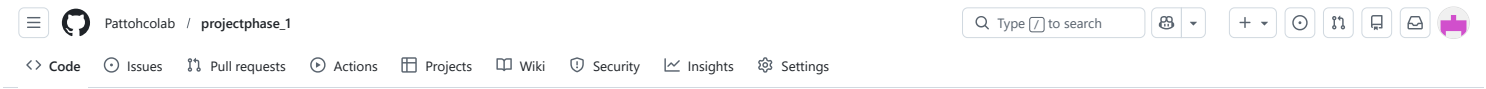


projectphase_1 Public

1 Branch 0 Tags Go to file Add file Code

Pattohcolab Add files via upload 310e535 · 3 minutes ago 8 Commits

.gitignore	Adding Jupyter notebook	yesterday
Accident Trends Over Time and ...	Add files via upload	12 hours ago
PROJECTPHASE_1.pptx	Add project presentation PowerPoint	yesterday
README.md	Update wording for Tableau dashboard...	5 minutes ago
flight.csv	Add files via upload	3 minutes ago
index.ipynb	Adding Jupyter notebook	yesterday

About

this is a data science project phase 1

Readme
Activity
0 stars
0 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

Jupyter Notebook 100.0%

README

AIRCRAFT ACCIDENT ANALYSIS AND SAFETY INSIGHTS

BUSINESS UNDERSTANDING

From a business / aviation safety perspective, stakeholders want to know:

- Are accidents increasing or decreasing over time?
- How severe are most accidents?
- How often do accidents involve fatalities?
- Which operators appear most frequently in accident reports?

These insights can help:

- Airlines improve safety procedures
- Regulators focus inspections
- Insurance companies assess risk

OBJECTIVES

- Understand accident trends over time
- Identify damage severity patterns
- Examine fatal vs non-fatal accidents
- Identify operators with higher accident counts

```
# import all the necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# loading the dataset
df = pd.read_csv('/content/flight.csv')
df
```

DATA UNDERSTANDING

The dataset contains 2,500 flight accident records with the following columns:

'Unnamed', 'acc.date', 'type', 'reg', 'operator', 'fat', 'location', 'dmg'.

Most columns are categorical.

```
# Checking for missing values
df.isna().sum()
```

we have 92 missing values in 'reg column', 14 in 'operator column' and 12 in 'fat column'. The rest have no missing values.

DATA PREPERATION AND CLEANING

This step ensures the data is usable and reliable.

```
# Remove unnecessary index column 'Unnamed: 0'
if 'Unnamed: 0' in df.columns:
    df = df.drop(columns=['Unnamed: 0'])
df
```

```
# Convert accident date to datetime
df['acc.date'] = pd.to_datetime(df['acc.date'], errors='coerce')
df
```

	acc.date	type	reg	operator	fat		
0	2022-01-03	British Aerospace 4121 Jetstream 41	ZS-NRJ	SA Airlink	0	near Venetia Mine Airport	sub
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
2	2022-01-05	Boeing 737-4H6	EP-CAP	Caspian Airlines	0	Isfahan-Shahid Beheshti Airport (IFN)	sub
3	2022-01-08	Tupolev Tu-204-100C	RA-64032	Cainiao, opb Aviastar-TU	0	Hangzhou Xiaoshan International Airport (HGH)	w/o
4	2022-01-12	Beechcraft 200 Super King Air	NaN	private	0	Machakilha, Toledo District, Grahem Creek area	w/o
...	...	...	...	...	...	...	...
2495	2018-12-20	Cessna 560 Citation V	N188CW	Chen Aircrafts LLC	4	2 km NE of Atlanta-Fulton County Airport, GA (...	w/o

- Handling missing values

Since this is a sensitive data, there are various ways to handle the missing data/values. So i've decided to handle each of them by the column they are in.

1. Accident Date (acc.date), drop if it has missing values. without date record can't be used for trend or time based analysis.

```
# dropping missing values in acc.date column
df = df.dropna(subset=['acc.date'])
```

2. Reg and Operator columns, to be filled with 'unknown'. dropping them would lose valuable incidents.

```
# Replacing missing values with 'unknown'
categorical_cols = ['reg', 'operator']

for col in categorical_cols:
    df[col] = df[col].fillna('Unknown')
```

3. Fatalities(fat) column best approach is median imputation since median is robust and realistic, and may contain outliers.

```
#using the .loc method and median to fill the missing values in the fatalities column
df.loc[:, 'fat'] = pd.to_numeric(df['fat'], errors='coerce')
df.loc[:, 'fat'] = df['fat'].fillna(df['fat'].median())
```

	acc.date	type	reg	operator	fat	location	dmg
0	2022-01-03	British Aerospace 4121 Jetstream 41	ZS-NRJ	SA Airlink	0.0	near Venetia Mine Airport	sub
1	2022-01-04	British Aerospace 3101 Jetstream 31	HR-AYY	LANHSA - Línea Aérea Nacional de Honduras S.A	0.0	Roatán-Juan Manuel Gálvez International Airpor...	sub
2	2022-01-05	Boeing 737-4H6	EP-CAP	Caspian Airlines	0.0	Isfahan-Shahid Beheshti Airport (IFN)	sub
3	2022-01-08	Tupolev Tu-204-100C	RA-64032	Cainiao, opb Aviastar-TU	0.0	Hangzhou Xiaoshan International Airport (HGH)	w/o
4	2022-01-12	Beechcraft 200 Super King Air	Unknown	private	0.0	Machakilha, Toledo District, Grahem Creek area	w/o
...	...	...	...	...	...	...	...
2494	2018-12-20	Antonov An-26B	9S-AGB	Gomair	7.0	ca 37 km from Kinshasa-N'Djili Airport (FIH)	w/o

```
# Ensure there are no missing values left
df.isna().sum()
```

	0
acc.date	0
type	0
reg	0
operator	0
fat	0
location	0
dmg	0

dtype: int64

- Handling of Duplicates

```
# finding the number of duplicates
df.duplicated().sum()
```

There are 1247 duplicated values in this data set, which may be because this data may be extracted from many sources and combined to one.

Here we shall keep the most complete record (partial duplicates) and drop the rest

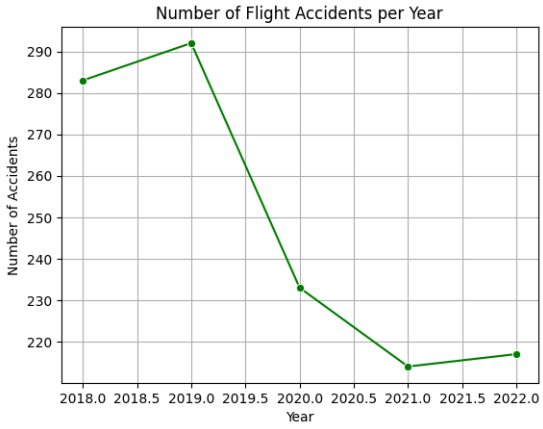
```
# the most complete duplicates will be left in the dataset
df['missing_count'] = df.isna().sum(axis=1)
```

```
df = df.sort_values('missing_count')
df = df.drop_duplicates(
    subset=['acc.date', 'type', 'location'],
    keep='first'
)

df = df.drop(columns='missing_count')


# confirming that we have no duplicated values in the dataset
df.duplicated().sum()
```

We have zero duplicates left

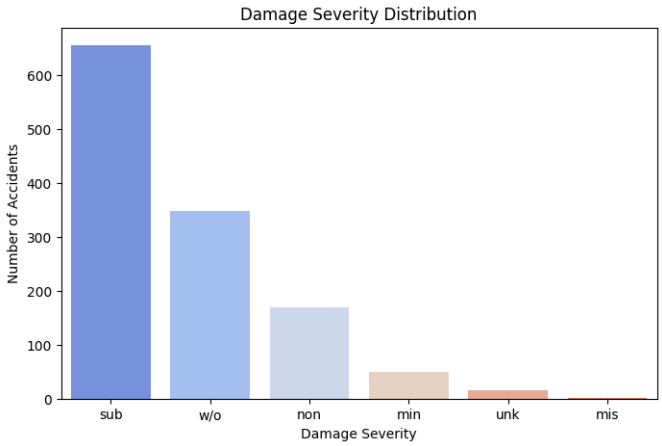
DATA ANALYSIS- EXPLORATORY DATA ANALYSIS(EDA)

- Finding out the number of Accidents per year and then visualizing it using a line chart.



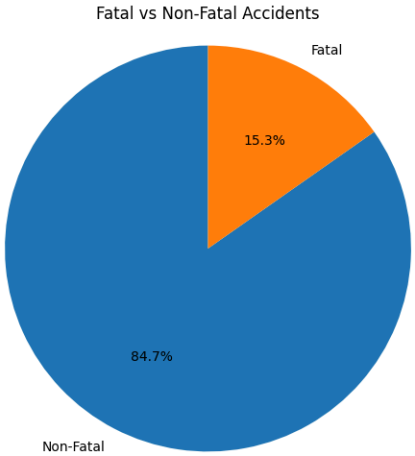
The line chart shows that the number of flight accidents peaked around 2018–2019 and declined afterward. This trend may be associated with reduced global air traffic during the COVID-19 pandemic.

- EDA: Damage Severity Distribution. Here we look at accidents that resulted in substantial damage,write-offs, Minor and no-damage



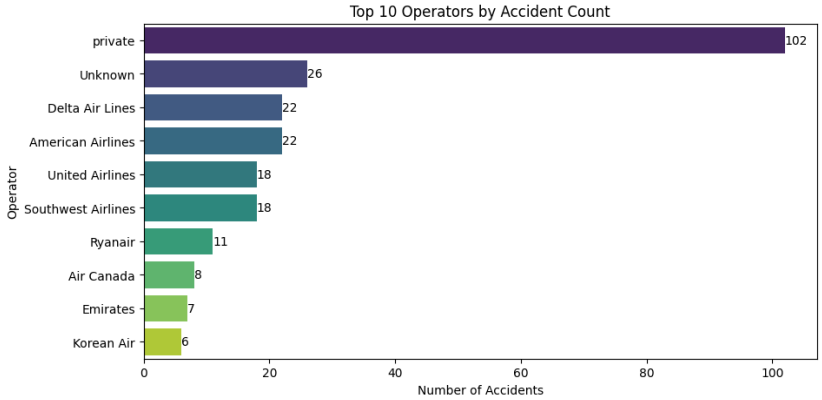
The damage severity distribution shows that most recorded accidents resulted in substantial damage, followed by write-offs. Minor and no-damage cases are relatively rare, indicating that reported accidents often involve significant aircraft damage.

- EDA: Fatal vs Non-Fatal Accidents using a pie chart visualization



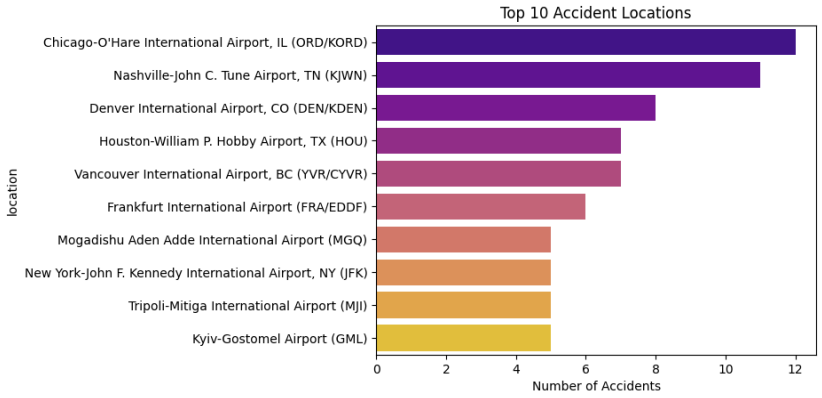
The visualization shows that the majority of flight accidents are non-fatal. Fatal accidents represent a smaller proportion of total incidents, indicating that while accidents occur, loss of life is relatively infrequent.

- EDA: Top 10 Operators by Accident Count

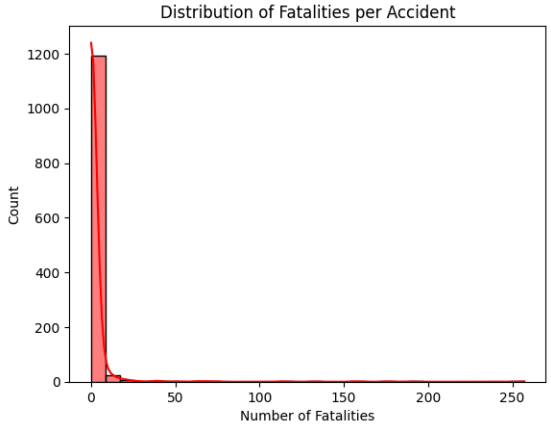


The visualization shows that private operators account for the highest number of recorded accidents. This may be due to a large number of small private flights rather than poorer safety performance. Therefore, accident counts should be normalized by flight volume for fair comparison.

- EDA Most Frequent Accident Locations



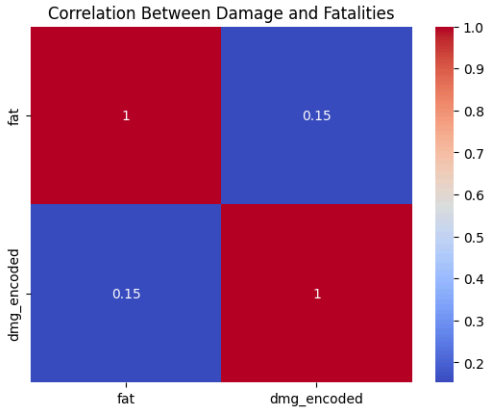
- EDA Distribution of Fatalities per Accident using a Histogram



Here we can see The distribution is heavily right-skewed Most accidents occur at 0 fatalities

Most flight accidents do not result in loss of life, but when fatalities occur, they can be severe.

- FINDING OUT THE CORRELATION BETWEEN DAMAGE TYPE AND FATALITIES



Higher damage severity correlates with fatalities. The more the damage to an aircraft the more its likely to have fatalities

- Finding out the Number of Accidents by Month