

MICH-MAINA / moringa\_project1

Q

Type / to search

+

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

moringa\_project1 / Project\_One.ipynb

MICH-MAINA Adding project details to notebook

3a5703f · 1 minute ago

History

Preview

Code

Blame

2976 lines (2976 loc) · 211 KB

Raw

# Final Project Submission

Student name: **Michelle Wangui Maina**

Student pace: **Part-time**

Scheduled project review date/time: **28th July 2025**

Instructor name: **Fidelis Wanalwenge**

Dashboard: [https://public.tableau.com/app/profile/michelle.maina/viz/Aircraft\\_Business\\_Risk\\_Assessment/Dashboard1](https://public.tableau.com/app/profile/michelle.maina/viz/Aircraft_Business_Risk_Assessment/Dashboard1)

Github Repository: [https://github.com/MICH-MAINA/moringa\\_project1.git](https://github.com/MICH-MAINA/moringa_project1.git)

## #Business questions to answer

Question 1: Safest aircraft (fewest fatal accidents)

Question 2: Most common purpose of flight during accidents

Question 3: Countries with highest number of accidents

### 1. Import the Relevant Library

```
In [2]: import pandas as pd
```

### 2. Load the Data into a DataFrame

```
In [12]: aviation_data = pd.read_csv(r"C:\Users\mmaina\Documents\Flatiron\Project\moringa_project1\DataSets\Aviation_Data.csv", low_memory=False)
          aviation_data.head()
```

```
Out[12]:
```

Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Country	Latitude	Longitude	Airport.Code	Air
----------	--------------------	-----------------	------------	----------	---------	----------	-----------	--------------	-----

0	20001218X45444	Accident	SEA87LA080	1948-10-24	MOOSE CREEK, ID	United States	NaN	NaN	NaN
1	20001218X45447	Accident	LAX94LA336	1962-07-19	BRIDGEPORT, CA	United States	NaN	NaN	NaN
2	20061025X01555	Accident	NYC07LA005	1974-08-30	Saltville, VA	United States	36.922223	-81.878056	NaN
3	20001218X45448	Accident	LAX96LA321	1977-06-19	EUREKA, CA	United States	NaN	NaN	NaN
4	20041105X01764	Accident	CHI79FA064	1979-08-02	Canton, OH	United States	NaN	NaN	NaN

5 rows × 31 columns



3. Get the information on the file

In [6]:

```
aviation_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 90348 entries, 0 to 90347
Data columns (total 31 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Event.Id              88889 non-null  object
1   Investigation.Type     90348 non-null  object
2   Accident.Number       88889 non-null  object
3   Event.Date            88889 non-null  object
4   Location              88837 non-null  object
5   Country              88663 non-null  object
6   Latitude              34382 non-null  object
7   Longitude             34373 non-null  object
8   Airport.Code          50132 non-null  object
9   Airport.Name          52704 non-null  object
10  Injury.Severity        87889 non-null  object
11  Aircraft.damage        85695 non-null  object
12  Aircraft.Category      32287 non-null  object
13  Registration.Number    87507 non-null  object
14  Make                  88826 non-null  object
```

```

14  Name                33225 non-null object
15  Model               88797 non-null object
16  Amateur.Built       88787 non-null object
17  Number.of.Engines    82805 non-null float64
18  Engine.Type         81793 non-null object
19  FAR.Description      32023 non-null object
20  Schedule            12582 non-null object
21  Purpose.of.flight   82697 non-null object
22  Air.carrier         16648 non-null object
23  Total.Fatal.Injuries 77488 non-null float64
24  Total.Serious.Injuries 76379 non-null float64
25  Total.Minor.Injuries 76956 non-null float64
26  Total.Uninjured     82977 non-null float64
27  Weather.Condition   84397 non-null object
28  Broad.phase.of.flight 61724 non-null object
29  Report.Status       82505 non-null object
30  Publication.Date    73659 non-null object

```

dtypes: float64(5), object(26)

memory usage: 21.4+ MB

#### 4. Data Cleaning

In [60]:

```
#Dropping columns with many missing values
```

```
aviation_data.drop(columns=['Latitude', 'Longitude', 'Schedule', 'Air.carrier', 'Airport.Code', 'Airport.Name' ], inplace=True)
```

In [61]:

```
aviation_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

Index: 32254 entries, 5 to 90345

Data columns (total 25 columns):

#	Column	Non-Null Count	Dtype
0	Event.Id	32254 non-null	object
1	Investigation.Type	32254 non-null	object
2	Accident.Number	32254 non-null	object
3	Event.Date	32254 non-null	datetime64[ns]
4	Location	32245 non-null	object
5	Country	32242 non-null	object
6	Injury.Severity	31372 non-null	object
7	Aircraft.damage	30798 non-null	object
8	Aircraft.Category	32254 non-null	object

```
8 Aircraft.Category      32254 non-null object
9 Registration.Number    31970 non-null object
10 Make                  32245 non-null object
11 Model                 32254 non-null object
12 Amateur.Built         32235 non-null object
13 Number.of.Engines      28801 non-null float64
14 Engine.Type           26708 non-null object
15 Purpose.of.flight      27815 non-null object
16 Total.Fatal.Injuries   32254 non-null float64
17 Total.Serious.Injuries 32254 non-null float64
18 Total.Minor.Injuries   32254 non-null float64
19 Total.Uninjured        32254 non-null float64
20 Weather.Condition      28600 non-null object
21 Report.Status          26301 non-null object
22 Publication.Date        29606 non-null datetime64[ns]
23 total_injuries          32254 non-null float64
24 Year                   32254 non-null int32
dtypes: datetime64[ns](2), float64(6), int32(1), object(16)
memory usage: 6.3+ MB
```

In [63]:

```
#Add new column
aviation_data['total_injuries'] = aviation_data['Total.Fatal.Injuries'] + aviation_data['Total.Serious.Injuries'] + aviatio

#Create a new column for year to get year trend
aviation_data['Year'] = pd.DatetimeIndex(aviation_data['Event.Date']).year

aviation_data.head()
```

Out[63]:

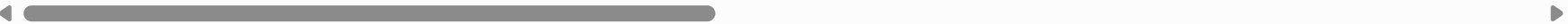
	Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Country	Injury.Severity	Aircraft.damage	Aircraft.C
5	20170710X52551	Accident	NYC79AA106	1979-09-17	BOSTON, MA	United States	Non-Fatal	Substantial	
7	20020909X01562	Accident	SEA82DA022	1982-01-01	PULLMAN, WA	United States	Non-Fatal	Substantial	
8	20020909X01561	Accident	NYC82DA015	1982-01-01	EAST HANOVER, NJ	United States	Non-Fatal	Substantial	
12	20020917X02148	Accident	FTW82FRJ07	1982-01-02	HOMER, LA	United States	Non-Fatal	Destroyed	

13	20020917X02134	Accident	FTW82FRA14	1982-01-02	HEARNE, TX	United States	Fatal(1)	Destroyed
----	----------------	----------	------------	------------	------------	---------------	----------	-----------

5 rows × 25 columns



```
In [65]: #replace null with 0 total.Fatal.Injuries      Total.Serious.Injuries      Total.Minor.Injuries      Total.Uninjured
aviation_data[['Total.Fatal.Injuries','Total.Serious.Injuries','Total.Minor.Injuries', 'Total.Uninjured', 'Number.of.Engi
aviation_data.head()
```



Out[65]:

	Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Country	Injury.Severity	Aircraft.damage	Aircraft.C
5	20170710X52551	Accident	NYC79AA106	1979-09-17	BOSTON, MA	United States	Non-Fatal	Substantial	
7	20020909X01562	Accident	SEA82DA022	1982-01-01	PULLMAN, WA	United States	Non-Fatal	Substantial	
8	20020909X01561	Accident	NYC82DA015	1982-01-01	EAST HANOVER, NJ	United States	Non-Fatal	Substantial	
12	20020917X02148	Accident	FTW82FRJ07	1982-01-02	HOMER, LA	United States	Non-Fatal	Destroyed	
13	20020917X02134	Accident	FTW82FRA14	1982-01-02	HEARNE, TX	United States	Fatal(1)	Destroyed	

5 rows × 25 columns



```
In [64]: #Delete rows where event ID is null
aviation_data = aviation_data.dropna(subset=['Event.Id', 'Model', 'Aircraft.Category'])
aviation_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 32254 entries, 5 to 90345
Data columns (total 25 columns):
```

```
Data columns (total 23 columns):
#   Column                               Non-Null Count  Dtype
---  -
0   Event.Id                             32254 non-null  object
1   Investigation.Type                     32254 non-null  object
2   Accident.Number                       32254 non-null  object
3   Event.Date                           32254 non-null  datetime64[ns]
4   Location                              32245 non-null  object
5   Country                              32242 non-null  object
6   Injury.Severity                       31372 non-null  object
7   Aircraft.damage                       30798 non-null  object
8   Aircraft.Category                     32254 non-null  object
9   Registration.Number                   31970 non-null  object
10  Make                                  32245 non-null  object
11  Model                                 32254 non-null  object
12  Amateur.Built                         32235 non-null  object
13  Number.of.Engines                     28801 non-null  float64
14  Engine.Type                           26708 non-null  object
15  Purpose.of.flight                     27815 non-null  object
16  Total.Fatal.Injuries                  32254 non-null  float64
17  Total.Serious.Injuries                32254 non-null  float64
18  Total.Minor.Injuries                  32254 non-null  float64
19  Total.Uninjured                       32254 non-null  float64
20  Weather.Condition                     28600 non-null  object
21  Report.Status                         26301 non-null  object
22  Publication.Date                      29606 non-null  datetime64[ns]
23  total_injuries                        32254 non-null  float64
24  Year                                  32254 non-null  int32
dtypes: datetime64[ns](2), float64(6), int32(1), object(16)
memory usage: 6.3+ MB
```

```
In [66]: # Convert date columns
aviation_data['Event.Date'] = pd.to_datetime(aviation_data['Event.Date'], errors='coerce')
aviation_data['Publication.Date'] = pd.to_datetime(aviation_data['Publication.Date'], errors='coerce')
aviation_data.head()
```

Out[66]:

	Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Country	Injury.Severity	Aircraft.damage	Aircraft.C
5	20170710X52551	Accident	NYC79AA106	1979-09-17	BOSTON, MA	United States	Non-Fatal	Substantial	
7	20020909X01562	Accident	SEA82DA022	1982-01-01	PULLMAN, WA	United States	Non-Fatal	Substantial	

8	20020909X01561	Accident	NYC82DA015	1982-01-01	EAST HANOVER, NJ	United States	Non-Fatal	Substantial
12	20020917X02148	Accident	FTW82FRJ07	1982-01-02	HOMER, LA	United States	Non-Fatal	Destroyed
13	20020917X02134	Accident	FTW82FRA14	1982-01-02	HEARNE, TX	United States	Fatal(1)	Destroyed

5 rows × 25 columns



```
In [67]: risk_summary = aviation_data.groupby('Make').agg({
        'Event.Id': 'count',
        'Total.Fatal.Injuries': 'sum',
        'total_injuries': 'sum'
    }).rename(columns={'Event.Id': 'accident_count'}).sort_values(by='accident_count', ascending=False)
    risk_summary.head()
```

Out[67]:

	accident_count	Total.Fatal.Injuries	total_injuries
Make			
CESSNA	4864	1886.0	3865.0
Cessna	3607	1170.0	2514.0
PIPER	2804	1237.0	2339.0
Piper	1910	661.0	1346.0
BOEING	1036	2056.0	3529.0

```
In [68]: aviation_data.info()

<class 'pandas.core.frame.DataFrame'>
Index: 32254 entries, 5 to 90345
Data columns (total 25 columns):
#   Column                                Non-Null Count  Dtype
---  -
---
```



```
0 Event.Id 32254 non-null object
1 Investigation.Type 32254 non-null object
2 Accident.Number 32254 non-null object
3 Event.Date 32254 non-null datetime64[ns]
4 Location 32245 non-null object
5 Country 32242 non-null object
6 Injury.Severity 31372 non-null object
7 Aircraft.damage 30798 non-null object
8 Aircraft.Category 32254 non-null object
9 Registration.Number 31970 non-null object
10 Make 32245 non-null object
11 Model 32254 non-null object
12 Amateur.Built 32235 non-null object
13 Number.of.Engines 32254 non-null float64
14 Engine.Type 26708 non-null object
15 Purpose.of.flight 27815 non-null object
16 Total.Fatal.Injuries 32254 non-null float64
17 Total.Serious.Injuries 32254 non-null float64
18 Total.Minor.Injuries 32254 non-null float64
19 Total.Uninjured 32254 non-null float64
20 Weather.Condition 28600 non-null object
21 Report.Status 26301 non-null object
22 Publication.Date 29606 non-null datetime64[ns]
23 total_injuries 32254 non-null float64
24 Year 32254 non-null int32
```

dtypes: datetime64[ns](2), float64(6), int32(1), object(16)  
memory usage: 6.3+ MB

```
In [70]: aviation_data['Injury.Severity'] = aviation_data['Injury.Severity'].str.upper()
         aviation_data['Injury.Severity'] = aviation_data['Injury.Severity'].replace(
           to_replace=r'FATAL\\(\\d+\\)|FATAL', value='FATAL', regex=True)

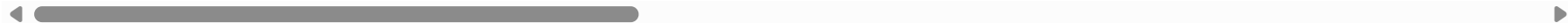
         aviation_data.head()
```

Out[70]:

	Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Country	Injury.Severity	Aircraft.damage	Aircraft.C
5	20170710X52551	Accident	NYC79AA106	1979-09-17	BOSTON, MA	United States	NON-FATAL	Substantial	
7	20020909X01562	Accident	SEA82DA022	1982-01-01	PULLMAN, WA	United States	NON-FATAL	Substantial	
				1982-01-	EAST	United			

8	20020909X01561	Accident	NYC82DA015	01	HANOVER, NJ	States	NON-FATAL	Substantial
12	20020917X02148	Accident	FTW82FRJ07	1982-01-02	HOMER, LA	United States	NON-FATAL	Destroyed
13	20020917X02134	Accident	FTW82FRA14	1982-01-02	HEARNE, TX	United States	FATAL	Destroyed

5 rows × 25 columns



In [79]:

```
#fill null vales in weather conditions with unknown and replace unk with unknown
aviation_data['Weather.Condition'] = aviation_data['Weather.Condition'].fillna('Unknown')
aviation_data['Weather.Condition'] = aviation_data['Weather.Condition'].replace('UNK', 'Unknown')

#replace null and unavilable in injuries severity with unkown
aviation_data['Injury.Severity'] = aviation_data['Injury.Severity'].fillna('Unknown')
aviation_data['Injury.Severity'] = aviation_data['Injury.Severity'].replace('UNAVAILABLE', 'Unknown')

aviation_data['Aircraft.damage'] = aviation_data['Aircraft.damage'].fillna('Unknown')

aviation_data['Report.Status'] = aviation_data['Report.Status'].fillna('Unknown')

aviation_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 32254 entries, 5 to 90345
Data columns (total 25 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Event.Id                             32254 non-null  object
1   Investigation.Type                    32254 non-null  object
2   Accident.Number                      32254 non-null  object
3   Event.Date                           32254 non-null  datetime64[ns]
4   Location                             32245 non-null  object
5   Country                             32242 non-null  object
6   Injury.Severity                      32254 non-null  object
7   Aircraft.damage                      32254 non-null  object
8   Aircraft.Category                    32254 non-null  object
9   Registration.Number                  31970 non-null  object
10  Make                                 32245 non-null  object
11  Model                                32254 non-null  object
```

```

12 Amateur.Built      32235 non-null object
13 Number.of.Engines  32254 non-null float64
14 Engine.Type        26708 non-null object
15 Purpose.of.flight  27815 non-null object
16 Total.Fatal.Injuries 32254 non-null float64
17 Total.Serious.Injuries 32254 non-null float64
18 Total.Minor.Injuries 32254 non-null float64
19 Total.Uninjured     32254 non-null float64
20 Weather.Condition   32254 non-null object
21 Report.Status        32254 non-null object
22 Publication.Date     29606 non-null datetime64[ns]
23 total_injuries       32254 non-null float64
24 Year                 32254 non-null int32
dtypes: datetime64[ns](2), float64(6), int32(1), object(16)
memory usage: 6.3+ MB

```

```

In [81]: #Copy of the csv to export
aviation_data.to_csv('aviation_data_cleaned.csv', index=False)

aviation_data.info()

```

```

<class 'pandas.core.frame.DataFrame'>
Index: 32254 entries, 5 to 90345
Data columns (total 25 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Event.Id               32254 non-null  object
1   Investigation.Type      32254 non-null  object
2   Accident.Number        32254 non-null  object
3   Event.Date             32254 non-null  datetime64[ns]
4   Location                32245 non-null  object
5   Country                32242 non-null  object
6   Injury.Severity        32254 non-null  object
7   Aircraft.damage        32254 non-null  object
8   Aircraft.Category      32254 non-null  object
9   Registration.Number    31970 non-null  object
10  Make                   32245 non-null  object
11  Model                  32254 non-null  object
12  Amateur.Built          32235 non-null  object
13  Number.of.Engines      32254 non-null  float64
14  Engine.Type            26708 non-null  object
15  Purpose.of.flight      27815 non-null  object
16  Total.Fatal.Injuries    32254 non-null  float64

```

```

17 Total.Serious.Injuries 32254 non-null float64
18 Total.Minor.Injuries 32254 non-null float64
19 Total.Uninjured 32254 non-null float64
20 Weather.Condition 32254 non-null object
21 Report.Status 32254 non-null object
22 Publication.Date 29606 non-null datetime64[ns]
23 total_injuries 32254 non-null float64
24 Year 32254 non-null int32
dtypes: datetime64[ns](2), float64(6), int32(1), object(16)
memory usage: 6.3+ MB

```

In [83]:

```

# Question 1: Safest aircraft (fewest fatal accidents)
safe_aircraft = (
    aviation_data.groupby(['Make', 'Model'], dropna=True)['Total.Fatal.Injuries']
    .sum()
    .reset_index()
    .sort_values(by='Total.Fatal.Injuries', ascending=True)
)
safe_aircraft_top5 = safe_aircraft[safe_aircraft['Total.Fatal.Injuries'] == 0].head(5)
print("Top 5 Aircraft with 0 Fatal Injuries:\n", safe_aircraft_top5)

# Question 2: Most common purpose of flight during accidents
purpose_counts = aviation_data['Purpose.of.flight'].value_counts().head(5)
print("\nTop 5 Purposes of Flight with Most Accidents:\n", purpose_counts)

# Question 3: Countries with highest number of accidents
top_countries = aviation_data['Country'].value_counts().head(5)
print("\nTop 5 Countries with Most Aircraft Accidents:\n", top_countries)

```

Top 5 Aircraft with 0 Fatal Injuries:

	Make	Model	Total.Fatal.Injuries
5540	HUGHES HELICOPTERS INC	369	0.0
5556	Hagerty	Glasair Super IIS-TD	0.0
5558	Hahn	R-W22 Tiger Moth Rep	0.0
5559	Haines	Searey	0.0
5560	Halbrook	Rans S-6S	0.0

Top 5 Purposes of Flight with Most Accidents:

Purpose.of.flight	
Personal	17718
Instructional	3867
Aerial Application	1384
Unknown	1131
Business	946

Name: count, dtype: int64

Top 5 Countries with Most Aircraft Accidents:

Country	
United States	28133
Brazil	304
United Kingdom	256
Mexico	244
Canada	218

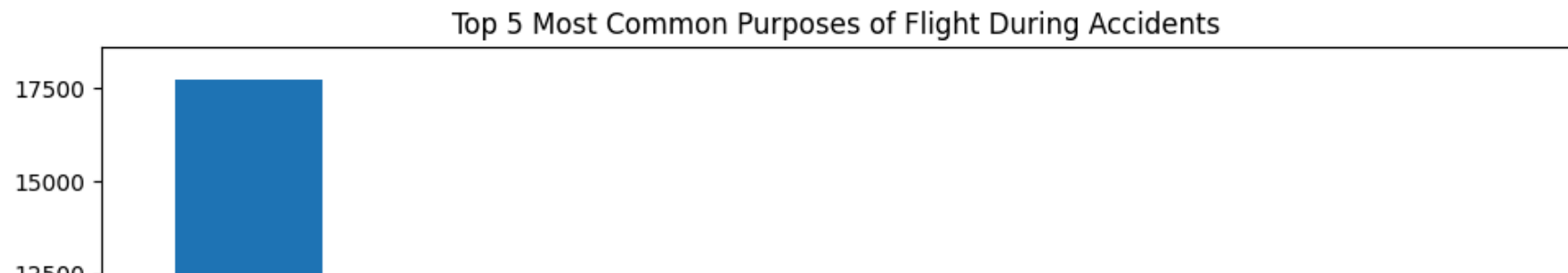
Name: count, dtype: int64

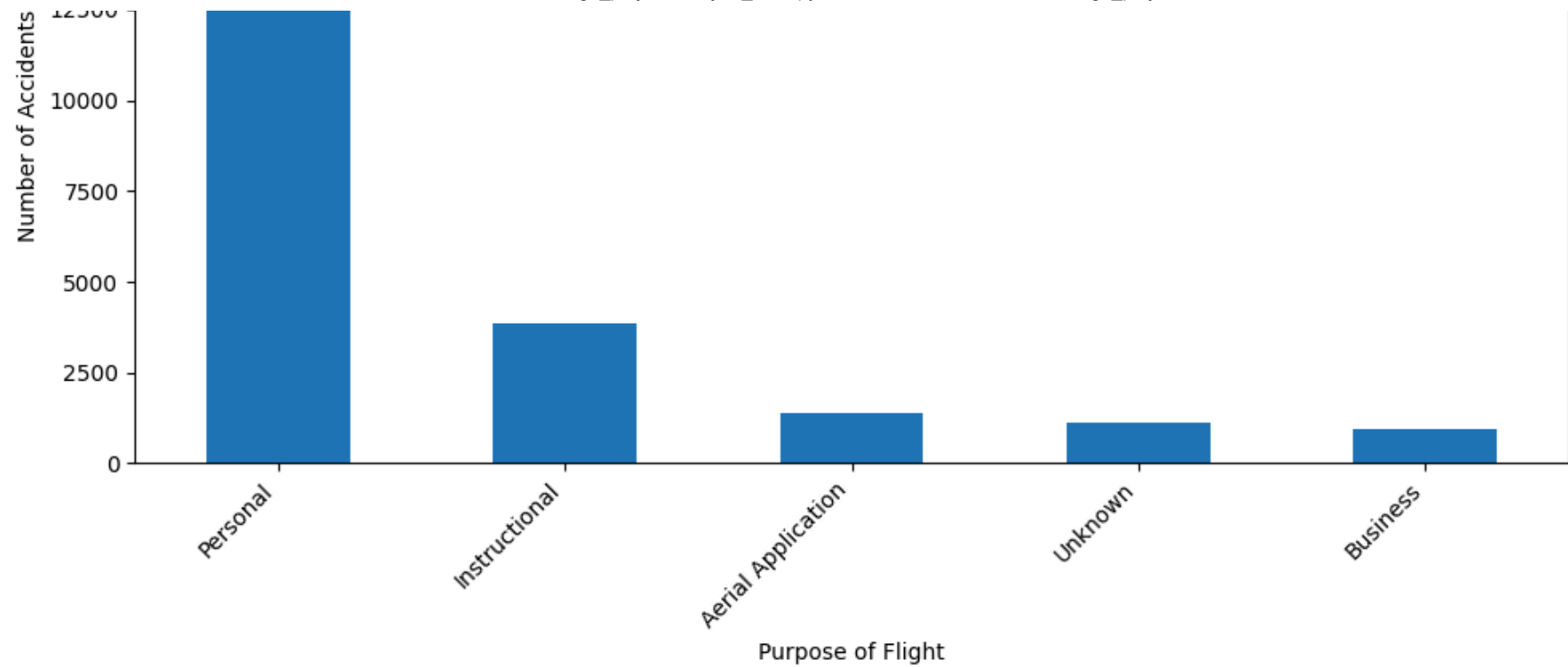
In [87]:

```
import matplotlib.pyplot as plt

# Chart 2: Top 5 Purposes of Flight with Most Accidents
plt.figure(figsize=(10, 6))
purpose_counts.plot(kind='bar')
plt.xlabel('Purpose of Flight')
plt.ylabel('Number of Accidents')
plt.title('Top 5 Most Common Purposes of Flight During Accidents')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

# Chart 3: Top 5 Countries with Most Aircraft Accidents
plt.figure(figsize=(10, 6))
top_countries.plot(kind='bar')
plt.xlabel('Country')
plt.ylabel('Number of Accidents')
plt.title('Top 5 Countries with the Most Aircraft Accidents')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```







In [ ]: