Risk Evaluation for Aviation Division Expansion

Overview

This analysis evaluates aviation accident data to identify the lowest-risk airplanes for commercial and private enterprises to provide actionable insights for the company's new aviation division to guide aircraft purchasing decisions.

Business Understanding

The company is entering the aviation industry, will be focusing on operating airplanes for commercial and private enterprises. As a new player, it aims to minimize risks—safety (accidents/fatalities). This analysis will leverage historical accident data to identify the safest airplanes for these uses. Insights will guide the head of the new aviation division in making secure, cost-effective aircraft purchasing decisions.

In [13]:

```
# Imports here
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

Data Understanding

The dataset contains aviation accident records, which will be used to identify airplanes with the lowest safety risks, the following steps explore the dataset’s structure, content, and quality to understand its potential for risk analysis.

In [14]:

```
# 1. Load dataset
df = pd.read_csv('Data/AviationData.csv', encoding='latin-1', low_memory=False)

# 2. Preview data
print("************ Head ************")
print(df.head())

print("\n\n************ Sample ************")
print(df.sample())

# 3. Explore Dataset Structure
print("\n\n************ Shape ************")
print(df.shape)

print("\n\n************ Columns ************")
print(df.columns.tolist())

print("\n\n************ Info ************")
print(df.info())

# 4. Missing data overview
print("\n\n===== Missing Data =====")
missing = df.isnull().sum()
print(missing[missing > 0].sort_values(ascending=False))

# 5. Summarize Numeric and Categorical Data
print("\n\n************ Describe (Numerical) ************")
print(df.describe())

print("\n\n************ Describe (Categorical) ************")
print(df.describe(include='object'))

# 6. Inspect categorical columns and their top categories
cat_cols = df.select_dtypes(include='object').columns
print("\n\n************ Categorical Columns ************")
print(cat_cols)

print("\n\n************ Top 5 Categories per Categorical Column ************")
for col in cat_cols:
    print(f"\n{col}:")
    print(df[col].value_counts(dropna=False).head(5))

# 7. Check for duplicates
print("\n\n************ Check for Duplicates ************")
print(f"Total Duplicates: {df.duplicated().sum()}")

# 8. Unique values of important columns to understand domain better
print("\n\n************ Unique Values ************")
print("Injury Severity:", df['Injury.Severity'].unique().tolist())
print("Aircraft Damage:", df['Aircraft.damage'].unique().tolist())
print("Weather Condition:", df['Weather.Condition'].unique().tolist())
print("Broad Phase of Flight:", df['Broad.phase.of.flight'].unique().tolist())
print("Purpose of Flight:", df['Purpose.of.flight'].unique().tolist())
```

```
************ Head ************
         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   

          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   

  Airport.Name  ... Purpose.of.flight Air.carrier Total.Fatal.Injuries  \
0          NaN  ...          Personal         NaN                  2.0   
1          NaN  ...          Personal         NaN                  4.0   
2          NaN  ...          Personal         NaN                  3.0   
3          NaN  ...          Personal         NaN                  2.0   
4          NaN  ...          Personal         NaN                  1.0   

  Total.Serious.Injuries Total.Minor.Injuries Total.Uninjured  \
0                    0.0                  0.0             0.0   
1                    0.0                  0.0             0.0   
2                    NaN                  NaN             NaN   
3                    0.0                  0.0             0.0   
4                    2.0                  NaN             0.0   

  Weather.Condition  Broad.phase.of.flight   Report.Status Publication.Date
0               UNK                  Cruise  Probable Cause              NaN
1               UNK                 Unknown  Probable Cause       19-09-1996
2               IMC                  Cruise  Probable Cause       26-02-2007
3               IMC                  Cruise  Probable Cause       12-09-2000
4               VMC                Approach  Probable Cause       16-04-1980

[5 rows x 31 columns]


************ Sample ************
             Event.Id Investigation.Type Accident.Number  Event.Date  \
23085  20001213X28387           Accident      MIA89LA163  1989-05-30   

               Location        Country Latitude Longitude Airport.Code  \
23085  JACKSONVILLE, FL  United States      NaN       NaN          CRG   

          Airport.Name  ... Purpose.of.flight Air.carrier  \
23085  CRAIG MUNICIPAL  ...          Personal         NaN   

      Total.Fatal.Injuries Total.Serious.Injuries Total.Minor.Injuries  \
23085                  0.0                    0.0                  0.0   

      Total.Uninjured Weather.Condition  Broad.phase.of.flight  \
23085             1.0               VMC               Standing   

        Report.Status Publication.Date
```

```
23085  Probable Cause       22-08-1990

[1 rows x 31 columns]


************ Shape ************
(88889, 31)


************ Columns ************
['Event.Id', 'Investigation.Type', 'Accident.Number', 'Event.Date', 'Location', 'Cou
ntry', 'Latitude', 'Longitude', 'Airport.Code', 'Airport.Name', 'Injury.Severity',
'Aircraft.damage', 'Aircraft.Category', 'Registration.Number', 'Make', 'Model', 'Ama
teur.Built', 'Number.of.Engines', 'Engine.Type', 'FAR.Description', 'Schedule', 'Pur
pose.of.flight', 'Air.carrier', 'Total.Fatal.Injuries', 'Total.Serious.Injuries', 'T
otal.Minor.Injuries', 'Total.Uninjured', 'Weather.Condition', 'Broad.phase.of.fligh
t', 'Report.Status', 'Publication.Date']


************ Info ************
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 88889 entries, 0 to 88888
Data columns (total 31 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Event.Id                88889 non-null  object 
 1   Investigation.Type      88889 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        75118 non-null  object 
```

```
dtypes: float64(5), object(26)
memory usage: 21.0+ MB
None


===== Missing Data =====
Schedule                  76307
Air.carrier               72241
FAR.Description           56866
Aircraft.Category         56602
Longitude                 54516
Latitude                  54507
Airport.Code              38757
Airport.Name              36185
Broad.phase.of.flight     27165
Publication.Date          13771
Total.Serious.Injuries    12510
Total.Minor.Injuries      11933
Total.Fatal.Injuries      11401
Engine.Type                7096
Report.Status              6384
Purpose.of.flight          6192
Number.of.Engines          6084
Total.Uninjured            5912
Weather.Condition          4492
Aircraft.damage            3194
Registration.Number        1382
Injury.Severity            1000
Country                     226
Amateur.Built               102
Model                        92
Make                         63
Location                     52
dtype: int64


************ Describe (Numerical) ************
       Number.of.Engines  Total.Fatal.Injuries  Total.Serious.Injuries  \
count       82805.000000          77488.000000            76379.000000   
mean            1.146585              0.647855                0.279881   
std             0.446510              5.485960                1.544084   
min             0.000000              0.000000                0.000000   
25%             1.000000              0.000000                0.000000   
50%             1.000000              0.000000                0.000000   
75%             1.000000              0.000000                0.000000   
max             8.000000            349.000000              161.000000   

       Total.Minor.Injuries  Total.Uninjured  
count          76956.000000     82977.000000  
mean               0.357061         5.325440  
std                2.235625        27.913634  
min                0.000000         0.000000  
25%                0.000000         0.000000  
50%                0.000000         1.000000  
75%                0.000000         2.000000  
max              380.000000       699.000000  
```

```
************ Describe (Categorical) ************
              Event.Id Investigation.Type Accident.Number  Event.Date  \
count            88889              88889           88889       88889   
unique           87951                  2           88863       14782   
top     20001214X45071           Accident      ERA22LA103  1982-05-16   
freq                 3              85015               2          25   

             Location        Country Latitude Longitude Airport.Code  \
count           88837          88663    34382     34373        50132   
unique          27758            219    25589     27154        10374   
top     ANCHORAGE, AK  United States  332739N  0112457W         NONE   
freq              434          82248       19        24         1488   

       Airport.Name  ... Amateur.Built    Engine.Type FAR.Description  \
count         52704  ...         88787          81793           32023   
unique        24870  ...             2             12              31   
top         Private  ...            No  Reciprocating             091   
freq            240  ...         80312          69530           18221   

       Schedule Purpose.of.flight Air.carrier Weather.Condition  \
count     12582             82697       16648             84397   
unique        3                26       13590                 4   
top        NSCH          Personal       Pilot               VMC   
freq       4474             49448         258             77303   

       Broad.phase.of.flight   Report.Status Publication.Date  
count                  61724           82505            75118  
unique                    12           17074             2924  
top                  Landing  Probable Cause       25-09-2020  
freq                   15428           61754            17019  

[4 rows x 26 columns]


************ Categorical Columns ************
Index(['Event.Id', 'Investigation.Type', 'Accident.Number', 'Event.Date',
       'Location', 'Country', 'Latitude', 'Longitude', 'Airport.Code',
       'Airport.Name', 'Injury.Severity', 'Aircraft.damage',
       'Aircraft.Category', 'Registration.Number', 'Make', 'Model',
       'Amateur.Built', 'Engine.Type', 'FAR.Description', 'Schedule',
       'Purpose.of.flight', 'Air.carrier', 'Weather.Condition',
       'Broad.phase.of.flight', 'Report.Status', 'Publication.Date'],
      dtype='object')


************ Top 5 Categories per Categorical Column ************

Event.Id:
Event.Id
20001214X45071    3
20001212X19172    3
20001214X45064    2
20001212X17570    2
20001214X37556    2
```

```
Name: count, dtype: int64

Investigation.Type:
Investigation.Type
Accident    85015
Incident     3874
Name: count, dtype: int64

Accident.Number:
Accident.Number
ERA22LA103    2
DCA22WA089    2
DCA22WA167    2
ERA22LA119    2
CEN22LA149    2
Name: count, dtype: int64

Event.Date:
Event.Date
1982-05-16    25
1984-06-30    25
2000-07-08    25
1983-08-05    24
1984-08-25    24
Name: count, dtype: int64

Location:
Location
ANCHORAGE, AK      434
MIAMI, FL          200
ALBUQUERQUE, NM    196
HOUSTON, TX        193
CHICAGO, IL        184
Name: count, dtype: int64

Country:
Country
United States     82248
Brazil              374
Canada              359
Mexico              358
United Kingdom      344
Name: count, dtype: int64

Latitude:
Latitude
NaN          54507
332739N         19
335219N         18
334118N         17
32.815556       17
Name: count, dtype: int64

Longitude:
Longitude
NaN            54516
```

```
0112457W          24
1114342W          18
1151140W          17
-104.673056       17
Name: count, dtype: int64

Airport.Code:
Airport.Code
NaN     38757
NONE     1488
PVT       485
APA       160
ORD       149
Name: count, dtype: int64

Airport.Name:
Airport.Name
NaN                 36185
Private               240
PRIVATE               224
Private Airstrip      153
NONE                  146
Name: count, dtype: int64

Injury.Severity:
Injury.Severity
Non-Fatal    67357
Fatal(1)      6167
Fatal         5262
Fatal(2)      3711
Incident      2219
Name: count, dtype: int64

Aircraft.damage:
Aircraft.damage
Substantial    64148
Destroyed      18623
NaN             3194
Minor           2805
Unknown          119
Name: count, dtype: int64

Aircraft.Category:
Aircraft.Category
NaN           56602
Airplane      27617
Helicopter     3440
Glider          508
Balloon         231
Name: count, dtype: int64

Registration.Number:
Registration.Number
NaN      1382
NONE      344
UNREG     126
```

```
UNK        13
USAF        9
Name: count, dtype: int64

Make:
Make
Cessna    22227
Piper     12029
CESSNA     4922
Beech      4330
PIPER      2841
Name: count, dtype: int64

Model:
Model
152          2367
172          1756
172N         1164
PA-28-140     932
150           829
Name: count, dtype: int64

Amateur.Built:
Amateur.Built
No     80312
Yes     8475
NaN      102
Name: count, dtype: int64

Engine.Type:
Engine.Type
Reciprocating    69530
NaN               7096
Turbo Shaft       3609
Turbo Prop        3391
Turbo Fan         2481
Name: count, dtype: int64

FAR.Description:
FAR.Description
NaN                          56866
091                          18221
Part 91: General Aviation     6486
NUSN                          1584
NUSC                          1013
Name: count, dtype: int64

Schedule:
Schedule
NaN     76307
NSCH     4474
UNK      4099
SCHD     4009
Name: count, dtype: int64

Purpose.of.flight:
```

```
Purpose.of.flight
Personal              49448
Instructional         10601
Unknown                6802
NaN                    6192
Aerial Application     4712
Name: count, dtype: int64

Air.carrier:
Air.carrier
NaN                  72241
Pilot                  258
American Airlines       90
United Airlines         89
Delta Air Lines         53
Name: count, dtype: int64

Weather.Condition:
Weather.Condition
VMC    77303
IMC     5976
NaN     4492
UNK      856
Unk      262
Name: count, dtype: int64

Broad.phase.of.flight:
Broad.phase.of.flight
NaN            27165
Landing        15428
Takeoff        12493
Cruise         10269
Maneuvering     8144
Name: count, dtype: int64

Report.Status:
Report.Status
Probable Cause    61754
NaN                6384
Foreign            1999
<br /><br />        167
Factual             145
Name: count, dtype: int64

Publication.Date:
Publication.Date
25-09-2020    17019
NaN           13771
26-09-2020     1769
03-11-2020     1155
31-03-1993      452
Name: count, dtype: int64


************ Check for Duplicates ************
Total Duplicates: 0
```

```
************ Unique Values ************
Injury Severity: ['Fatal(2)', 'Fatal(4)', 'Fatal(3)', 'Fatal(1)', 'Non-Fatal', 'Inci
dent', 'Fatal(8)', 'Fatal(78)', 'Fatal(7)', 'Fatal(6)', 'Fatal(5)', 'Fatal(153)', 'F
atal(12)', 'Fatal(14)', 'Fatal(23)', 'Fatal(10)', 'Fatal(11)', 'Fatal(9)', 'Fatal(1
7)', 'Fatal(13)', 'Fatal(29)', 'Fatal(70)', 'Unavailable', 'Fatal(135)', 'Fatal(3
1)', 'Fatal(256)', 'Fatal(25)', 'Fatal(82)', 'Fatal(156)', 'Fatal(28)', 'Fatal(18)',
'Fatal(43)', 'Fatal(15)', 'Fatal(270)', 'Fatal(144)', 'Fatal(174)', 'Fatal(111)', 'F
atal(131)', 'Fatal(20)', 'Fatal(73)', 'Fatal(27)', 'Fatal(34)', 'Fatal(87)', 'Fatal
(30)', 'Fatal(16)', 'Fatal(47)', 'Fatal(56)', 'Fatal(37)', 'Fatal(132)', 'Fatal(6
8)', 'Fatal(54)', 'Fatal(52)', 'Fatal(65)', 'Fatal(72)', 'Fatal(160)', 'Fatal(189)',
'Fatal(123)', 'Fatal(33)', 'Fatal(110)', 'Fatal(230)', 'Fatal(97)', 'Fatal(349)', 'F
atal(125)', 'Fatal(35)', 'Fatal(228)', 'Fatal(75)', 'Fatal(104)', 'Fatal(229)', 'Fat
al(80)', 'Fatal(217)', 'Fatal(169)', 'Fatal(88)', 'Fatal(19)', 'Fatal(60)', 'Fatal(1
13)', 'Fatal(143)', 'Fatal(83)', 'Fatal(24)', 'Fatal(44)', 'Fatal(64)', 'Fatal(92)',
'Fatal(118)', 'Fatal(265)', 'Fatal(26)', 'Fatal(138)', 'Fatal(206)', 'Fatal(71)', 'F
atal(21)', 'Fatal(46)', 'Fatal(102)', 'Fatal(115)', 'Fatal(141)', 'Fatal(55)', 'Fata
l(121)', 'Fatal(45)', 'Fatal(145)', 'Fatal(117)', 'Fatal(107)', 'Fatal(124)', 'Fatal
(49)', 'Fatal(154)', 'Fatal(96)', 'Fatal(114)', 'Fatal(199)', 'Fatal(89)', 'Fatal(5
7)', 'Fatal', nan, 'Minor', 'Serious']
Aircraft Damage: ['Destroyed', 'Substantial', 'Minor', nan, 'Unknown']
Weather Condition: ['UNK', 'IMC', 'VMC', nan, 'Unk']
Broad Phase of Flight: ['Cruise', 'Unknown', 'Approach', 'Climb', 'Takeoff', 'Landin
g', 'Taxi', 'Descent', 'Maneuvering', 'Standing', 'Go-around', 'Other', nan]
Purpose of Flight: ['Personal', nan, 'Business', 'Instructional', 'Unknown', 'Ferr
y', 'Executive/corporate', 'Aerial Observation', 'Aerial Application', 'Public Aircr
aft', 'Skydiving', 'Other Work Use', 'Positioning', 'Flight Test', 'Air Race/show',
'Air Drop', 'Public Aircraft - Federal', 'Glider Tow', 'Public Aircraft - Local', 'E
xternal Load', 'Public Aircraft - State', 'Banner Tow', 'Firefighting', 'Air Race sh
ow', 'PUBS', 'ASHO', 'PUBL']
```

Key Observations

The dataset covering accidents from 1948 to recent years, with key columns like 'Aircraft.Category' (to distinguish airplanes from helicopters), 'Make' and 'Model' (to identify aircraft), 'Injury.Severity' and 'Total.Fatal.Injuries' (to assess safety risks), and 'Broad.phase.of.flight' and 'Weather.Condition' (to analyze accident causes). Significant missing data exists in 'Aircraft.Category' (56,602 missing), 'Broad.phase.of.flight' (27,165 missing), and injury columns (e.g., 11,401 missing for 'Total.Fatal.Injuries'), requiring cleaning in the next phase. Numerical summaries show most accidents have zero fatalities (mean 0.65, max 349), indicating non-fatal incidents dominate. No duplicates ensure data integrity.

Data Preparation

This section cleans and preprocesses the aviation accident dataset to ensure data quality for analyzing low-risk airplanes and comparing them to another aircraft type. Steps include handling missing values, standardizing text, filtering for relevant aircraft categories, and creating derived columns for risk analysis.

In [15]:

```
# Standardize columns
df.columns = df.columns.str.strip().str.replace('.', '_').str.lower()

pd.set_option('display.max_rows', 2000)

# Select essential columns
selected_cols = [
    'event_id', 'event_date', 'make', 'model', 'aircraft_category',
    'engine_type', 'amateur_built', 'number_of_engines', 'purpose_of_flight',
    'schedule', 'aircraft_damage', 'injury_severity', 'total_fatal_injuries',
    'total_serious_injuries', 'total_minor_injuries', 'total_uninjured',
    'weather_condition', 'broad_phase_of_flight'
]
df = df[selected_cols]

# Fill missing data with defaults or placeholders
df['weather_condition'] = df['weather_condition'].fillna('UNK').str.upper()
df['broad_phase_of_flight'] = df['broad_phase_of_flight'].fillna('Unknown')
df[['total_fatal_injuries', 'total_serious_injuries', 'total_minor_injuries', 'tota


# Drop rows with missing critical data
critical_cols = ['aircraft_category', 'make', 'model', 'injury_severity', 'total_fa
df = df.dropna(subset=critical_cols)


# Filter rows by valid categories for aircraft damage and amateur built
valid_damage = ['Destroyed', 'Substantial', 'Minor']
df = df[df['aircraft_damage'].isin(valid_damage)]
df = df[df['amateur_built'].isin(['Yes', 'No'])]


injury_cols = ['total_fatal_injuries', 'total_serious_injuries',
               'total_minor_injuries', 'total_uninjured']
df[injury_cols] = df[injury_cols].fillna(0)


injury_cols = ['total_fatal_injuries', 'total_serious_injuries', 'total_minor_injur
for col in injury_cols:
    df[col] = pd.to_numeric(df[col], errors='coerce').fillna(0)

# Standardize text columns
df['make'] = df['make'].str.upper().str.replace(r'[^A-Z0-9 ]', '', regex=True).str.
df['model'] = df['model'].str.upper().str.replace(r'[^A-Z0-9 ]', '', regex=True).st
df['aircraft_category'] = df['aircraft_category'].str.upper().str.replace(r'[^A-Z0-
df['weather_condition'] = df['weather_condition'].str.upper()

# Normalize 'make' names with mapping
normalize_make = {
   'AEROFAB INC.':  'AEROFAB INC',
    'AEROFAB':  'AEROFAB INC',
     'AERO SP Z O O GOBOSH': 'AERO SP ZOO',
    'AERO SP Z O O': 'AERO SP ZOO',
 'AEROPRO CZ S R O': 'AEROPRO CZ',
 'AIR TRACTOR INC': 'AIR TRACTOR',
 'AIR TRACTOR INC.': 'AIR TRACTOR',
 'AIRBUS HELICOPTERS': 'AIRBUS',
 'AIRBUS HELICOPTERS INC': 'AIRBUS',
 'AIRBUS Helicopters': 'AIRBUS',
 'AIRCRAFT MFG & DVLPMT CO':'AIRCRAFT MFG & DEVELOPMENT CO',
  'AIRPLANE FACTORY (PTY) LTD THE':'AIRPLANE FACTORY (PTY) LTD',
 'AMS Flight': 'AMS FLIGHT',
 'ANKESTAR, BRADLEY D.': 'ANKERSTAR BRADLEY D',
 'ANTONOV':'ANTONOVICH ANTON B',
     'AUTOGYRO': 'AUTOGYRO GMBH',
 'AVIAT':'AVIAT AIRCRAFT INC',
 'AVIAT AIRCRAFT':'AVIAT AIRCRAFT INC',
 'AVIAT INC':'AVIAT AIRCRAFT INC',
 'AVIATE':'AVIAT AIRCRAFT INC',
'AEROTEK': 'AEROTEK INC',
 'AGUSTAWESTLAND PHILADELPHIA': 'AGUSTAWESTLAND PHILADELPHIA CO',
      'AERO VODOCHODY': 'AERO VODOCHODY WORKS',
 'AERO VODOCHODY AERO WORKS': 'AERO VODOCHODY WORKS',
 'AERO WORKS': 'AERO VODOCHODY WORKS',
    'AERONCA CHAMP': 'AERONCA CHAMPION',
 'AERO AT SP ZOO': 'AERO SP ZOO',
'AEROSTAR S A': 'AEROSTAR SA',

 'AIRCRAFT MFG  DEV CO': 'AIRCRAFT MFG  DEVELOPMENT CO',
 'AIRCRAFT MFG  DEV CO AMD': 'AIRCRAFT MFG  DEVELOPMENT CO',

 'AIRBORNE WINDSPORT': 'AIRBORNE WINDSPORTS',

 'AIRBORNE WINDSPORTS LTD': 'AIRBORNE WINDSPORTS PTY LTD',

 'AIRPLANE FACTORY': 'AIRPLANE FACTORY PTY LTD',

     'AMERICAN CHAMPION': 'AMERICAN CHAMPION AIRCRAFT',
 'AMERICAN CHAMPION ACAC': 'AMERICAN CHAMPION AIRCRAFT',
 'AMERICAN CHAMPION AIRCRAFT': 'AMERICAN CHAMPION AIRCRAFT',
 'AMERICAN CHAMPION AIRCRAFT COR': 'AMERICAN CHAMPION AIRCRAFT',
 'AVIONS MUDRY ET CIE': 'AVIONS MUDRY CIE',

 'BEECH': 'BEECH AIRCRAFT CORPORATION',
 'BEECH AIRCRAFT': 'BEECH AIRCRAFT CORPORATION',
 'BEECH AIRCRAFT CORP': 'BEECH AIRCRAFT CORPORATION',
 'BEECHCRAFT': 'BEECH AIRCRAFT CORPORATION',
 'BEECHCRAFT CORPORATION': 'BEECH AIRCRAFT CORPORATION',

 'BELLTRANSWORLD HELICOPTER COR': 'BELLTRANSWORLD HELICOPTERS',
 'BENNET': 'BENNETT',

 'BOWER': 'BOWERS FLY BABY',
 'BOWERS': 'BOWERS FLY BABY',
 'BOWERS FLYBABY': 'BOWERS FLY BABY',
 'BRITISH AEROSPACE': 'BRITISH AIRCRAFT CORP',
 'BRITISH AIRCRAFT CORP BAC': 'BRITISH AIRCRAFT CORP',
 'BRITTENNORMAN': 'BRITTEN NORMAN',

 'BUCKEYE':'BUCKEYE AVIATION INC',
 'BUCKEYE AVIATION':'BUCKEYE AVIATION INC',
 'BUCKEYE INDUSTRIES':'BUCKEYE INDUSTRIES INC',
 'BURKHART GROB FLUGZEUGBAH':'BURKHART GROB',
 'BURKHART GROB FLUGZEUGBAU':'BURKHART GROB',
 'BURKHARTGROB':'BURKHART GROB',

   'C A TECNAM SRL': 'CA TECNAM SRL',

 'CENTRAL OHIO DRAGONFLY CLUB':'CENTRAL OHIO DRAGONFLY CLUB LLC',
 'CENTRAL OHIO DRAGONFLY CLUBLLC':'CENTRAL OHIO DRAGONFLY CLUB LLC',
    'CESSNA':'CESSNA AIRCRAFT COMPANY',
 'CESSNA AIRCRAFT':'CESSNA AIRCRAFT COMPANY',
 'CESSNA AIRCRAFT CO':'CESSNA AIRCRAFT COMPANY',
     'CGS':'CGS AVIATION LLC',
 'CGS AVIATION':'CGS AVIATION LLC',
  'CHRISTEN INDUSTRIES':'CHRISTEN INDUSTRIES INC',
  'CIRRUS':'CIRRUS DESIGN CORPORATION',
 'CIRRUS DESIGN':'CIRRUS DESIGN CORPORATION',
 'CIRRUS DESIGN CORP':'CIRRUS DESIGN CORPORATION',
 'CLARK':  'CLARKE COLIN',
 'CLARKE COLIN A':  'CLARKE COLIN',
    'COMMANDER': 'COMMANDER AIRCRAFT COMPANY',
 'COMMANDER AIRCRAFT CO': 'COMMANDER AIRCRAFT COMPANY',
'CONSOLIDATED  AERONAUTICS INC': 'CONSOLIDATED AERONAUTICS INC',
 'CONSOLIDATED AERO': 'CONSOLIDATED AERONAUTICS INC',
 'CONSOLIDATED AERONAUTICS': 'CONSOLIDATED AERONAUTICS INC',
     'CONTINENTAL':'CONTINENTAL COPTERS INC',
 'CONTINENTAL COPTERS':'CONTINENTAL COPTERS INC',
  'COSTRUZIONI AERONAUTICHE':'COSTRUZIONI AERONAUTICHE TECNA',
 'COSTRUZIONI AERONAUTICHETECNAM': 'COSTRUZIONI AERONAUTICHE TECNA',

 'CUB CRAFTER':'CUB CRAFTERS INC',
 'CUB CRAFTERS':'CUB CRAFTERS INC',
 'CUBCRAFTER':'CUB CRAFTERS INC',
 'CUBCRAFTERS':'CUB CRAFTERS INC',
 'CUBCRAFTERS INC':'CUB CRAFTERS INC',
 'CULVER':'CULVER GLENN',
  'CURTISS': 'CURTISS MOSES',

 'CURTISSWRIGHT':'CURTISS WRIGHT',
 'CURTISWRIGHT':'CURTISS WRIGHT',
 'CZECH AIRCRAFT WORKS SPOL SRO':'CZECH AIRCRAFT WORKS',

 'CZECH SPORT AIRCRAFT':'CZECH SPORT AIRCRAFT AS',
 'CZECH SPORT AIRCRAFT A S':'CZECH SPORT AIRCRAFT AS',
  'AEROS': 'AEROS LTD',

 'AEROS LTDSKYRANGER AIRCRAFT': 'AEROS LTD',
     'AEROSPATIALE': 'AEROSPATIALE ALENIA',

 'AMATEUR BUILT': 'AMATEUR BUILT AIRCRAFT',
 'AMERICAN AIR RACING': 'AMERICAN AIR RACING LTD',
 'ARCTIC':'ARCTIC AIRCRAFT CO INC',
 'ARCTIC AIRCRAFT':'ARCTIC AIRCRAFT CO INC',
 'ARROWFALCON EXPORTERS INC':'ARROW FALCON EXPORTERS INC',
    'ATKINS': 'ATKINS JOHN',
 'BAE':'BAE SYSTEMS',

 'BAE SYSTEMS OPERATIONS LIMIT':'BAE SYSTEMS',

 'BORDELON':'BORDELON BRUCE',

 'DESTINY POWERED PARACHUTES':'DESTINY POWERED PARACHUTES LLC',

 'DG FLUGZEUGBAU': 'DG FLUGZEUGBAU GMBH',
 'DG FLUGZEUGBAU GMBH':'DG FLUGZEUGBAU GMBH',
 'DGFLUGZEUGBAU GMBH':'DG FLUGZEUGBAU GMBH',

    'DIAMOND':'DIAMOND AIRCRAFT INDUSTRY INC',
 'DIAMOND AICRAFT INDUSTRIES INC':'DIAMOND AIRCRAFT INDUSTRY INC',
 'DIAMOND AIRCRAFT':'DIAMOND AIRCRAFT INDUSTRY INC',
 'DIAMOND AIRCRAFT IND GMBH':'DIAMOND AIRCRAFT INDUSTRY INC',
 'DIAMOND AIRCRAFT IND INC':'DIAMOND AIRCRAFT INDUSTRY INC',
 'DIAMOND AIRCRAFT INDUSTRIES':'DIAMOND AIRCRAFT INDUSTRY INC',
 'DIAMOND AIRCRAFT INDUSTRIES IN':'DIAMOND AIRCRAFT INDUSTRY INC',
 'EMBRAER':'EMBRAER AIRCRAFT',
 'EMBRAER EXECUTIVE AIRCRAFT INC':'EMBRAER AIRCRAFT',

 'EMBRAER S A': 'EMBRAER SA',


 'EUROCOPTER':'EUROCOPTER DEUTSCHLAND GMBH',
 'EUROCOPTER DEUTSCHLAND':'EUROCOPTER DEUTSCHLAND GMBH',
 'EVEKTOR':'EVEKTOR AEROTECHNIC',
 'EVEKTOR AEROTECHNIK':'EVEKTOR AEROTECHNIC',
 'EVEKTOR AEROTECHNIK AS':'EVEKTOR AEROTECHNIC',
 'EVEKTORAEROTECHNIK':'EVEKTOR AEROTECHNIC',
 'EVEKTORAEROTECHNIK AS':'EVEKTOR AEROTECHNIC',

 'GATES LEAR JET':'GATES LEARJET CORPORATION',
 'GATES LEARJET':'GATES LEARJET CORPORATION',
 'GATES LEARJET CORP':'GATES LEARJET CORPORATION',

 'GLASAIR':'GLASAIR AVIATION USA LLC',
 'GLASAIR AVIATION LLC':'GLASAIR AVIATION USA LLC',
 'GRUMMAN':'GRUMMAN AIRCRAFT ENG CORP',
 'GRUMMAN ACFT ENG':'GRUMMAN AIRCRAFT ENG CORP',
 'GRUMMAN ACFT ENG COR':'GRUMMAN AIRCRAFT ENG CORP',
 'GRUMMAN ACFT ENG CORSCHWEIZER':'GRUMMAN AIRCRAFT ENG CORP',
 'GRUMMAN AIRCRAFT':'GRUMMAN AIRCRAFT ENG CORP',
 'GRUMMAN AIRCRAFT CORSCHWEIZER':'GRUMMAN AIRCRAFT ENG CORP',
 'GULFSTREAM AM CORP COMM DIV':'GULFSTREAM AMERICAN CORP',
 'GULFSTREAM AMERICAN':'GULFSTREAM AMERICAN CORP'
}
df['make'] = df['make'].replace(normalize_make).fillna('UNKNOWN')

# Map damage severity for scoring
damage_map = {'Minor': 1, 'Substantial': 2, 'Destroyed': 3}
df['damage_severity'] = df['aircraft_damage'].map(damage_map)
```

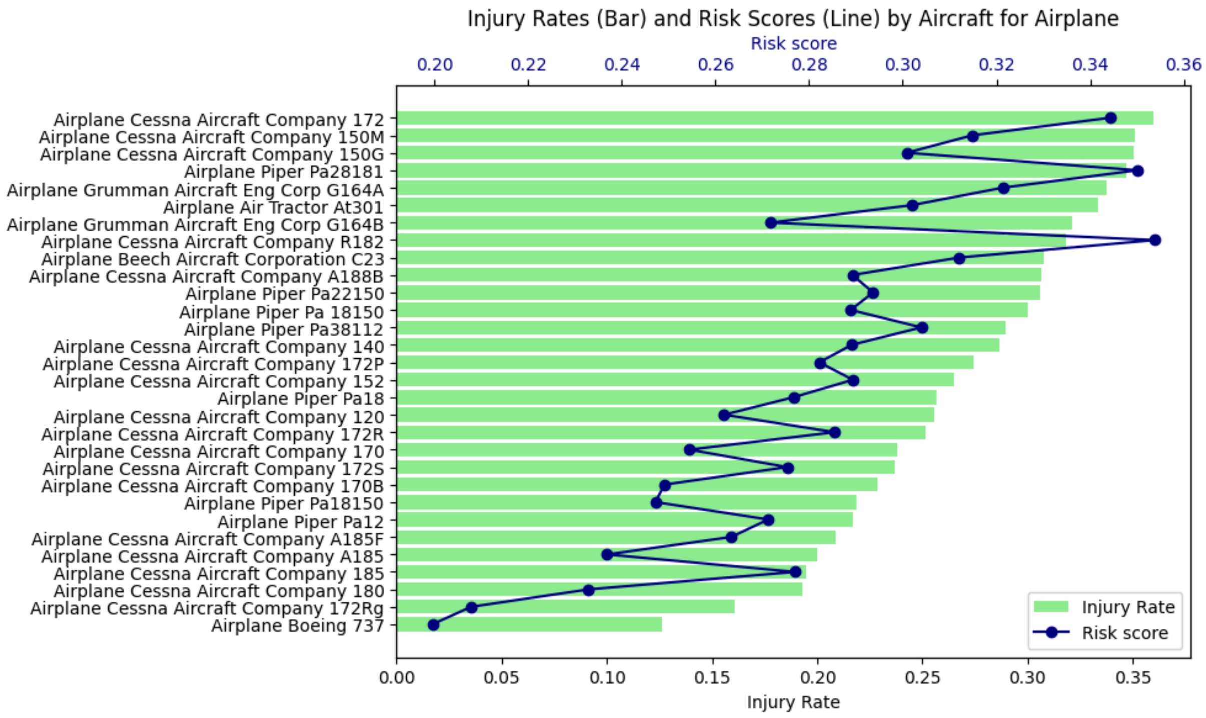
```
# Convert injury columns to numeric and fill missing
df['total_fatal_injuries'] = pd.to_numeric(df['total_fatal_injuries'], errors='coer
df['total_serious_injuries'] = pd.to_numeric(df['total_serious_injuries'], errors='
df['total_minor_injuries'] = pd.to_numeric(df['total_minor_injuries'], errors='coer
df['total_uninjured'] = pd.to_numeric(df['total_uninjured'], errors='coerce').filln

# Calculate total injuries and occupants
df['total_injuries'] = df['total_fatal_injuries'] + df['total_serious_injuries'] +
df['total_occupants'] = df['total_injuries'] + df['total_uninjured']
```

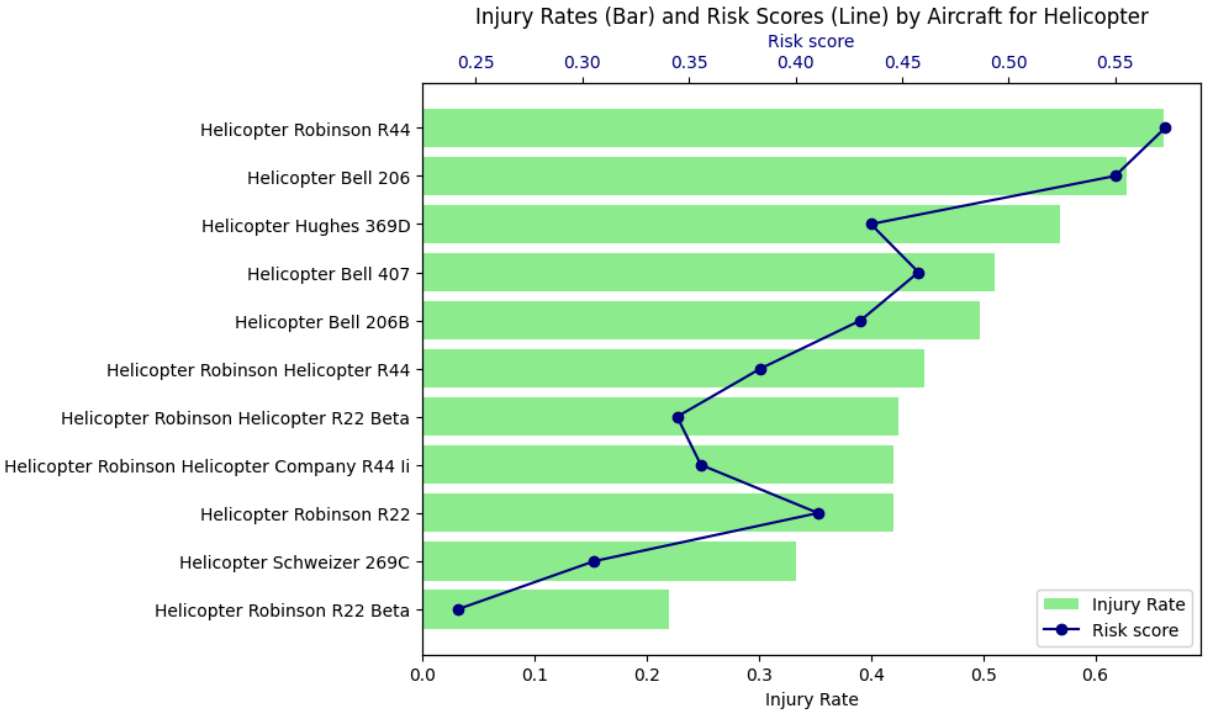
Analysis and Results

The analysis focused on identifying low-risk airplanes and comparing them to helicopters. A risk score will be calculated for each aircraft type based on injury rate, fatality rate, and damage severity.

In [16]:

```
# Create aircraft_key for grouping: category_make_model
df['aircraft_key'] = (
    df['aircraft_category'].str.title().replace(' ', '', regex=False) + ' ' +
    df['make'].str.title().replace(' ', '', regex=False) + ' ' +
    df['model'].str.title().replace(' ', '', regex=False)
)

# Group by aircraft_key
grouped = df.groupby('aircraft_key').agg(
    event_count=('aircraft_key', 'count'),
    total_injuries=('total_injuries', 'sum'),
    total_fatalities=('total_fatal_injuries', 'sum'),
    total_uninjured=('total_uninjured', 'sum'),
    total_occupants=('total_occupants', 'sum'),
    avg_damage_severity=('damage_severity', 'mean')
).reset_index()


# Calculate group level rates
grouped['injury_rate'] = grouped['total_injuries'] / grouped['total_occupants']
grouped['fatality_rate'] = grouped['total_fatalities'] / grouped['total_occupants']
grouped['risk_score'] = (
    0.4 * grouped['injury_rate'] +
    0.4 * grouped['fatality_rate'] +
    0.2 * (grouped['avg_damage_severity'] / 3)
)

#filter groups with sufficient data
min_event_count = 50
grouped = grouped[grouped['event_count'] >= min_event_count]


import seaborn as sns
import matplotlib.pyplot as plt


def showInjuryRateAndRiskScorePlot(aircraft_type):
    plot_data = grouped[grouped['aircraft_key'].str.startswith(aircraft_type)].sort
    plot_data = plot_data.sort_values(by='injury_rate')
    plot_data = plot_data.head(30)

    fig, ax1 = plt.subplots(figsize=(10, 6))

    # Bar plot for injury rate
    ax1.barh(plot_data['aircraft_key'], plot_data['injury_rate'], color='lightgreen
    ax1.set_xlabel('Injury Rate')
    ax1.set_title(f'Injury Rates (Bar) and Risk Scores (Line) by Aircraft for {airc

    # Line plot for risk rate (overlaid)
    ax2 = ax1.twiny()
    ax2.plot(plot_data['risk_score'], plot_data['aircraft_key'], color='navy', mark
    ax2.set_xlabel('Risk score', color='navy')
    ax2.tick_params(axis='x', labelcolor='navy')

    # Combine legends
    lines_labels = ax1.get_legend_handles_labels()
    lines_labels2 = ax2.get_legend_handles_labels()
    ax1.legend(lines_labels[0] + lines_labels2[0], lines_labels[1] + lines_labels2[

    plt.tight_layout()
    plt.savefig(f'injury_rates_risk_scores_{aircraft_type}.png')
    plt.show()

showInjuryRateAndRiskScorePlot('Airplane')
showInjuryRateAndRiskScorePlot('Helicopter')
```





The charts above show the injury rates (bars) and risk scores (line) for the top 30 low-risk airplanes and helicopters. The Airplane Boeing 737, Cessna aircraft company 172Rg and the helicopter Robinson R22 Beta and Schweizer 269C consistently show low injury rates and risk scores, making them ideal candidates.

Business Recommendations

Based on the analysis, the following recommendations are provided to the head of the new aviation division:

Business Recommendation 1: Prioritize Boeing 737 for Airplane Operations

The Boeing 737 has the lowest injury rate (0.126193) and risk score (0.199628) among airplanes, reflecting a strong safety record for commercial use. Purchase 2–3 newer models Boeing 737 or Airplane Cessna Aircraft Company 172Rg with updated safety systems, invest in extensive pilot and maintenance training, and begin with a small fleet for controlled operations.

Business Recommendation 2: Select Robinson R22 Beta for Helicopter Operations

The Robinson R22 Beta has the lowest injury rate (0.219780) and risk score (0.241442) among helicopters, suitable for training or light commercial roles like sightseeing. Acquire R22 Betas, provide specialized pilot and maintenance training due to higher complexity, and limit initial use to low-risk environments.

Conclusion

This analysis has enabled data-driven insights into aircraft safety profiles. Aircraft such as the Boeing 737 demonstrate high operational safety with many incidents resulting in no or minor injuries.

Next Steps

1. Procure newer 737s and R22 Betas from reliable suppliers.
2. Develop comprehensive training for pilots and crews.
3. Analyze demand for commercial routes (737) and niche services (R22 Beta).
4. Implement a safety tracking system for ongoing risk assessment.