Aircraft Risk Analysis

- Student name: Leo Rono
- Student pace: Full Time Online
- Instructor name: Kaleha Lucille

Business Understanding

Overview

The company plans to diversify its portfolio by entering the aviation industry, focusing on acquiring and operating airplanes for commercial and private clients. This project involves analyzing aviation accident data obtained from Kaggle to identify aircraft with the lowest risk profile. By conducting a descriptive analysis of factors such as accident frequency, severity, and related variables, the study will provide insights into which aircraft present the least safety, financial, and operational risks. This analysis will guide the company's decision-making process in selecting the most suitable airplanes for purchase.

Business Problem

To reduce safety and financial liabilities in aircraft operations, the company can focus on selecting the safest and most reliable airplane models. This project aims to:

1. Identify airplane models with minimal risk profiles
2. Analyze the severity and frequency of aviation accidents.
3. Examine factors contributing to these accidents.
4. Offer data-driven recommendations for choosing the most suitable airplanes.

Doing so will help the company to make informed decisions on which airplanes to purchase.

Data Understanding

The [aviation accident dataset (https://www.kaggle.com/datasets/khsamaha/aviation-accident-database-synopses)](https://www.kaggle.com/datasets/khsamaha/aviation-accident-database-synopses) sourced from Kaggle originally obtained from the [National Transportation Safety Board (https://www.ntsb.gov/Pages/home.aspx)](https://www.ntsb.gov/Pages/home.aspx) contains comprehensive records of airplane accidents. Each accident is uniquely identified by its 'Accident Number' and includes key details such as the date, location, aircraft make and model, and injury severity. Additionally, the dataset records factors such as weather conditions and the flight phase, enabling a thorough analysis of accident patterns, risk factors, and the relationships between aircraft models, accident severity, and environmental influences.

In [1]:
```
# Import standard packages

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

%matplotlib inline
```

In [2]:
```
# load dataset

aviation_data = pd.read_csv('data/Aviation_Data.csv', low_memory=False)
aviation_data.head()
```

Out[2]:

	Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Countr
0	20001218X45444	Accident	SEA87LA080	1948-10-24	MOOSE CREEK, ID	Unite State
1	20001218X45447	Accident	LAX94LA336	1962-07-19	BRIDGEPORT, CA	Unite State
2	20061025X01555	Accident	NYC07LA005	1974-08-30	Saltville, VA	Unite State
3	20001218X45448	Accident	LAX96LA321	1977-06-19	EUREKA, CA	Unite State
4	20041105X01764	Accident	CHI79FA064	1979-08-02	Canton, OH	Unite State

5 rows × 31 columns

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 90348 entries, 0 to 90347
Data columns (total 31 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Event.Id                88889 non-null  object 
 1   Investigation.Type      90348 non-null  object 
 2   Accident.Number         88889 non-null  object 
 3   Event.Date              88889 non-null  object 
 4   Location                88837 non-null  object 
 5   Country                 88663 non-null  object 
 6   Latitude                34382 non-null  object 
 7   Longitude               34373 non-null  object 
 8   Airport.Code            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        73659 non-null  object 
dtypes: float64(5), object(26)
memory usage: 21.4+ MB
```

In [4]:
```
aviation_data.shape
```

Out[4]:
```
(90348, 31)
```

In [5]:
```
aviation_data.columns
```

Out[5]:
```
Index(['Event.Id', 'Investigation.Type', 'Accident.Number', 'Event.Dat
e',
       '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.Descrip
tion',
       'Schedule', 'Purpose.of.flight', 'Air.carrier', 'Total.Fatal.Inju
ries',
       'Total.Serious.Injuries', 'Total.Minor.Injuries', 'Total.Uninjure
d',
       'Weather.Condition', 'Broad.phase.of.flight', 'Report.Status',
       'Publication.Date'],
      dtype='object')
```

In [6]:
```
aviation_data.describe()
```

Out[6]:

	Number.of.Engines	Total.Fatal.Injuries	Total.Serious.Injuries	Total.Minor.Injuries	Tota
count	82805.000000	77488.000000	76379.000000	76956.000000	82
mean	1.146585	0.647855	0.279881	0.357061	
std	0.446510	5.485960	1.544084	2.235625	
min	0.000000	0.000000	0.000000	0.000000	
25%	1.000000	0.000000	0.000000	0.000000	
50%	1.000000	0.000000	0.000000	0.000000	
75%	1.000000	0.000000	0.000000	0.000000	
max	8.000000	349.000000	161.000000	380.000000	

Data Preparation

Data Cleaning

In [7]:
```
# Make column names easier to use

aviation_data.columns = aviation_data.columns.str.lower().str.replace('.',
```

In [8]:
```
# Check for duplicate entries based on accident_number column since it is

aviation_data.duplicated('accident_number').sum()
```

Out[8]: 1484

In [9]:
```
# Remove duplicates

aviation_data = aviation_data.drop_duplicates(subset='accident_number', ke
```

In [10]:
```
# Check for null values

aviation_data.isna().sum()
```

Out[10]:
```
event_id                      1
investigation_type            0
accident_number               1
event_date                    1
location                     53
country                     227
latitude                  54501
longitude                 54510
airport_code              38631
airport_name              36090
injury_severity             991
aircraft_damage            3186
aircraft_category         56601
registration_number        1318
make                         64
model                        93
amateur_built               103
number_of_engines          6075
engine_type                7058
far_description           56867
schedule                  76288
purpose_of_flight          6182
air_carrier               72229
total_fatal_injuries      11402
total_serious_injuries    12511
total_minor_injuries      11934
total_uninjured            5913
weather_condition          4482
broad_phase_of_flight     27140
report_status              6362
publication_date          15219
dtype: int64
```

In [11]:
```
# Drop rows with null values in the primary key column; 'accident_number'

aviation_data = aviation_data.dropna(subset = ['accident_number'])
```

In [12]:

```
# Check the percentage of mising values for every column

aviation_data.isna().sum()/len(aviation_data)*100
```

Out[12]:

```
event_id                   0.000000
investigation_type         0.000000
accident_number            0.000000
event_date                 0.000000
location                   0.058517
country                    0.254324
latitude                  61.330362
longitude                 61.340490
airport_code              43.471411
airport_name              40.611953
injury_severity            1.114074
aircraft_damage            3.584169
aircraft_category         63.693551
registration_number        1.482057
make                       0.070896
model                      0.103530
amateur_built              0.114783
number_of_engines          6.835241
engine_type                7.941438
far_description           63.992888
schedule                  85.847878
purpose_of_flight          6.955651
air_carrier               81.280173
total_fatal_injuries      12.829862
total_serious_injuries    14.077850
total_minor_injuries      13.428536
total_uninjured            6.652938
weather_condition          5.042594
broad_phase_of_flight     30.540270
report_status              7.158210
publication_date          17.125238
dtype: float64
```

In [13]:

```
# Drop columns that have more than 35% of their data missing

drop_columns =  ['latitude', 'longitude', 'airport_code', 'airport_name',
                'air_carrier']
aviation_data = aviation_data.drop(columns = drop_columns)
```

In [14]:

```
# Drop columns that are irrelevant to my analysis

drop_columns_2 = ['event_id', 'accident_number', 'location', 'country', 'r
                  'publication_date']
aviation_data = aviation_data.drop(columns = drop_columns_2)
```

In [15]:
```
aviation_data.isna().sum()
```

Out[15]:
```
investigation_type            0
event_date                    0
injury_severity             990
aircraft_damage            3185
make                         63
model                        92
amateur_built               102
number_of_engines          6074
engine_type                7057
purpose_of_flight          6181
total_fatal_injuries      11401
total_serious_injuries    12510
total_minor_injuries      11933
total_uninjured            5912
weather_condition          4481
dtype: int64
```

In [16]:
```
# Drop the rows with missing values

aviation_data = aviation_data.dropna(subset=['make', 'model', 'amateur_bui
                                             'total_serious_injuries', 'tot
```

In [17]:
```
# Fill missing values of dtype object columns with 'Unknown'

aviation_data.fillna('Unknown', inplace=True)
```

In [18]:
```
aviation_data.isna().sum()
```

Out[18]:
```
investigation_type        0
event_date                0
injury_severity           0
aircraft_damage           0
make                      0
model                     0
amateur_built             0
number_of_engines         0
engine_type               0
purpose_of_flight         0
total_fatal_injuries      0
total_serious_injuries    0
total_minor_injuries      0
total_uninjured           0
weather_condition         0
dtype: int64
```

In [19]:
```
# Format the columns with entries of type string

columns = ['investigation_type', 'aircraft_damage', 'make', 'amateur_built
           'weather_condition']

for column in columns:
    aviation_data[column] = aviation_data[column].str.strip()
    aviation_data[column] = aviation_data[column].str.lower()
```

In [20]:
```
# Standardize missing data representations

aviation_data['injury_severity'].replace('unavailable', 'unknown', inplace
aviation_data['engine_type'].replace('unk', 'unknown', inplace=True)
aviation_data['weather_condition'].replace('unk', 'unknown', inplace=True)
aviation_data['model'].replace('unk', 'unknown', inplace=True)
```

In [21]:
```
# Create year column for future analysis

aviation_data['year'] = [date[:4] for date in aviation_data['event_date']]
```

In [22]:
```
# Reset the index of the dataframe

aviation_data.reset_index(drop=True, inplace=True)
```

In [23]:

```
aviation_data
```

Out[23]:

	investigation_type	event_date	injury_severity	aircraft_damage	make	mode
0	accident	1948-10-24	Fatal(2)	destroyed	stinson	108-
1	accident	1962-07-19	Fatal(4)	destroyed	piper	PA24 18
2	accident	1977-06-19	Fatal(2)	destroyed	rockwell	11
3	accident	1981-08-01	Fatal(4)	destroyed	cessna	18
4	accident	1982-01-01	Non-Fatal	substantial	cessna	14
...	...	...	...	...	...	..
69574	accident	2022-12-13	Non-Fatal	substantial	piper	PA4
69575	accident	2022-12-14	Non-Fatal	substantial	cirrus design corp	SR2
69576	accident	2022-12-15	Non-Fatal	substantial	swearingen	SA226T
69577	accident	2022-12-16	Minor	substantial	cessna	R172
69578	accident	2022-12-26	Non-Fatal	substantial	american champion aircraft	8GCB

69579 rows × 16 columns

In [24]:

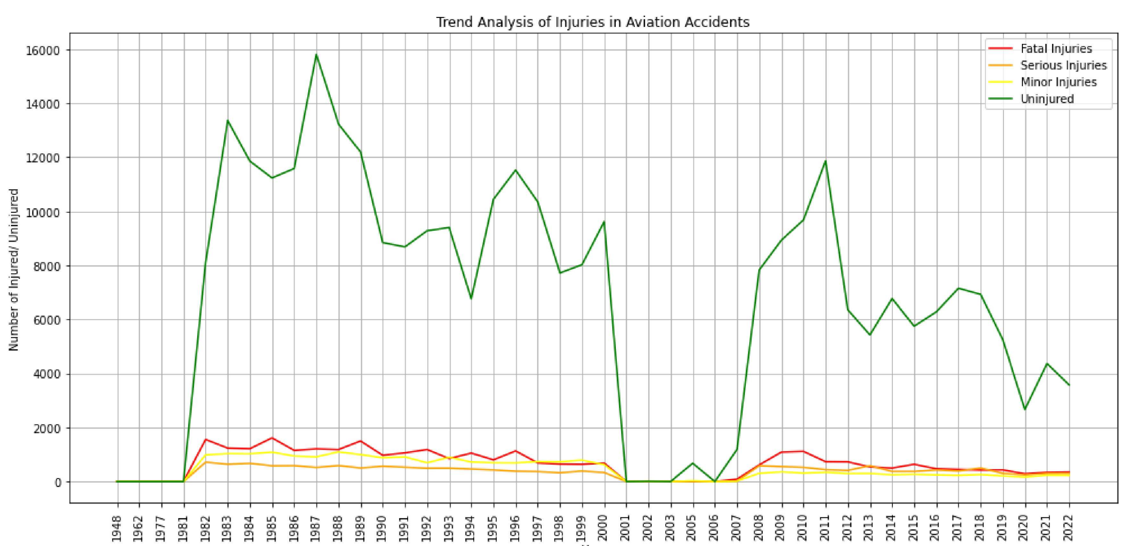
```
# Save cleaned data as excel
aviation_data.to_csv('./data/cleaned_aviation_data.csv', index=False)
```

Data Analysis

Trend Analysis of Injuries and Uninjured Passengers in Aviation Accidents Over Time

In [25]:

```
# Group by year
year_grouped = aviation_data.groupby('year').agg({
    'total_fatal_injuries': 'sum',
    'total_serious_injuries': 'sum',
    'total_minor_injuries': 'sum',
    'total_uninjured': 'sum'
}).reset_index()

# Plot
plt.figure(figsize=(14, 7))

plt.plot(year_grouped['year'], year_grouped['total_fatal_injuries'], label
plt.plot(year_grouped['year'], year_grouped['total_serious_injuries'], lab
plt.plot(year_grouped['year'], year_grouped['total_minor_injuries'], label
plt.plot(year_grouped['year'], year_grouped['total_uninjured'], label='Uni

# Adding labels and title
plt.title('Trend Analysis of Injuries in Aviation Accidents')
plt.xlabel('Year')
plt.ylabel('Number of Injured/ Uninjured')
plt.legend()
plt.xticks(rotation=90)

plt.tight_layout()
plt.grid()
```



It appears that the total number of uninjured passengers has consistently been higher than the total number of fatal, serious, or minor injuries from 1948 to 2022, which is a positive sign. While there have been fluctuations throughout this period, it's worth noting that the number of uninjured passengers has generally decreased since 1987.

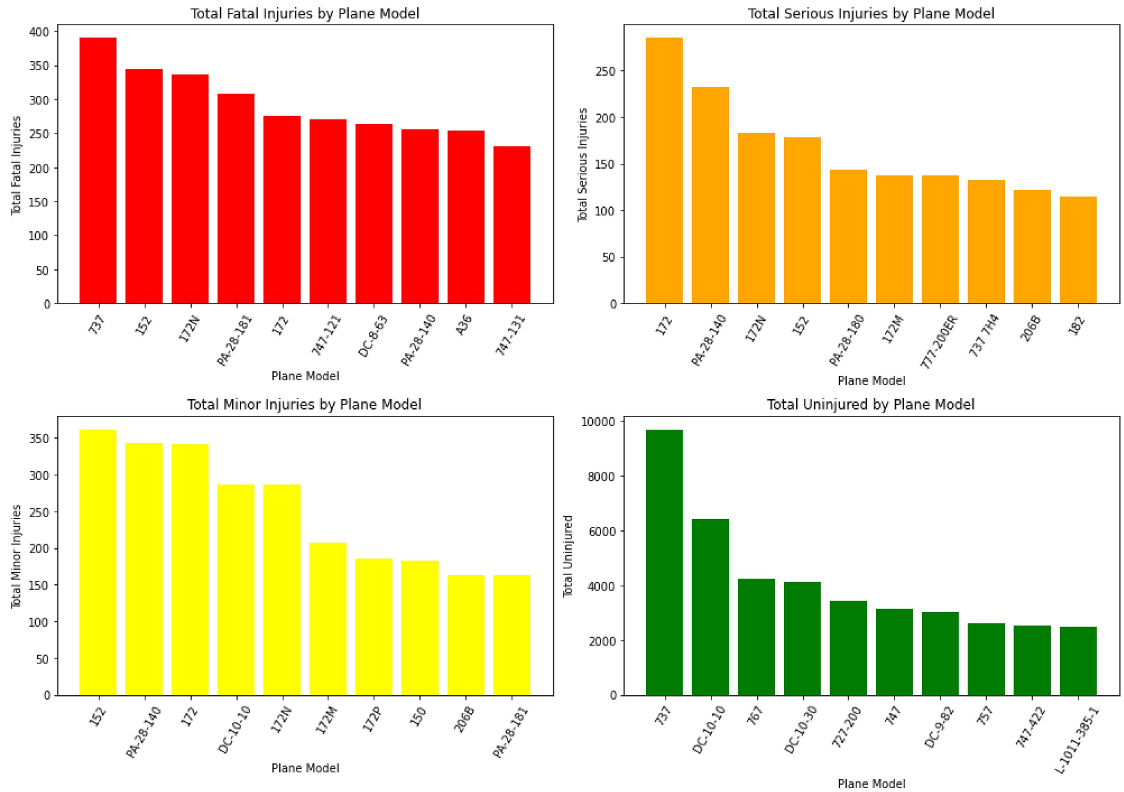
In contrast, the number of fatal, serious, and minor injuries has remained relatively low, staying below 2,000 people over the 74-year period. This steady trend suggests that overall, injuries have decreased over the years, indicating that safety standards in aviation have improved significantly.

These trends highlight the effectiveness of safety measures and innovations over time, which have contributed to reducing the severity and frequency of injuries in aviation accidents.

Injured Passengers by the Plane Model

In [26]:

```
# Group by model
model_grouped = aviation_data.groupby('model').agg({
    'total_fatal_injuries': 'sum',
    'total_serious_injuries': 'sum',
    'total_minor_injuries': 'sum',
    'total_uninjured': 'sum'
}).reset_index()

# Create subplots
fig, axes = plt.subplots(2, 2, figsize = (14, 10))

# Plot total fatal injuries
fatal_injuries = model_grouped.sort_values(by='total_fatal_injuries', asce
axes[0, 0].bar(fatal_injuries['model'], fatal_injuries['total_fatal_injur
axes[0, 0].set_title('Total Fatal Injuries by Plane Model')
axes[0, 0].set_xlabel('Plane Model')
axes[0, 0].set_ylabel('Total Fatal Injuries')
axes[0, 0].tick_params(axis='x', rotation=60)

# Plot total serious injuries
serious_injuries = model_grouped.sort_values(by='total_serious_injuries',
axes[0, 1].bar(serious_injuries['model'], serious_injuries['total_serious_
axes[0, 1].set_title('Total Serious Injuries by Plane Model')
axes[0, 1].set_xlabel('Plane Model')
axes[0, 1].set_ylabel('Total Serious Injuries')
axes[0, 1].tick_params(axis='x', rotation=60)

# Plot total minor injuries
minor_injuries = model_grouped.sort_values(by='total_minor_injuries', asce
axes[1, 0].bar(minor_injuries['model'], minor_injuries['total_minor_injur
axes[1, 0].set_title('Total Minor Injuries by Plane Model')
axes[1, 0].set_xlabel('Plane Model')
axes[1, 0].set_ylabel('Total Minor Injuries')
axes[1, 0].tick_params(axis='x', rotation=60)

# Plot total uninjured
uninjured = model_grouped.sort_values(by='total_uninjured', ascending=Fals
axes[1, 1].bar(uninjured['model'], uninjured['total_uninjured'], color='gr
axes[1, 1].set_title('Total Uninjured by Plane Model')
axes[1, 1].set_xlabel('Plane Model')
axes[1, 1].set_ylabel('Total Uninjured')
axes[1, 1].tick_params(axis='x', rotation=60)

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



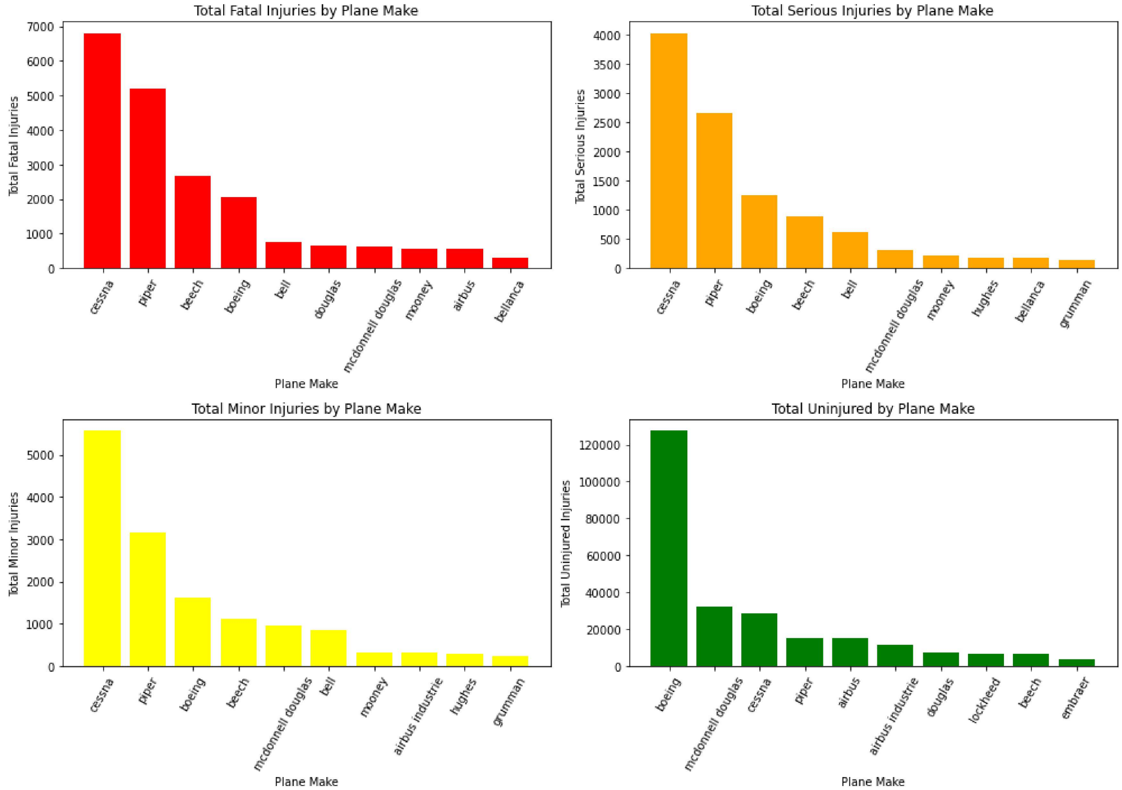
- Model 737 has the most fatal injuries.
- Model 172 has the most serious injuries.
- Model 152 has the most minor injuries.
- Model 737 has the most uninjured.

Even though model 737 has the most fatal injuries, it also has the most uninjured which is positive.

Injured Passengers by Plane Make

In [27]:

```
# Group by make
make_grouped = aviation_data.groupby('make').agg({
    'total_fatal_injuries': 'sum',
    'total_serious_injuries': 'sum',
    'total_minor_injuries': 'sum',
    'total_uninjured': 'sum'
}).reset_index()

# Create subplots
fig, axes = plt.subplots(2, 2, figsize = (14, 10))

# Plot total fatal injuries
fatal_injuries = make_grouped.sort_values(by='total_fatal_injuries', ascen
axes[0, 0].bar(fatal_injuries['make'], fatal_injuries['total_fatal_injurie
axes[0, 0].set_title('Total Fatal Injuries by Plane Make')
axes[0, 0].set_xlabel('Plane Make')
axes[0, 0].set_ylabel('Total Fatal Injuries')
axes[0, 0].tick_params(axis='x', rotation=60)

# Plot total serious injuries
serious_injuries = make_grouped.sort_values(by='total_serious_injuries', a
axes[0, 1].bar(serious_injuries['make'], serious_injuries['total_serious_i
axes[0, 1].set_title('Total Serious Injuries by Plane Make')
axes[0, 1].set_xlabel('Plane Make')
axes[0, 1].set_ylabel('Total Serious Injuries')
axes[0, 1].tick_params(axis='x', rotation=60)

# Plot total minor injuries
minor_injuries = make_grouped.sort_values(by='total_minor_injuries', ascen
axes[1, 0].bar(minor_injuries['make'], minor_injuries['total_minor_injurie
axes[1, 0].set_title('Total Minor Injuries by Plane Make')
axes[1, 0].set_xlabel('Plane Make')
axes[1, 0].set_ylabel('Total Minor Injuries')
axes[1, 0].tick_params(axis='x', rotation=60)

# Plot total uninjured
uninjured = make_grouped.sort_values(by='total_uninjured', ascending=False
axes[1, 1].bar(uninjured['make'], uninjured['total_uninjured'], color='gre
axes[1, 1].set_title('Total Uninjured by Plane Make')
axes[1, 1].set_xlabel('Plane Make')
axes[1, 1].set_ylabel('Total Uninjured Injuries')
axes[1, 1].tick_params(axis='x', rotation=60)

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

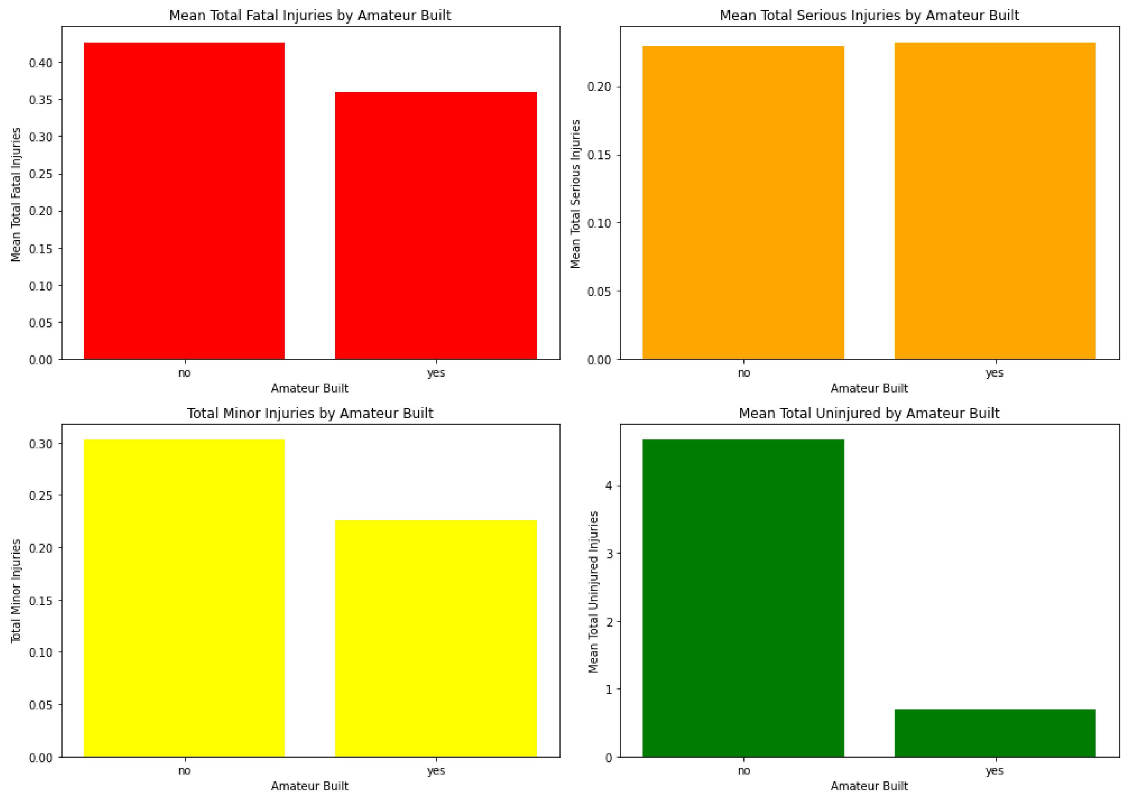


- Cessna has the most fatal, serious and minor injuries reported.
- Boeing has the most uninjured reported.

Injured Passengers by Amateur Built

In [28]:

```
# Group by amateur built
amateur_built_grouped = aviation_data.groupby('amateur_built').agg({
    'total_fatal_injuries': 'mean',
    'total_serious_injuries': 'mean',
    'total_minor_injuries': 'mean',
    'total_uninjured': 'mean'
}).reset_index()

# Create subplots
fig, axes = plt.subplots(2, 2, figsize = (14, 10))

# Plot total fatal injuries
axes[0, 0].bar(amateur_built_grouped['amateur_built'], amateur_built_grour
axes[0, 0].set_title('Mean Total Fatal Injuries by Amateur Built')
axes[0, 0].set_xlabel('Amateur Built')
axes[0, 0].set_ylabel('Mean Total Fatal Injuries')

# Plot total serious injuries
axes[0, 1].bar(amateur_built_grouped['amateur_built'], amateur_built_grour
axes[0, 1].set_title('Mean Total Serious Injuries by Amateur Built')
axes[0, 1].set_xlabel('Amateur Built')
axes[0, 1].set_ylabel('Mean Total Serious Injuries')

# Plot total minor injuries
axes[1, 0].bar(amateur_built_grouped['amateur_built'], amateur_built_grour
axes[1, 0].set_title('Total Minor Injuries by Amateur Built')
axes[1, 0].set_xlabel('Amateur Built')
axes[1, 0].set_ylabel('Total Minor Injuries')

# Plot total amateur_built_grouped
axes[1, 1].bar(amateur_built_grouped['amateur_built'], amateur_built_grour
axes[1, 1].set_title('Mean Total Uninjured by Amateur Built')
axes[1, 1].set_xlabel('Amateur Built')
axes[1, 1].set_ylabel('Mean Total Uninjured Injuries')

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



It is shown that:

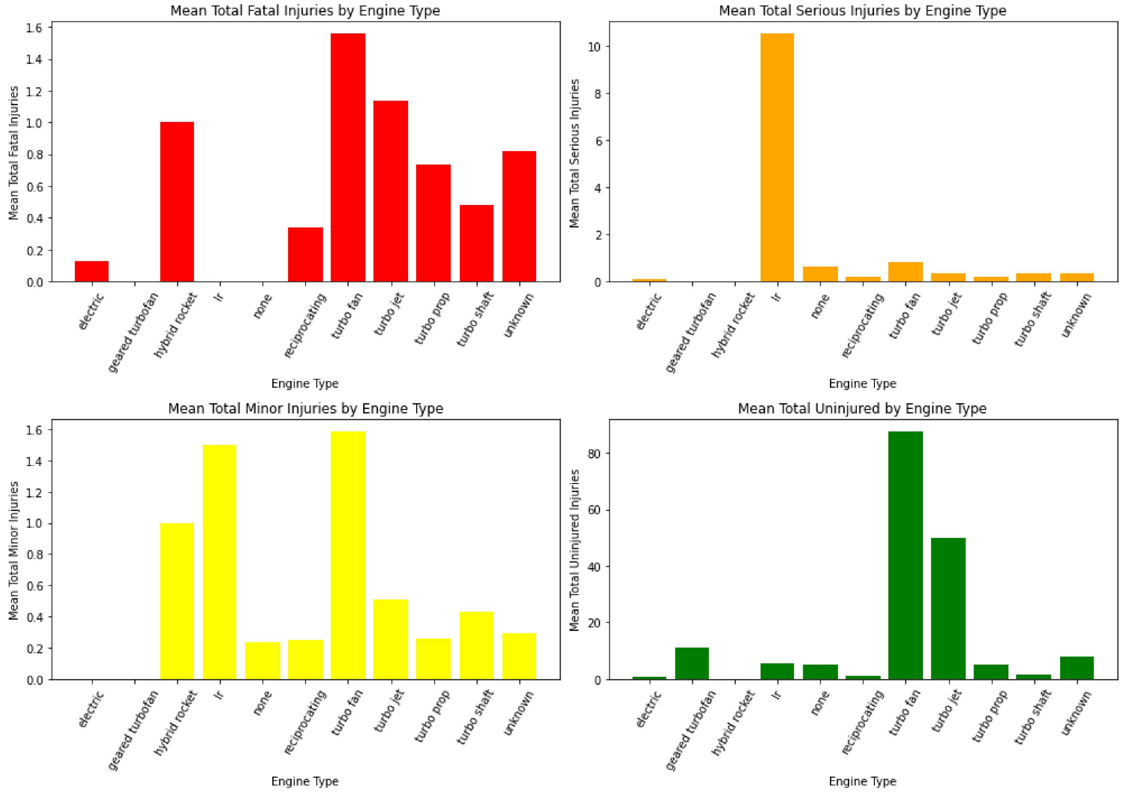
- Non-amateur-built planes have more fatal, minor and uninjured.
- Amateur-built planes have more serious injuries. As shown in the data, the gap between non-amateur-built and amateur-built planes in terms of uninjured passengers is more significant than the gap for fatal injuries. This suggests that non-amateur-built planes tend to have a higher number of uninjured passengers, which implies that they are generally safer, even though they report a higher number of fatal injuries.

While non-amateur-built planes may experience more fatal injuries, the overall safety profile, as indicated by the higher number of uninjured passengers, suggests that professionally built aircraft are better equipped to protect passengers in the event of an accident. This further supports the idea that the safety features and design of professionally built planes contribute to minimizing the severity of injuries.

Injured Passengers by Type of Engine of the Plane

In [29]:

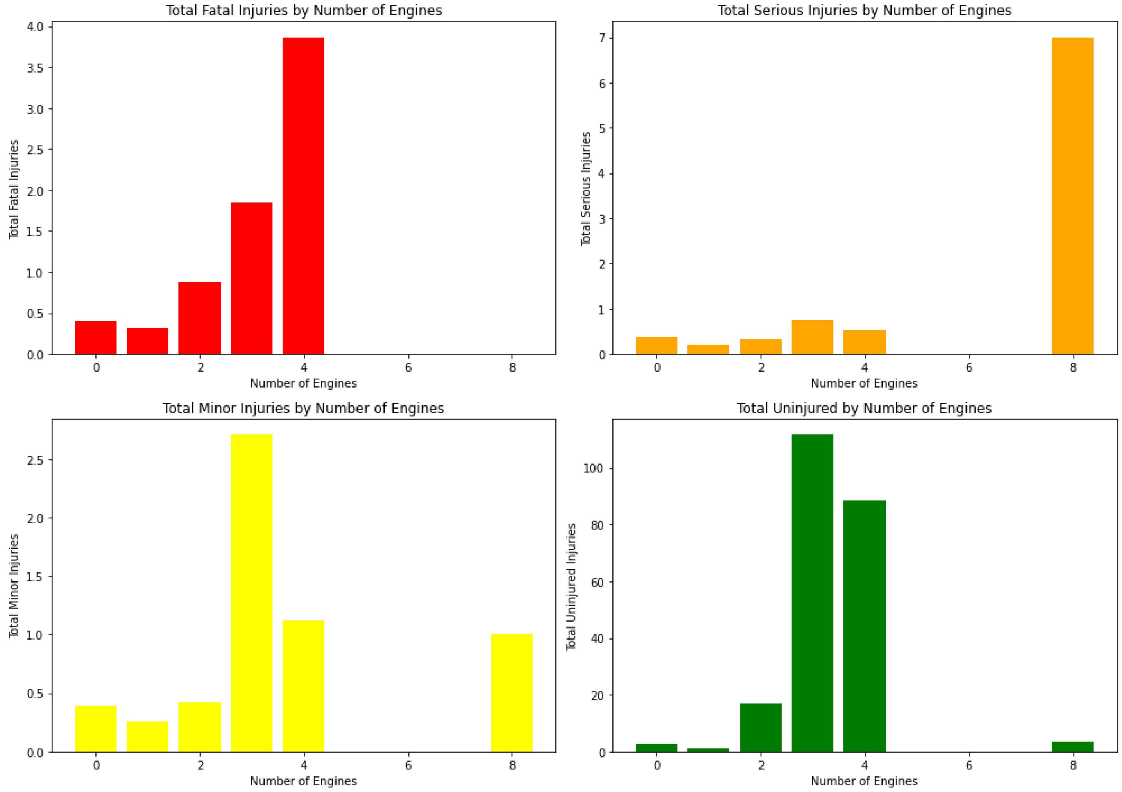
```
# Group by engine type
engine_type_grouped = aviation_data.groupby('engine_type').agg({
    'total_fatal_injuries': 'mean',
    'total_serious_injuries': 'mean',
    'total_minor_injuries': 'mean',
    'total_uninjured': 'mean'
}).reset_index()

# Create subplots
fig, axes = plt.subplots(2, 2, figsize = (14, 10))

# Plot total fatal injuries
axes[0, 0].bar(engine_type_grouped['engine_type'], engine_type_grouped['to
axes[0, 0].set_title('Mean Total Fatal Injuries by Engine Type')
axes[0, 0].set_xlabel('Engine Type')
axes[0, 0].set_ylabel('Mean Total Fatal Injuries')
axes[0, 0].tick_params(axis='x', rotation=60)

# Plot total serious injuries
axes[0, 1].bar(engine_type_grouped['engine_type'], engine_type_grouped['to
axes[0, 1].set_title('Mean Total Serious Injuries by Engine Type')
axes[0, 1].set_xlabel('Engine Type')
axes[0, 1].set_ylabel('Mean Total Serious Injuries')
axes[0, 1].tick_params(axis='x', rotation=60)

# Plot total minor injuries
axes[1, 0].bar(engine_type_grouped['engine_type'], engine_type_grouped['to
axes[1, 0].set_title('Mean Total Minor Injuries by Engine Type')
axes[1, 0].set_xlabel('Engine Type')
axes[1, 0].set_ylabel('Mean Total Minor Injuries')
axes[1, 0].tick_params(axis='x', rotation=60)

# Plot total uninjured
axes[1, 1].bar(engine_type_grouped['engine_type'], engine_type_grouped['to
axes[1, 1].set_title('Mean Total Uninjured by Engine Type')
axes[1, 1].set_xlabel('Engine Type')
axes[1, 1].set_ylabel('Mean Total Uninjured Injuries')
axes[1, 1].tick_params(axis='x', rotation=60)

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



- Turbo tan engine has the most fatal, minor injuries and uninjured.
- Lr engine has the most serious injuries.

Injured Passengers by Number of Engines in a Plane

In [30]:
```
# Group by number of engines
number_of_engines_grouped = aviation_data.groupby('number_of_engines').agg
    'total_fatal_injuries': 'mean',
    'total_serious_injuries': 'mean',
    'total_minor_injuries': 'mean',
    'total_uninjured': 'mean'
}).reset_index()

# Create subplots
fig, axes = plt.subplots(2, 2, figsize = (14, 10))

# Plot total fatal injuries
axes[0, 0].bar(number_of_engines_grouped['number_of_engines'], number_of_e
axes[0, 0].set_title('Total Fatal Injuries by Number of Engines')
axes[0, 0].set_xlabel('Number of Engines')
axes[0, 0].set_ylabel('Total Fatal Injuries')

# Plot total serious injuries
axes[0, 1].bar(number_of_engines_grouped['number_of_engines'], number_of_e
axes[0, 1].set_title('Total Serious Injuries by Number of Engines')
axes[0, 1].set_xlabel('Number of Engines')
axes[0, 1].set_ylabel('Total Serious Injuries')

# Plot total minor injuries
axes[1, 0].bar(number_of_engines_grouped['number_of_engines'], number_of_e
axes[1, 0].set_title('Total Minor Injuries by Number of Engines')
axes[1, 0].set_xlabel('Number of Engines')
axes[1, 0].set_ylabel('Total Minor Injuries')

# Plot total uninjured
axes[1, 1].bar(number_of_engines_grouped['number_of_engines'], number_of_e
axes[1, 1].set_title('Total Uninjured by Number of Engines')
axes[1, 1].set_xlabel('Number of Engines')
axes[1, 1].set_ylabel('Total Uninjured Injuries')

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

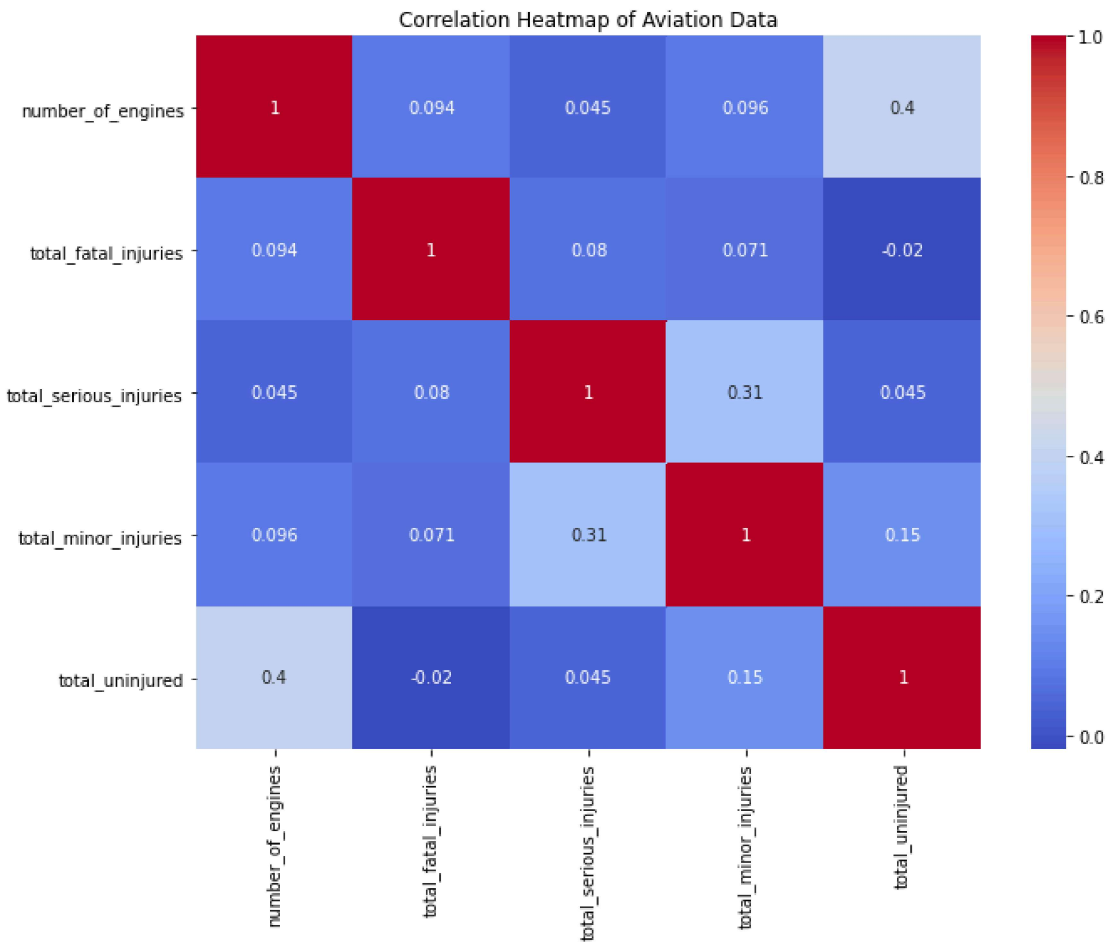


- Planes with 4 engines have the most fatal injuries.
- Plane with 8 engines have the most serious injuries.
- Planes with 3 engines have the most minor injuries and uninjured.

However, I do not think the number of engines in a plane affects the number of injured/ uninjured. Let's find the correlation between the number of engines and the injuries/ uninjured.

Correlation Between Number of Engines and Injuries/ Uninjured

In [31]:
```
# Create subplots
fig, ax = plt.subplots(figsize = (10, 8))

# Plot
sns.heatmap(aviation_data.corr(), annot=True, cmap='coolwarm')
ax.set_title('Correlation Heatmap of Aviation Data')

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



Upon examining the heatmap, the highest correlation coefficient, apart from 1, is 0.4, indicating a weak positive correlation between certain variables. The lowest correlation is -0.02, representing a very weak negative correlation.

Notably, when focusing on the number of engines column, its correlation with other variables is consistently below 0.1. This suggests that the number of engines does not have a significant relationship with the number of injuries or uninjured passengers. Therefore, it cannot be concluded that the number of engines in an aircraft directly affects the severity of injuries or the likelihood of passengers being uninjured.

This observation reinforces the idea that other factors, such as aircraft model, make, and engine type, are more influential in determining injury outcomes, rather than the number of

Conclusion

Accidents are an unfortunate reality in aviation, and while some may be preventable, it's unrealistic to claim that any aircraft can be entirely accident-proof. Factors such as the make, model, and other features of a plane may influence the likelihood and severity of accidents, but it is equally important for the company to implement additional safety measures to minimize risk.

Based on the analysis, the following conclusions can be drawn:

- **Model, Make, and Engine Type**: These factors do influence the number of injuries and uninjured passengers. Different aircraft characteristics can impact safety outcomes, highlighting the importance of selecting the right combinations for better risk management.
- **Number of Engines**: There is no significant correlation between the number of engines in a plane and the number of injuries or uninjured passengers. This suggests that other factors, beyond engine count, play a more substantial role in influencing safety outcomes.

These insights emphasize that while aircraft specifications are important, safety measures and other preventive strategies should also be prioritized by the company.

Recommendations

I would suggest the company consider the following points:

- **Aircraft Model**: Consider the model 737. While it has a high number of fatalities, it also reports the highest number of uninjured passengers. By implementing additional safety measures, it's possible to reduce injuries and potentially increase the number of uninjured passengers.
- **Aircraft Make**: Boeing emerges as the safest option, with a high number of uninjured passengers and moderate injury figures. This make demonstrates strong safety performance overall.
- **Professionally Built Planes**: Opt for professionally built aircraft, as they tend to have a higher number of uninjured passengers compared to amateur-built planes, highlighting the benefits of expert craftsmanship in ensuring passenger safety.
- **Engine Type**: A turbojet engine should be prioritized, as it is associated with the highest number of uninjured passengers. This engine type offers a safer profile in terms of passenger injury outcomes.

If interested in a number of options, consider the following makes:

- Boeing
- Mcdonnel
- Douglas
- Piper
- Airbus

These recommendations aim to optimize safety by focusing on proven models, makes, and engineering standards.