Aviation Safety Risk Analysis for Aircraft Procurement

Author: Reeves Gonah
Date:
Project: Phase 1 End Project
Data Source: Aviation Safety Network (Scraped)

Introduction.

As the organization considers expanding into aerospace operations, a clear understanding of patterns in accident frequency and severity is critical for minimizing operational risk.This project analyzes historical aviation accident data from the National Transportation Safety Board (NTSB) to support data-driven aircraft procurement decisions.
Through exploratory data analysis and visualization, this notebook identifies aircraft characteristics and operational factors associated with safer outcomes. The findings are used to develop practical recommendations to guide aircraft selection and risk management strategy.

The CRISP-DM method, a structured, six-phase framework, has been appropriately selected to guide data analysis being carried out as follows below for the entirety of the project, each phase numbered and highlighted as concisely as possible.

1. Business Understanding

Business Context

The company is considering entry into the aviation market by purchasing and operating aircraft. To support procurement decisions, the aim is to analyze historical aviation accident data to identify aircraft that are associated with lower operational risk.

Business Objective

Identify aircraft that are associated with lower accident risk and severity, or good durability, in order to guide safer aircraft acquisition and operational planning.

Key Business Questions

The following are the key questions this analysis attempts to answer as they would directly inform procurement decisions:

- How has accident risk changed over time?
- Which aircraft or aircraft types are more prone to accidents?
- Which aircraft provide the most safety and are the most durable in accidents?

Success Criteria

The aforementioned questions will be considered as well answered by three data-backed procurement recommendations with visual/graphical evidence as support.

2. Data Understanding

Data Source and Description.

This dataset has been scrapped from the National Transportation Safety Board that includes aviation accident data from 1962 to 2023 about civil aviation accidents and selected incidents in the United States and

international waters. It has been downloaded from this kaggle repository as a csv file, published by Aditya Kushawa 2025* and saved as 'flight.csv' in this project's directory.

**latest update by date of download*

Initial Data Load

Preliminary analysis on existing data to identify the form and metrics of our data is necessary

In [1]:
```
# import necessary libraries likely to be used

import pandas as pd     #for data structures
import numpy as np      #for numerical operations
import seaborn as sns   #for visualization
import matplotlib.pyplot as plt     #for visualization
#%matplotlib.inline
```

In [2]:
```
#load the dataset as a pandas dataframe, df, and display the first few rows, as well as the shape

df = pd.read_csv('flight.csv')

#get the total number of rows and columns
print("no. of rows,no. of columns:", df.shape)

#identify the columns that exist in our dataset
print("column names:",df.columns)

#preview the loaded dataset in a dataframe
df
```

```
no. of rows,no. of columns: (2500, 8)
column names: Index(['Unnamed: 0', 'acc.date', 'type', 'reg', 'operator', 'fat', 'location',
       'dmg'],
      dtype='object')
```

Out[2]:

	Unnamed: 0	acc.date	type	reg	operator	fat	location	dmg
0	0	3 Jan 2022	British Aerospace 4121 Jetstream 41	ZS-NRJ	SA Airlink	0	near Venetia Mine Airport	sub
1	1	4 Jan 2022	British Aerospace 3101 Jetstream 31	HR-AYY	LANHSA - Línea Aérea Nacional de Honduras S.A	0	Roatán-Juan Manuel Gálvez International Airpor...	sub
2	2	5 Jan 2022	Boeing 737-4H6	EP-CAP	Caspian Airlines	0	Isfahan-Shahid Beheshti Airport (IFN)	sub
3	3	8 Jan 2022	Tupolev Tu-204-100C	RA-64032	Cainiao, opb Aviastar-TU	0	Hangzhou Xiaoshan International Airport (HGH)	w/o
4	4	12 Jan 2022	Beechcraft 200 Super King Air	NaN	private	0	Machakilha, Toledo District, Grahem Creek area	w/o
...	...	...	...	...	...	...	...	...
2495	1245	20 Dec 2018	Cessna 560 Citation V	N188CW	Chen Aircrafts LLC	4	2 km NE of Atlanta-Fulton County Airport, GA (...	w/o
2496	1246	22 Dec	PZL-Mielec	GNB-	Guardia Nacional	0	Kamarata Airport	sub

2496	1246	2018	M28 Skytruck	96107	Bolivariana de Venezuela - GNBV	0	(KTV)	sub
2497	1247	24 Dec 2018	Antonov An-26B	9T-TAB	Air Force of the Democratic Republic of the Congo	0	Beni Airport (BNC)	w/o
2498	1248	31 Dec 2018	Boeing 757-2B7 (WL)	N938UW	American Airlines	0	Charlotte-Douglas International Airport, NC (C...	sub
2499	1249	unk. date 2018	Rockwell Sabreliner 80	N337KL	private	0	Eugene Airport, OR (EUG)	sub

2500 rows × 8 columns

The author of the kaggle dataset provides additional context by providing a preview of key fields as follows:

- **Date**: The date of the crash
- **Type**: Aircraft model/type
- **Registration**: Aircraft registration code
- **Operator**: Airline or organization operating the aircraft
- **fat**: Number of fatalities reported in the crash (passengers + crew)
- **Location**: Where the crash occurred
- **dmg**: Damage severity (encoded)

Similarly, the damage severity can be observed to have categorical values like sub and w/o. These are decoded as follows, representing an assesment of aircraft damage:

- **sub** → Substantial Damage
- **w/o** → Write-Off (Total Loss)
- **non** → No Damage / Minor

Further assesing our data for summary statistics can be done for full data understanding.

In [3]:
```
#Statistics for all columns
df.info() # to get each columns data type and non null values
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2500 entries, 0 to 2499
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Unnamed: 0  2500 non-null   int64 
 1   acc.date    2500 non-null   object
 2   type        2500 non-null   object
 3   reg         2408 non-null   object
 4   operator    2486 non-null   object
 5   fat         2488 non-null   object
 6   location    2500 non-null   object
 7   dmg         2500 non-null   object
dtypes: int64(1), object(7)
memory usage: 156.4+ KB
```

In [4]:
```
#Check for total duplicate values (true =1, False =0)
df.duplicated().sum()
```

Out[4]: 1250

Data Quality Assesment

In totality, the following preliminary observations on the provided data can be made:

In [5]:
```
"""
    1. The data contains 2500 data entry rows with 8 columns detailing the date,location, aircra
       as well as the operator, aircraft damage level and fatalities.
    2. Half the dataset is a duplicate of the original. 1250 duplicate values could indicate tha
       and further analysis is required.
    2. There are a few errors: wrong daata types for certain columns e.g date, missing values, p
       registration column as well as a few others.
    3. Column names are not easily deduced. fat,dmg and reg cannot be easily interpretted.
    4. Decoding of the damage column could make for easier interpretation.
"""
```

Out[5]:
```
'\n    1. The data contains 2500 data entry rows with 8 columns detailing the date,location, air
craft registration, type and damage\n       as well as the operator, aircraft damage level and
fatalities.\n    2. Half the dataset is a duplicate of the original. 1250 duplicate values could
indicate that the dataset is repeated \n       and further analysis is required.\n    2. There
are a few errors: wrong daata types for certain columns e.g date, missing values, prominently in
the aircraft\n       registration column as well as a few others.\n    3. Column names are not
easily deduced. fat,dmg and reg cannot be easily interpretted.\n    4. Decoding of the damage co
lumn could make for easier interpretation.\n'
```

Important information that could be beneficial to this project can also be noticed to be absent. This would include columns such as phase of flight *(takeoff, cruise or landing)*, passenger capacity, probable cause *(aircraft malfunction or human error)* , cost, e.t.c that could have allowed for a more thorough analysis onto best fits for the company.

3.Data Preparation

Inconsistencies noticed earlier need to be ammended.These are done stepwise as follows:

3.1. Changing column titles

In [6]:
```
#create a list containing the new values
new_titles = ['I.d','Date', 'Aircraft type', 'Registration','Operator','Fatalities','Location','

#assign the new values
df.columns = new_titles

#df.tail()
```

3.2 Removing duplicates

The I.d and the index are inconsistent, likely because each row was duplicated as preliminary results suggested.

In [7]:
```
#check the I.d column for value counts
df['I.d'].value_counts()
```

Out[7]:
```
1249    2
838     2
824     2
826     2
828     2
       ..
417     2
419     2
421     2
423     2
0       2
Name: I.d, Length: 1250, dtype: int64
```

In [8]:
```
# Appears to be true. Load the middle of the table to check
df[1249: 1255]
```

Out[8]:

	I.d	Date	Aircraft type	Registration	Operator	Fatalities	Location	Damage
1249	1249	unk. date 2018	Rockwell Sabreliner 80	N337KL	private	0	Eugene Airport, OR (EUG)	sub
1250	0	3 Jan 2022	British Aerospace 4121 Jetstream 41	ZS-NRJ	SA Airlink	0	near Venetia Mine Airport	sub
1251	1	4 Jan 2022	British Aerospace 3101 Jetstream 31	HR-AYY	LANHSA - Línea Aérea Nacional de Honduras S.A	0	Roatán-Juan Manuel Gálvez International Airpor...	sub
1252	2	5 Jan 2022	Boeing 737-4H6	EP-CAP	Caspian Airlines	0	Isfahan-Shahid Beheshti Airport (IFN)	sub
1253	3	8 Jan 2022	Tupolev Tu-204-100C	RA-64032	Cainiao, opb Aviastar-TU	0	Hangzhou Xiaoshan International Airport (HGH)	w/o
1254	4	12 Jan 2022	Beechcraft 200 Super King Air	NaN	private	0	Machakilha, Toledo District, Grahem Creek area	w/o

In [9]:
```
# Conclusion: Table is repeated, hence duplicates
#remove duplicates and reassign the new unique values to the dataframe
df = df.drop_duplicates()

#check dataframe shape
df.shape

# ->duplicates dropped!
```

Out[9]: (1250, 8)

In [10]:
```
# Lastly assign I.d column to be the new index:
df.set_index('I.d', inplace=True)
```

3.3. Column by Column Cleaning & Preperation

Approach each column individually. Identify patterns to errors, rectify and deal with missing values on a row by row basis.

In [11]:
```
#Preview the dataframe as is
df
```

Out[11]:

I.d	Date	Aircraft type	Registration	Operator	Fatalities	Location	Damage
0	3 Jan 2022	British Aerospace 4121 Jetstream 41	ZS-NRJ	SA Airlink	0	near Venetia Mine Airport	sub

1	4 Jan 2022	British Aerospace 3101 Jetstream 31	HR-AYY	LANHSA - Línea Aérea Nacional de Honduras S.A	0	Roatán-Juan Manuel Gálvez International Airpor...	sub
2	5 Jan 2022	Boeing 737-4H6	EP-CAP	Caspian Airlines	0	Isfahan-Shahid Beheshti Airport (IFN)	sub
3	8 Jan 2022	Tupolev Tu-204-100C	RA-64032	Cainiao, opb Aviastar-TU	0	Hangzhou Xiaoshan International Airport (HGH)	w/o
4	12 Jan 2022	Beechcraft 200 Super King Air	NaN	private	0	Machakilha, Toledo District, Grahem Creek area	w/o
...	...	...	...	...	...	...	...
1245	20 Dec 2018	Cessna 560 Citation V	N188CW	Chen Aircrafts LLC	4	2 km NE of Atlanta-Fulton County Airport, GA (...	w/o
1246	22 Dec 2018	PZL-Mielec M28 Skytruck	GNB-96107	Guardia Nacional Bolivariana de Venezuela - GNBV	0	Kamarata Airport (KTV)	sub
1247	24 Dec 2018	Antonov An-26B	9T-TAB	Air Force of the Democratic Republic of the Congo	0	Beni Airport (BNC)	w/o
1248	31 Dec 2018	Boeing 757-2B7 (WL)	N938UW	American Airlines	0	Charlotte-Douglas International Airport, NC (C...	sub
1249	unk. date 2018	Rockwell Sabreliner 80	N337KL	private	0	Eugene Airport, OR (EUG)	sub

1250 rows × 7 columns

In [12]:
```
#Create a function that can remove whitespace and convert to lowercase that can be reused for mu
def strip_and_lower(column):
    column = column.str.lower()
    column = column.str.strip()
    return column
```

a)Date Column

In [13]:
```
#Research based standard practice to prevent subtle bugs, ensuring modification is done on df da
df = df.copy()

#remove whitespace and lower
df['Date'] = strip_and_lower(df['Date'])

#get summative statistics
print(df['Date'].describe())

#get unique values
print(df['Date'].value_counts())

#Convert the data type to date
df['Date'] = pd.to_datetime(df['Date'], errors='coerce')

df = df.copy()
```

```
count           1250
unique           871
top       3 mar 2020
freq              12
Name: Date, dtype: object
3 mar 2020     12
24 feb 2022     6
15 apr 2018     6
10 jun 2018     5
10 mar 2019     5
                ..
26 sep 2018     1
22 jul 2021     1
30 oct 2021     1
18 jul 2021     1
5 may 2021      1
Name: Date, Length: 871, dtype: int64
```

In [14]:
```
#Check for total missing values
df['Date'].isna().sum()
```

Out[14]: 3

In [15]:
```
"""
Consider dropping these rows as this is an integral field, especially if other fields are missing
because only three rows lack dates.
"""
#Filter those rows to view those with missing dates
missing_dates = df['Date'].isna() #create a df with the missing dates bool
df.loc[missing_dates] #locate and print
```

Out[15]:

I.d	Date	Aircraft type	Registration	Operator	Fatalities	Location	Damage
101	NaT	Antonov An-2R	RA-17951	Voskhod LLC	0	near Oymyakon Airfield	sub
233	NaT	Antonov An-2	NaN	Korean People's Army Air and Anti-Air Force (K...	5	near Taechon Air Base	w/o
1249	NaT	Rockwell Sabreliner 80	N337KL	private	0	Eugene Airport, OR (EUG)	sub

In [16]:
```
"""All other columns are filled. Also these would skew data slightly for the case of antonov air
    2 are present"""

#Best option is to forward fill or backward fill
df['Date'].fillna(method='ffill',inplace=True)
```

b) Aircraft Type Column

In [17]:
```
#strip and lower
df['Aircraft type'] = strip_and_lower(df['Aircraft type'])

#Preview unique values
df['Aircraft type'].unique()
```

Out[17]:
```
array(['british aerospace 4121 jetstream 41',
       'british aerospace 3101 jetstream 31', 'boeing 737-4h6',
       'tupolev tu-204-100c', 'beechcraft 200 super king air',
       'airbus a320-214 (wl)', 'cessna 208b grand caravan ex',
       'airbus a320-232', 'bombardier cl-600-2b16 challenger 604',
```

```
'beechcraft b300 king air 350', 'hawker 1000',
'cessna 208b grand caravan', 'embraer erj-190-100lr',
'antonov an-26', 'cessna 501 citation i/sp', 'antonov an-2r',
'let l-410uvp-e3', 'atr 42-500', 'britten-norman bn-2a-9 islander',
'swearingen sa226-at merlin iv', 'embraer emb-500 phenom 100e',
'de havilland canada dhc-3t texas turbine otter',
'raytheon hawker 800xp', 'antonov an-2', 'antonov an-22a',
'antonov an-26-100', 'antonov an-74t', 'cessna 208b supervan 900',
'antonov an-124-100', 'antonov an-225',
'embraer erj 170-200 lr (erj-175lr)', 'shaanxi y-8q',
'embraer emb-500 phenom 100', 'boeing 737-8as (wl)',
'dornier 228-101', 'cessna 525b citationjet cj3',
'cessna 208 caravan i', 'honda ha-420 hondajet',
'beechcraft 1900d', 'cessna 208b super cargomaster',
'boeing 737-89p (wl)', 'boeing 767-346er',
'embraer emb-110p2 bandeirante', 'learjet 75',
'boeing 757-27a (pcf)', 'aero modifications ami dc-3-65tp',
'beechcraft a100 king air',
'gulfstream american g-1159 gulfstream ii b',
'basler bt-67 turbo 67', 'fairchild sa227-dc metro 23',
'embraer emb-505 phenom 300', 'antonov an-26b-100',
'beechcraft 99a airliner', 'shorts 360-300', 'boeing 737-82r (wl)',
'boeing 737-883 (wl)', 'viking air dhc-6 twin otter 400',
'airbus a319-115 (wl)', 'embraer emb-110p1 bandeirante',
'boeing 737-7h4 (wl)', 'de havilland canada dhc-3t turbine otter',
'boeing 787-9 dreamliner', 'beechcraft b200 king air',
'de havilland canada dhc-6 twin otter 300',
'cessna 525 citationjet', 'shorts sc.7 skyvan 3-200',
'bombardier bd-100-1a10 challenger 300', 'airbus a319',
'cessna 550 citation ii', 'beechcraft b200 super king air',
'boeing 737-3m8 (qc)', 'boeing 777-228er', 'airbus a330-202',
'mcdonnell douglas md-82', 'learjet 55c', 'antonov an-30m',
'fokker f-27 friendship 500', 'ilyushin il-76md',
'de havilland canada dhc-8-402q dash 8', 'cessna 208 supervan 900',
'antonov an-12bk', 'learjet 35a', 'gulfstream g450',
'raytheon beechjet 400a', 'boeing 737-81d (wl)', 'fokker 50',
'airbus a321-271nx', 'cessna 680 citation sovereign',
'boeing 717-2bd', 'airbus a320-214',
'de havilland canada dhc-3t vazar turbine otter',
'casa c-212 aviocar 200', 'boeing 737-86n (wl)', 'boeing 777-fdz',
'boeing 757-251 (wl)', 'iai 1124 westwind',
'cessna 560 citation encore', 'antonov an-24b',
'boeing 737-824 (wl)', 'beechcraft 99', 'airbus a320-251n',
'shorts sc.7 skyvan 3m-400', 'airbus a220-100', 'boeing 777-f1h',
'cessna 551 citation ii/sp', 'cessna 750 citation x', 'learjet 36',
'cirrus sf50 vision jet', 'antonov an-28',
'fairchild sa227-ac metro iii', 'beechcraft 1900c-1',
'dassault falcon 2000', 'british aerospace 3201 jetstream 32ep',
'britten norman bn-2a-8 islander', 'rockwell sabreliner 65',
'boeing 737-8hc (wl)', 'boeing 737-436 (sf)',
'boeing 737-8v3 (wl)', 'boeing 757-256 (wl)', 'boeing 777-3b5er',
'boeing 737-7ct (wl)', 'boeing 737-8gj (wl)',
'boeing 737-8q3 (wl)', 'embraer emb-545 legacy 450',
'britten-norman bn-2a-21 islander', 'cirrus sf50 vision jet g2',
'beechcraft 99 airliner', 'de havilland canada dhc-8-314 dash 8',
'airbus a330-322', 'canadair cl-215-6b11 (cl-415)',
'embraer emb-110c bandeirante', 'airbus a319-132', 'eclipse 550',
'let l-410uvp-e20', 'atr 72-600 (72-212a)', 'boeing 737-823 (wl)',
'boeing 737 max 8', 'boeing b-17g flying fortress',
'fairchild sa227-at expediter', 'airbus a320-271n',
'airbus a320-211', 'boeing 737-4q8 (sf)', 'airbus a330-243',
'learjet 45', 'embraer erj-175lr', 'british aerospace bae-146-200',
'boeing 767-322er (wl)', 'swearingen sa226-tc metro ii',
'de havilland canada dhc-8-402qpf dash 8',
'gulfstream giv-x (g450)', 'boeing 737-860 (wl)',
'boeing 737 max 8-200', 'boeing 737-3z0',
'canadair cl-600s challenger 600',
'embraer erj-175lr (erj-170-200 lr)', 'learjet 31a',
'cessna 560 citation v', 'boeing 737-524 (wl)', 'harbin y-12-ii',
'boeing 777-31her', 'avro rj85', 'canadair cl-600 challenger 600',
'north american rockwell sabreliner 60', 'harbin y-12e',
```

```
'beechcraft 100 king air', 'boeing 747-8kzf',
'dassault falcon 900ex easy', 'learjet 45xr', 'antonov an-2t',
'douglas dc-3c', 'atr 72-500 (72-212a)',
'beechcraft b300 king air 350i', 'let l-410uvp-e', 'fokker 100',
'embraer emb-120er brasilia', 'boeing 777-223er',
'bombardier crj-701er', 'boeing 737-4y0', 'saab 340af', 'embraer',
'embraer emb-145lr', 'boeing c-17a globemaster iii',
'cessna 525a citationjet cj2+', 'beechcraft b99 airliner',
'learjet 25b', 'gulfstream g150', 'beechcraft b200gt king air 250',
'hawker 800', 'iai 1125 astra', 'boeing 737-7h4', 'boeing 767-38e',
'beechcraft 1900c', 'boeing 737-8al (wl)',
'boeing 787-8 dreamliner', 'gulfstream g280',
'lockheed l-100-30 hercules', 'airbus a321-231', 'airbus a350-941',
'convair cv-580 airtanker', 'boeing 737-275c adv.',
'lockheed c-130h hercules', 'iai 1124a westwind ii', 'atr 42-320',
'british aerospace bae-125-700a', 'cessna 560xl citation excel',
'de havilland canada dhc-8-106', 'airbus a321-211',
'boeing 737-8jp (wl)', 'bombardier cl-600-2b16 challenger 605',
'airbus a300b4-622r (f)',
'gulfstream american g-1159a gulfstream iii',
'mcdonnell douglas md-11f', 'shijiazhuang y-5b(d)',
'britten-norman bn-2a-iii-2 trislander',
'bombardier dhc-8-402q dash 8', 'beriev be-200chs',
'ilyushin il-112v', 'gulfstream g-iv', 'boeing 737-990er',
'bombardier crj-900lr', 'boeing 767-332er (wl)',
'cessna 560xls+ citation xls+', 'raytheon 390 premier i',
'british aerospace bae-125', 'canadair crj-200lr',
'antonov an-26kpa', 'airbus a321-271n', 'saab 340b',
'boeing 757-224 (wl)', 'britten-norman bn-2b-26 islander',
'dassault falcon 20c', 'boeing 737-8h4 (wl)',
'bombardier bd-700-1a10 global 6000', 'rockwell sabreliner 60',
'mcdonnell douglas md-87', 'beechcraft 200 king air',
'beechcraft super king air', 'transall c-160ng', 'gulfstream ?',
'gulfstream aerospace g-1159a gulfstream iii',
'britten-norman bn-2a-6 islander', 'iai 1125 astra sp',
'bombardier bd-100-1a10 challenger 350', 'boeing 747-4b5f',
'cessna 510 citation mustang', 'swearingen sa-226at merlin iv',
'gulfstream g-ivsp', 'airbus a330-343', 'embraer erj-170se',
'airbus a330-303', 'beechcraft 300 super king air',
'irma/pilatus britten-norman bn-2t islander', 'gulfstream g iv',
'antonov an-12a', 'hawker 900xp', 'de havilland canada dhc-8-202q',
'boeing 737-8kv (wl)', 'lockheed c-130bz hercules',
'airbus a321-231 (wl)', 'boeing 737-8q8 (wl)', 'airbus a380-861',
'boeing 767-232 (bdsf)', 'cessna s550 citation s/ii',
'lockheed ec-130q hercules', 'bombardier e-11a (global express)',
'mcdonnell douglas md-83', 'boeing 737-6ct', 'boeing 737-76n (wl)',
'boeing 747-412f (scd)', 'boeing 747-412 (bcf)',
'boeing 767-375er', 'boeing 737-86j (wl)', 'boeing 737-8kn (wl)',
'boeing 737-301 (bdsf)', 'learjet 55', 'antonov an-2p',
'dassault falcon 50', 'airbus a300b4-203 (f)',
'dassault falcon 7x', 'cessna 525c citation cj4',
'cessna 525 citationjet cj1',
'de havilland canada dhc-6 vista liner 300', 'boeing 767-3q8er',
'boeing 757-223 (wl)', 'hawker siddeley hs-125-f400b',
'boeing 737-881 (wl)', 'british aerospace bae-125-800a',
'lockheed c-130j-30 super hercules', 'learjet 25d',
'embraer emb-120rt brasilia', 'hawker siddeley hs-125',
'ilyushin il-78', 'learjet 60xr', 'let l-410ma',
'beechcraft 400a beechjet', 'embraer c-95bm bandeirante (emb-110)',
'de havilland canada dhc-6 twin otter 100', 'dornier 228-212',
'lockheed c-130h3 hercules', 'beechcraft 300 super king air 350',
'grumman g-1159 gulfstream ii', 'embraer erj-145ep',
'cessna 560xl citation xls', 'grumman g-1159 gulfstream ii sp',
'airbus a319-111', 'hawker 800xp',
'britten-norman bn-2a-27 islander', 'boeing 737-8s3 (wl)',
'de havilland canada dhc-8-402q (pf) dash 8', 'boeing 777-f60',
'airbus a310-304', 'boeing 737-3z0 (sf)', 'airbus a321-211 (wl)',
'antonov an-74tk-100', 'boeing 737-8hg (wl)',
'canadair cl-215-1a10', 'rockwell sabreliner 75a',
'boeing 767-3s2fer', 'antonov an-26b', 'boeing 747sp-21',
'boeing 767-324er', 'dassault falcon 200',
```

```
'gulfstream g200 galaxy', 'learjet', 'learjet 35',
'antonov an-26sh', 'canadair cl-600-1a11 challenger 600',
'lockheed kc-130j hercules', 'boeing 777-232er', 'hawker 850xp',
'boeing 737-4b6 (sf)', 'douglas c-54e (dc-4)', 'airbus a320-200',
'antonov an-32a', 'learjet 60', 'embraer erj-145lr',
'britten-norman bn-2a-26 islander',
'cessna ac-208b combat caravan', 'british aerospace bae-125-800b',
'de havilland canada dhc-7-110', 'atr 42-300', 'boeing 737-529',
'shijiazhuang y-5b', 'eclipse 500', 'cessna 525 citation m2',
'canadair cl-600-2a12 challenger 601', 'boeing 707-3j9c',
'boeing 777-212er', 'boeing 737-8lj (wl)',
'de havilland canada dhc-8-102 dash 8', 'boeing 727-2b6 adv. (f)',
'boeing 737-332', 'airbus a321-251n', 'convair c-131b samaritan',
'embraer erj 190ar (erj-190-100 igw)', 'boeing 747-406m',
'boeing 737-832 (wl)', 'boeing 767-375er (bcf) (wl)',
'boeing 737-8e9 (wl)', 'de havilland canada eo-5c (dhc-7-102)',
'embraer erj-195lr (erj-190-200 lr)',
'bombardier bd-100 challenger 350', 'embraer emb-145xr',
'douglas dc-3', 'bombardier cl-600-2c10 regional jet crj-702er',
'hawker siddeley hs-125-600a',
'de havilland canada cc-138 twin otter (dhc-6)',
'cessna 650 citation iii', 'mcdonnell douglas md-88',
'british aerospace 4100 jetstream 41', 'boeing 737-924er (wl)',
'bombardier bd-700-1a11 global 5000',
'pilatus britten-norman bn-2b-27 islander', 'airbus a320-232 (wl)',
'boeing 737-33a (wl)', 'boeing 737-73s (wl)',
'douglas c-47a-25-dk (dc-3c)', 'boeing 737-81q (wl)',
'canadair cl-600-2b16 challenger 601-3a',
'sukhoi superjet 100-95b', 'de havilland canada dhc-8-311 dash 8',
'embraer erj-190lr', 'boeing 737-9b5',
'gulfstream american g-1159 gulfstream ii sp', 'antonov an-32',
'basler bt-67 turbo 67 (dc-3t)', 'hawker siddeley hs-125-400',
'british aerospace 3212 jetstream 31', 'antonov an-24rv',
'saab 340a', 'dassault falcon 900b', 'boeing 737-85r',
'boeing 737-8k2 (wl)', 'boeing 757-324 (wl)',
'boeing 737-36n (wl)', 'boeing 777-333er', 'ilyushin il-76td',
'cessna 208 caravan 675', 'airbus a319-114', 'unknown',
'bombardier crj-900er', 'douglas c-118a liftmaster (dc-6a)',
'airbus a340-313', 'british aerospace 3201 jetstream 32',
'hs-125/bae-125', 'canadair cl-600-2b16 challenger 601',
'de havilland canada dhc-6 twin otter 200',
'cessna 680a citation latitude', 'de havilland canada dhc-8-200',
'airbus a340-642', 'lockheed c-130a hercules',
'cessna 525 citationjet cj1+', 'airbus a319-112', 'convair cv-440',
'cessna 208 caravan', 'antonov an-72', 'atr 42-600',
'bombardier global express', 'saab 2000',
'airbus cc-150 polaris (a310-300)',
'canadair cl-600-2b19 regional jet crj-200lr',
'de havilland canada dhc-3t/m601 turbine otter', 'airbus a330-341',
'let l-410uvp', 'boeing 747-428erf',
'embraer emb-145lr (erj-145lr)', 'bombardier crj-900',
'boeing 737-8f2 (wl)', 'boeing 737-401', 'boeing 737-800',
'dornier 228-201', 'grumman american g-1159 gulfstream iisp',
'embraer erj-140lr', 'embraer erj-190ar',
'cessna 525a citationjet cj2', 'boeing 737-7bd (wl)',
'airbus a319-131', 'boeing 737-8fh (wl)', 'boeing 737-8ct (wl)',
'boeing 777-369er', 'antonov an-2tp', 'gulfstream g200',
'fairchild sa227-at merlin ivc',
'british aerospace bae-748-347 srs. 2a', 'iptn/casa cn-235m-100',
'convair cv-580f', 'airbus a320', 'shaanxi y-8gx-3',
'antonov an-148-100b', 'airbus a330-223', 'atr 72-212',
'beechcraft b100 king air', 'airbus a330-323',
'boeing 737-322 (sf)', 'learjet 31', 'beechcraft c99 commuter',
'canadair cl-600-2b19 regional jet crj-200er', 'gulfstream v',
'boeing 737-8k5 (wl)', 'casa c-212 aviocar', 'boeing 737-76j (wl)',
'dassault falcon 2000ex', 'learjet 40xr', 'lockheed p-3c orion',
'antonov an-74-200', 'cessna 550 citation bravo',
'airbus a321-213', 'boeing 737-8gp (wl)',
'lockheed wc-130h hercules', 'dassault falcon 900ex',
'embraer kc-390', 'boeing 737-8bk (wl)', 'boeing 737-201 advanced',
'pilatus britten-norman bn-2b-26 islander',
```

```
'de havilland canada dhc-8-202q dash 8',
'embraer erj 170lr (erj-170-100 lr)', 'boeing 777-346',
'ilyushin il-76', 'boeing 777-2b5er',
'irma/pilatus britten-norman bn-2a-26 islander', 'dornier 228-202',
'convair cv-340', 'boeing 737-8eh (wl)',
'curtiss c-46f-1-cu commando', 'douglas c-47b (dc-3)',
'boeing 757-204', 'boeing 767-38eer',
'irma/britten-norman bn-2a-27 islander',
'britten-norman bn-2a-8 islander', 'antonov an-2sx',
'boeing 757-223', 'embraer erj 190ar', 'junkers ju-52/3mg4e',
'dassault falcon 20d', 'boeing 777-3f2er', 'boeing 777-367er',
'boeing 737-85c (wl)', 'boeing 757-2q8 (wl)',
'grumman hu-16c albatross', 'boeing 767-333er (wl)',
'boeing 777-328er', 'boeing 747-428fer', 'ilyushin il-20m',
'airbus a320-216', 'boeing 747-4d7', 'atr 72-202',
'boeing 737-7l9 (wl)', 'boeing 777-337er', 'airbus a330-203',
'boeing 747-412f', 'lockheed c-130e hercules', 'boeing 757-23n',
'embraer erj-190lr (erj-190-100 lr)',
'hawker siddeley hs-125-700a',
'de havilland canada dhc-8-315q dash 8',
'british aerospace 3212 jetstream 32', 'boeing 737-53c',
'bombardier crj-200lr', 'lockheed martin kc-130j',
'pilatus britten-norman bn-2a-20 islander',
'pzl-mielec m28 skytruck', 'boeing 757-2b7 (wl)',
'rockwell sabreliner 80'], dtype=object)
```

In [18]:

```
# Observation: The data is highly granular.
# i.e each aircraft would be represented as its own type which wouldn't inform us of anything ab

""" Idea is to extract a make and model by splitting the 'aircraft type' into two new columns
    (maker and model) based on the first set of text/word that contains a number """

#create a function that could do this
def manufacturer_and_model(aircraft_type):

    #Assign null values for those with missing aircraft types
    if pd.isna(aircraft_type):
        return None,None

    #For non-null values, split text into individual words
    words = str(aircraft_type).split()

    #create a new variable to store the index position of the first digit
    digit_index = None

    #Loop through each word
    for i,word in enumerate(words):
        #and each character to find out if it is a number or not
        for character in word:
            if character.isdigit():
                digit_index = i
                break

        #stop the character iteration if a number is found
        if digit_index is not None:
            break

    #If no digit is found, return all the words as the manufacturer as long as the first word is
    if digit_index is None:
        make = words[0]
        model = " ".join(words [1:]) if len(words) > 1 else None
        return make, model

    #If a digit is found, all the words before the word with the digit are manufacturer, rest ar
    make = " ".join(words[:digit_index])
    model = " ".join(words[digit_index:])

    return make, model
```

In [19]:
```
# Now to use this function to clean the data into the two new columns:

# Create empty lists for the manufacturer and model
aircraft_manufacturer = []
aircraft_model =[]

#Loop through the data column, appending the manufacturer and model to the list
for aircraft in df['Aircraft type']:
    make, model= manufacturer_and_model(aircraft)

    #add to list
    aircraft_manufacturer.append(make)
    aircraft_model.append(model)

#Create new columns in the original dataframe for the manufacturer and aircraft model of each rou
df['Manufacturer'] = aircraft_manufacturer
df['Model'] = aircraft_model

#Ensure changes are made into the database
df = df.copy()

# Show a few rows of the dataframe
df.head()
```

Out[19]:

	Date	Aircraft type	Registration	Operator	Fatalities	Location	Damage	Manufacturer	Model
I.d									
0	2022-01-03	british aerospace 4121 jetstream 41	ZS-NRJ	SA Airlink	0	near Venetia Mine Airport	sub	british aerospace	4121 jetstream 41
1	2022-01-04	british aerospace 3101 jetstream 31	HR-AYY	LANHSA - Línea Aérea Nacional de Honduras S.A	0	Roatán-Juan Manuel Gálvez International Airpor...	sub	british aerospace	3101 jetstream 31
2	2022-01-05	boeing 737-4h6	EP-CAP	Caspian Airlines	0	Isfahan-Shahid Beheshti Airport (IFN)	sub	boeing	737-4h6
3	2022-01-08	tupolev tu-204-100c	RA-64032	Cainiao, opb Aviastar-TU	0	Hangzhou Xiaoshan International Airport (HGH)	w/o	tupolev	tu-204-100c
4	2022-01-12	beechcraft 200 super king air	NaN	private	0	Machakilha, Toledo District, Grahem Creek area	w/o	beechcraft	200 super king air

In [20]:
```
#Check the edited values
df['Manufacturer'].describe()
#df['Manufacturer'].unique()
```

Out[20]:
```
count       1250
unique        67
top       boeing
```

```
freq         209
Name: Manufacturer, dtype: object
```

In [21]:
```
#some companies appear to have gone through mergers and others have been mislabelled
df['Manufacturer'].replace({'britten norman': 'britten-norman','irma/pilatus britten-norman': 'b
    'irma/britten-norman': 'britten-norman','pilatus britten-norman': 'britten-norman'}, inplace
df['Manufacturer'].replace({'gulfstream american': 'gulfstream','gulfstream giv-x': 'gulfstream'
    'gulfstream aerospace': 'gulfstream','pilatus britten-norman': 'gulfstream'}, inplace=True)
df['Manufacturer'].replace({'raytheon': 'raytheon hawker','hawker': 'raytheon hawker','hawker si
    'raytheon beechjet': 'raytheon hawker'}, inplace=True)
df['Manufacturer'].replace({'north american rockwell sabreliner':'rockwell sabreliner','embraer
                        'douglas':'mcdonnell douglas','lockheed':'lockheed martin','grumman amer
                        'iptn/casa':'casa','':'unknown'}, inplace=True)

#confirm new values
df['Manufacturer'].describe()
#df['Manufacturer'].unique()
```

Out[21]:
```
count       1250
unique        49
top       boeing
freq         209
Name: Manufacturer, dtype: object
```

In [22]:
```
#Check for total missing values
df['Manufacturer'].isna().sum()
```

Out[22]:
```
0
```

In [23]:
```
#Because the data is highly granular and due to time constraints, the top n normalization is imp
# We can check to see what the top 20 values account for in terms of a percentage of the total,
vc = df['Manufacturer'].value_counts()  #create a new variable that stres the value counts table
coverage = vc.cumsum() / vc.sum()   #calculate the running totals as a percentage of total rows
coverage.head(25)  # display the top half to see what percentage we have reached
```

Out[23]:
```
boeing                0.1672
cessna                0.3168
airbus                0.4144
beechcraft            0.4952
antonov               0.5744
de havilland canada   0.6288
embraer               0.6704
learjet               0.6992
british aerospace     0.7256
gulfstream            0.7512
bombardier            0.7768
atr                   0.8000
raytheon hawker       0.8184
mcdonnell douglas     0.8352
britten-norman        0.8496
lockheed martin       0.8624
let                   0.8744
canadair              0.8848
fairchild             0.8944
dassault falcon       0.9040
rockwell sabreliner   0.9128
fokker                0.9208
ilyushin              0.9280
iai                   0.9344
honda                 0.9400
Name: Manufacturer, dtype: float64
```

c) Registration Column

In [24]:

```
# Standardize the columns as previously done
df['Registration'] = strip_and_lower(df['Registration'])

#Should all be in uppercase
df['Registration'] = df['Registration'].str.upper()

#see values to notice if some could be replaced
df['Registration'].value_counts()[:10]
```

Out[24]:
```
UNREG.        3
5Y-SAV        2
N817NW        2
C-FKWE        2
N233SW        2
FALSE REG.    2
P2-PXE        2
OB-2152       2
VH-...        2
N601VH        1
Name: Registration, dtype: int64
```

In [25]:
```
#combine those with missing or incomplete registration
df['Registration'].replace({'UNREG.': np.nan,'FALSE REG.': np.nan,'VH-...':np.nan}, inplace=True
```

In [26]:
```
#Check for total missing values
df['Registration'].isna().sum()
#Column may be dropped due to large amount of missing data
#This column also likely will not influence business intelligence outcomes anyway, hence is
# unlikely to be used in analysis.
```

Out[26]:
```
53
```

d) Operator Column

In [27]:
```
#Using the strip and lower function
df['Operator'] = strip_and_lower(df['Operator'])

#create a function that cleans the operated by and operated for
def clean_operator_name(aircraft_operator):
    if pd.isna(aircraft_operator):
        return None

    text = aircraft_operator.strip()

    # operated by → take AFTER
    if 'opb' in text:
        parts = text.split('opb')
        return parts[-1].strip()

    # operated for → take BEFORE
    if 'opf' in text:
        parts = text.split('opf')
        return parts[0].strip()

    return text

#Run the function through each item in the series, appending the output to a  new list
new_operator = []
for op in df['Operator']:
    cleaned = clean_operator_name(op)
    new_operator.append(cleaned)

#Assign the created list to the operator column
df['Operator'] =new_operator
```

In [28]:
```
# Remove all commas resulting from the opb and opf
```

```
df['Operator'] = df['Operator'].str.replace(r'[,.]+', '', regex=True)

#Remove all ltd inc and llc that may cause issues
df['Operator'] = df['Operator'].str.replace(r'\b(inc|llc|ltd)\b', '', regex=True)

#Remove everything in brackets Square or round including the brackets
df['Operator'] = df['Operator'].str.replace(r'\[.*?\]|\(.*?\)', '', regex=True).str.strip()

#Remove everything that comes after a hyphen, but only those with abbreviations i.e text - abb.(
df['Operator'] = df['Operator'].str.split(' - ').str[0].str.strip()

# Replaces any instance of "air lines" with "airlines"
df['Operator'] = df['Operator'].str.replace(r'air\s+lines', 'airlines', regex=True)

# Replaces any instance of "air line" with "airline"
df['Operator'] = df['Operator'].str.replace(r'air\s+line', 'airline', regex=True)
```

In [29]:

```
# Replace visible errors
df['Operator'].replace({'thy turkish airlines':'turkish airlines','lionair':'lion air'},inplace=
```

In [30]:

```
df['Operator'].unique()
```

Out[30]:

```
array(['sa airlink', 'lanhsa', 'caspian airlines', 'aviastar-tu',
       'private', 'star flyer', 'mahan air', 'jetblue airways',
       'volare aviation', 'lima delta co trustee', 'air tindi', 'skyview',
       'care aviation', 'finnair', 'south sudanese air force',
       'aviajet sa', 'kamchatsky krechet', 'doren air congo',
       'japan air commuter', 'air flamenco', 'colcharter',
       'eclipse transport', 'bald mountain air services',
       'roper aviation', 'air serv limited', 'taraz zhana alem',
       'antonov airlines', 'gojump oceanside', 'russian air force',
       'ukraine air force', 'ab aviation', 'revolution flight',
       'republic airlines', 'spirit avia sentosa',
       'china naval air force', 'flyzar', 'ryanair', 'indian coast guard',
       'hlaf aeta', 'ozark air services', 'bamaji air', 'jet it',
       'bocas air', 'martinaire', 'china eastern airlines',
       'japan airlines', "skydive costa d'argento lsf gladwings",
       'tag airlines', 'georgia crown distributing co',
       'dhl aero expreso', 'aliansa colombia', 'thunder airlines',
       'servicio aéreo de policia', 'gem air', 'denver air connection',
       'voar cooperativa', 'gp aviation', 'constanta airlines',
       'freight runners express',
       "centre ecole de parachutisme sportif de l'ariège", 'spicejet',
       'air cargo carriers', 'sociedad de transporte aéreo movi air spa',
       'blue air', 'sas scandinavian airlines', 'caverton helicopters',
       'lbk serviço aéreo especializado', 'tibet airlines',
       'sales taxi aéreo', 'southwest airlines',
       'yakutat coastal airlines', 'united airlines', 'etihad airways',
       'unknown', 'tara air', 'kam aviation', 'connecticut parachutists',
       'alaskan air charter', 'pacific dental services', None,
       'northern meridian', 'asl', 'gomair', 'air france', 'ita airways',
       'red air', 'techservice', 'npp mir', 'icon aviation', 'ana wings',
       'alaska seaplanes', 'voskhod', 'motor sich', 'flying america sa',
       'alpha star aviation services', 'corporate air',
       'aircraft management group', 'solaseed air', 'meridian',
       'jubba airways', 'wizz air', 'anderson air',
       'crop protection company', 'delta airlines', 'avianca', 'vistajet',
       "rust's flying service", 'rampart aviation', 'nok air',
       'qatar airways cargo', 'brasil vida táxi aéreo',
       'raber flight services', 'angara airlines', 'pacc air',
       'smart cakrawala aviation', 'tap air portugal',
       '2nd arkhangelsk united aviation division', 'skyforce',
       'aerologic', 'friday harbor seaplane tours', 'gg rent', 'bvs',
       'strategic airborne operations', 'tac9', 'tracep-congo aviation',
       'piquiatuba táxi aéreo', 'aeronaves tsm', 'equaflight services',
       'penken group', 'pacific cataract & laser institute, pc',
```

```
'pepkor group', 'pacific cataract & laser institute pc',
'saeta peru', 'jags aviation', 'sunexpress', 'swiftair',
'copa airlines', 'freedom airline express',
'abs equipment leasing', 'icelandair flugfélag islands',
'korean air', 'transavia france', 'japan transocean air',
'partee aviation', 'torres strait air', 'sky express',
'jupiter aviation', 'true north airways',
'panamerican training center', 'flamingo air charter',
'perimeter aviation', 'reven global transpor', 'vigili del fuoco',
'fuerza aérea uruguaya', 'spirit airlines', 'elite flight travel',
'goma express', 'aviación militar bolivariana', 'precision air',
'american airlines', 'american airpower heritage museum',
'risk mondial aviation & recovery', 'andersen air', 'ameriflight',
'leair charter services', 'penjet pty', 'latam perú',
'raisbeck engineering', 'bluebird cargo', 'verijet',
'azul linhas aéreas brasileiras', 'jett aircraft', 'aery aviation',
'reva', 'mesa airlines', 'antarctic airways', 'key lime air',
'blue bird aviation', 'hawaiian airlines', 'air canada',
'rimbun air', 'flexjet', 'ethiopian airlines',
'tarco aviation lsf mid africa aviation', 'skystallion',
'envoy air', 'bar aviation', 'vagus group', 'sx transport',
'sriwijaya air', 'kenya air force', 'emirates', 'jota aviation',
'west atlantic uk', 'global avionics',
'aviacion ejecutiva del bajio', 'martinair', 'zambian air force',
'mm-air', "korean people's army air and anti-air force",
'nippon cargo airlines', 'luxwing', 'flyadeal', 'aerospike iron',
'manta air', "armée de l'air et de l'espace", 'royal air freight',
'air india express', 'wings over kississing',
'fuerza aérea mexicana', 'nigerian air force',
'asia continental airlines', 'air algérie',
'beidahuang general airlines', 'south sudan supreme airlines',
'iran air', 'berry aviation', 'mission aviation fellowship',
'grant aviation', 'kazakhstan border guards', 'skywest airlines',
'viva aerobus', 'venezuela flight canaima', 'aeronav air services',
'trigana air service', 'sprintair',
'mauritania airlines international', 'west wind aviation',
'piedmont airlines', 'malta air', 'everts air cargo',
'united states air force', 'med air', 'lake clark air',
'eletric power construção', 'haugland group aviation',
'stb aviation', 'tld aviation i', 'government of madhya pradesh',
'uni air', 'skydive andes', 'utair', 'asiana airlines',
'jl&gl productions lp', 'alpine air express', 'dabi air nusantara',
'gf aviation', 'paraclub wiener neustadt', 'vistara',
'tatmadaw lei', 'indigo airlines', 'kin avia', 'british airways',
'dosaaf', 'ineos aviation', 'ethiopian air force', 'skydive teuge',
'conair', 'transair', 'hellenic caa', 'elisa',
'hukbong himpapawid ng pilipinas', 'transenergie', 'transwest air',
'kamchatka aviation enterprise', 'ryan air', 'condor flugdienst',
'northwestern air', 'air tunilik', 'harbour air',
'seair seaplanes', 'sila', 'goskydive', 'aeronova', 'pv transport',
'skyward express', 'norwegian air sweden', 'aeolus air charter',
'skydive binz', 'abyssinian flight services',
'european air transport', 'united parcel service',
'northeast general aviation co', 'roraima airways',
'congo airways', 'russian navy', 'wright air service',
'united aircraft corporation', 'brooks range aviation', 'sn 1124',
'alaska airlines', 'n1gg', 'skyjet', 'nicholas services',
'brook haven properties', 'pb air', 'fly exclusive', 'aeroservice',
'meander air ii', 'csm agropecuária', 'austrian airlines',
'air wisconsin', 'impuma group', 'lps flight checks & systems',
'paracentrum vlaanderen', 'stp airways lsf aerojet', 'svg air',
'sierra west airlines', 'air services limited',
'lowcountry aviation co brokerage', 'aeroural', '987 investments',
'scout about', 'pal airlines', 'optimum aviation',
'grodno aircompany', 'safe air company',
'asociatia club sportiv skydiving center', 'island airways',
'emd astra holdings', 'netjets', 'western wings corp',
'kalitta air', 'lake & peninsula airlines',
'air caraïbes atlantique', 'air charter scotland',
'castle aviation', 'frontier airlines',
'dimed sa distribuidora da medicamentos',
```

```
'hellidosa aviation group', 'keewatin air',
'cottingham & butler insurance services', 'malu aviation',
'aeromedevac air ambulance', 'sky-bound aviation',
'halsted aviation corporation', 'turkish airlines',
'empresas ginro sa', 'royal moroccan gendarmerie',
'executive charters', 'al quwwat al-jawwiya as-sudaniya',
'million air san juan', 'sri lanka air force', 'lc whitford co',
'us special operations command', 'l3 technologies',
'ukraine international airlines', 'south african air force',
'nordwind airlines', 'jin air', 'lifemed alaska',
'west air sweden', 'air inuit',
'south african civil aviation authority', 'coulson aviation',
'compass aviation', 'us air force', 'westjet', 'ale',
'westjet encore', 'act airlines', 'pegasus airlines',
'redding aero enterprises', 'psa airlines', 'mountain air cargo',
'transportes torreon', 'remonia air', 'sc cole aviation',
'eastar jet', 'lauren engineers & constructors', 't-cement',
'canadian pacific railway company', 'trans maldivian airways',
'expectra aviation', 'orlan 2000', 'sigma airlines',
'east coast jets', 'bbr air', 'jetleg management', 'at aero',
'executive business aviation', 'legacy air', 'textron',
'crye-leike south', 'carol 1', 'caridad aviation', 'csa air',
'grand canyon airlines', 'omni air international',
'thai airways international', 'platinum jet co',
'martini aviation', 'servicios 5250084', 'gemini air group',
'planemasters', 'rbr development', 'hog air aviation', 'lion air',
'libyan national army', "mike's oilfield services",
'inversiones sc 2012', 'buffalo airways',
'province of saskatchewan ministry of central services',
'dos mil aerosistema', 'african express airways', 'md fly',
'libyan air force', 'united aviation', 'mcwilliams', 'swift air',
'sanjiv goenka group', 'força aérea brasileira',
'pakistan international airlines', 'kapowsin air sports',
'netherlands coast guard', 'sunwest aviation',
'gama aviation signature', 'ural airlines', 's7 airlines',
'loganair', 'nordic aviation capital', 'jindal steel & power',
'chair airlines', 'latam airlines brasil', 'aeroparadise sa',
'tropic air charters', 'zeus-avia', 'mango', 'turkish police',
'skyjet aviation', 'phoenix', 'n425bj', 'ax transporter',
'aeroflot russian international airlines', 'utair-cargo',
'babcock', 'iberia', 'city link', 'air senegal', 'fedex',
'bomac air', 'south west aviation lsf skyway air',
'las vegas sands corporation', 'skymark airlines',
'wrv empreendimentos e participacoes ltda', 'connair consulting',
'silverstone air services', 'united states marine corps',
'ratchthani leasing public company limited', 'aercaribe cargo',
'alaska air fuel', 'duk air travel', 'aercaribe peru',
'caroline islands air', 'worldwide jet charter',
'oriental air bridge', 'st bernard parish government', 'tracbel',
'al quwwat al jawwiya al iraqiya', 'georgia jet',
'volga-dnepr airlines', 'trujet', 'air-glaciers',
'voyageur airways', 'british antarctic survey',
'calm air international', 'cubana de aviación',
'royal bahamas defence force', 'air sanga', 'aviation star s ii',
'air djibouti', 'airco aircraft charters',
'xinjiang general aviation', 'executive aviation investors',
'talon air', 'la barca empreendimentos ltda', 'blue wing airlines',
'raf-avia', 'utair aviation', 'tw 601-c investment',
'saha air lsf islamic republic of iran air force',
'singapore airlines', 'priority air charter', 'air creebec',
'onur air', 'kalitta charters', 'guardian flight', 'tarco air',
'novair', 'conquest air cargo', 'rico taxi aéreo', 'kenya airways',
"stein's aircraft services", 'compass airlines', 'winair',
'klm royal dutch airlines', 'atlas air',
'biman bangladesh airlines', 'united states army',
'silk way business aviation', 'flybe', 'avianca brasil',
'banyan jet service', 'tassili airlines', 'amik aviation',
'commutair', 'baires fly', 'waffle house', 'laser aéreo colombia',
'royal canadian air force', 'golden wings aviation',
'rp sales and leasing', 'sundance airport fbo', 'berjaya air',
'afrijet', 'major blue air',
```

```
'empresa nacional de servicios aéreos-ensa', 'easyjet',
'sky high aviation services', 'classic aviation', 'summit air',
'luftwaffe', 'archipiélagos servicios aéreos', 'thai smile',
'asia airways', 'jet2', "t'way air", 'miami air international',
'skyservice business aviation',
'compañia de aviación y logística empresarial', 'jazz aviation',
'parachutisme nouvel air', 'myanmar national airlines',
'taquan air', 'jet sales', 'batik air', 'libyan air cargo',
'indian air force', 'tso ukrayiny', 'feniks', 'ep aviation',
'map linhas aéreas', 'banco macro', 'hong kong airlines',
'navigator', 'north star air', 'phoenix air', 'yankee dawdle',
'mueller aero', 'transmandu', 'western air', 'isr aviation',
'warren transportation', 'dosaaf gvardeysky atsk', 'ee operations',
'gouvernement du québec', 'afriqiyah airways', 'easyjet europe',
'omicron business services', 'avia jet', 'air peace', 'europe air',
'cobham aviation services', 'lufthansa',
'fuerza aérea de guinea ecuatorial',
'pakistan army aviation corps', 'bush air fuel',
'thomas cook airlines', 'liaoning star general aviation',
'edelweiss air', 'dosaaf tajikistan', 'alkan air', 'tropical air',
'skyaviatrans', 'aerowest', 'sarpa', 'norwegian air uk',
'fuerza aérea argentina', 'all nippon airways', 'jrm air',
'safarilink', 'mokulele airlines', 'virgin atlantic airways',
'fuerza aérea venezolana', 'morningstar air express',
'delta private jets', 'club tiroler adler',
'international air response', 'air china', 'air charter services',
'lenox handels und speditions gmbh', 'star jet', 'libyan airlines',
'jet 24 gmbh', 'regional express américas', 'eurowings', 'gojump',
'ferreteria e implementos san francisco', 'twoflex',
'carpediem aviation', 'flymontserrat', 'auric air',
'gobernación del estado bolívar',
'department of royal rainmaking and agricultural aviation',
'cavok air', 'the collings foundation', 'ukraine air alliance',
'pro by air', 'air force of the democratic republic of the congo',
'skydive georgia', 'pineapple air', 'elite air', 'penair',
'atlantic air cargo', 'endeavor air', 'fuerza aérea salvadoreña',
'blue water aviation services', 'garuda indonesia airways',
'abeer air services', 'horizon air', 'usmx airlink',
'josé joão abdalla filho', 'etg aviation',
'pilot point consultancy', 'air namibia', 'tahe havacilik',
'avior airlines', 'busy bee congo', 'proflight air services',
'tropicair', 'business jet managers', 'fuerza aérea de chile',
'qantas', 'air europa', 'ritch air', 'wl aviation',
'c-ghgr holdings', 'representaciones aero3', 'calafia airlines',
'tigerair taiwan', 'bek air', 'air fast congo', 'sunny sky',
'private airlines germany', 'sunwing airlines', 'kuwait airways',
'lot polskie linie lotnicze', 'air taxi ph',
'kazakhstan air defence force', 'bismillah airlines', 'fastjet',
'türk hava kuvvetleri', 'air tribe', 'galeyr airline',
'delegation of the eu to somalia lsf airtraffic africa', 'nestoil',
"people's liberation army", 'travel management',
'saratov airlines', 'qeshm air', 'iran aseman airlines',
'dana air', 'army parachute association', 'island express air',
'strait air', 'smartlynx airlines estonia', 'serve air', 'n500mp',
'mc aviation', 'us-bangla airlines', 'west coast air services',
'peyton holdings', 'smartwings', 'bighorn airways',
'el al israel airlines', 'germania', 'aeromar',
'línea aérea amaszonas', '2m leasing', 'skydive the ranch',
'avis industrial corporation', 'henning aviation',
'mtc enterprises', 'union gas air ventures i', 'erg holdings',
'mega aircompany', 'airwing', 'hageland aviation services',
"al quwwat al-jawwiya al-jaza'eriya", 'penial air',
'united states navy', 'ayk avia', 'morningstar partners',
'herbert waldmann lichttechnik gmbh & co kg',
'world atlantic airlines', 'szemp air kft', 'alpine aviation',
'tui fly belgium', 'blueport trade 121', 'aluminios cortizo',
'embraer', 'air niugini', 'lynden air cargo',
'irish parachute club lsf parachuting caravan leasing',
'makalu air', 'cubana de aviación lsf global air',
'saudi arabian airlines lsf onur air', 'silver air',
'amazonaves táxi aéreo', 'titan drilling', 'air katanga',
```

```
       'asure air', 'fly-sax', 'aeroserv', 'proair',
       'falkland islands government air service', 'bravo airways',
       'linkpng', 'norwegian', 'dimonim air', 'eagle air',
       'islamic republic of iran air force',
       'federal aviation administration', 'maya island air',
       'air choice one', 'gam', 'flex air charters', 'white airways',
       'ram express', 'ookpik aviation', 'rovos air', 'air colombia',
       'gol linhas aéreas', 'flybondi', 'everts air fuel',
       'air canada rouge', 'kazairtrans airline',
       'commemorative air force highland lakes squadron',
       'ozair charter service', 'sunday airlines', 'air kasai',
       'air taxi vanuatu', 'unity airlines', 'air vanuatu',
       'aeroméxico connect', 'atlantic transportation of wilmington',
       'heilongjiang kunpeng general aviation co', 'ju-air',
       'alliance air charter', 'supreme airlines', 'royal air maroc',
       'vancon holdings', 'cathay pacific airways', 'xiamen airlines',
       'lc perú', 'ostthüringer fallschirmsportclub', 'aer lingus',
       'stargazer aero', 'satena', 'bush air safaris', 'vanilla air',
       'air mizia', 'beijing capital airlines', 'boydak air',
       'west coast aviation services', 'voar aviação',
       'south west aviation lsf slav-air',
       'government of newfoundland and labrador',
       'saudi arabian airlines lsf act airlines', 'thai airasia',
       'sts aviatsija', 'starjet', 'américa latina tecnologia agrícola',
       'philippine airlines', 'air transat', 'cgg aviation australia',
       'air america flight services', 'inversiones moraima',
       'frontier flying service', 'jhonlin air transport',
       'yakutia airlines', 'overland airways', 'air india',
       'sky lease cargo', 'pakistan air force', 'fly jamaica airways',
       'air astana', 'songbird aviation', 'aeroflot', 'peruvian airlines',
       'arg ltda', 'dirt dynamics', 'estoair', 'arrow aviation',
       'ellinair', 'desert-air safaris', 'par-avion', 'aerounion',
       "africa's connection stp", 'chen aircrafts',
       'guardia nacional bolivariana de venezuela'], dtype=object)
```

In [31]:
```
#Check for total missing values
df['Operator'].isna().sum()
```

Out[31]:
```
7
```

In [32]:
```
#Replace missing values with unknown since this column's data will likely not influence results
df['Operator'] = df['Operator'].fillna('unknown')
df['Operator'].value_counts().head()
```

Out[32]:
```
private              107
unknown               27
american airlines     22
delta airlines        22
united airlines       18
Name: Operator, dtype: int64
```

e) Fatalities Column

In [33]:
```
df['Fatalities'].unique()
```

Out[33]:
```
array(['0', '2', nan, '5', '14', '11', '132', '1', '22', '6', '4', '8',
       '0+2', '10', '3', '0+1', '19', '5+1', '62', '7', '12', '50+3',
       '28', '16', '9', '1+1', '18', '176', '97+1', '21', '26', '15',
       '157', '1+2', '13', '41', '1+5', '5+14', '21+6', '38', '71', '66',
       '39', '51', '257', '112', '20', '189'], dtype=object)
```

In [34]:
```
#Take a look at the values in the column
#df['Fatalities'].unique()

#major error is that there appears to be a plus in some of the columns
#create a function that will do the sum using python's inbuilt eval feature
def evaluate_fatalities(figure):
```

```
def evaluate_fatalities(figure):
    #If there is already an null value
    if pd.isna(figure):
        return np.nan
    try:
        summation = eval(str(figure))
        return summation
    except(ValueError, TypeError, SyntaxError):
        #In case of errors
        return np.nan

#Assign the created list to the operator column
df['Fatalities'] =df['Fatalities'].apply(evaluate_fatalities)

#Convert to integer format
df['Fatalities'] = df['Fatalities'].astype('Int64')
```

In [35]:

```
#Look at descriptive data to decide on how to handle missing values
df['Fatalities'].describe()
```

Out[35]:

```
count    1244.000000
mean        2.015273
std        13.472021
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max       257.000000
Name: Fatalities, dtype: float64
```

In [36]:

```
"""
Consider dropping these rows as this is an integral field, especially if other fields are missing
because only six rows lack dates.
"""
#Filter those rows to view those with missing dates
missing_dates = df['Fatalities'].isna() #create a df with the missing dates bool
df.loc[missing_dates] #locate and print
```

Out[36]:

	Date	Aircraft type	Registration	Operator	Fatalities	Location	Damage	Manufacturer	Model
I.d									
30	2022-02-24	antonov an-26	RF-36074	russian air force	<NA>	near Ostrogozhsk, Voronezh Region	w/o	antonov	an-26
307	2021-06-23	lockheed l-100-30 hercules	NaN	ethiopian air force	<NA>	near Gijet	w/o	lockheed martin	l-100-30 hercules
521	2020-04-05	antonov an-26	UP-AN601	libyan national army	<NA>	near Tarhuna	w/o	antonov	an-26
614	2020-09-25	british aerospace bae-125	NaN	private	<NA>	Catatumbo, Edo, Zulia	w/o	british aerospace	bae-125
658	2020-12-05	british aerospace bae-125-800a	N484AR	private	<NA>	Jesús María Semprúm	w/o	british aerospace	bae-125-800a
1223	2018-11-23	british aerospace bae-125-700a	N422X	private	<NA>	S of Curaçao [Caribbean Sea]	mis	british aerospace	bae-125-700a

In [37]:
```
# All but one of these are write-offs
""" Consideration: Filling in the average value of fatalities (in integer form),
    of fatalities in write-off accidents, as all other rows contain data. """
missing_fatalities = df.groupby('Damage')[['Fatalities']].mean()
missing_fatalities #to view the mean, particularly for write-off damage
```

Out[37]:

Damage	Fatalities
min	0.061224
mis	NaN
non	0.017751
sub	0.078195
unk	0.000000
w/o	7.078035

In [38]:
```
df['Fatalities'].fillna(7, inplace=True)

#Confirm negligent change in descriptive statistics
df['Fatalities'].describe()
        #Note: Barely any change
```

Out[38]:
```
count    1250.000000
mean        2.039200
std        13.444042
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max       257.000000
Name: Fatalities, dtype: float64
```

f) Location Column

In [39]:
```
# Initial assesment is that this column might not be of the highest importance for current use c
# Remove initials for standardization
#Remove everything that comes after a comma and space, similar to operators column
df['Location'] = df['Location'].str.split(', ').str[0].str.strip()

#Remove everything in brackets Square or round including the brackets
df['Location'] = df['Location'].str.replace(r'\[.*?\]|\(.*?\)', '', regex=True).str.strip()

#Remove the airport and international airport from the name

df['Location'] = df['Location'].str.replace(r'Airport|International Airport|Airstrip|Airfield',

#round up locations, ie near,within, etc
df['Location'] = df['Location'].str.replace(r'near|within', '', regex=True).str.strip()

#Remove everything that comes after hyphen or forward slash
df['Location'] = df['Location'].str.split('-').str[0].str.strip()
df['Location'] = df['Location'].str.split('/').str[0].str.strip()
```

In [40]:
```
# A few rows have their distances put as --km NNE of -----
df['Location'] = (
    df['Location']
      .str.split('off ').str[-1] #remove everything before 'off'
```

```
      .str.split(' of ').str[-1]  #remove everything before 'of'
      .str.split(' from ').str[-1]  #remove everything before 'from'
      .str.split(' and ').str[-1] #For instances of between this and that, take the latter
)
```

In [41]:
```
#Remove descriptive words that some instances may lack
df['Location'] = df['Location'].str.replace(r'Mine|City|Regional|Island|Municipal|intersection|H

# As well as double space
df['Location'] = df['Location'].str.replace(r'  ', '', regex=True).str.strip()

#Replace stubborn values
df['Location'].replace({'Toronto-Lester B. Pearson':'Toronto','Parque Nacional Laguna del Tigre'
                        'western Venezuela':'Venezuela'},inplace=True)

#Finally remove whitespace
df['Location'] = df['Location'].str.strip()
```

In [42]:
```
#Fill in missing values as unknown
df['Location'] = df['Location'].fillna('Unknown')
```

g) Damage Column

In [43]:
```
#use function for strip and lower
df['Damage'] = strip_and_lower(df['Damage'])
#Check to confirm what values exist
df['Damage'].unique()
```

Out[43]:
```
array(['sub', 'w/o', 'non', 'min', 'unk', 'mis'], dtype=object)
```

In [44]:
```
# Replace Values with those from the dictionary provided
df['Damage'].replace({'sub':'substantial','w/o':'write-off','non':'none',
                      'unk':'unknown','min':'minor','mis':'unknown'},inplace=True)

#Since unknown columns exist, categorized null values as unknown
df['Damage'] = df['Damage'].fillna('unknown')
```

In [45]:
```
#Confirm changes
df['Damage'].value_counts()
```

Out[45]:
```
substantial    665
write-off      351
none           169
minor           49
unknown         16
Name: Damage, dtype: int64
```

In [46]:
```
#Check for missing values
df['Damage'].isna().sum()
```

Out[46]:
```
0
```

h) Created Column: Model

In [47]:
```
#During the aircraft type stage, two new columns were created:Aircraft and Model
df.head()
```

Out[47]:

	Date	Aircraft type	Registration	Operator	Fatalities	Location	Damage	Manufacturer	Model

I.d									
0	2022-01-03	british aerospace 4121 jetstream 41	ZS-NRJ	sa airlink	0	Venetia	substantial	british aerospace	4121 jetstream 41
1	2022-01-04	british aerospace 3101 jetstream 31	HR-AYY	lanhsa	0	Roatán	substantial	british aerospace	3101 jetstream 31
2	2022-01-05	boeing 737-4h6	EP-CAP	caspian airlines	0	Isfahan	substantial	boeing	737-4h6
3	2022-01-08	tupolev tu-204-100c	RA-64032	aviastar-tu	0	Hangzhou Xiaoshan	write-off	tupolev	tu-204-100c
4	2022-01-12	beechcraft 200 super king air	NaN	private	0	Machakilha	write-off	beechcraft	200 super king air

In [48]:
```
#Check for missing values
df['Model'].isna().sum()
```

Out[48]: 3

In [49]:
```
# Filter to view the errant rows as previously done
missing_models = df['Model'].isna() #create a df with the missing dates bool
df.loc[missing_models] #locate and print
```

Out[49]:

I.d	Date	Aircraft type	Registration	Operator	Fatalities	Location	Damage	Manufacturer	Model
265	2021-03-25	embraer	NaN	mauritania airlines international	0	Nouakchott	none	embraer	None
609	2020-09-18	learjet	NaN	private	0	Zanja	write-off	learjet	None
821	2019-07-30	unknown	NaN	fuerza aérea de guinea ecuatorial	0	Grand Batanga	write-off	unknown	None

In [50]:
```
""" All other rows contain valid data, therefore avoid deleting."""

# Replace with 'unknown' appears to be the best decision
df['Model'].fillna('unknown', inplace=True)
```

3.4. Creating New Columns

A new year column can be created from the date column to allow for easier time trend analysis.

In [51]:
```
# since 'Date' column is already in datetime format, the .year can be used
df['Year'] = df['Date'].dt.year

#Ensure integer type
```

```
df['Year'] = df['Year'].astype('Int64')
```

3.5. Dropping Columns

Recall that the 'Registration' column had 53 null values. Considering the scope of the project and the kind of data stored in this column, it is unlikely that this data is useful.
Similarly, new aircraft may be assigned previously used registration, as well as aircraft that may still be functional. As a result it is best to consider dropping it.

The cleaned dataset can first be exported for analysis and visualization in Tableau.

In [52]:
```
# Export the DataFrame to a CSV file named 'cleaned_flight_data.csv
df.to_csv("cleaned_flight_data.csv", index=False)
```

Taking the opportunity to also assign the new cleaned data a new variable name:

In [53]:
```
#Drop the 'Registration' Column
#rename the dataset as 'ad' for aircraft data

ad = df.drop(['Registration','Aircraft type'],axis=1)
```

Finally, Take one more look at the dataframe info to ascertain no missing values and correct data type.

In [54]:
```
ad.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1250 entries, 0 to 1249
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype         
---  ------        --------------  -----         
 0   Date          1250 non-null   datetime64[ns]
 1   Operator      1250 non-null   object        
 2   Fatalities    1250 non-null   Int64         
 3   Location      1250 non-null   object        
 4   Damage        1250 non-null   object        
 5   Manufacturer  1250 non-null   object        
 6   Model         1250 non-null   object        
 7   Year          1250 non-null   Int64         
dtypes: Int64(2), datetime64[ns](1), object(5)
memory usage: 90.3+ KB
```

3.6. Data Cleaning Summary

The following is a generalization of the cleaning procedures done as well as an assesment on its feasibility for use in the next stage.

In [55]:
```
""" The dataset looks clean and ready for further analysis. It contains 1250 rows and 8 columns.
        -> Duplicate values (1250) were dropped, halving the size of the dataset.
        -> The registration column was dropped due to missing values, along with the fact that i
        -> The aircraft type column was also dropped as it eas split into the new columns: Manuf
           This is to ease analysis as the data in that column was highly granular and was diffi
        -> The date column was changed to datetype and the (3) missing values were forward fille
            all other columns contained data, and there were only a few missing rows, meaning ag
        -> The Fatalities column was changed to integer form, the missing values were filled wit
            categorized as write off as 6 of the 7 rows with missing values had this categorizat
        -> The Location column wa cleaned by removing all the airport initials, descriptive word
            which allowed for generalization rather than specifics, making data less granular.
        -> The damage column was rewritten using the data dictionary provided by the dataset aut
            as unknown, similar to the 'unk' category. There were no missing values.
        -> The year column was created as an extraction of the date column, in integer format.
        -> The cleaned dataset was renamed as 'ad'
```

```
    Conclusion: Data can be used for next stage.
"""
```

Out[55]:
```
" The dataset looks clean and ready for further analysis. It contains 1250 rows and 8 columns.
\n        -> Duplicate values (1250) were dropped, halving the size of the dataset.\n        ->
The registration column was dropped due to missing values, along with the fact that it provides
no useful info.\n        -> The aircraft type column was also dropped as it eas split into the n
ew columns: Manufacturer and Model.\n           This is to ease analysis as the data in that col
umn was highly granular and was difficult to aggregate.\n        -> The date column was changed
to datetype and the (3) missing values were forward filled. This was chosen because\n
all other columns contained data, and there were only a few missing rows, meaning aggregate stat
istics would not be skewed much with the new changes.\n        -> The Fatalities column was chan
ged to integer form, the missing values were filled with the average value of fatalities in airc
raft whose damage was\n            categorized as write off as 6 of the 7 rows with missing valu
es had this categorization. This prevented skewing of aggregation statistics as well.\n        -
> The Location column wa cleaned by removing all the airport initials, descriptive words like ci
ty, park e.t.c and the distance near a particular place.\n            which allowed for generali
zation rather than specifics, making data less granular.\n        -> The damage column was rewri
tten using the data dictionary provided by the dataset author. The category 'mis' was taken as m
issing and hence filled in \n            as unknown, similar to the 'unk' category. There were n
o missing values.\n        -> The year column was created as an extraction of the date column, i
n integer format.\n        -> The cleaned dataset was renamed as 'ad'\n\n    Conclusion: Data ca
n be used for next stage.\n"
```

4.Exploratory Analysis

Using the cleaned data, analysis can be done with the aim to answer key business questions as follows:

4.1. Overall Accident Trend Over Time

Business question: Has aviation safety improved over time?

Explore the accidents per year over the five years in the dataset and examine the trend.

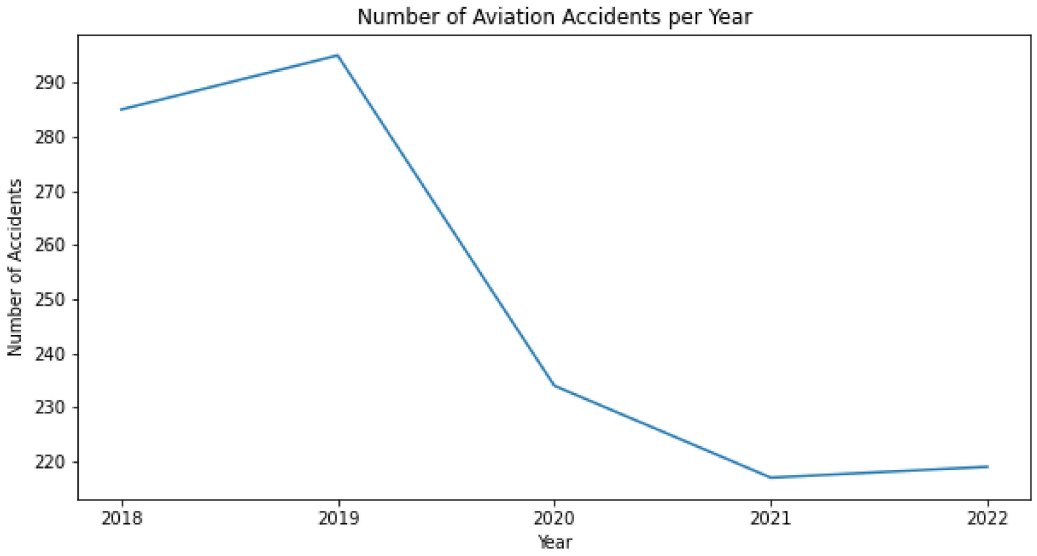
In [56]:
```
# Aggregate the accidents per year using the 'groupby' feature
acc_per_year = (ad.groupby('Year').size().reset_index(name='Accident_Count').sort_values('Year')
acc_per_year
```

Out[56]:

	Year	Accident_Count
0	2018	285
1	2019	295
2	2020	234
3	2021	217
4	2022	219

In [57]:
```
#Create figure
plt.figure(figsize=(10, 5))

# Create x and y
x = acc_per_year['Year']
y = acc_per_year['Accident_Count']

#create plot and labels
plt.plot(x,y)
plt.title('Number of Aviation Accidents per Year')
plt.xlabel('Year')
plt.ylabel('Number of Accidents')

#Plots correctly but with a decimal point between the years i.e. 2018.0, 2018.5 etc.
# Force integer ticks (Researched solved problem code)
import matplotlib.ticker as mticker
```

```
plt.gca().xaxis.set_major_locator(mticker.MaxNLocator(integer=True))
```



4.2. Accident Frequency by Aircraft Type.

Business question: Which aircraft types are involved in the most accidents?

Explore the top highest and fewest aircraft manufacturers in accidents

In [58]:

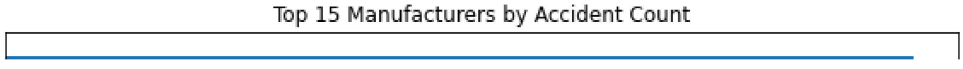
```
# Create a new dataframe that contains the value counts in descending order of the manufacturer
manuf_counts = (ad['Manufacturer'].value_counts().reset_index())

# Assign column names to the new dataframe
manuf_counts.columns = ['Manufacturer', 'Accident_Count']

# Display the first few rows
manuf_counts.head()
```

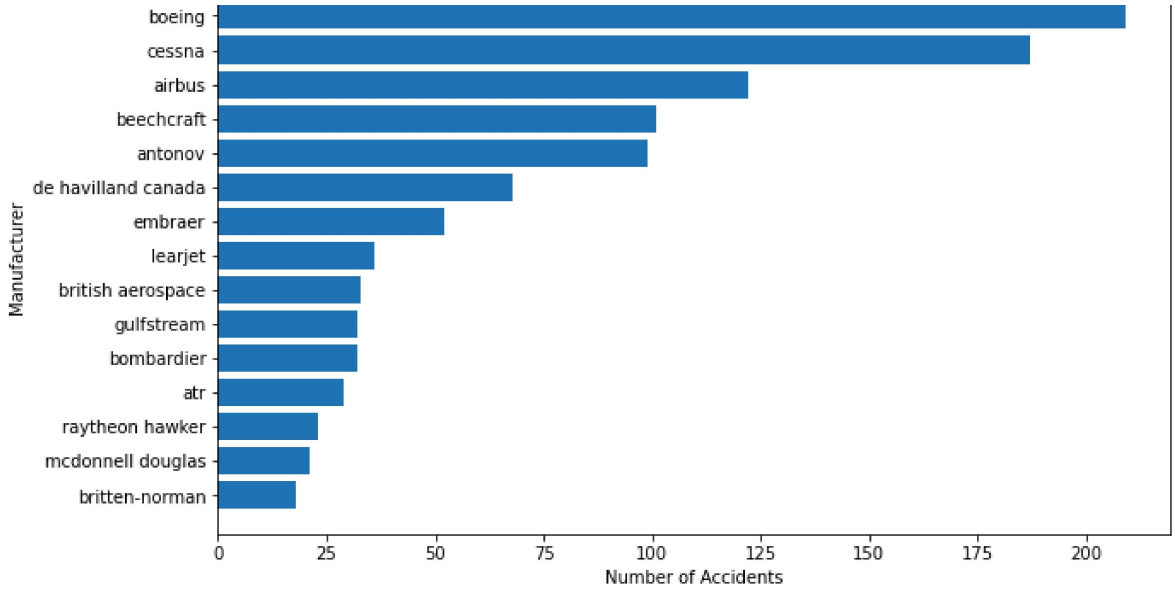
Out[58]:

	Manufacturer	Accident_Count
0	boeing	209
1	cessna	187
2	airbus	122
3	beechcraft	101
4	antonov	99

In [59]:

```
# Create the plot figure
plt.figure(figsize=(10, 6))

#Create the x and y axis values
x = manuf_counts['Manufacturer'].head(15)
y = manuf_counts['Accident_Count'].head(15)
plt.barh(x,y)

plt.title('Top 15 Manufacturers by Accident Count')
plt.xlabel('Number of Accidents')
plt.ylabel('Manufacturer')

# Invert y-axis so highest is on top for visual clarity
plt.gca().invert_yaxis()
```





In [60]:
```
"""Interpretation:     A small number of manufacturers account for a large proportion of reporte
                       This pattern likely reflects relative market share and fleet size and usa
                       and should be interpreted alongside severity and trend analyses."""
```

Out[60]:
```
'Interpretation:     A small number of manufacturers account for a large proportion of reported
accidents. \n                       This pattern likely reflects relative market share and fleet
size and usage rather than inherent aircraft risk, \n                         and should be interp
reted alongside severity and trend analyses.'
```

4.3 Accident Severity by Aircraft Type

Business question: Which aircraft manufacturers have the best survivability?

Explore accident severity and identify which manufacturer comes out of accidents relatively unscathed

4.3.1) Aircraft Damage (Plane Survivability)

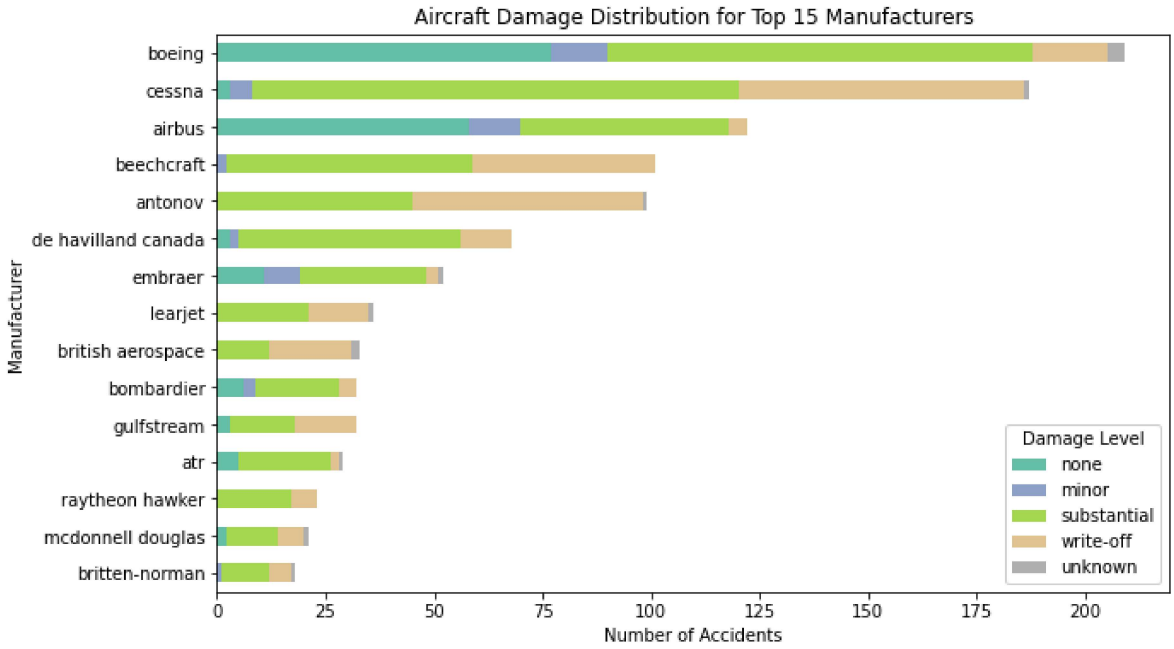
In [61]:
```
# Aggregate damage by manufacturer

#First get top count accidents per manufacturer similar as before but restrict to top 10 manufac
manuf_counts = ad['Manufacturer'].value_counts().head(15).index.tolist()

# Filter the dataframe
top_manuf = ad[ad['Manufacturer'].isin(manuf_counts)]

# Then aggregate damage by manufacturer
#group the new dataframe by manufacturer and damage
#count and fill zeros where absent
damage_counts = top_manuf.groupby(['Manufacturer', 'Damage']).size().unstack(fill_value=0)

# Add total accidents column
damage_counts['total'] = damage_counts.sum(axis=1)

# Sort by total accidents descending so it's similar to the first graph
damage_counts = damage_counts.sort_values('total', ascending=False)

# Drop the helper column for plotting
damage_counts = damage_counts.drop(columns='total')

#preview the dataframe
#damage_counts
```

In [62]:
```
#The damage seems to be in the wrong order
```

```
#Create a list for the correct order
damage_order = ['none', 'minor', 'substantial', 'write-off','unknown']  # from least to most sev

#reassign the dataframe in this order
damage_counts = damage_counts[damage_order]

#Preview the dataframe again
#damage_counts
```

In [63]:

```
#Create the new plot using the existing dataframe
damage_counts.plot(
    kind='barh', #horizontal bar chart, stacked
    stacked=True,
    figsize=(10, 6),
    colormap='Set2' #cheeky feature to select colour visuals
)

plt.xlabel('Number of Accidents')
plt.ylabel('Manufacturer')
plt.title(f'Aircraft Damage Distribution for Top 15 Manufacturers')
plt.legend(title='Damage Level')

# handy feature to plot highest at the top
plt.gca().invert_yaxis()
```



In [64]:

```
"""Interpretation:  This stacked bar chart shows aircraft damage distribution for the top manufa
                    Manufacturers with a higher proportion of none minor damage (cyan and viole
                    providing useful insight for procurement decisions.
                   Boeing Airbus and Embraer have a good ratio of minor damage to total damage r
```

Out[64]:

```
'Interpretation:  This stacked bar chart shows aircraft damage distribution for the top manufact
urers. \n                    Manufacturers with a higher proportion of none minor damage (cyan
and violet) demonstrate stronger structural survivability in accidents,\n                    pr
oviding useful insight for procurement decisions.\n                    Boeing Airbus and Embraer
have a good ratio of minor damage to total damage making them good candidates.'
```

4.3.2) Fatality Rate (Passenger survivability)

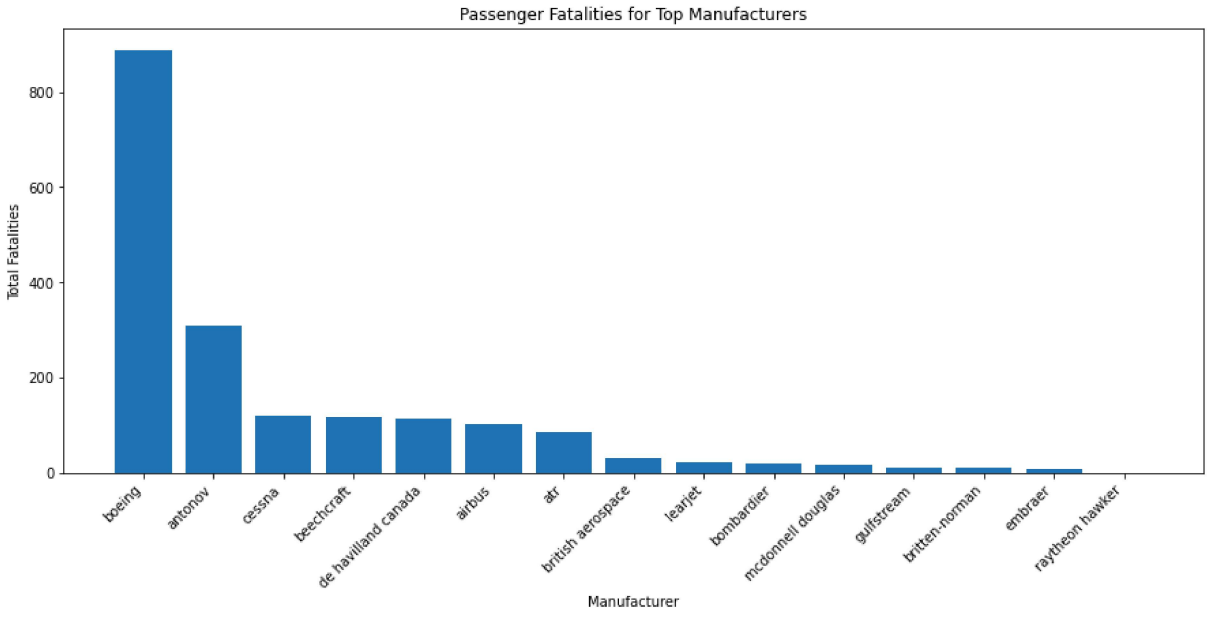
In [65]:

```
# Using top 15 manufacturers by accident count (same as before for consistency)
# Aggregate fatalities by manufacturer
```

```
fatalities_by_manufacturer = top_manuf.groupby('Manufacturer')['Fatalities'].sum().sort_values(a

#preview the dataframe
fatalities_by_manufacturer
```

Out[65]:
```
Manufacturer
boeing                889
antonov               310
cessna                120
beechcraft            117
de havilland canada   114
airbus                102
atr                    85
british aerospace      31
learjet                24
bombardier             20
mcdonnell douglas      17
gulfstream             13
britten-norman         11
embraer                 8
raytheon hawker         1
Name: Fatalities, dtype: Int64
```

In [66]:
```
#Create the plot figure
plt.figure(figsize=(15, 6))

#Create the x and y axes values
x = fatalities_by_manufacturer.index
y =fatalities_by_manufacturer.values

plt.bar(x, y)
plt.xlabel('Manufacturer')
plt.ylabel('Total Fatalities')
plt.title('Passenger Fatalities for Top Manufacturers')

# Rotate x-axis labels (e.g., 45 degrees)
plt.xticks(rotation=45, ha='right')

#plt.show to remove printed out values
plt.show()
```



In [67]:
```
"""Interpretation:  This chart shows total passenger fatalities for the top manufacturers. Lower
                    complementing the aircraft damage analysis.
                    Together, these metrics provide a clearer picture of which manufacturers pro
```

```
    For instance, it was noted previously that boeing airbus and embreaer had go
    boeing has poor passenger survivability in general.
"""
```

Out[67]:
```
'Interpretation:  This chart shows total passenger fatalities for the top manufacturers. Lower f
atality counts suggest stronger passenger survivability,\n                    complementing the
aircraft damage analysis. \n                    Together, these metrics provide a clearer pictur
e of which manufacturers produce the safest aircraft.\n                    For instance, it was
noted previously that boeing airbus and embreaer had good aircraft survivability, however, \n
boeing has poor passenger survivability in general. \n'
```

4.4 Most survivable aircraft models

Business question: Which specific aircraft models should be prioritized for acquisition based on survivability?

Explore models of aircraft by passenger and plane survivability to prioritize a few

In [68]:
```
"""Based on the visualizations above, the aircraft manufacturers can be narrowed down based on h
    Plane survivability: 'boeing', 'airbus' , 'embraer'
    Human survivability: 'britten-norman', 'raytheon hawker', 'gulfstream'
    Taking a better look at these models
"""
```

Out[68]:
```
"Based on the visualizations above, the aircraft manufacturers can be narrowed down based on hum
an survivability and plane survivability\n    Plane survivability: 'boeing', 'airbus' , 'embrae
r'\n    Human survivability: 'britten-norman', 'raytheon hawker', 'gulfstream'\n    Taking a bet
ter look at these models\n"
```

In [69]:
```
#Putting these into a list
top_survivable_makes = ['boeing','airbus','embraer','britten-norman','raytheon hawker','gulfstre

# Filter the dataframe
selected_models = ad[ad['Manufacturer'].isin(top_survivable_makes)]
```

In [70]:
```
#Aggregate by model

#group the selected models manufacturer by model as well
model_stats = selected_models.groupby(['Manufacturer', 'Model']).agg(
    total_accidents=('Fatalities', 'size'),  #use size as we don't have a unique id
    total_fatalities=('Fatalities', 'sum') #sum by total fatalities
).reset_index() #reset the index to get a good new dataframe

#preview the df
model_stats
```

Out[70]:

	Manufacturer	Model	total_accidents	total_fatalities
0	airbus	a220-100	1	0
1	airbus	a300b4-203 (f)	2	0
2	airbus	a300b4-622r (f)	1	0
3	airbus	a310-304	2	0
4	airbus	a319	1	0
...	...	...	...	...
233	raytheon hawker	hs-125	1	0
234	raytheon hawker	hs-125-400	1	0
235	raytheon hawker	hs-125-600a	1	1

236	raytheon hawker	hs-125-700a	1	0
237	raytheon hawker	hs-125-f400b	1	0

238 rows × 4 columns

In [71]:
```
#Create a survivability metric
#total fatalities divided by total accidents for our case =fatalities per accident for each mode
model_stats['fatality_score'] = model_stats['total_fatalities'] / model_stats['total_accidents']

#Select top n models per manufacturer, for our case n=3
best_buy = (
    model_stats
    .sort_values(['Manufacturer', 'fatality_score']) #sort by manufacturer and fatalities
    .groupby('Manufacturer') #group by manufacturer
    .head(3)  # top 3 models per manufacturer
)
best_buy
```

Out[71]:

	Manufacturer	Model	total_accidents	total_fatalities	fatality_score
0	airbus	a220-100	1	0	0.0
1	airbus	a300b4-203 (f)	2	0	0.0
2	airbus	a300b4-622r (f)	1	0	0.0
44	boeing	717-2bd	2	0	0.0
45	boeing	727-2b6 adv. (f)	1	0	0.0
47	boeing	737 max 8-200	1	0	0.0
172	britten-norman	bn-2a-21 islander	2	0	0.0
173	britten-norman	bn-2a-26 islander	2	0	0.0
174	britten-norman	bn-2a-27 islander	4	0	0.0
182	embraer	170-200 lr (erj-175lr)	1	0	0.0
183	embraer	170lr (erj-170-100 lr)	1	0	0.0
184	embraer	190ar	1	0	0.0
212	gulfstream	(g450)	1	0	0.0
213	gulfstream	?	1	0	0.0
214	gulfstream	g iv	1	0	0.0
226	raytheon hawker	1000	1	0	0.0
227	raytheon hawker	390 premier i	1	0	0.0
228	raytheon hawker	400a	6	0	0.0

In [72]:
```
"""Plotting this would be pointless. However, it can be observed that in all of these, the fatal
    even for above 4 accidents per aircraft model indicating very good survivability.
"""
```

Out[72]:
```
'Plotting this would be pointless. However, it can be observed that in all of these, the fatalit
ies are none, \n    even for above 4 accidents per aircraft model indicating very good survivabi
lity.\n'
```

Lets see the ones with poorest survivability for each of these models instead, to see the ones bringing down the average

In [73]:
```
poor_buy = (
    model_stats
    .sort_values(['Manufacturer', 'fatality_score']) #sort by manufacturer and fatalities
    .groupby('Manufacturer') #group by manufacturer
    .tail(3)  # top 3 models per manufacturer
)
poor_buy
```

Out[73]:

	Manufacturer	Model	total_accidents	total_fatalities	fatality_score
19	airbus	a320-251n	4	2	0.500000
20	airbus	a320-271n	4	2	0.500000
14	airbus	a320-214	11	98	8.909091
46	boeing	737 max 8	3	346	115.333333
90	boeing	737-89p (wl)	1	132	132.000000
108	boeing	737-8kv (wl)	1	176	176.000000
171	britten-norman	bn-2a-20 islander	1	1	1.000000
175	britten-norman	bn-2a-6 islander	1	4	4.000000
180	britten-norman	bn-2b-27 islander	1	6	6.000000
204	embraer	erj-175lr (erj-170-200 lr)	2	1	0.500000
206	embraer	erj-190ar	2	1	0.500000