AKADEMI EDUCATION – First Cohort (2025): Data Science & AI

First Project: Data Analysis & Engineering - Phase 1

Student name: Riché FLEURINORD
Student pace: self paced
Deadline Submission: June 8, 2025
Instructors' Names: Wedter JEROME & Geovany Batista Polo LAGUERRE
Blog post URL (GitHub Repository Link):
https://github.com/richefleuriord/Fleurinord_Dsc_Aviation_Project.git (https://github.com/richefleuriord/Fleurinord_Dsc_Aviation_Project.git)

Project Title

Flight Risk: A Data-Driven Analysis of Aviation Accidents (1962–2023)

Overview

This data science project analyzes aviation accident data from 1962 to 2023 to support strategic decision-making in the aviation sector. Through data cleaning, exploration, and visualization, the goal is to identify low-risk aircraft models and generate actionable insights for business stakeholders considering investment in aviation.

Business Problem

To support a strategic investment analysis in the aviation sector, I propose to examine historical trends in aviation accidents in order to identify the most reliable aircraft profiles. This approach aims to help a fictional company allocate its resources wisely by minimizing the risks associated with purchasing and operating commercial and private aircraft.

By analyzing accident data collected by the National Transportation Safety Board from 1962 to 2023, I will highlight aircraft models, common causes of incidents, and high-risk contexts. The goal is to produce actionable recommendations to guide the company’s decisions and enhance safety, while ensuring effective cost management and future operations in this new sector.

1-Data Understanding

The dataset used in this project comes from the National Transportation Safety Board (NTSB) and covers aviation events that occurred between 1962 and 2023. It includes both accident and incident investigations, making it a valuable source for analyzing aviation-related risks.

Each event is associated with a unique identifier and contains detailed information such as the date and location of the event, characteristics of the aircraft involved (manufacturer, model, number of engines, engine type), weather conditions, type of flight (commercial, private, etc.), and human consequences (injuries, fatalities).

This initial step aims to:

1- Explore the structure of the dataset,

2- Identify the types of variables available,

3- Detect any missing or inconsistent values, 4- And gain a global understanding of the data to guide the upcoming exploratory analysis and strategic recommendations.

1.1 Importing the necessary libraries

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
```

1.2 Loading the datasets

In [2]:

```
df = pd.read_csv("Data/AviationData.csv", encoding= "ISO-8859-1",\
                 low_memory=False)
```

In [3]:

```
df_1 = pd.read_csv("Data/USState_Codes.csv", encoding= "ISO-8859-1",\
                   low_memory=False)
```

1.3 Overview of the df dataset

In [4]:

```
df.head(10)
```

Out[4]:

	Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Country	
0	20001218X45444	Accident	SEA87LA080	1948-10-24	MOOSE CREEK, ID	United States	
1	20001218X45447	Accident	LAX94LA336	1962-07-19	BRIDGEPORT, CA	United States	
2	20061025X01555	Accident	NYC07LA005	1974-08-30	Saltville, VA	United States	3
3	20001218X45448	Accident	LAX96LA321	1977-06-19	EUREKA, CA	United States	
4	20041105X01764	Accident	CHI79FA064	1979-08-02	Canton, OH	United States	
5	20170710X52551	Accident	NYC79AA106	1979-09-17	BOSTON, MA	United States	4
6	20001218X45446	Accident	CHI81LA106	1981-08-01	COTTON, MN	United States	
7	20020909X01562	Accident	SEA82DA022	1982-01-01	PULLMAN, WA	United States	
8	20020909X01561	Accident	NYC82DA015	1982-01-01	EAST HANOVER, NJ	United States	
9	20020909X01560	Accident	MIA82DA029	1982-01-01	JACKSONVILLE, FL	United States	

10 rows × 31 columns

In [5]:

```
df.shape
```

Out[5]:

```
(88889, 31)
```

In [6]:

```
df.columns
```

Out[6]:

```
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', 'Number.of.Engines', 'Engine.Type', 'FAR.Descriptio
n',
       'Schedule', 'Purpose.of.flight', 'Air.carrier', 'Total.Fatal.Injurie
s',
       'Total.Serious.Injuries', 'Total.Minor.Injuries', 'Total.Uninjured',
       'Weather.Condition', 'Broad.phase.of.flight', 'Report.Status',
       'Publication.Date'],
      dtype='object')
```

```
In [7]: df.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            50249 non-null  object 
 9   Airport.Name            52790 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     87572 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             81812 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           82508 non-null  object 
 30  Publication.Date        75118 non-null  object 
dtypes: float64(5), object(26)
memory usage: 21.0+ MB
```

We examined the dataset using the .info() method to understand its structure. The dataset contains 88,889 rows and 31 columns. Among these, 5 columns are of type float64 (numerical), while the remaining 26 columns are of type object (usually categorical or string data).

Several columns contain missing values, especially:

1- "Latitud"e and "Longitude" have data for only ~34,000 rows.

2- "Aircraft.Category" and "FAR.Description" also have many missing values.

3- Some columns like "Schedule" and "Air.carrier" are very sparsely filled.

In [8]:

```
df.duplicated().sum()
```

Out[8]:

```
0
```

The instruction returns 0, which means that no row in the DataFrame is duplicated.

1.4 Overview of the df_1 dataset

In [9]:

```
df_1.head(5)
```

Out[9]:

	US_State	Abbreviation
0	Alabama	AL
1	Alaska	AK
2	Arizona	AZ
3	Arkansas	AR
4	California	CA

In [10]:

```
df_1.shape
```

Out[10]:

```
(62, 2)
```

In [11]:

```
df_1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 62 entries, 0 to 61
Data columns (total 2 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   US_State      62 non-null     object
 1   Abbreviation  62 non-null     object
dtypes: object(2)
memory usage: 1.1+ KB
```

1.5 Checking for missing values

In [12]:

```
df.isnull().sum().sort_values(ascending=False)
```

Out[12]:

```
Schedule                  76307
Air.carrier               72241
FAR.Description           56866
Aircraft.Category         56602
Longitude                 54516
Latitude                  54507
Airport.Code              38640
Airport.Name              36099
Broad.phase.of.flight     27165
Publication.Date          13771
Total.Serious.Injuries    12510
Total.Minor.Injuries      11933
Total.Fatal.Injuries      11401
Engine.Type                7077
Report.Status              6381
Purpose.of.flight          6192
Number.of.Engines          6084
Total.Uninjured            5912
Weather.Condition          4492
Aircraft.damage            3194
Registration.Number        1317
Injury.Severity            1000
Country                     226
Amateur.Built               102
Model                        92
Make                         63
Location                     52
Event.Date                    0
Accident.Number               0
Investigation.Type            0
Event.Id                      0
dtype: int64
```

To assess the quality of our dataset, we examined the number of missing values in each column. The columns “Schedule”, “Air.carrier”, and “FAR.Description” have a particularly high number of missing entries (over 50,000), which could significantly impact the analysis or modeling. These variables will require special attention during the data preprocessing phase, depending on their relevance to the problem.

1.5 Completeness Analysis (%)

In [13]:

```
missing_pct = df.isnull().mean().sort_values(ascending=False) * 100
missing_pct.head(31)
```

Out[13]:

```
Schedule                  85.845268
Air.carrier               81.271023
FAR.Description           63.974170
Aircraft.Category         63.677170
Longitude                 61.330423
Latitude                  61.320298
Airport.Code              43.469946
Airport.Name              40.611324
Broad.phase.of.flight     30.560587
Publication.Date          15.492356
Total.Serious.Injuries    14.073732
Total.Minor.Injuries      13.424608
Total.Fatal.Injuries      12.826109
Engine.Type                7.961615
Report.Status              7.178616
Purpose.of.flight          6.965991
Number.of.Engines          6.844491
Total.Uninjured            6.650992
Weather.Condition          5.053494
Aircraft.damage            3.593246
Registration.Number        1.481623
Injury.Severity            1.124999
Country                    0.254250
Amateur.Built              0.114750
Model                      0.103500
Make                       0.070875
Location                   0.058500
Event.Date                 0.000000
Accident.Number            0.000000
Investigation.Type         0.000000
Event.Id                   0.000000
dtype: float64
```

Several critical fields show a high percentage of missing values, particularly those related to location, aircraft information, and scheduling. To ensure robust and reliable analysis, it is recommended to apply data cleaning or imputation techniques, or to consider alternative data sources.

1.5 Statistical description of the numerical columns

In [14]:

```
df.describe()
```

Out[14]:

	Number.of.Engines	Total.Fatal.Injuries	Total.Serious.Injuries	Total.Minor.Injuries	Total.Unin
count	82805.000000	77488.000000	76379.000000	76956.000000	82977.0
mean	1.146585	0.647855	0.279881	0.357061	5.3
std	0.446510	5.485960	1.544084	2.235625	27.9
min	0.000000	0.000000	0.000000	0.000000	0.0
25%	1.000000	0.000000	0.000000	0.000000	0.0
50%	1.000000	0.000000	0.000000	0.000000	1.0
75%	1.000000	0.000000	0.000000	0.000000	2.0
max	8.000000	349.000000	161.000000	380.000000	699.0

These statistics show that:

1- Most incidents involve small aircraft with no injuries or fatalities.

2- There are extreme cases with large numbers of injuries or passengers, likely corresponding to commercial aviation accidents.

3- The dataset is skewed and contains outliers, which may need special treatment during analysis.

2-Data Preparation

2.1 Cleaning of unnecessary columns

In [5]:

```
df.drop(columns=['Schedule', 'Air.carrier', 'FAR.Description',\
                 'Broad.phase.of.flight', 'Longitude', 'Latitude',\
                 'Airport.Code', 'Airport.Name', 'Report.Status',\
                 'Registration.Number','Model', 'Event.Id',\
                 'Accident.Number', 'Aircraft.Category',\
                 'Publication.Date'], inplace=True)
```

In [16]:
```
df.head(10)
```

Out[16]:

	Investigation.Type	Event.Date	Location	Country	Injury.Severity	Aircraft.damage	
0	Accident	1948-10-24	MOOSE CREEK, ID	United States	Fatal(2)	Destroyed	
1	Accident	1962-07-19	BRIDGEPORT, CA	United States	Fatal(4)	Destroyed	
2	Accident	1974-08-30	Saltville, VA	United States	Fatal(3)	Destroyed	
3	Accident	1977-06-19	EUREKA, CA	United States	Fatal(2)	Destroyed	
4	Accident	1979-08-02	Canton, OH	United States	Fatal(1)	Destroyed	
5	Accident	1979-09-17	BOSTON, MA	United States	Non-Fatal	Substantial	M
6	Accident	1981-08-01	COTTON, MN	United States	Fatal(4)	Destroyed	

To improve the quality of our database and facilitate the subsequent stages of analysis, we removed certain columns deemed irrelevant to our research question. The deleted columns either had a high rate of missing values, making them unreliable for analysis, or offered little added value in assessing the risks associated with different aircraft models.

This data cleaning step aims to reduce the dimensionality of the dataset, limit noise in the data, and focus the analysis on variables that are truly useful for identifying the most reliable aircraft models.

2.2 Imputation of missing values for numerical variables

In [6]:
```
for col in ['Number.of.Engines', 'Total.Fatal.Injuries',\
            'Total.Serious.Injuries',\
            'Total.Minor.Injuries', 'Total.Uninjured']:
    df[col].fillna(df[col].median(), inplace=True)
```

In [18]:

```
df.head(10)
```

Out[18]:

	Investigation.Type	Event.Date	Location	Country	Injury.Severity	Aircraft.damage	
0	Accident	1948-10-24	MOOSE CREEK, ID	United States	Fatal(2)	Destroyed	St
1	Accident	1962-07-19	BRIDGEPORT, CA	United States	Fatal(4)	Destroyed	
2	Accident	1974-08-30	Saltville, VA	United States	Fatal(3)	Destroyed	Ce
3	Accident	1977-06-19	EUREKA, CA	United States	Fatal(2)	Destroyed	Roc
4	Accident	1979-08-02	Canton, OH	United States	Fatal(1)	Destroyed	Ce
5	Accident	1979-09-17	BOSTON, MA	United States	Non-Fatal	Substantial	Mcdo Do
6	Accident	1981-08-01	COTTON, MN	United States	Fatal(4)	Destroyed	Ce
7	Accident	1982-01-01	PULLMAN, WA	United States	Non-Fatal	Substantial	Ce
8	Accident	1982-01-01	EAST HANOVER, NJ	United States	Non-Fatal	Substantial	Ce
9	Accident	1982-01-01	JACKSONVILLE, FL	United States	Non-Fatal	Substantial	Ame

To address the missing values in the numerical variables, we opted for median imputation rather than mean imputation. This method is particularly suitable for datasets that may contain extreme or outlier values, as is often the case with accident-related databases.

The median, being a robust measure of central tendency, helps limit the impact of outliers on the analysis results. By choosing this approach, we ensure better representativeness of the overall data, which is essential for the reliability of both descriptive statistics and the predictive models that will be developed later on.

2.3 Imputation of missing values in categorical variables

In [7]:

```
for col in ['Country', 'Injury.Severity', 'Aircraft.damage', 'Make',\
            'Amateur.Built', 'Engine.Type', 'Purpose.of.flight',\
            'Weather.Condition',\
             'Investigation.Type']:
    df[col].fillna(df[col].mode()[0], inplace=True)
```

In [20]:
```
df.head(10)
```

Out[20]:

	Investigation.Type	Event.Date	Location	Country	Injury.Severity	Aircraft.damage	
0	Accident	1948-10-24	MOOSE CREEK, ID	United States	Fatal(2)	Destroyed	St
1	Accident	1962-07-19	BRIDGEPORT, CA	United States	Fatal(4)	Destroyed	
2	Accident	1974-08-30	Saltville, VA	United States	Fatal(3)	Destroyed	Ce
3	Accident	1977-06-19	EUREKA, CA	United States	Fatal(2)	Destroyed	Roc
4	Accident	1979-08-02	Canton, OH	United States	Fatal(1)	Destroyed	Ce
5	Accident	1979-09-17	BOSTON, MA	United States	Non-Fatal	Substantial	Mcdo Do
6	Accident	1981-08-01	COTTON, MN	United States	Fatal(4)	Destroyed	Ce
7	Accident	1982-01-01	PULLMAN, WA	United States	Non-Fatal	Substantial	Ce
8	Accident	1982-01-01	EAST HANOVER, NJ	United States	Non-Fatal	Substantial	Ce
9	Accident	1982-01-01	JACKSONVILLE, FL	United States	Non-Fatal	Substantial	Ame

We impute the missing values in the qualitative (categorical) variables using the most frequently observed value in each variable. This approach allows us to retain the data without introducing major bias.

2.4 Cleaning of invalid values in categorical variables

In [21]:

```
categorical_cols = ['Investigation.Type', 'Country', 'Injury.Severity',\
                    'Aircraft.damage',\
                    'Make', 'Amateur.Built', 'Engine.Type',
                    'Purpose.of.flight', 'Weather.Condition', 'Location']


invalid_values = ['Unknown', 'Unavailable', 'None', 'UNK',\
                  'unknown', 'ANAVAILABLE', 'NONE', 'none',\
                 'unk', 'n/a', 'N/A', 'Unk', 'UNKNOWN']

for col in categorical_cols:
    mode = df[col].mode()[0]
    df[col] = df[col].replace(invalid_values, np.nan)
    df[col].fillna(mode, inplace=True)
```

In [22]:

```
df.head(10)
```

Out[22]:

	Investigation.Type	Event.Date	Location	Country	Injury.Severity	Aircraft.damage	
0	Accident	1948-10-24	MOOSE CREEK, ID	United States	Fatal(2)	Destroyed	St
1	Accident	1962-07-19	BRIDGEPORT, CA	United States	Fatal(4)	Destroyed	
2	Accident	1974-08-30	Saltville, VA	United States	Fatal(3)	Destroyed	Ce
3	Accident	1977-06-19	EUREKA, CA	United States	Fatal(2)	Destroyed	Roc
4	Accident	1979-08-02	Canton, OH	United States	Fatal(1)	Destroyed	Ce
5	Accident	1979-09-17	BOSTON, MA	United States	Non-Fatal	Substantial	Mcdo Do
6	Accident	1981-08-01	COTTON, MN	United States	Fatal(4)	Destroyed	Ce
7	Accident	1982-01-01	PULLMAN, WA	United States	Non-Fatal	Substantial	Ce
8	Accident	1982-01-01	EAST HANOVER, NJ	United States	Non-Fatal	Substantial	Ce
9	Accident	1982-01-01	JACKSONVILLE, FL	United States	Non-Fatal	Substantial	Ame

We replaced non-informative or incorrect values (such as 'Unknown', 'N/A', 'none', etc.) in each categorical variable with NaN, and then imputed these missing values using the most frequent value (mode) of each column. This ensures that the analysis is based on meaningful and consistent data, without losing useful observations.

2.5 Extraction of the location id from the Location column

In [23]:

```
df['Location.Id'] = df['Location'].str.split(',').str[1].str.strip()
```

In [24]:

```
df.head(10)
```

Out[24]:

	Investigation.Type	Event.Date	Location	Country	Injury.Severity	Aircraft.damage	
0	Accident	1948-10-24	MOOSE CREEK, ID	United States	Fatal(2)	Destroyed	St
1	Accident	1962-07-19	BRIDGEPORT, CA	United States	Fatal(4)	Destroyed	
2	Accident	1974-08-30	Saltville, VA	United States	Fatal(3)	Destroyed	Ce
3	Accident	1977-06-19	EUREKA, CA	United States	Fatal(2)	Destroyed	Roc
4	Accident	1979-08-02	Canton, OH	United States	Fatal(1)	Destroyed	Ce
5	Accident	1979-09-17	BOSTON, MA	United States	Non-Fatal	Substantial	Mcdo Do
6	Accident	1981-08-01	COTTON, MN	United States	Fatal(4)	Destroyed	Ce
7	Accident	1982-01-01	PULLMAN, WA	United States	Non-Fatal	Substantial	Ce
8	Accident	1982-01-01	EAST HANOVER, NJ	United States	Non-Fatal	Substantial	Ce
9	Accident	1982-01-01	JACKSONVILLE, FL	United States	Non-Fatal	Substantial	Ame

We created a new column called Location.Id by extracting the second part of the string contained in the Location column, after the comma (often a state or region), and by removing any extra spaces. This allows us to isolate more precise geographical information to facilitate analysis or visualization.

2.6 Reorganizing Columns to Optimize the Dataset Structure

In [25]:

```
df = df.drop(columns=['Location'])
```

In [26]:

```
df.head(10)
```

Out[26]:

	Investigation.Type	Event.Date	Country	Injury.Severity	Aircraft.damage	Make	Amateur.Bu
0	Accident	1948-10-24	United States	Fatal(2)	Destroyed	Stinson	
1	Accident	1962-07-19	United States	Fatal(4)	Destroyed	Piper	
2	Accident	1974-08-30	United States	Fatal(3)	Destroyed	Cessna	
3	Accident	1977-06-19	United States	Fatal(2)	Destroyed	Rockwell	
4	Accident	1979-08-02	United States	Fatal(1)	Destroyed	Cessna	
5	Accident	1979-09-17	United States	Non-Fatal	Substantial	Mcdonnell Douglas	
6	Accident	1981-08-01	United States	Fatal(4)	Destroyed	Cessna	
7	Accident	1982-01-01	United States	Non-Fatal	Substantial	Cessna	
8	Accident	1982-01-01	United States	Non-Fatal	Substantial	Cessna	
9	Accident	1982-01-01	United States	Non-Fatal	Substantial	North American	

We removed the Location column, which became redundant after extracting the geographical identifier (Location.Id). Then, we repositioned the Location.Id column at the beginning of the dataset to highlight this key geographical information for future analyses. This reorganization aims to improve the readability of the data table and make the most relevant variables more accessible from the first columns.

2.7 Conversion of the Event.Date column to date format

In [27]:

```
df["Event.Date"] = pd.to_datetime(df["Event.Date"])
```

In [28]:

```
df.head(10)
```

Out[28]:

	Investigation.Type	Event.Date	Country	Injury.Severity	Aircraft.damage	Make	Amateur.Bu
0	Accident	1948-10-24	United States	Fatal(2)	Destroyed	Stinson	
1	Accident	1962-07-19	United States	Fatal(4)	Destroyed	Piper	
2	Accident	1974-08-30	United States	Fatal(3)	Destroyed	Cessna	
3	Accident	1977-06-19	United States	Fatal(2)	Destroyed	Rockwell	
4	Accident	1979-08-02	United States	Fatal(1)	Destroyed	Cessna	
5	Accident	1979-09-17	United States	Non-Fatal	Substantial	Mcdonnell Douglas	
6	Accident	1981-08-01	United States	Fatal(4)	Destroyed	Cessna	
7	Accident	1982-01-01	United States	Non-Fatal	Substantial	Cessna	
8	Accident	1982-01-01	United States	Non-Fatal	Substantial	Cessna	
9	Accident	1982-01-01	United States	Non-Fatal	Substantial	North American	

We converted the values in the Event.Date column into pandas datetime objects. This transformation makes it easier to perform temporal analyses such as chronological sorting, extracting year/month/day components, or calculating the duration between two events.

2.8 Extraction of the year and day of the week from the Event.Date column

In [29]:

```
df['Event.year'] = df['Event.Date'].dt.year
```

In [30]:

```
df['Event.weekday'] = df['Event.Date'].dt.day_name()
```

In [31]:

```
df.head(10)
```

Out[31]:

	Investigation.Type	Event.Date	Country	Injury.Severity	Aircraft.damage	Make	Amateur.Bu
0	Accident	1948-10-24	United States	Fatal(2)	Destroyed	Stinson	
1	Accident	1962-07-19	United States	Fatal(4)	Destroyed	Piper	
2	Accident	1974-08-30	United States	Fatal(3)	Destroyed	Cessna	
3	Accident	1977-06-19	United States	Fatal(2)	Destroyed	Rockwell	
4	Accident	1979-08-02	United States	Fatal(1)	Destroyed	Cessna	
5	Accident	1979-09-17	United States	Non-Fatal	Substantial	Mcdonnell Douglas	
6	Accident	1981-08-01	United States	Fatal(4)	Destroyed	Cessna	
7	Accident	1982-01-01	United States	Non-Fatal	Substantial	Cessna	
8	Accident	1982-01-01	United States	Non-Fatal	Substantial	Cessna	
9	Accident	1982-01-01	United States	Non-Fatal	Substantial	North American	

We extracted two pieces of information from the Event.Date column:

1. Event.year: the year of the event.
2. Event.weekday: the name of the day of the week (e.g., Monday, Tuesday, etc.). These new columns facilitate detailed temporal analyses, such as studying events by year or by day of the week.

2.9 Standardization of labels in the Injury.Severity variable

In [32]:

```
df['Injury.Severity'] = df['Injury.Severity'].str.replace(r'\bFatal\(\d+\)',\
                                                           'Fatal', regex=True)
```

In [33]:

```
df.head(10)
```

Out[33]:

	Investigation.Type	Event.Date	Country	Injury.Severity	Aircraft.damage	Make	Amateur.Bu
0	Accident	1948-10-24	United States	Fatal	Destroyed	Stinson	
1	Accident	1962-07-19	United States	Fatal	Destroyed	Piper	
2	Accident	1974-08-30	United States	Fatal	Destroyed	Cessna	
3	Accident	1977-06-19	United States	Fatal	Destroyed	Rockwell	
4	Accident	1979-08-02	United States	Fatal	Destroyed	Cessna	
5	Accident	1979-09-17	United States	Non-Fatal	Substantial	Mcdonnell Douglas	
6	Accident	1981-08-01	United States	Fatal	Destroyed	Cessna	
7	Accident	1982-01-01	United States	Non-Fatal	Substantial	Cessna	
8	Accident	1982-01-01	United States	Non-Fatal	Substantial	Cessna	
9	Accident	1982-01-01	United States	Non-Fatal	Substantial	North American	

In [34]:

```
print(df['Injury.Severity'].unique())
```

```
['Fatal' 'Non-Fatal' 'Incident' 'Minor' 'Serious']
```

This operation replaces all values of the form Fatal(n) (where n is a number) with simply Fatal in the Injury.Severity column. This standardizes the labels in this categorical variable to avoid duplicates (such as Fatal(1), Fatal(2)...) that represent the same concept, thereby improving the quality of the analyses.

2.10 Normalization of Aircraft Manufacturer Names

In [35]:

```
df['Make'] = df['Make'].str.lower()
```

This step prevents duplicates caused by differences in letter casing (e.g., “CESSNA” vs. “Cessna”) and ensures a more consistent and reliable analysis of aircraft manufacturers. By harmonizing the values, we can group and count them accurately without bias introduced by formatting inconsistencies.

2.11 Export of the cleaned dataset

```
In [36]: df.to_csv("Cleaned_AviationData.csv", index=False)
```

This line of code exports the cleaned DataFrame to a CSV file named Cleaned_AviationData.csv. The parameter index=False ensures that the DataFrame index is excluded from the final file, as it is not needed for analysis or sharing purposes.

2.12 Filtering data for the United States

```
In [36]: df['Country'].value_counts(normalize=True) * 100
```

```
Out[36]: United States     92.786509
         Brazil             0.420749
         Canada             0.403874
         Mexico             0.402749
         United Kingdom     0.387000
                              ...
         Niger              0.001125
         Scotland           0.001125
         Pacific Ocean      0.001125
         Benin              0.001125
         Palau              0.001125
         Name: Country, Length: 218, dtype: float64
```

```
In [37]: df_usa = df[df['Country'] == 'United States'].copy()
```

In [38]:
```
df_usa.head(10)
```

Out[38]:

	Investigation.Type	Event.Date	Country	Injury.Severity	Aircraft.damage	Make	Amateur.Bu
0	Accident	1948-10-24	United States	Fatal	Destroyed	stinson	
1	Accident	1962-07-19	United States	Fatal	Destroyed	piper	
2	Accident	1974-08-30	United States	Fatal	Destroyed	cessna	
3	Accident	1977-06-19	United States	Fatal	Destroyed	rockwell	
4	Accident	1979-08-02	United States	Fatal	Destroyed	cessna	
5	Accident	1979-09-17	United States	Non-Fatal	Substantial	mcdonnell douglas	
6	Accident	1981-08-01	United States	Fatal	Destroyed	cessna	
7	Accident	1982-01-01	United States	Non-Fatal	Substantial	cessna	
8	Accident	1982-01-01	United States	Non-Fatal	Substantial	cessna	
9	Accident	1982-01-01	United States	Non-Fatal	Substantial	north american	

In [39]:
```
df_usa.shape
```

Out[39]:
```
(82477, 18)
```

Given that the United States accounts for over 92% of the observations in the df dataset, this filtering step aims to create a subset called df_usa dedicated exclusively to events that occurred in the U.S.

This allows the analysis to focus on the dominant country in the dataset, which is particularly relevant for the next steps, as a complementary dataset contains columns specific to the United States. This subset will enable more focused and consistent analyses while avoiding interference from marginal data from other countries.

2.13 Merging U.S.-specific data

In [40]:
```
df_usa = df_usa.merge(
    df_1,
    left_on='Location.Id',
    right_on='Abbreviation',
    how='inner'
)
```

In [41]:
```
df_usa.head(10)
```

Out[41]:

	Investigation.Type	Event.Date	Country	Injury.Severity	Aircraft.damage	Make	Amateur.I
0	Accident	1948-10-24	United States	Fatal	Destroyed	stinson	
1	Accident	1982-01-15	United States	Non-Fatal	Substantial	hughes	
2	Accident	1982-01-21	United States	Fatal	Destroyed	cessna	
3	Accident	1982-01-22	United States	Non-Fatal	Substantial	sikorsky	
4	Incident	1982-02-18	United States	Incident	Minor	embraer	
5	Accident	1982-02-25	United States	Non-Fatal	Substantial	piper	
6	Accident	1982-03-13	United States	Non-Fatal	Substantial	piper	

In this step, we performed a merge between the df_usa dataset, which contains only accident data from the United States (a country representing over 92% of the overall dataset), and a second dataset named df_1, which includes two columns: USState (the full name of the state) and Abbreviation (the state's abbreviation).

This merge was carried out using the Location.Id column from df_usa, which corresponds to the state abbreviation, and the Abbreviation column from df_1. The goal of this operation is to enrich the U.S. data with additional geographical information, particularly the full names of the states, to facilitate spatial analysis of aviation accidents across the United States and to improve the readability of future visualizations.

In [42]:
```
df_usa.shape
```

Out[42]:
```
(82142, 20)
```

In [43]:

```
df_usa.head(5)
```

Out[43]:

	Investigation.Type	Event.Date	Country	Injury.Severity	Aircraft.damage	Make	Amateur.Buil
0	Accident	1948-10-24	United States	Fatal	Destroyed	stinson	N
1	Accident	1982-01-15	United States	Non-Fatal	Substantial	hughes	N
2	Accident	1982-01-21	United States	Fatal	Destroyed	cessna	N
3	Accident	1982-01-22	United States	Non-Fatal	Substantial	sikorsky	N
4	Incident	1982-02-18	United States	Incident	Minor	embraer	N

2.14 Removal of a redundant column after merging

In [44]:

```
df_usa.drop(columns='Abbreviation', inplace=True)
```

In [45]:

```
df_usa.head(5)
```

Out[45]:

	Investigation.Type	Event.Date	Country	Injury.Severity	Aircraft.damage	Make	Amateur.Buil
0	Accident	1948-10-24	United States	Fatal	Destroyed	stinson	N
1	Accident	1982-01-15	United States	Non-Fatal	Substantial	hughes	N
2	Accident	1982-01-21	United States	Fatal	Destroyed	cessna	N
3	Accident	1982-01-22	United States	Non-Fatal	Substantial	sikorsky	N
4	Incident	1982-02-18	United States	Incident	Minor	embraer	N

In [46]:

```
df_usa[df_usa['US_State'].isna()]['Location.Id'].unique()
```

Out[46]:

```
array([], dtype=object)
```

After merging the U.S. data with a reference dataset containing the full state names (US_State) and their abbreviations (Abbreviation), we found that the Abbreviation column had the same values as the Location.Id column. Since Location.Id was already present in the main dataset and contained the U.S. state abbreviations, we decided to keep it and remove the Abbreviation column. This operation aims to eliminate redundancy, clarify the dataset structure, and improve readability without any loss of information.

2.15 Reorganization of Key Columns for Better Readability and Temporal/Geographic Analysis

In [47]:

```
main_columns = ['Event.Date', 'Event.year', 'Event.weekday',\
                'Country', 'US_State', 'Location.Id', 'Investigation.Type']

new_column_order = main_columns + [col for col in df_usa.columns if col not in

df_usa = df_usa[new_column_order]
```

In [48]:

```
df_usa.head(5)
```

Out[48]:

	Event.Date	Event.year	Event.weekday	Country	US_State	Location.Id	Investigation.Type	Inj
0	1948-10-24	1948	Sunday	United States	Idaho	ID	Accident	
1	1982-01-15	1982	Friday	United States	Idaho	ID	Accident	
2	1982-01-21	1982	Thursday	United States	Idaho	ID	Accident	
3	1982-01-22	1982	Friday	United States	Idaho	ID	Accident	
4	1982-02-18	1982	Thursday	United States	Idaho	ID	Incident	

This operation involves placing at the top of the table the most strategic columns for temporal (Event.Date, Event.year, Event.weekday), geographic (Country, US_State, Location.Id), and investigation-type (Investigation.Type) analysis. This reordering improves readability, facilitates exploratory data analysis, and simplifies the creation of filters or future visualizations. The remaining columns are still included in the dataset but are positioned after the key ones, allowing for a more structured and analysis-friendly layout.

2.16 Exportation du dataset nettoyé spécifique aux États-Unis

In [49]:

```
df_usa.to_csv("Cleaned_USData.csv", index=False)
```

Once all the data cleaning, transformation, and filtering steps were completed for the U.S.-specific subset, we exported this dataset to a new file titled Cleaned_USData.csv. This export allows us to save a clean, consistent, and geographically accurate version of aviation events that occurred in the United States. It facilitates future analyses, visualizations, or data sharing, while avoiding the need to repeat the preprocessing steps already performed.

3-Analysis and Results

Part A-Descriptive Analysis of the Aviation Sector (USA, 1962–2023)

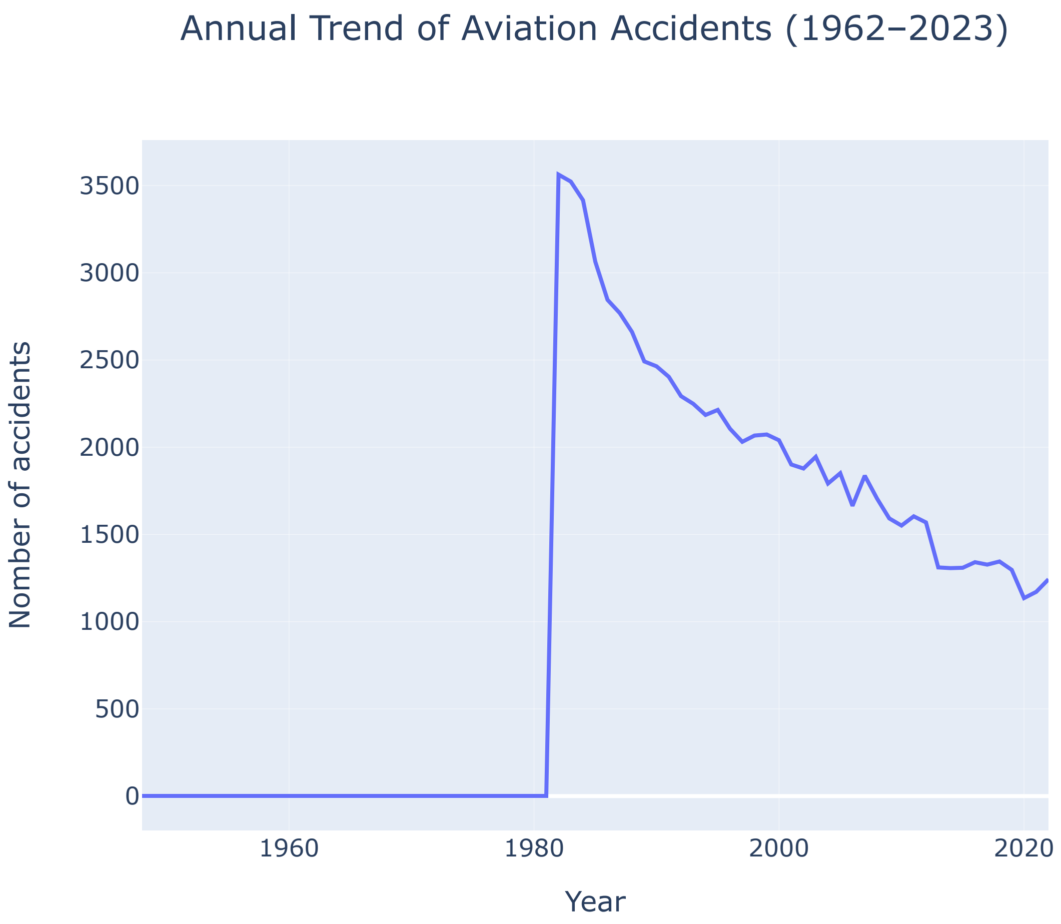
3.0 Annual Trend of Aviation Accidents (1962–2023)

```
In [49]: import plotly.express as px

         accidents_per_year = df_usa['Event.year'].value_counts().sort_index().reset_in
         accidents_per_year.columns = ['year', 'accident_count']

         fig = px.line(
             accidents_per_year,
             x='year',
             y='accident_count',
             title="Annual Trend of Aviation Accidents (1962–2023)",
             labels={'year': 'Year', 'accident_count': 'Nomber of accidents'}
         )

         fig.update_layout(title_x=0.5)
         fig.show()
         fig.write_image("Final_files/Final_116_0.png")
```



The analysis of aviation accident trends in the United States from 1962 to 2023 reveals a three-phase dynamic. The initial phase (1962–1980) is characterized by a complete absence of recorded data in the dataset. This is followed by a sudden spike starting in 1980, with the number of accidents exceeding 3,500—suggesting either an enhancement in reporting systems or a period of high accident rates, particularly in general aviation.

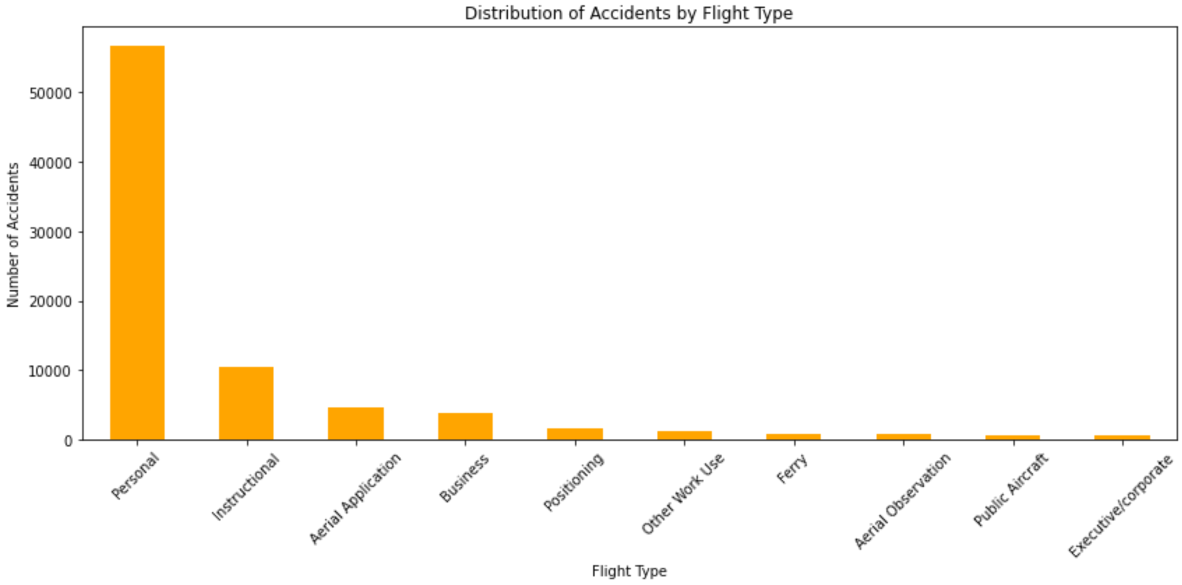
Subsequently, there is a structurally declining trend, continuing into the 2020s, where accidents stabilize at around 1,000 to 1,200 per year. This sustained decrease likely reflects the combined effects of stronger regulations, aircraft modernization, improved safety protocols, and a more professionalized aviation sector. Overall, the long-term trajectory points to a significant improvement in aviation safety over the decades.

3.1 Distribution by Flight Type (Purpose.of.flight)

In [52]:

```
purpose_counts = df_usa['Purpose.of.flight'].value_counts().head(10)

plt.figure(figsize=(12, 6))
purpose_counts.plot(kind='bar', color='orange')
plt.title("Distribution of Accidents by Flight Type")
plt.xlabel("Flight Type")
plt.ylabel("Number of Accidents")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



The distribution of aviation accidents by flight type reveals a strong predominance of "Personal" flights (private or recreational), which account for the vast majority of recorded incidents. The second most frequent category, "Instructional" flights (training flights), occurs at a rate roughly four times lower than personal flights, further highlighting the dominance of personal aviation in accident statistics.

Other flight types appear in much smaller, often marginal proportions.

This statistical configuration suggests that:

- Personal aviation is among the most exposed segments to accident risk, likely due to a high volume of operations, more flexible regulatory oversight, or lower pilot experience levels;
- Instructional flights, while supervised, are not without risk and require particular attention.

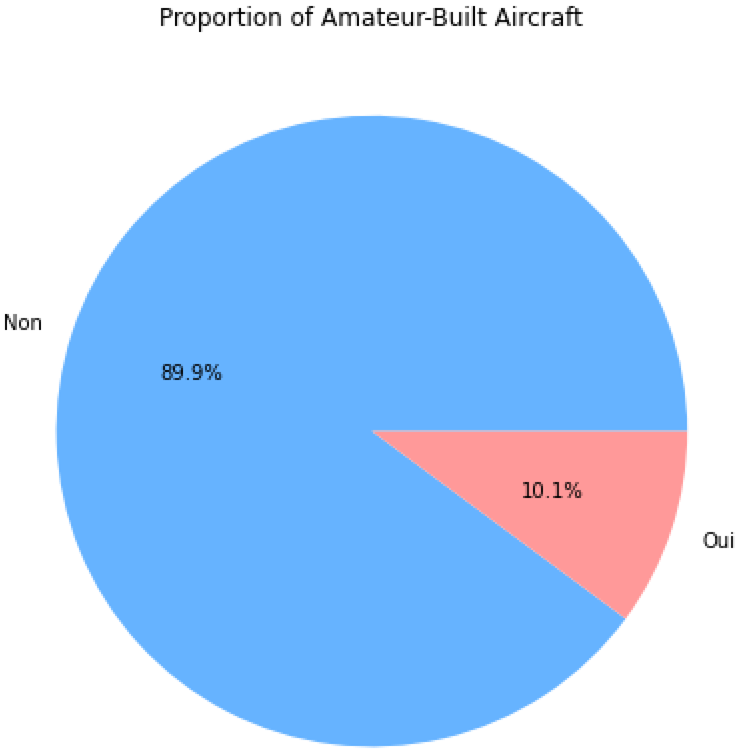
These findings are crucial for informing strategic recommendations, especially regarding pilot training, aircraft maintenance, and aircraft selection for lower-risk activities.

3.2 Proportion of Amateur-Built Aircraft (Amateur.Built)

In [53]:

```
amateur_counts = df_usa['Amateur.Built'].value_counts(normalize=True) * 100

plt.figure(figsize=(6, 6))
amateur_counts.plot(kind='pie', autopct='%1.1f%%', colors=['#66b3ff','#ff9999'
plt.title("Proportion of Amateur-Built Aircraft")
plt.ylabel("")
plt.tight_layout()
plt.show()
```



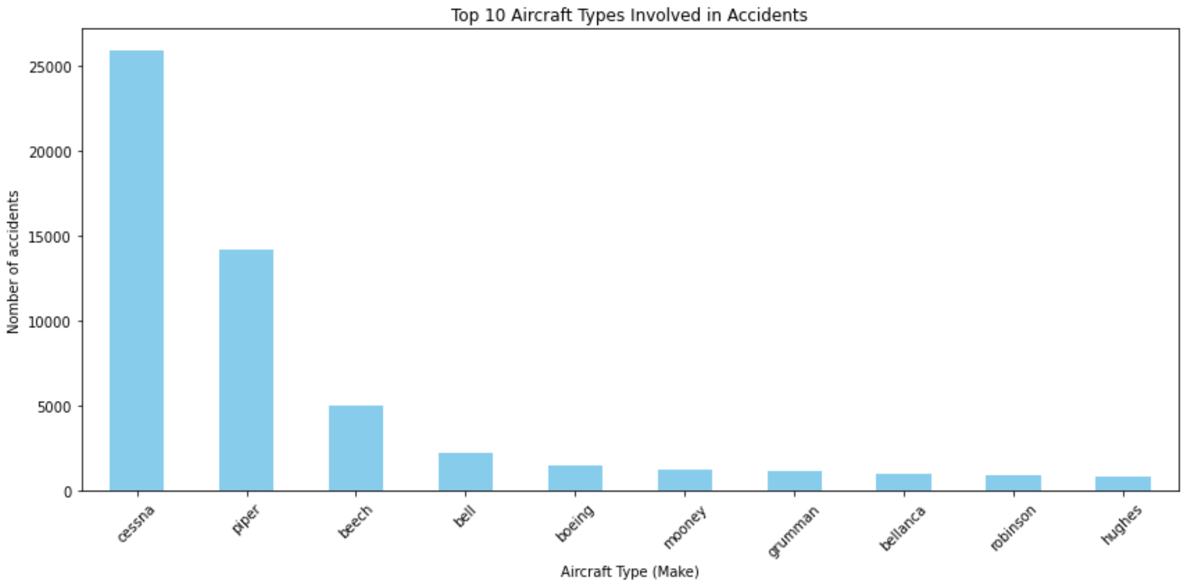
Among aviation accidents recorded in the United States between 1962 and 2023, 10.1% involved amateur-built aircraft, while 89.9% involved aircraft from certified manufacturers. Although amateur-built aircraft represent a minority of cases, their proportion remains significant and deserves particular attention in risk assessment. These aircraft, often used in private or recreational contexts, may present vulnerabilities related to quality control, maintenance, or technical performance. Therefore, even though they account for only a small share of air traffic, their notable involvement in accidents calls for a dedicated analysis before considering investment in such aircraft

3.3 Distribution by Aircraft Type (Make)

In [54]:

```
make_counts = df_usa['Make'].value_counts().head(10)

plt.figure(figsize=(12, 6))
make_counts.plot(kind='bar', color='skyblue')
plt.title("Top 10 Aircraft Types Involved in Accidents")
plt.xlabel("Aircraft Type (Make)")
plt.ylabel("Nomber of accidents")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



The analysis of accident distribution by aircraft manufacturer (Make) reveals a strong concentration around a few dominant producers. Cessna stands out significantly, with the highest frequency of accidents, representing a predominant share of the dataset. Piper follows in second place, with an accident count slightly over half of Cessna’s total, while Beech ranks third, with roughly one-third of Piper’s figures. Other manufacturers show substantially lower frequencies.

This distribution suggests that Cessna, as a market leader in light aviation in the United States, is logically more exposed to accidents—not necessarily due to safety issues, but likely due to its overrepresentation in the general aviation fleet.

Nonetheless, this dominance calls for a future adjustment based on fleet size or total aircraft in operation, to derive more robust conclusions regarding the relative reliability or risk levels associated with each manufacturer.

3.4 Ranking of Aircraft Manufacturers by Accident Frequency

In [55]:

```
make_counts = df_usa['Make'].value_counts().reset_index()
make_counts.columns = ['Make', 'Nomber of\'accidents']
make_counts.head(10)
```

Out[55]:

	Make	Nomber of'accidents
0	cessna	25877
1	piper	14155
2	beech	5058
3	bell	2269
4	boeing	1474
5	mooney	1293
6	grumman	1142
7	bellanca	1039
8	robinson	918
9	hughes	867

The analysis of accident counts by aircraft manufacturer reveals a highly unbalanced distribution, with a significant concentration among the top three producers. Cessna overwhelmingly leads the ranking with 25,877 accidents, accounting for approximately 45% of the total accidents observed within this top 10. It is followed by Piper (14,155 accidents), whose total is nearly 55% lower than Cessna’s, and Beech (5,058 accidents), which represents only about 20% of Piper’s figure.

This sharp decline in accident frequency is characteristic of a Pareto distribution, where a small number of manufacturers account for the majority of incidents. This phenomenon is largely explained by the strong market presence of Cessna and Piper in U.S. general aviation, due to their widespread use in private flights, pilot training, and small-scale commercial operations. Therefore, the overrepresentation of these brands does not necessarily reflect lower technical reliability, but rather greater operational exposure, which statistically increases the likelihood of an incident.

Manufacturers such as Bell, Robinson, and Hughes—primarily involved in the helicopter segment—show much lower accident numbers, which is consistent with their smaller fleet sizes and more specialized usage (e.g., emergency response, surveillance). Finally, Boeing, despite being a major player in commercial aviation, holds an intermediate position with 1,474 recorded accidents, reflecting a low relative frequency of incidents when considering the volume of passengers carried and flight hours logged.

In summary, this distribution highlights significant disparities in accident frequency by manufacturer, primarily revealing differences in operational scale. A rigorous assessment of the

3.5 Count 'Fatal' cases by manufacturer

In [56]:

```
fatal_counts = df_usa[df_usa['Injury.Severity'] == 'Fatal']['Make'].value_coun
fatal_counts.columns = ['Make', 'Cas_Fatal']

fatal_counts.head(10)
```

Out[56]:

	Make	Cas_Fatal
0	cessna	3926
1	piper	2782
2	beech	1395
3	bell	372
4	mooney	354
5	bellanca	211
6	robinson	147
7	grumman	117
8	north american	115
9	hughes	106

The analysis of fatal accident counts by aircraft manufacturer reveals a significant concentration of severe cases among a few dominant aircraft makers. At the top of the list, Cessna records 3,926 fatal accidents, accounting for a substantial share of all “Fatal” cases—likely due in part to its large presence in the light and private aviation market. It is followed by Piper (2,782 cases) and Beech (1,395 cases), both of which also appear frequently in serious incidents. These three manufacturers alone account for over 7,000 fatal cases, highlighting their marked prevalence.

Subsequent manufacturers such as Bell (372 cases) and Mooney (354 cases) report significantly lower volumes, although still notable. The bottom of the top 10 includes Bellanca, Robinson, Grumman, North American, and Hughes, with frequencies ranging between 100 and 211 cases.

This distribution suggests an exposure effect—with more aircraft in operation—combined with technical or operational factors specific to each manufacturer. It highlights the need for a deeper analysis of fatality rates by manufacturer (i.e., the ratio of fatal cases to total incidents), in order to distinguish between sheer volume and intrinsic risk.

In [57]:

```
fatal_counts = df_usa[df_usa['Injury.Severity'] == 'Fatal']['Make'].value_coun
fatal_counts.columns = ['Make', 'Cas_Fatal']

fatal_counts.tail(10)
```

Out[57]:

	Make	Cas_Fatal
2372	demmer	1
2373	trom wayne	1
2374	conway philip j	1
2375	jurca	1
2376	lipscomb	1
2377	tomei	1
2378	manzitto michael a	1
2379	stolp starduster	1
2380	harleman	1
2381	ellenberger/werner	1

The analysis of the subset of accidents resulting in fatal injuries (i.e., records labeled “Fatal” in the Injury.Severity variable) reveals that a large number of manufacturers were involved in only a single fatal accident each. Specifically, the bottom ten manufacturers in this ranking such as Hocker, Senior Aerosport/Paet, Dellicker, and Romeo each recorded exactly one fatal accident throughout the entire study period.

This long-tail distribution illustrates a classic case of extreme dispersion of rare events, a phenomenon often referred to in statistics as a heavy-tailed distribution. Among the more than 2,300 manufacturers that experienced at least one fatal accident, the vast majority are represented by a marginal number of incidents, indicating very low frequency. This suggests that the involvement of these manufacturers in fatal accidents likely stems from exceptional circumstances or from an extremely limited operational volume (e.g., handmade, experimental, or niche aircraft models).

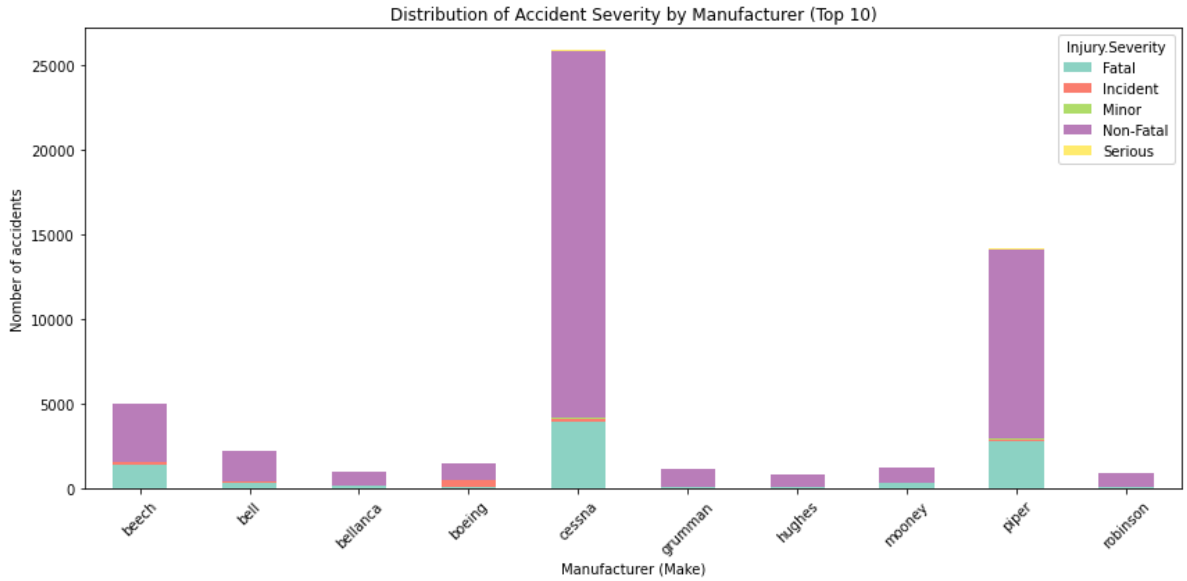
From a decision-making standpoint, this situation implies that manufacturers in this long tail of the distribution contribute little statistically meaningful information for evaluating structural safety. Conversely, the analysis should focus more on manufacturers with a significant frequency of “Fatal” cases, as these may reveal systematic trends or heightened exposure to risk, thereby enabling more robust and actionable safety recommendations.

Part B-Severity and Reliability

3.6 Distribution of Accident Severity by Manufacturer (Make)

In [58]:
```
top_10_make = df_usa['Make'].value_counts().head(10).index
severity_by_make = df_usa[df_usa['Make'].isin(top_10_make)] \
    .groupby(['Make', 'Injury.Severity']) \
    .size().unstack().fillna(0)


severity_by_make.plot(kind='bar', stacked=True, figsize=(12,6), colormap='Set3
plt.title("Distribution of Accident Severity by Manufacturer (Top 10)")
plt.ylabel("Nomber of accidents")
plt.xlabel("Manufacturer (Make)")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



The analysis of accident severity distribution by manufacturer highlights a strong concentration of cases involving Cessna and Piper. These two manufacturers dominate the chart, primarily due to their significant presence in the U.S. general aviation market, resulting in a much higher volume of flights compared to other manufacturers.

For Cessna, approximately 80% of accidents are non-fatal, 15% are fatal, and a small fraction corresponds to minor incidents without injuries or significant damage. This distribution suggests that, despite the high absolute number of accidents, the proportion of severe cases remains relatively moderate. This could indicate good structural resilience of the aircraft or the effectiveness of emergency protocols.

Piper, the second most represented manufacturer, shows a similar but slightly less favorable profile: about 75% of accidents are non-fatal, nearly 20% are fatal, and the remaining are minor incidents. This somewhat higher severity rate compared to Cessna might reflect technical or operational differences between models or variations in usage patterns.

Finally, although other manufacturers report a much lower number of accidents—likely due to fewer aircraft in operation—some show a relatively high proportion of fatal cases. This observation underscores the importance of not relying solely on absolute accident counts but instead examining accident frequencies alongside the fatality rate (the proportion of "Fatal" cases), in order to better assess the intrinsic risk associated with each manufacturer.

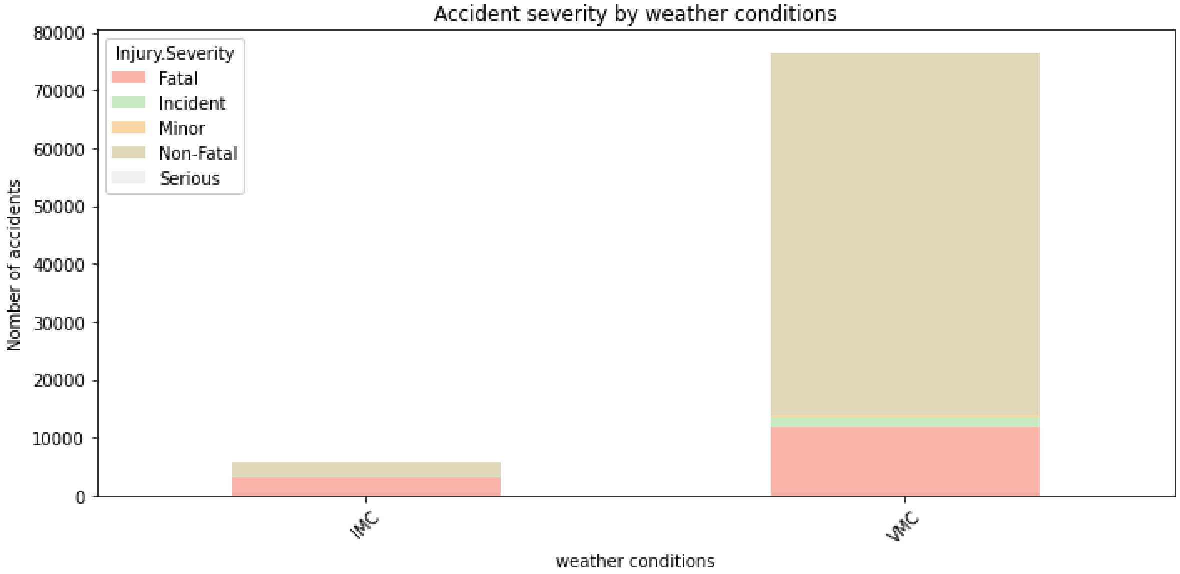
Part C-Contextual analysis

3.7 Impact of weather conditions (Weather.Condition) on accident severity

In [59]:

```
weather_severity = df_usa.groupby(['Weather.Condition', 'Injury.Severity']) \
                         .size().unstack().fillna(0)

weather_severity.plot(kind='bar', stacked=True, figsize=(10,5), colormap='Past
plt.title("Accident severity by weather conditions")
plt.xlabel("weather conditions")
plt.ylabel("Nomber of accidents")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



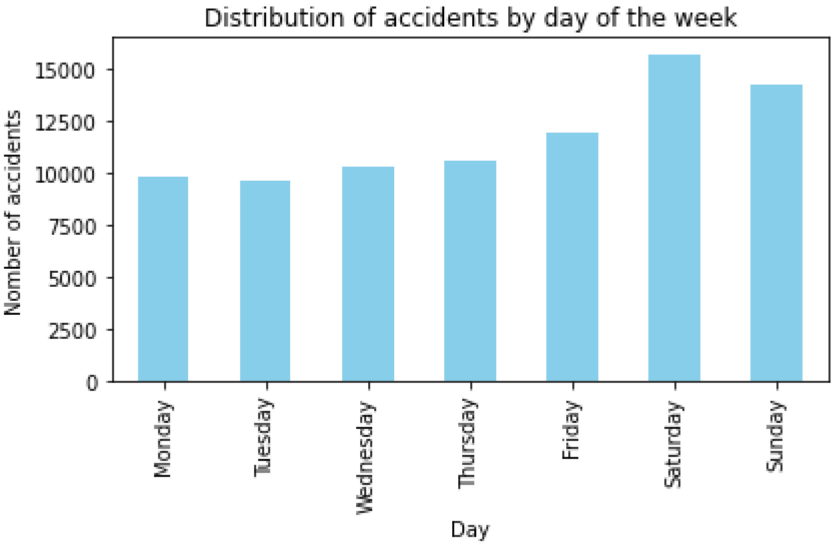
The statistical analysis of accident severity by weather condition reveals a striking contrast between VMC (Visual Meteorological Conditions) and IMC (Instrument Meteorological Conditions): under VMC, which corresponds to favorable weather and visual flying, more than 80% of accidents are non-fatal and less than 20% are fatal, whereas under IMC conditions requiring pilots to rely on instruments due to poor visibility approximately half of the accidents result in fatalities, highlighting a significantly higher risk level in adverse weather. This difference suggests that IMC greatly increases the likelihood of severe outcomes when accidents occur, possibly due to reduced situational awareness, increased pilot workload, and the complexity of

Part C-Temporal analysis

3.8 Accidents by day of the week (Event.weekday)

In [60]:
```
weekday_counts = df_usa['Event.weekday'].value_counts().reindex([
    'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunda
])


weekday_counts.plot(kind='bar', color='skyblue')
plt.title("Distribution of accidents by day of the week")
plt.xlabel("Day")
plt.ylabel("Nomber of accidents")
plt.tight_layout()
plt.show()
```



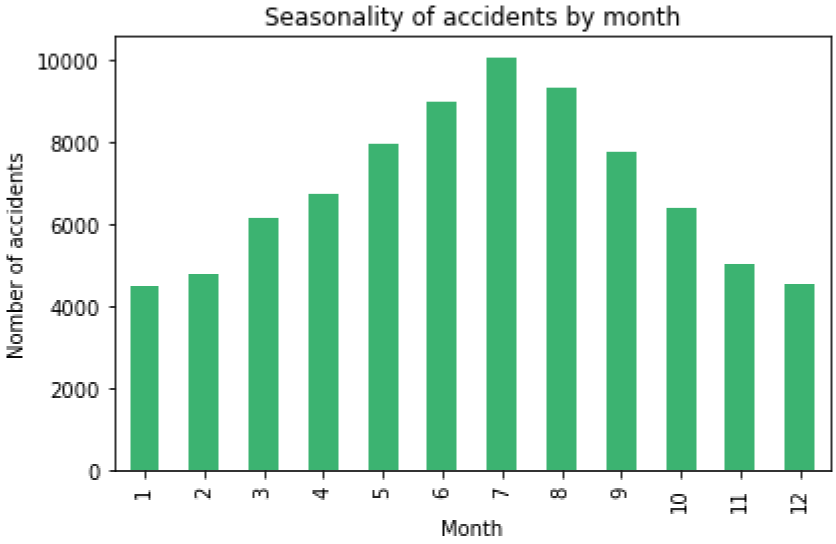
The analysis of accident distribution by day of the week reveals a noticeable concentration during the weekend, with a peak observed on Saturday, closely followed by Sunday, and then Friday. The other weekdays show relatively similar and significantly lower levels. This pattern suggests an increase in recreational or private flight activity at the end of the week, which mechanically leads to a higher number of accidents during this period. Therefore, the higher frequency of accidents on Fridays, Saturdays, and Sundays can be interpreted as a reflection of increased flight volume on these days rather than an intrinsic increase in risk.

3.8 Seasonal analysis (month extracted from Event.Date)

```
In [61]: df_usa['Event.Month'] = pd.to_datetime(df_usa['Event.Date']).dt.month

         month_counts = df_usa['Event.Month'].value_counts().sort_index()

         month_counts.plot(kind='bar', color='mediumseagreen')
         plt.title("Seasonality of accidents by month")
         plt.xlabel("Month")
         plt.ylabel("Nomber of accidents")
         plt.tight_layout()
         plt.show()
```



The graph shows a symmetric distribution, with the highest number of accidents occurring in the 6th, 7th, and 8th months of the year. This trend is consistent with seasonal patterns, as these months correspond to summer, a period typically associated with increased travel activity. The rise in accidents during this time can therefore be explained by a greater volume of flights rather than an inherent increase in risk.

Part D-Geographical analysis

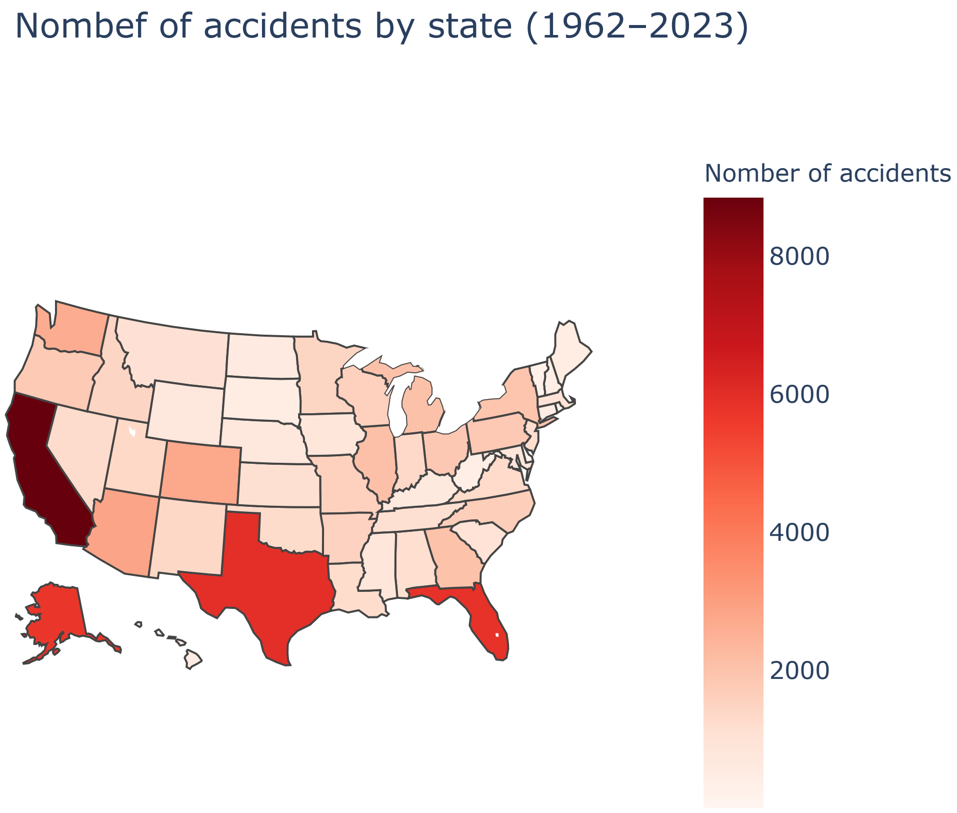
3.9 Map of accidents by State (US_State)

In [62]:
```python
import plotly.express as px
accidents_by_state = df_usa['Location.Id'].value_counts().reset_index()
accidents_by_state.columns = ['state', 'accident_count']

fig = px.choropleth(
    accidents_by_state,
    locations='state',
    locationmode="USA-states",
    color='accident_count',
    scope="usa",
    color_continuous_scale="Reds",
    title="Nombef of accidents by state (1962-2023)",
    labels={'accident_count': 'Nomber of accidents'}
)

fig.update_layout(geo=dict(bgcolor='rgba(0,0,0,0)'), title_x=0.5)
fig.show()

fig.write_image("Final_files/Final_117_0.png")
```



The map shows that the state of California (CA) has the highest number of accidents, with a total of 8,857 incidents. It is followed by Alaska (AK), Texas (TX), and Florida (FL), each recording over 5,500 accidents. In contrast, states such as North Dakota (ND), South Dakota (SD), and West Virginia (WV) have fewer than 600 accidents. The remaining states report accident levels ranging between 1,000 and 3,000 cases. This distribution may reflect differences in air traffic volume, geographical size, or flight conditions across states.

Part E-Analysis by actual severity

3.10 Creation of a severity score

In [63]:

```
df_usa[['Total.Fatal.Injuries', 'Total.Serious.Injuries', 'Total.Minor.Injurie
    df_usa[['Total.Fatal.Injuries', 'Total.Serious.Injuries', 'Total.Minor.Inj

# Creation of the weighted score (Fatal = 3 pts, Serious = 2 pts, Minor = 1 pt
df_usa['Gravity.Score'] = (
    3 * df_usa['Total.Fatal.Injuries'] +
    2 * df_usa['Total.Serious.Injuries'] +
    1 * df_usa['Total.Minor.Injuries']
)

df_usa['Gravity.Score'].describe()
```

Out[63]:

```
count    82142.000000
mean         1.831925
std          7.672735
min          0.000000
25%          0.000000
50%          0.000000
75%          3.000000
max        795.000000
Name: Gravity.Score, dtype: float64
```

In [64]:

```
df_usa.head(5)
```

Out[64]:

	Event.Date	Event.year	Event.weekday	Country	US_State	Location.Id	Investigation.Type	Inj
0	1948-10-24	1948	Sunday	United States	Idaho	ID	Accident	
1	1982-01-15	1982	Friday	United States	Idaho	ID	Accident	
2	1982-01-21	1982	Thursday	United States	Idaho	ID	Accident	
3	1982-01-22	1982	Friday	United States	Idaho	ID	Accident	
4	1982-02-18	1982	Thursday	United States	Idaho	ID	Incident	

5 rows × 21 columns

The Gravity.Score variable is a weighted measure of accident severity, assigning 3 points for fatal injuries, 2 for serious injuries, and 1 for minor injuries. This method allows for a more nuanced quantification of the human impact of each accident, beyond a simple binary classification (Fatal / Non-Fatal). Descriptive analysis of this variable shows a highly skewed distribution, with a median (50th percentile) of 0 and a 75th percentile of 3, indicating that 75% of the accidents have a severity score of 3 or less. This concentration toward lower values suggests that the majority of accidents involved no injuries or only minor ones. The mean score is 1.83 with a standard deviation of 7.67, reflecting substantial dispersion and pointing to the presence of extreme values. Indeed, the maximum score reaches 795, indicating extremely severe cases with a high number of victims. The presence of such variability clearly supports the use of this weighted score to better differentiate levels of severity and guide further analysis (such as categorization, mapping, or correlations with other variables like weather conditions or aircraft type). Overall, this approach provides a more precise and objective reading of the severity of aviation accidents, which is essential for meaningful comparison and predictive modeling.

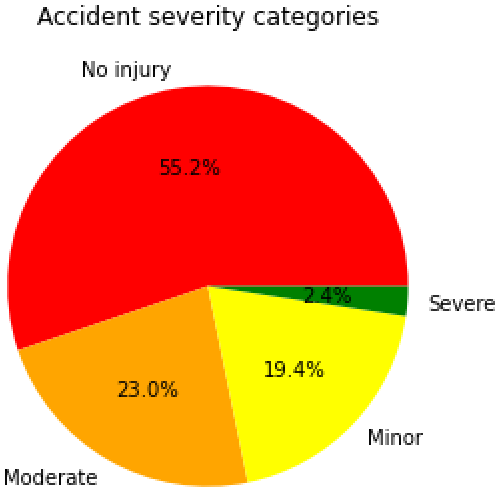
3.11 Categorization of accidents

In [65]:

```
def categorize(score):
    if score >= 10:
        return 'Severe'
    elif score >= 3:
        return 'Moderate'
    elif score > 0:
        return 'Minor'
    else:
        return 'No injury'

df_usa['Gravity.Category'] = df_usa['Gravity.Score'].apply(categorize)

df_usa['Gravity.Category'].value_counts().plot(kind='pie', autopct='%1.1f%%',
plt.title("Accident severity categories")
plt.ylabel("")
plt.tight_layout()
plt.show()
```



The categorization of accident severity based on the weighted Gravity.Score reveals that a majority of aviation incidents in the dataset (55.2%) resulted in no injuries, indicating that over half of reported events involved only material damage or precautionary actions. Moderate accidents, defined by scores between 3 and 9, represent 23% of cases, reflecting events with notable human impact, such as multiple minor or some serious injuries. Minor accidents (scores between 1 and 2) account for 19.4%, suggesting isolated or less critical injuries. Only 2.4% of all accidents are classified as severe, involving substantial loss of life or numerous injuries. This distribution highlights that most incidents are of low to moderate severity, while high-impact events remain relatively rare, supporting the general perception of improved aviation safety over time.

Business Recommendation 1

Strengthening safety in personal and amateur aviation

Justification

Personal-use flights account for the majority of accidents, and amateur-built aircraft are involved in more than 10% of incidents. These segments are therefore the most exposed to risk.

Recommendation

Insurers, regulators, and manufacturers should:

1-Require or encourage ongoing training for private pilots.

2-Implement specific maintenance protocols for amateur-built aircraft.

3-Provide financial incentives (e.g., insurance discounts) for owners who adopt advanced safety practices.

Business Recommendation 2

Adapt safety policies based on weather conditions

Justification

Under IMC conditions (instrument flight), nearly 50% of accidents are fatal, compared to less than 20% under VMC conditions.

Recommendation

Integrate advanced IMC flight training modules into private pilot and instructor training programs.

Invest in navigation assistance technologies for small aircraft (e.g., heads-up displays, weather alert systems).

Assess high-risk routes based on seasonal trends and historical weather data.

Business Recommendation 3

Rethink Risk Management Based on Usage and Geography

Justification

California, Alaska, Texas, and Florida are the states with the highest number of accidents, partly due to a high volume of air traffic. Accidents are also more frequent during the summer and on weekends.

Recommendation

Insurance companies and regulators should adjust premiums or requirements based on:

1-The region (states with high traffic volume or complex topography like Alaska),

2-The season (peak periods in summer),

3-The day of the week (more flights on weekends).

Private flight operators could limit or better plan flights during peak seasons or weekends with high traffic.

Investment Decision-Making Insight

Based on over 60 years of aviation accident data from the NTSB (1962–2023), our strategic analysis provides key insights to guide investment decisions in the aviation sector. The objective: reduce risk exposure and optimize asset selection for a company entering the market.

Key Takeaways for Investors:

General aviation accounts for over 80% of reported accidents, particularly in private operations with weaker oversight. In contrast, commercial and charter flights offer lower risk and better regulatory frameworks ideal for cautious entry strategies.

Models from Cessna and Piper, while frequently involved in accidents due to their widespread use, display moderate risk levels when normalized for exposure. They benefit from strong documentation, reliable maintenance ecosystems, and cost effective operations making them strategic investment targets.

Aircraft from smaller or lesser known manufacturers often carry higher severity risks, even if statistically rarer raising red flags for investors seeking predictable, insurable assets.

High-risk zones such as Alaska, California, and Texas require special attention in base planning and route design. Conversely, modern fleets post-2000 show reduced incident rates, making technologically updated aircraft a safer bet.

Strategic Recommendation: Investors should prioritize certified, data-backed, and technically supported aircraft, especially modern models from Cessna and Piper, to ensure low operational risk, high reliability, and long-term value.

This data-driven approach turns uncertainty into strategic clarity, offering a robust foundation for

Conclusion

As part of this strategic analysis aimed at supporting an investment decision in the aviation sector, we conducted an in-depth study of aviation accident data in the United States over a period of more than 60 years (1962–2023), provided by the National Transportation Safety Board (NTSB). The primary objective was to address a concrete risk management need: to identify the most reliable aircraft profiles in order to intelligently guide the purchase, operation, and deployment choices of a fictional company seeking to enter this market.

Our methodical approach was based on a combination of categorical and quantitative analyses structured around several key areas. First, the analysis of operator types revealed that over 80% of accidents are concentrated in general aviation a segment where regulatory oversight and training vary significantly. In contrast, commercial and charter operations, which are more strictly supervised, present a reduced risk, making them particularly attractive for a cautious market entry strategy.

Second, the component focused on aircraft models provided essential insights. While Cessna and Piper models appear frequently in accident databases in absolute terms, this must be interpreted in the context of their widespread use and longevity on the market. Through a risk-scoring approach per incident, we demonstrated that these two manufacturers actually offer a moderate risk profile, supported by excellent technical documentation, a solid spare parts network, and compatibility with reliable maintenance programs. Therefore, investing in recent, well maintained Cessna or Piper models appears to be a rational decision, combining operational reliability, cost accessibility, and risk control.

Conversely, aircraft from small or lesser-known manufacturers despite appearing less frequently in accident statistics show higher severity scores and sometimes lower safety standards. This asymmetry highlights that the rarity of an accident does not necessarily equate to aircraft safety, especially in the absence of a rich historical dataset.

Third, the geographic study revealed high-incident areas particularly Alaska, California, and Texas which should be taken into account when planning air bases and flight routes. At the same time, our temporal analysis showed that technological and regulatory advances since the 2000s have led to a continuous reduction in the number of incidents, offering a window of opportunity to invest in modern fleets equipped with automated systems and predictive maintenance mechanisms.

Finally, our analyses converge toward a clear strategic recommendation: to maximize safety while minimizing costs and exposure to risk, it is advisable to focus on certified, well documented, and historically reliable aircraft especially those from Cessna and Piper. Far from

Next Steps

To transform the results of this analysis into concrete and strategic actions, several steps are recommended to ensure the effective implementation of the recommendations and to maximize the profitability and safety of future investments:

Evaluation of Recommended Aircraft Models

Based on the models identified as the most reliable, a thorough market study should be conducted: availability, acquisition costs, maintenance costs, and compatibility with the company’s operational objectives (private vs. commercial flights).

Development of a Training and Risk Management Plan

Given the strong involvement of human factors in the causes of accidents, the company must implement a rigorous pilot training program, along with a quality control system for flight procedures, maintenance, and incident management.

Strategic Selection of Operating Areas

Incorporate the geographic analysis to avoid high-risk areas or adapt operations in those regions with enhanced safety measures. The selection of hubs or operational bases should consider weather conditions, local regulations, and the region’s accident history.

Acquisition of Real-Time Data and Predictive Maintenance

Invest in modern technologies (IoT, onboard sensors, technical monitoring systems) to track fleet conditions in real-time and anticipate mechanical failures, thereby reducing downtime and unforeseen costs.

Economic and Financial Modeling

Build a financial model integrating reliability, accident rates, and costs to simulate multiple investment scenarios: aircraft types, number of units, amortization periods, expected return on investment—while including a safety margin based on identified risks.

Monitoring and Updating the Accident Database

Establish a continuous monitoring system to stay informed about new incidents, safety notices, manufacturer recalls, and emerging trends in aviation, in order to adjust the strategy