillna(0).
 ↪astype(int)
```

1.3.10 3(b). Event date and Publication date:

For these columns, we have to first convert them to date times and then fill in the missing values

[17]:
```
# Convert to datetime (handles most common string formats automatically)
Aviation_data['Event.Date'] = pd.to_datetime(Aviation_data['Event.Date'],␣
 ↪errors='coerce', dayfirst=True)
Aviation_data['Publication.Date'] = pd.to_datetime(Aviation_data['Publication.
 ↪Date'], errors='coerce', dayfirst=True)
print(f"Event.Date missing after conversion: {Aviation_data['Event.Date'].
 ↪isna().mean():.1%}")
print(f"Publication.Date missing after conversion: {Aviation_data['Publication.
 ↪Date'].isna().mean():.1%}")
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_8244\2942137262.py:2: UserWarning:
Parsing dates in %Y-%m-%d format when dayfirst=True was specified. Pass
`dayfirst=False` or specify a format to silence this warning.
  Aviation_data['Event.Date'] = pd.to_datetime(Aviation_data['Event.Date'],
errors='coerce', dayfirst=True)
```

```
Event.Date missing after conversion: 1.6%
Publication.Date missing after conversion: 18.5%
```

Let us now handle them individually

For Event.Date (1.7% missing):

[18]:
```
# Fill with a logical estimate (e.g., publication date minus median delay)
if not Aviation_data['Publication.Date'].isna().all():
    median_delay = (Aviation_data['Publication.Date'] - Aviation_data['Event.
 ↪Date']).median()
    Aviation_data['Event.Date'] = Aviation_data['Event.Date'].
 ↪fillna(Aviation_data['Publication.Date'] - median_delay)
```

For Publication.Date (19% missing):

[19]:
```
# Option 1: Fill with event date plus median publication delay (if Event.Date␣
 ↪exists)
median_delay = (Aviation_data['Publication.Date'] - Aviation_data['Event.
 ↪Date']).median()
Aviation_data['Publication.Date'] = Aviation_data['Publication.Date'].
 ↪fillna(Aviation_data['Event.Date'] + median_delay)
```

[20]:
```
# Check if any publication dates predate event dates
invalid = Aviation_data[Aviation_data['Publication.Date'] <␣
 ↪Aviation_data['Event.Date']]
print(f"{len(invalid)} illogical records (published before event)")

# Optionally correct these cases
Aviation_data.loc[Aviation_data['Publication.Date'] < Aviation_data['Event.
 ↪Date'], 'Publication.Date'] = Aviation_data['Event.Date'] + pd.
 ↪Timedelta(days=7)
```

```
2 illogical records (published before event)
```

Great! we have dealt with two more columns! We can now deal with the remaining columns.

4. Remaining Columns

4(a). Categorical Columns (Low Missingness < 10%):

[21]:
```
categorical_cols = [
    'Engine.Type', 'Report.Status', 'Purpose.of.flight',
    'Weather.Condition', 'Aircraft.damage', 'Country',
     'Model', 'Make', 'Location'
]

# Fill with mode (most frequent category) per column
for col in categorical_cols:
    Aviation_data[col] = Aviation_data[col].fillna(Aviation_data[col].mode()[0])

# For binary categories (e.g., 'Amateur.Built'), consider:
Aviation_data['Amateur.Built'] = Aviation_data['Amateur.Built'].
 ↪fillna('Unknown')
```

4(b). Numeric Columns:

[22]:
```
# Fill with median (robust to outliers)
Aviation_data['Number.of.Engines'] = Aviation_data['Number.of.Engines'].
 ↪fillna(Aviation_data['Number.of.Engines'].median())
```

4(c). Identifier Columns:

[23]:
```
# Option 1: Fill with "UNKNOWN" (if IDs are critical for tracking)
Aviation_data['Registration.Number'] = Aviation_data['Registration.Number'].
 ↪fillna('UNKNOWN_REG')
Aviation_data['Accident.Number'] = Aviation_data['Accident.Number'].
 ↪fillna('UNKNOWN_ACC')
Aviation_data['Event.Id'] = Aviation_data['Event.Id'].fillna('UNKNOWN_EVT')
```

Now we have dealt with the missing values effectively. We can now get to the second phase of our project.

1.4 Data Analysis

Let's take another look at our data now that we have cleaned it and done some data imputation.

[24]:
```
Aviation_data.head(20)
```

[24]:

	Event.Id	Investigation.Type	Accident.Number	Event.Date	\
0	20001218X45444	Accident	SEA87LA080	1948-10-24	
1	20001218X45447	Accident	LAX94LA336	1962-07-19	
2	20061025X01555	Accident	NYC07LA005	1974-08-30	
3	20001218X45448	Accident	LAX96LA321	1977-06-19	
4	20041105X01764	Accident	CHI79FA064	1979-08-02	
5	20170710X52551	Accident	NYC79AA106	1979-09-17	
6	20001218X45446	Accident	CHI81LA106	1981-08-01	
7	20020909X01562	Accident	SEA82DA022	1982-01-01	
8	20020909X01561	Accident	NYC82DA015	1982-01-01	
9	20020909X01560	Accident	MIA82DA029	1982-01-01	
10	20020909X01559	Accident	FTW82DA034	1982-01-01	
11	20020909X01558	Accident	ATL82DKJ10	1982-01-01	
12	20020917X02148	Accident	FTW82FRJ07	1982-01-02	
13	20020917X02134	Accident	FTW82FRA14	1982-01-02	
14	20020917X02119	Accident	FTW82FPJ10	1982-01-02	
15	20020917X02117	Accident	FTW82FPG08	1982-01-02	
16	20020917X01962	Accident	DEN82DTM08	1982-01-02	
17	20020917X01656	Accident	ANC82FAG14	1982-01-02	
18	20020917X02481	Accident	NYC82DA016	1982-01-02	
19	20020917X02339	Accident	MIA82DA028	1982-01-02	

	Location	Country	Airport.Code	Airport.Name	\
0	MOOSE CREEK, ID	United States	NONE	Private	
1	BRIDGEPORT, CA	United States	NONE	Private	
2	Saltville, VA	United States	NONE	Private	

```
3         EUREKA, CA  United States         NONE             Private
4          Canton, OH  United States         NONE             Private
5          BOSTON, MA  United States         NONE             Private
6          COTTON, MN  United States         NONE             Private
7        PULLMAN, WA   United States         NONE  BLACKBURN AG STRIP
8   EAST HANOVER, NJ   United States          N58            HANOVER
9   JACKSONVILLE, FL   United States          JAX   JACKSONVILLE INTL
10         HOBBS, NM   United States         NONE             Private
11      TUSKEGEE, AL   United States         NONE           TUSKEGEE
12         HOMER, LA   United States         NONE             Private
13        HEARNE, TX   United States          T72    HEARNE MUNICIPAL
14     CHICKASHA, OK   United States         NONE             Private
15   LITTLE ROCK, AR   United States         NONE             Private
16        MIDWAY, UT   United States         NONE         FIELD RANCH
17       SKWENTA, AK   United States         NONE             Private
18       GALETON, PA   United States          5G6      CHERRY SPRINGS
19         MIAMI, FL   United States         NONE             Private

   Injury.Severity Aircraft.damage  …    Engine.Type Purpose.of.flight  \
0         Fatal(2)       Destroyed  …  Reciprocating          Personal
1         Fatal(4)       Destroyed  …  Reciprocating          Personal
2         Fatal(3)       Destroyed  …  Reciprocating          Personal
3         Fatal(2)       Destroyed  …  Reciprocating          Personal
4         Fatal(1)       Destroyed  …  Reciprocating          Personal
5        Non-Fatal     Substantial  …      Turbo Fan          Personal
6         Fatal(4)       Destroyed  …  Reciprocating          Personal
7        Non-Fatal     Substantial  …  Reciprocating          Personal
8        Non-Fatal     Substantial  …  Reciprocating          Business
9        Non-Fatal     Substantial  …  Reciprocating          Personal
10       Non-Fatal     Substantial  …  Reciprocating          Personal
11       Non-Fatal     Substantial  …  Reciprocating          Personal
12       Non-Fatal       Destroyed  …  Reciprocating          Personal
13        Fatal(1)       Destroyed  …  Reciprocating          Personal
14        Fatal(1)       Destroyed  …  Reciprocating          Personal
15        Fatal(2)       Destroyed  …  Reciprocating          Personal
16       Non-Fatal       Destroyed  …  Reciprocating          Personal
17        Fatal(3)       Destroyed  …  Reciprocating          Personal
18       Non-Fatal     Substantial  …  Reciprocating          Personal
19       Non-Fatal     Substantial  …  Reciprocating          Personal

   Total.Fatal.Injuries Total.Serious.Injuries Total.Minor.Injuries  \
0                   1.0                    0.0                   0.0
1                   1.0                    0.0                   0.0
2                   1.0                    0.0                   0.0
3                   1.0                    0.0                   0.0
4                   1.0                    0.0                   0.0
5                   0.0                    0.0                   1.0
```

6	1.0	0.0	0.0
7	0.0	0.0	1.0
8	0.0	0.0	1.0
9	0.0	0.0	1.0
10	0.0	0.0	1.0
11	0.0	0.0	1.0
12	0.0	0.0	1.0
13	1.0	0.0	0.0
14	1.0	0.0	0.0
15	1.0	0.0	0.0
16	0.0	0.0	1.0
17	1.0	0.0	0.0
18	0.0	0.0	1.0
19	0.0	0.0	1.0

	Total.Uninjured	Weather.Condition	Broad.phase.of.flight	Report.Status	\
0	0	UNK	Cruise	Probable Cause	
1	0	UNK	Unknown	Probable Cause	
2	0	IMC	Cruise	Probable Cause	
3	0	IMC	Cruise	Probable Cause	
4	0	VMC	Approach	Probable Cause	
5	44	VMC	Climb	Probable Cause	
6	0	IMC	Unknown	Probable Cause	
7	2	VMC	Takeoff	Probable Cause	
8	2	IMC	Landing	Probable Cause	
9	0	IMC	Cruise	Probable Cause	
10	1	VMC	Approach	Probable Cause	
11	1	VMC	Landing	Probable Cause	
12	0	IMC	Cruise	Probable Cause	
13	0	IMC	Takeoff	Probable Cause	
14	0	IMC	Cruise	Probable Cause	
15	0	IMC	Cruise	Probable Cause	
16	1	IMC	Taxi	Probable Cause	
17	0	VMC	Unknown	Probable Cause	
18	1	VMC	Taxi	Probable Cause	
19	2	VMC	Cruise	Probable Cause	

	Publication.Date
0	1950-02-06
1	1996-09-19
2	2007-02-26
3	2000-09-12
4	1980-04-16
5	2017-09-19
6	2001-11-06
7	1982-01-01
8	1982-01-01

```
9         1982-01-01
10        1982-01-01
11        1982-01-01
12        1983-01-02
13        1983-01-02
14        1983-01-02
15        1983-01-02
16        1983-01-02
17        1983-01-02
18        1983-01-02
19        1983-01-02

[20 rows x 26 columns]
```

For this stage of our project, we will need to address some key areas in relation to the problem being solved. The goal is to analyze the aviation data and identify:

- Which aircraft models have the best safety records
- What factors contribute to aviation accidents
- Recommendations for safest aircraft to purchase

If we take a good look at our data, we can see that there are a number of metrics that can be considered: 1. Injury Severity: Fatal vs. Non-Fatal incidents

2. Aircraft Damage: Destroyed vs. Substantial vs. Minor
3. Aircraft Make/Model: Which have the most incidents
4. Phase of Flight: When accidents occur
5. Weather Conditions: Impact on accidents
6. Purpose of Flight: Commercial vs. Private operations
7. Number of Engines: Safety implications

Having identified this as our key metrics for analysis, we can now go ahead and perform some aggregation techniques on the data and try to come up with optimal aggregates for our vizualizations. But before that, let us make our columns more presentable.

[25]:
```
Aviation_data.columns
```

[25]:
```
Index(['Event.Id', 'Investigation.Type', 'Accident.Number', 'Event.Date',
       'Location', 'Country', 'Airport.Code', 'Airport.Name',
       'Injury.Severity', 'Aircraft.damage', 'Aircraft.Category',
       'Registration.Number', 'Make', 'Model', 'Amateur.Built',
       'Number.of.Engines', 'Engine.Type', 'Purpose.of.flight',
       'Total.Fatal.Injuries', 'Total.Serious.Injuries',
       'Total.Minor.Injuries', 'Total.Uninjured', 'Weather.Condition',
       'Broad.phase.of.flight', 'Report.Status', 'Publication.Date'],
      dtype='object')
```

[26]:
```
Aviation_data.columns = (
    Aviation_data.columns
    .str.strip()                          # remove leading/trailing spaces
    .str.lower()                          # lowercase all
    .str.replace('[^a-z0-9]+', '_', regex=True)  # replace non-alphanumeric␣
 ↪with underscore
    .str.strip('_')                      # remove leading/trailing underscores
)

print(Aviation_data.columns)
```

```
Index(['event_id', 'investigation_type', 'accident_number', 'event_date',
       'location', 'country', 'airport_code', 'airport_name',
       'injury_severity', 'aircraft_damage', 'aircraft_category',
       'registration_number', 'make', 'model', 'amateur_built',
       'number_of_engines', 'engine_type', 'purpose_of_flight',
       'total_fatal_injuries', 'total_serious_injuries',
       'total_minor_injuries', 'total_uninjured', 'weather_condition',
       'broad_phase_of_flight', 'report_status', 'publication_date'],
      dtype='object')
```

Great! Now we can move on to our data aggregation.

1.4.1 Data Aggregation

(a). By Aircraft Make

[27]:
```
# Calculate fatality rate by aircraft model
safety_by_model = Aviation_data.groupby(['make', 'model']).agg(
    total_incidents=('event_id', 'count'),
    fatal_incidents=('total_fatal_injuries', lambda x: (x > 0).sum()),
    destruction_rate=('aircraft_damage', lambda x: (x == 'Destroyed').mean())
).reset_index()

safety_by_model['fatality_rate'] = safety_by_model['fatal_incidents'] /␣
 ↪safety_by_model['total_incidents']
```

[28]:
```
safety_by_model.head()
```

[28]:

	make	model	total_incidents	fatal_incidents	destruction_rate	fatality_rate
0	107.5 Flying Corporation	One Design DR 107	1	1	1.0	1.0
1	1200	G103	1			
2	177MF LLC	PITTS MODEL 12	1			
3	1977 Colfer-chan	STEEN SKYBOLT	1			
4	1st Ftr Gp	FOCKE-WULF 190	1			

1	0	0.0	0.0
2	0	0.0	0.0
3	0	0.0	0.0
4	1	1.0	1.0

Here, we have performed the following: - Calculated fatality rates by aircraft make/model

- Created a "Severity Index" combining injuries and damage levels
- Identified which aircraft models have:
  - Highest/Lowest fatality rates
  - Most "Destroyed" incidents vs "Minor" damage

(b). Operational Risk Factors `Columns to use: weather_condition, broad_phase_of_flight, purpose_of_flight`

[29]:
```
# Risk by flight phase
phase_risk = Aviation_data.groupby('broad_phase_of_flight').agg(
    incident_count=('event_id', 'count'),
    fatality_count=('total_fatal_injuries', 'sum')
).sort_values('fatality_count', ascending=False)

# Weather impact analysis
weather_impact = pd.crosstab(
    Aviation_data['weather_condition'],
    Aviation_data['aircraft_damage'],
    normalize='index'
)
```

[30]:
```
phase_risk.head()
```

[30]:
```
                       incident_count  fatality_count
broad_phase_of_flight
Landing                         44052          6500.0
Maneuvering                      8144          3183.0
Cruise                          10269          2786.0
Takeoff                         12493          1824.0
Approach                         6546          1599.0
```

[31]:
```
weather_impact.head()
```

[31]:
```
aircraft_damage    Destroyed     Minor  Substantial   Unknown
weather_condition
IMC                 0.559070  0.027945     0.412985  0.000000
UNK                 0.540888  0.080607     0.378505  0.000000
Unk                 0.278626  0.106870     0.599237  0.015267
VMC                 0.177121  0.030521     0.790977  0.001381
```

Here, we have analyzed accident distribution by:

- Flight phase (Takeoff, Cruise, Landing, etc.)
- Weather conditions (VMC, IMC, UNK)
- Flight purpose (Commercial, Personal, Training)

(c). Aircraft Technical Factors `Columns to use: number_of_engines, engine_type, amateur_built`

[32]:
```
engine_safety = Aviation_data.groupby(['number_of_engines', 'engine_type']).agg(
    incidents=('event_id', 'count'),
    fatality_rate=('total_fatal_injuries', lambda x: x.sum()/len(x))
).reset_index()
```

[33]:
```
engine_safety.head()
```

[33]:

	number_of_engines	engine_type	incidents	fatality_rate
0	0.0	NONE	2	0.000000
1	0.0	Reciprocating	544	0.128676
2	0.0	Unknown	680	0.139706
3	1.0	Electric	6	0.333333
4	1.0	Geared Turbofan	1	0.000000

Here, we have compared safety between:

- Single vs multi-engine aircraft
- Reciprocating vs Turbo engines
- Factory-built vs amateur-built

(d). Temporal Analysis `Columns to use: event_date`

[34]:
```
Aviation_data['year'] = pd.to_datetime(Aviation_data['event_date']).dt.year
yearly_trends = Aviation_data.groupby('year').agg(
    incident_count=('event_id', 'count'),
    fatality_rate=('total_fatal_injuries', lambda x: (x > 0).mean())
)
```

[35]:
```
yearly_trends.head()
```

[35]:

year	incident_count	fatality_rate
1948.0	1	1.0
1962.0	1	1.0
1974.0	1	1.0
1977.0	1	1.0
1979.0	2	0.5

Here we have: - Analyzed trends over time (yearly/monthly)

- Identified seasonal patterns in accidents

(e). Geographic Analysis `Columns to use: country, location`

[36]:
```
geo_risk = Aviation_data.groupby('country').agg(
    incident_count=('event_id', 'count'),
    fatality_rate=('total_fatal_injuries', lambda x: x.sum()/len(x))
).sort_values('fatality_rate', ascending=False)
```

[37]:
```
geo_risk.head()
```

[37]:

country	incident_count	fatality_rate
Liberia	1	1.0
Belarus	1	1.0
Bosnia And Herzegovina	1	1.0
Cambodia	1	1.0
St Lucia	1	1.0

Here, we have - Identified high-risk regions

- Compared safety records by country/state

Now that we have been able to aggregate into key metrics that will serve well for the analysis, we can now proceed to vizualize these key metrics using various plots.

1.5 Data Vizualization

We can now proceed to the next stage of our project which is the vizualization. We can use the various aggregates to plot various kinds of graphs which will make it easier to decide on which aircraft to purchase for the expansion venture.

Obviously, for this segment we have to import the various python in-built libraries that are suited to this kind of work by their standard aliases.

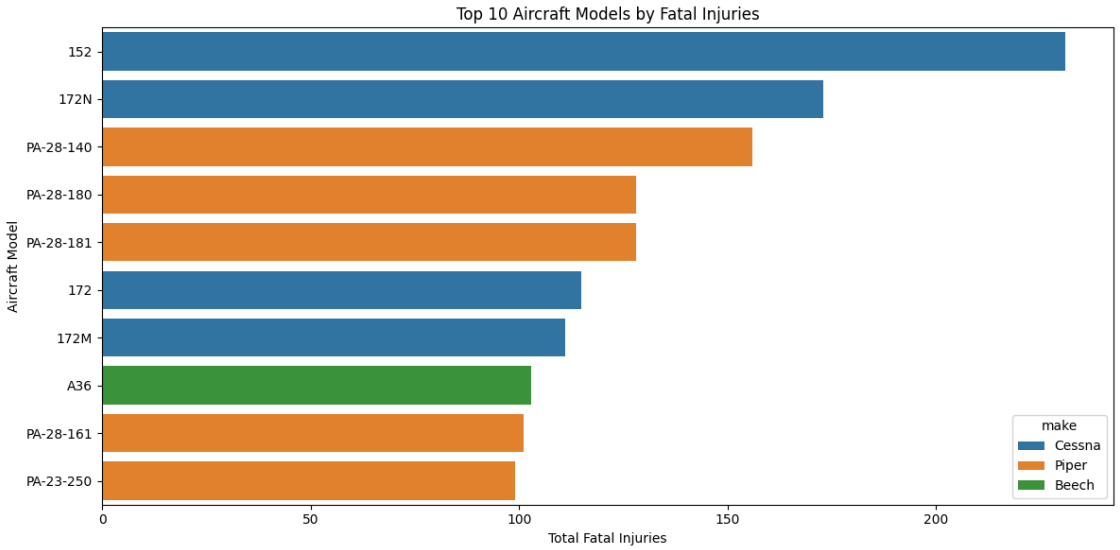
[38]:
```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Now we can proceed to plot by the various aggregates.

1.5.1 (a). Aircraft Make

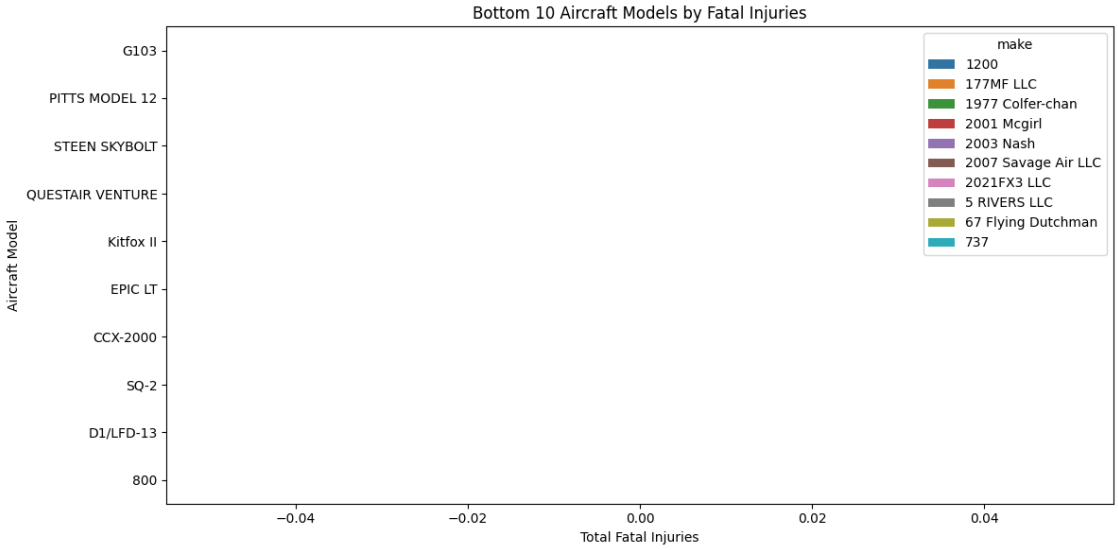
[39]:
```
# Top 10 aircraft by fatal incidents
top_fatal = safety_by_model.groupby(['make', 'model'])['fatal_incidents'].sum().
 ↪nlargest(10).reset_index()
plt.figure(figsize=(12,6))
sns.barplot(data=top_fatal, x='fatal_incidents', y='model', hue='make',␣
 ↪dodge=False)
plt.title('Top 10 Aircraft Models by Fatal Injuries')
plt.xlabel('Total Fatal Injuries')
```

```
plt.ylabel('Aircraft Model')
plt.tight_layout()
plt.show()
```



This graph clearly shows us that the ***Cessna 152*** is clearly not a good aircraft to invest in as it has led to the most fatalities. Moreover the ***Cessna*** and the ***Piper*** make are very unreliable models for safety hence not a good idea to invest in them.

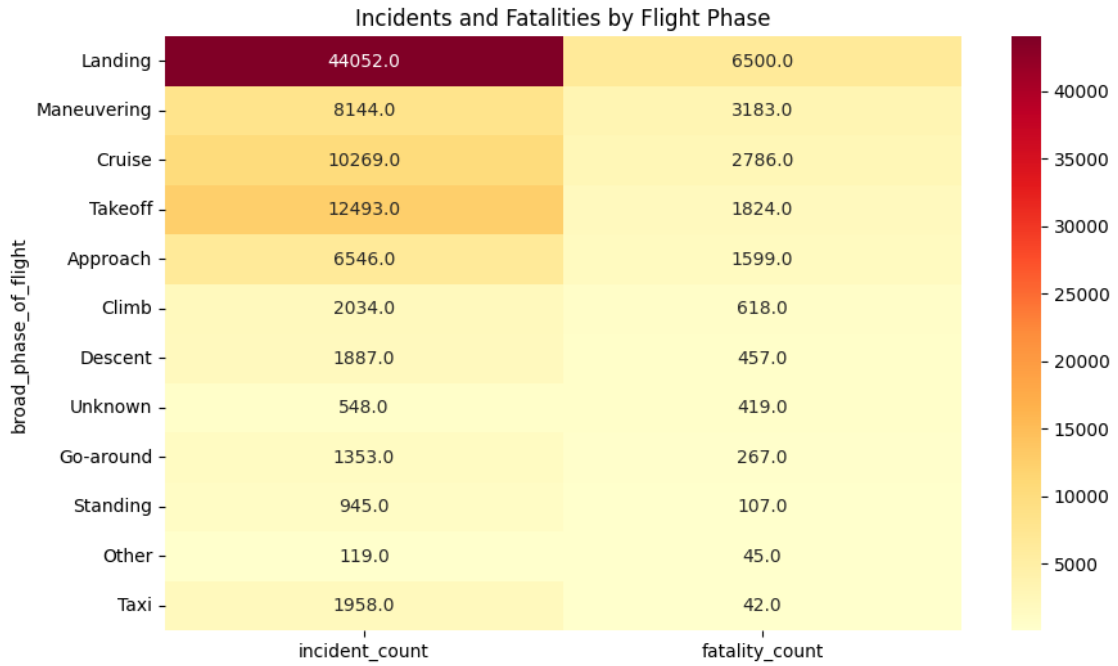
[40]:
```
# Top 10 aircraft by fatal incidents
least_fatal = safety_by_model.groupby(['make', 'model'])['fatal_incidents'].
 ↪sum().nsmallest(10).reset_index()
plt.figure(figsize=(12,6))
sns.barplot(data=least_fatal, x='fatal_incidents', y='model', hue='make',␣
 ↪dodge=False)
plt.title('Bottom 10 Aircraft Models by Fatal Injuries')
plt.xlabel('Total Fatal Injuries')
plt.ylabel('Aircraft Model')
plt.tight_layout()
plt.show()
```



Conversely, this shows that the ***1200*** and ***177MF LLC*** are among the most reliable models of air travel for purchase.

1.5.2 (b) Operational Risk Factors

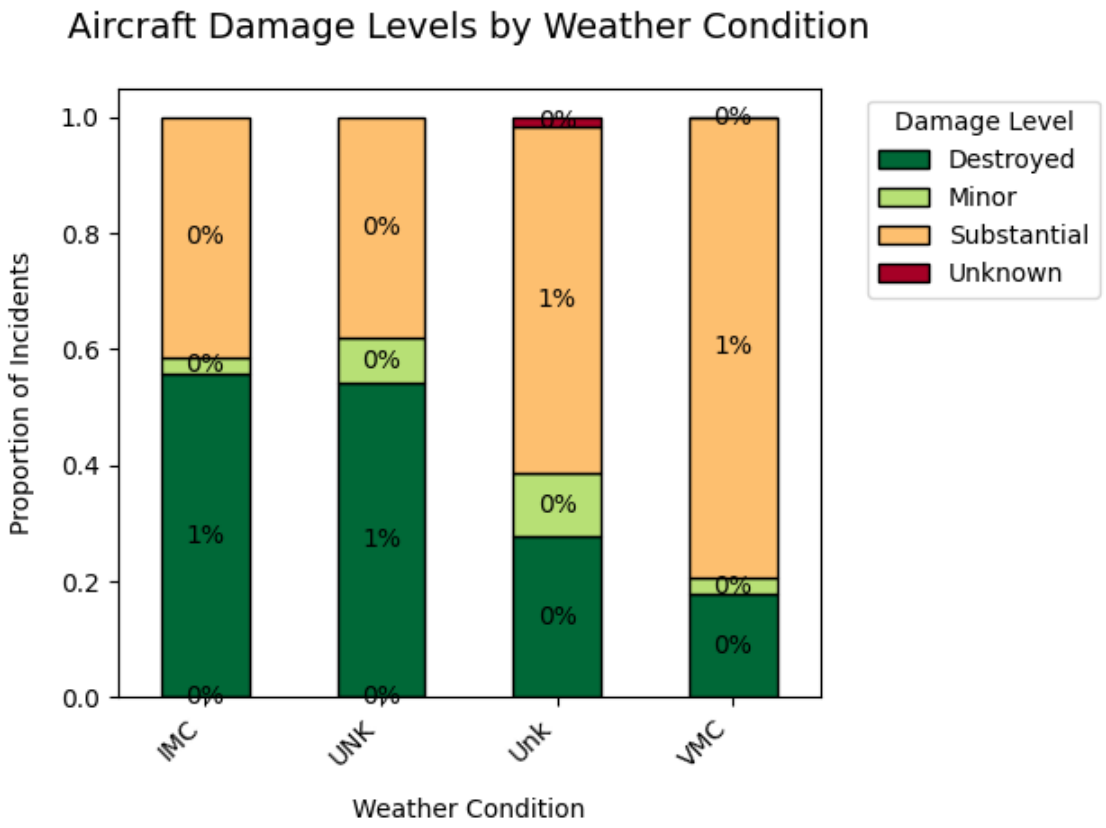
[41]:
```
plt.figure(figsize=(10,6))
sns.heatmap(phase_risk, annot=True, fmt='.1f', cmap='YlOrRd')
plt.title('Incidents and Fatalities by Flight Phase')
plt.show()
```



The heatmap above clearly shows that most incidents occur at the landing phase of flying with more than 44000 incidents being recorded leading to more than 6500 fatalities.

[42]:
```
# Sort by most dangerous weather condition first
weather_impact = weather_impact.sort_values('Destroyed', ascending=False)

# Create stacked bar chart
plt.figure(figsize=(12, 6))
ax = weather_impact.plot(kind='bar',
                         stacked=True,
                         colormap='RdYlGn_r',  # Red-Yellow-Green (reversed)
                         edgecolor='black')

# Formatting
plt.title('Aircraft Damage Levels by Weather Condition', pad=20, fontsize=14)
plt.xlabel('Weather Condition', labelpad=10)
plt.ylabel('Proportion of Incidents', labelpad=10)
plt.xticks(rotation=45, ha='right')
plt.legend(title='Damage Level', bbox_to_anchor=(1.05, 1), loc='upper left')
# Add percentage labels
for container in ax.containers:
    ax.bar_label(container,
                 label_type='center',
                 fmt='%.0f%%',  # Show as percentages
                 color='black',
                 fontsize=10)

plt.tight_layout()
plt.show()
```

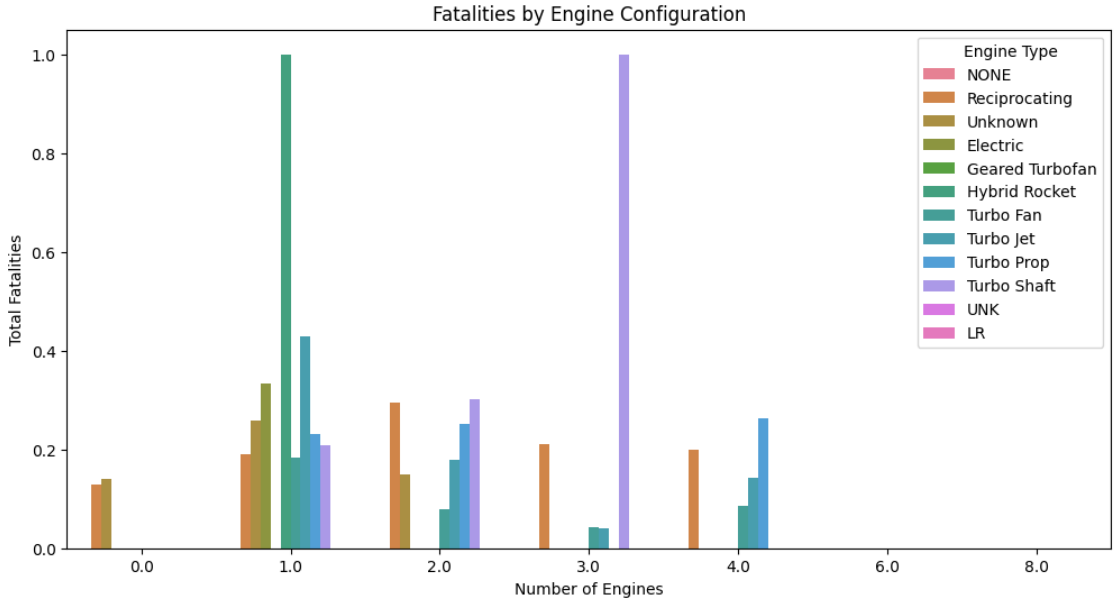
```
<Figure size 1200x600 with 0 Axes>
```



This stacked bar chart shows us first of all that ***IMC(Instrument Meteorological Conditions)*** contribute most to aircraft incidents and also contribute to great damage of aircrafts due to poor visibility.

1.5.3 (c). Aircraft Technical Factors

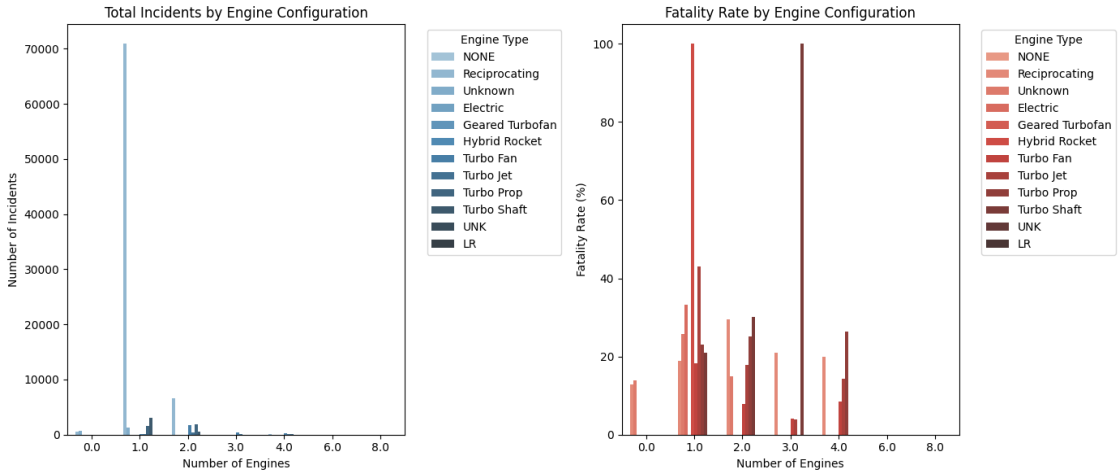
[43]:
```
plt.figure(figsize=(12,6))
sns.barplot(data=engine_safety, x='number_of_engines', y='fatality_rate',␣
↪hue='engine_type')
plt.title('Fatalities by Engine Configuration')
plt.xlabel('Number of Engines')
plt.ylabel('Total Fatalities')
plt.legend(title='Engine Type')
plt.show()
```



The bar graph abobe shows that planes with 1 jet engine of a ***Hybrid Rocket*** type and planes with 3 jet engines of a ***Turbo Shaft*** type have been involved in the most incidents hence making them the ones to avoid the most.

[44]:
```
# Convert to percentages for readability
engine_safety['fatality_pct'] = engine_safety['fatality_rate'] * 100

# Create figure with two subplots
plt.figure(figsize=(14, 6))

# Subplot 1: Incident Count by Engine Configuration
plt.subplot(1, 2, 1)
sns.barplot(data=engine_safety,
            x='number_of_engines',
            y='incidents',
            hue='engine_type',
            palette='Blues_d')
plt.title('Total Incidents by Engine Configuration')
plt.xlabel('Number of Engines')
plt.ylabel('Number of Incidents')
plt.legend(title='Engine Type', bbox_to_anchor=(1.05, 1), loc='upper left')
# Subplot 2: Fatality Rate by Engine Configuration
plt.subplot(1, 2, 2)
sns.barplot(data=engine_safety,
            x='number_of_engines',
            y='fatality_pct',
            hue='engine_type',
```

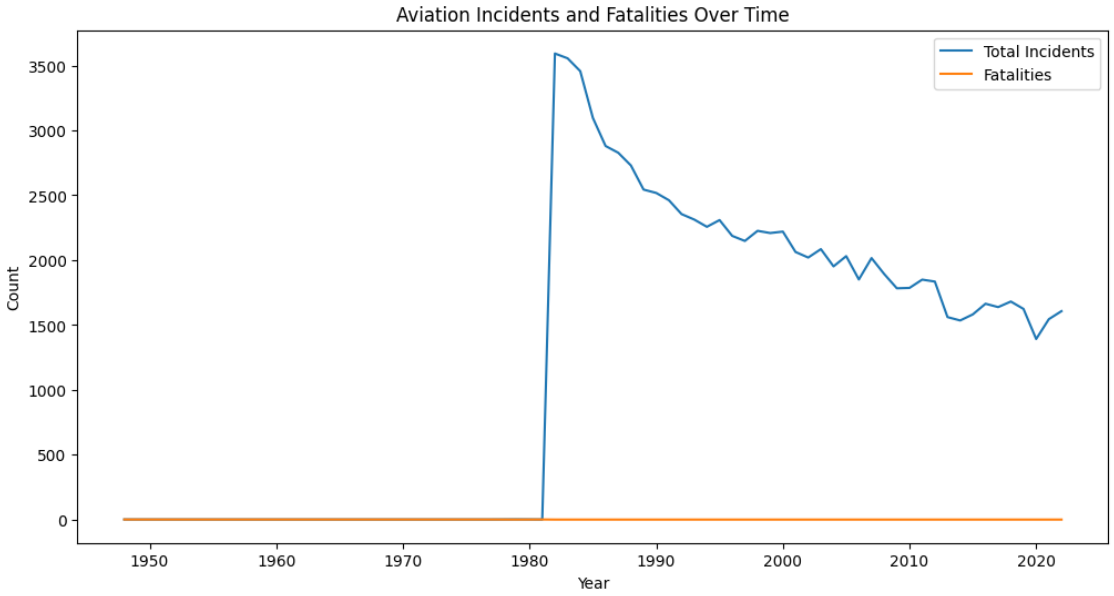
```
            palette='Reds_d')
plt.title('Fatality Rate by Engine Configuration')
plt.xlabel('Number of Engines')
plt.ylabel('Fatality Rate (%)')
plt.legend(title='Engine Type', bbox_to_anchor=(1.05, 1), loc='upper left')

plt.tight_layout()
plt.show()
```



Same case as above.

1.5.4 (d). Temporal Analysis

[45]:
```
plt.figure(figsize=(12,6))
sns.lineplot(data=yearly_trends, x='year', y='incident_count', label='Total␣
 ↪Incidents')
sns.lineplot(data=yearly_trends, x='year', y='fatality_rate',␣
 ↪label='Fatalities')
plt.title('Aviation Incidents and Fatalities Over Time')
plt.xlabel('Year')
plt.ylabel('Count')
plt.legend()
plt.show()
```



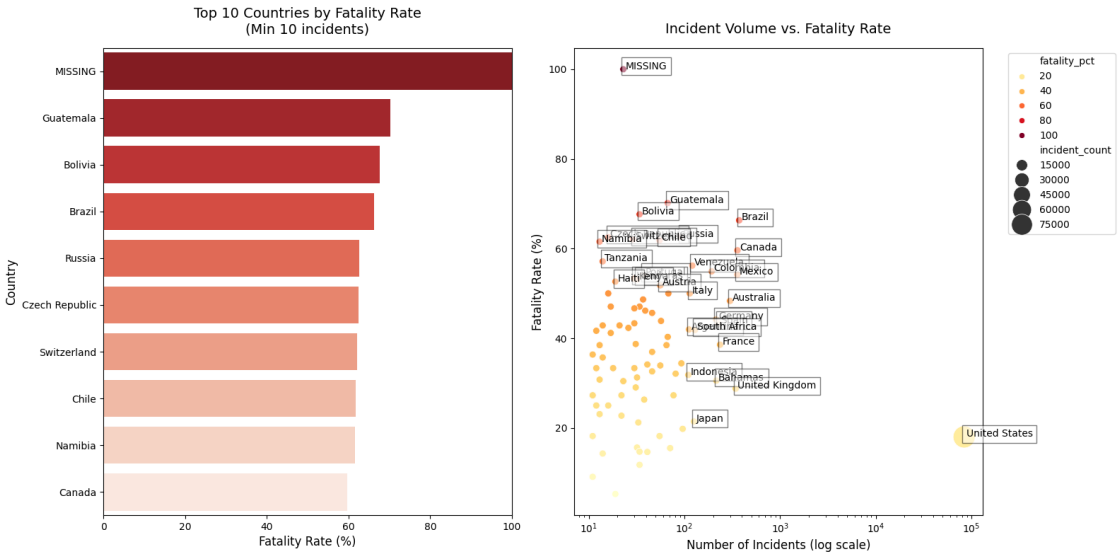
The yearly trend can be seen to be generally decreasing from 1980 showing an improvent in aviation safety with fatalities being low.

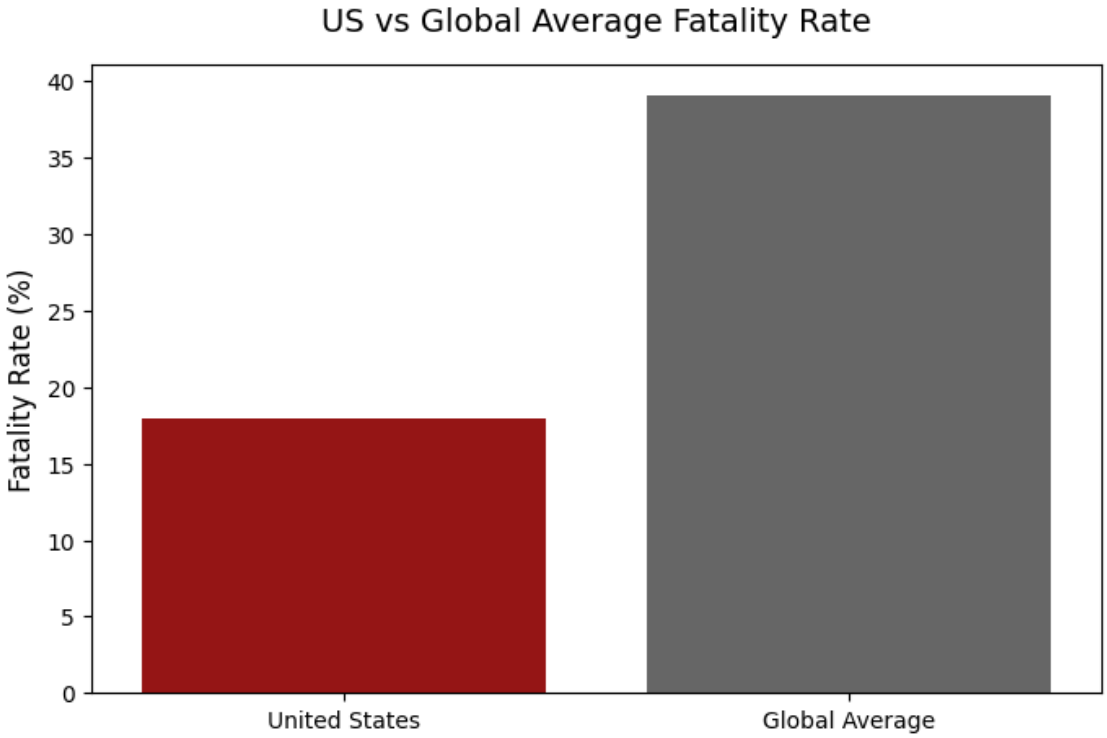
1.5.5 (e). Geographical Analysis

[46]:
```
# Convert fatality rate to percentage
geo_risk['fatality_pct'] = geo_risk['fatality_rate'] * 100

# Filter countries with significant incident counts (e.g., > 10 incidents)
significant_countries = geo_risk[geo_risk['incident_count'] > 10]

# Create figure with 2 subplots (removed Plotly for compatibility)
plt.figure(figsize=(16, 8))

# 1. Top 10 Highest Fatality Rate Countries (corrected)
plt.subplot(1, 2, 1)
top_fatal = significant_countries.head(10).reset_index()
sns.barplot(data=top_fatal,
            x='fatality_pct',
            y='country',
            hue='country',  # Added to address warning
            palette='Reds_r',
            legend=False)  # Disable legend for cleaner look
plt.title('Top 10 Countries by Fatality Rate\n(Min 10 incidents)', pad=15,␣
 ↪fontsize=14)
plt.xlabel('Fatality Rate (%)', fontsize=12)
plt.ylabel('Country', fontsize=12)
```

```
plt.xlim(0, 100)

# 2. Incident Volume vs. Fatality Rate (improved)
plt.subplot(1, 2, 2)
scatter = sns.scatterplot(
    data=significant_countries.reset_index(),
    x='incident_count',
    y='fatality_pct',
    size='incident_count',
    sizes=(50, 500),
    hue='fatality_pct',
    palette='YlOrRd',
    legend='brief'
)

# Add country labels for outliers
for line in range(significant_countries.shape[0]):
    row = significant_countries.iloc[line]
    if row['incident_count'] > 100 or row['fatality_pct'] > 50:  # Label␣
 ↪thresholds
        plt.text(
            x=row['incident_count']*1.05,  # Offset slightly
            y=row['fatality_pct'],
            s=row.name,
            fontdict=dict(color='black', size=10),
            bbox=dict(facecolor='white', alpha=0.5)
        )

plt.title('Incident Volume vs. Fatality Rate', pad=15, fontsize=14)
plt.xlabel('Number of Incidents (log scale)', fontsize=12)
plt.ylabel('Fatality Rate (%)', fontsize=12)
plt.xscale('log')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')

plt.tight_layout()
plt.show()

# US Comparison (if data exists)
if 'United States' in geo_risk.index:
    plt.figure(figsize=(8, 5))
    sns.barplot(
        x=['United States', 'Global Average'],
        y=[geo_risk.loc['United States','fatality_pct'],
           geo_risk['fatality_pct'].mean()],
        hue=['United States', 'Global Average'],  # Added to address warning
        palette=['#aa0000', '#666666'],
        legend=False
```

```
    )
    plt.title('US vs Global Average Fatality Rate', pad=15, fontsize=14)
    plt.ylabel('Fatality Rate (%)', fontsize=12)
    plt.xlabel('')
    plt.show()
```





Finally! We have been able to complete our vizualizations and we can now draw conclusions and recommendations based on this outpput

---

1.6 Conclusion

Based on our findings, we have concluded the following: - Aviation is a viable opportunity to venture into. As seen in our graphs, most incidents which occur do not direclty result in mass fatalities. Moreover, most incidents have not resulted directly in massive damage of aircrafts meaning no substantial losses will be incurred.

- Even though most incidents were brought about by IMC weather conditions, we cannot rule out VMC errors brought about due to pilot errors.
- The USA has shown that their fatality rate when compared to the global average is significantly high highlighting that the key safety measures and endeavors that have been undertaken across the country to ensure air safety have fallen short of standards.
- Cessna and Piper aircraft models have shown incredible unreliability when it comes to passenger safety and fatal incidents.
- Yearly trends have shown that air incidents are gradually reducing making it safe to invest in the aviation industry.

1.7 Recommendations

The following are the recommendations to be made: - We recommend that you expand into the aviation industry since it has proven to be a reliable source of transport.

- Conduct thorough background checks in regards to hiring pilots and cabin crew as they can be the difference between gains and losses in this industry.
- We highly recommend investing in the Japan and the United Kingdom as they have shown reliability in the aviation industry in terms of safety.
- Avoid the ***Cessna and Piper*** aircraft models. Moreover avoid ***Hybrid Rocket and Turbo Shaft*** type of engines when purchasing an aircraft model.

---

2 Epilogue

This project has done due research on the topic and has proven that aviation is an investable venture.

3 Further Work

Lets take a look at our cleaned data once more

[49]:
```
Aviation_data
```

[49]:

```
            event_id investigation_type accident_number event_date  \
0      20001218X45444           Accident      SEA87LA080 1948-10-24
1      20001218X45447           Accident      LAX94LA336 1962-07-19
2      20061025X01555           Accident      NYC07LA005 1974-08-30
3      20001218X45448           Accident      LAX96LA321 1977-06-19
4      20041105X01764           Accident      CHI79FA064 1979-08-02
…                 …                  …             …          …
90343  20221227106491           Accident      ERA23LA093 2022-12-26
90344  20221227106494           Accident      ERA23LA095 2022-12-26
90345  20221227106497           Accident      WPR23LA075 2022-12-26
90346  20221227106498           Accident      WPR23LA076 2022-12-26
90347  20221230106513           Accident      ERA23LA097 2022-12-29

               location        country airport_code airport_name  \
0      MOOSE CREEK, ID  United States         NONE      Private
1       BRIDGEPORT, CA  United States         NONE      Private
2        Saltville, VA  United States         NONE      Private
3           EUREKA, CA  United States         NONE      Private
4           Canton, OH  United States         NONE      Private
…                  …             …          …           …
90343    Annapolis, MD  United States         NONE      Private
90344      Hampton, NH  United States         NONE      Private
90345       Payson, AZ  United States          PAN       PAYSON
90346       Morgan, UT  United States         NONE      Private
90347       Athens, GA  United States         NONE      Private

      injury_severity aircraft_damage  … purpose_of_flight  \
0            Fatal(2)       Destroyed  …           Personal
1            Fatal(4)       Destroyed  …           Personal
2            Fatal(3)       Destroyed  …           Personal
3            Fatal(2)       Destroyed  …           Personal
4            Fatal(1)       Destroyed  …           Personal
…                 …               …  …                 …
90343           Minor     Substantial  …           Personal
90344             NaN     Substantial  …           Personal
90345      Non-Fatal     Substantial  …           Personal
90346             NaN     Substantial  …           Personal
90347           Minor     Substantial  …           Personal

      total_fatal_injuries total_serious_injuries total_minor_injuries  \
0                      1.0                   0.0                   0.0
1                      1.0                   0.0                   0.0
2                      1.0                   0.0                   0.0
3                      1.0                   0.0                   0.0
4                      1.0                   0.0                   0.0
```

```
…                    …                    …                    …
90343                  0.0                  1.0                  0.0
90344                  0.0                  0.0                  0.0
90345                  0.0                  0.0                  1.0
90346                  0.0                  0.0                  0.0
90347                  0.0                  1.0                  0.0

      total_uninjured  weather_condition broad_phase_of_flight  \
0                   0                UNK                Cruise
1                   0                UNK               Unknown
2                   0                IMC                Cruise
3                   0                IMC                Cruise
4                   0                VMC              Approach
…                 …                  …                    …
90343               0                VMC               Landing
90344               0                VMC               Landing
90345               1                VMC               Landing
90346               0                VMC               Landing
90347               1                VMC               Landing

        report_status  publication_date    year
0      Probable Cause       1950-02-06  1948.0
1      Probable Cause       1996-09-19  1962.0
2      Probable Cause       2007-02-26  1974.0
3      Probable Cause       2000-09-12  1977.0
4      Probable Cause       1980-04-16  1979.0
…                …               …     …
90343  Probable Cause       2022-12-29  2022.0
90344  Probable Cause       2024-04-09  2022.0
90345  Probable Cause       2022-12-27  2022.0
90346  Probable Cause       2024-04-09  2022.0
90347  Probable Cause       2022-12-30  2022.0

[90348 rows x 27 columns]
```

We can now export it to our local device for further work on Tableau.

[50]:
```
Aviation_data.to_csv('C:/Users/USER/Documents/aviation_cleaned_data.csv',␣
↪index=False)
```

We can go over to tableau and work now on our data.