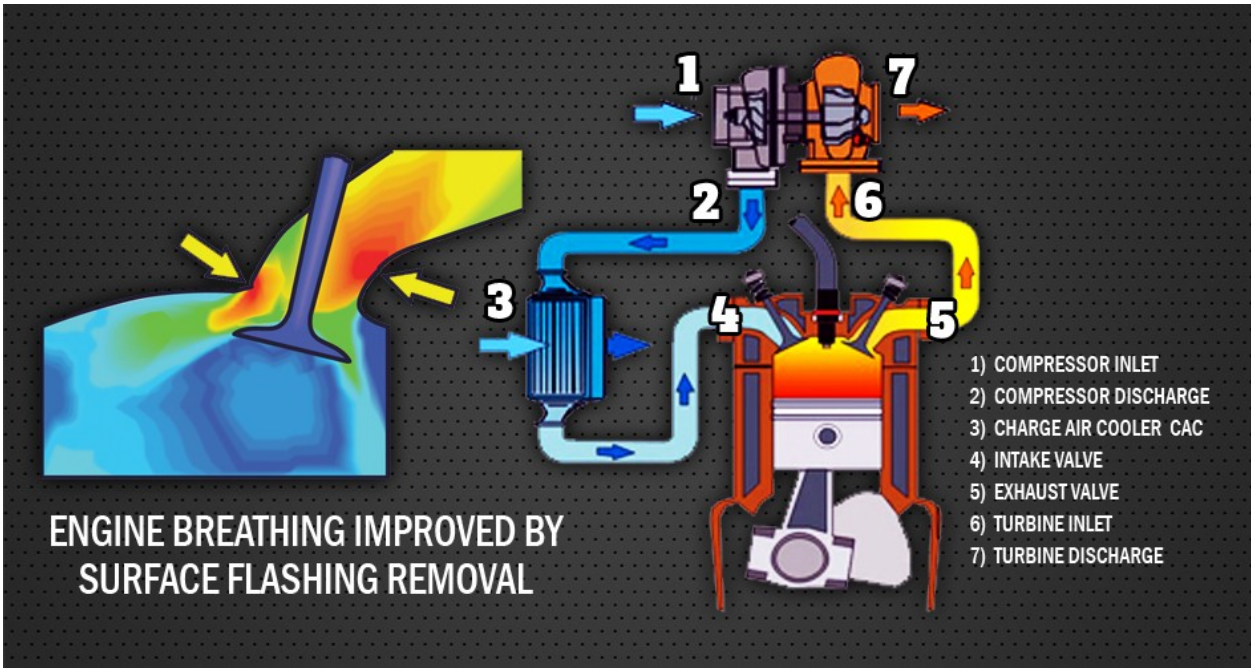
Flying Through The Years

Overview

This Project Analyzes aviation data accidents about civil aviation accidents and selected incidents in the United States and international waters. The project will focus on airplane accidents and how many engines each plane has.The aviation comapanies that make airplanes can use this analysis to improve and or add more engines to their airplanes for more safer flying and reduction of injury incidents.

Business Understanding



Aviation companies in America can reduce accidents and incidents by adding more engines. Doing so will reduce incidents in the air and or crashing. Using Aviation data set I'll describe patterns of accidents and incidents through the years on how many engines each plane had in each injury incident.

Data Understanding

The Aviation accident database has a long list of years from the 1960's to 2023 about the accidents and how many engines they have with the injury reports per incident.

```
import pandas as pd
import numpy as np
import csv

aviation_data = pd.read_csv("./data/Aviation_Data.csv",
low_memory=False)

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
 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
```

```
aviation_data.head(10)
```

```
         Event.Id Investigation.Type Accident.Number  Event.Date  \
0  20001218X45444           Accident      SEA87LA080  1948-10-24
1  20001218X45447           Accident      LAX94LA336  1962-07-19
2  20061025X01555           Accident      NYC07LA005  1974-08-30
3  20001218X45448           Accident      LAX96LA321  1977-06-19
4  20041105X01764           Accident      CHI79FA064  1979-08-02
5  20170710X52551           Accident      NYC79AA106  1979-09-17
6  20001218X45446           Accident      CHI81LA106  1981-08-01
7  20020909X01562           Accident      SEA82DA022  1982-01-01
8  20020909X01561           Accident      NYC82DA015  1982-01-01
9  20020909X01560           Accident      MIA82DA029  1982-01-01

           Location        Country   Latitude   Longitude Airport.Code
\
0   MOOSE CREEK, ID  United States        NaN         NaN          NaN
```

```
1    BRIDGEPORT, CA  United States        NaN         NaN          NaN
2     Saltville, VA  United States  36.922223  -81.878056          NaN
3        EUREKA, CA  United States        NaN         NaN          NaN
4        Canton, OH  United States        NaN         NaN          NaN
5        BOSTON, MA  United States  42.445277  -70.758333          NaN
6        COTTON, MN  United States        NaN         NaN          NaN
7       PULLMAN, WA  United States        NaN         NaN          NaN
8  EAST HANOVER, NJ  United States        NaN         NaN          N58
9  JACKSONVILLE, FL  United States        NaN         NaN          JAX

         Airport.Name  ... Purpose.of.flight Air.carrier
Total.Fatal.Injuries  \
0                 NaN  ...          Personal         NaN
2.0
1                 NaN  ...          Personal         NaN
4.0
2                 NaN  ...          Personal         NaN
3.0
3                 NaN  ...          Personal         NaN
2.0
4                 NaN  ...          Personal         NaN
1.0
5                 NaN  ...               NaN  Air Canada
NaN
6                 NaN  ...          Personal         NaN
4.0
7  BLACKBURN AG STRIP  ...          Personal         NaN
0.0
8             HANOVER  ...          Business         NaN
0.0
9   JACKSONVILLE INTL  ...          Personal         NaN
0.0

  Total.Serious.Injuries Total.Minor.Injuries Total.Uninjured  \
0                    0.0                  0.0             0.0
1                    0.0                  0.0             0.0
2                    NaN                  NaN             NaN
3                    0.0                  0.0             0.0
4                    2.0                  NaN             0.0
5                    NaN                  1.0            44.0
6                    0.0                  0.0             0.0
```

```
7                    0.0                  0.0             2.0
8                    0.0                  0.0             2.0
9                    0.0                  3.0             0.0

  Weather.Condition  Broad.phase.of.flight   Report.Status
Publication.Date
0                UNK                   Cruise  Probable Cause
NaN
1                UNK                  Unknown  Probable Cause       19-
09-1996
2                IMC                   Cruise  Probable Cause       26-
02-2007
3                IMC                   Cruise  Probable Cause       12-
09-2000
4                VMC                 Approach  Probable Cause       16-
04-1980
5                VMC                    Climb  Probable Cause       19-
09-2017
6                IMC                  Unknown  Probable Cause       06-
11-2001
7                VMC                  Takeoff  Probable Cause       01-
01-1982
8                IMC                  Landing  Probable Cause       01-
01-1982
9                IMC                   Cruise  Probable Cause       01-
01-1982

[10 rows x 31 columns]
```

Event Date of Accidents

```
aviation_data['Accident Date'] =
pd.to_datetime(aviation_data['Event.Date'])
aviation_data['Accident Date'].describe()

count                            88889
mean     1999-09-17 17:13:39.354476032
min                1948-10-24 00:00:00
25%                1989-01-15 00:00:00
50%                1998-07-18 00:00:00
75%                2009-07-01 00:00:00
max                2022-12-29 00:00:00
Name: Accident Date, dtype: object
```

Number of Engines data

```
aviation_data['Number.of.Engines'].value_counts()

Number.of.Engines
1.0    69582
```

```
2.0    11079
0.0     1226
3.0      483
4.0      431
8.0        3
6.0        1
Name: count, dtype: int64
```

Total Number of Injury Severity Incidents data

```
aviation_data['Injury.Severity'].value_counts()
```

```
Injury.Severity
Non-Fatal     67357
Fatal(1)       6167
Fatal          5262
Fatal(2)       3711
Incident       2219
              ...  
Fatal(270)        1
Fatal(60)         1
Fatal(43)         1
Fatal(143)        1
Fatal(230)        1
Name: count, Length: 109, dtype: int64
```

Data Preperation

Data cleaning

After getting the data for the number of engines and injury severity incidents data and accident dates, I will be prepping the data to focus on number of engines and injury severity.

```
#look up columns
aviation_data.columns
```

```
Index(['Event.Id', 'Investigation.Type', 'Accident.Number',
'Event.Date',
       'Location', 'Country', 'Latitude', 'Longitude', 'Airport.Code',
       'Airport.Name', 'Injury.Severity', 'Aircraft.damage',
       'Aircraft.Category', 'Registration.Number', 'Make', 'Model',
       'Amateur.Built', 'Number.of.Engines', 'Engine.Type',
'FAR.Description',
       'Schedule', 'Purpose.of.flight', 'Air.carrier',
'Total.Fatal.Injuries',
       'Total.Serious.Injuries', 'Total.Minor.Injuries',
'Total.Uninjured',
       'Weather.Condition', 'Broad.phase.of.flight', 'Report.Status',
```

```
       'Publication.Date', 'Accident Date'],
      dtype='object')

#Narrowing down the dataset to 3 columns
aviation_data = aviation_data[['Accident Date', 'Injury.Severity',
'Number.of.Engines']]

aviation_data.head(10)

  Accident Date Injury.Severity  Number.of.Engines
0    1948-10-24        Fatal(2)                1.0
1    1962-07-19        Fatal(4)                1.0
2    1974-08-30        Fatal(3)                1.0
3    1977-06-19        Fatal(2)                1.0
4    1979-08-02        Fatal(1)                NaN
5    1979-09-17       Non-Fatal                2.0
6    1981-08-01        Fatal(4)                1.0
7    1982-01-01       Non-Fatal                1.0
8    1982-01-01       Non-Fatal                2.0
9    1982-01-01       Non-Fatal                1.0

# I will drop NaN from the dataset
aviation_data = aviation_data.dropna()

aviation_data.head(10)

   Accident Date Injury.Severity  Number.of.Engines
0     1948-10-24        Fatal(2)                1.0
1     1962-07-19        Fatal(4)                1.0
2     1974-08-30        Fatal(3)                1.0
3     1977-06-19        Fatal(2)                1.0
5     1979-09-17       Non-Fatal                2.0
6     1981-08-01        Fatal(4)                1.0
7     1982-01-01       Non-Fatal                1.0
8     1982-01-01       Non-Fatal                2.0
9     1982-01-01       Non-Fatal                1.0
10    1982-01-01       Non-Fatal                1.0
```

Looking Duplicates

Looking for duplicates to reduce any inaccurate data.

```
#look for duplicates in dataset
aviation_data.duplicated()

0        False
1        False
2        False
3        False
5        False
```

```
            ...
90328    False
90332    False
90335    False
90336    False
90345    False
Length: 82526, dtype: bool
```

There are no duplicates

```
aviation_data.head(6)
```

```
  Accident Date Injury.Severity  Number.of.Engines
0    1948-10-24        Fatal(2)                1.0
1    1962-07-19        Fatal(4)                1.0
2    1974-08-30        Fatal(3)                1.0
3    1977-06-19        Fatal(2)                1.0
5    1979-09-17       Non-Fatal                2.0
6    1981-08-01        Fatal(4)                1.0
```

```
# Save cleaned dataset as csv
aviation_data.to_csv('numeng_injsev.csv')
```

Exploratory Data Analysis

After cleaning the data I'll be creating a bar chart showing the average number of engines and injury severity.

```
import matplotlib
import matplotlib.pyplot as plt

%matplotlib inline
```

Showing Statistics of Each Selected Column

```
aviation_data['Injury.Severity'].describe()
```

```
count         82526
unique           67
top       Non-Fatal
freq          64484
Name: Injury.Severity, dtype: object
```

```
aviation_data['Number.of.Engines'].describe()
```

```
count    82526.000000
mean         1.143785
std          0.442978
```

```
min          0.000000
25%          1.000000
50%          1.000000
75%          1.000000
max          8.000000
Name: Number.of.Engines, dtype: float64
```

```
aviation_data['Accident Date'].describe()
```

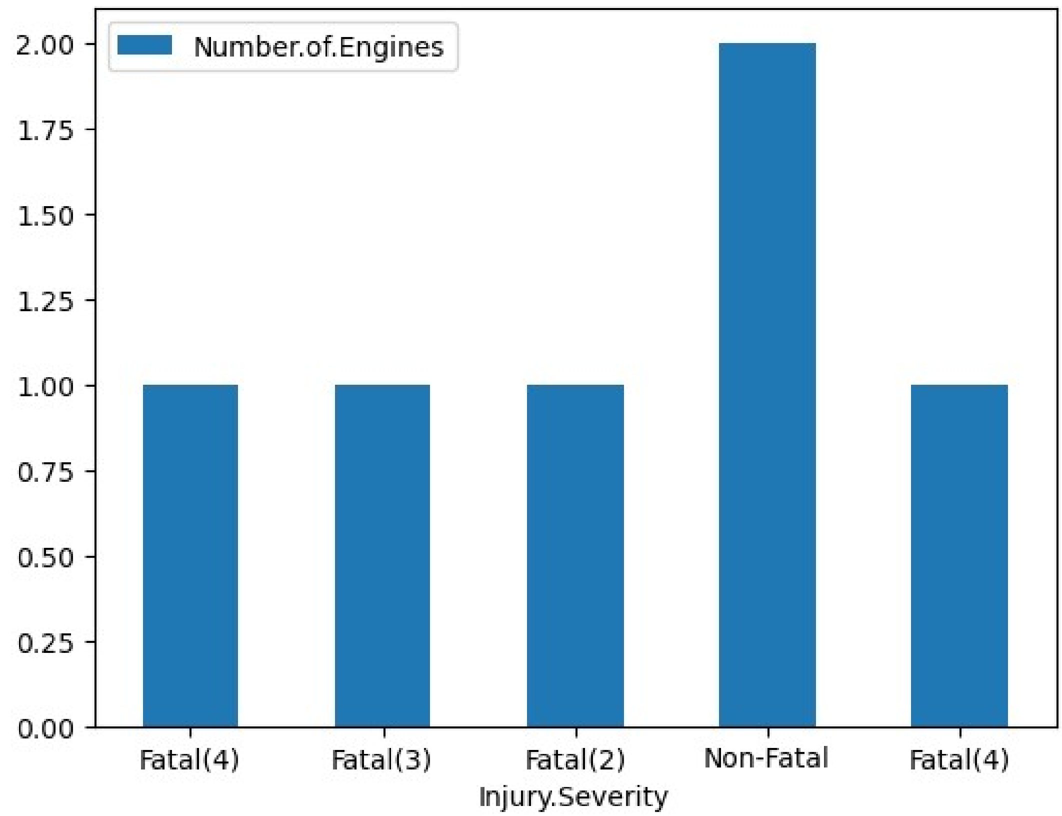
```
count                              82526
mean     1998-11-28 10:32:48.426920064
min                1948-10-24 00:00:00
25%                1988-07-17 00:00:00
50%                1997-06-13 00:00:00
75%                2008-05-11 00:00:00
max                2022-12-26 00:00:00
Name: Accident Date, dtype: object
```

Barplot of Number of Engine and Injury Severity

I will be creating a barplot to visually show the data.

```
# # Create a bar chart
aviation_data = pd.core.frame.DataFrame({'Injury.Severity':
['Fatal(4)','Fatal(3)','Fatal(2)','Non-Fatal','Fatal(4)'],
'Number.of.Engines':[1.0,1.0,1.0,2.0,1.0],})

ax = aviation_data.plot.bar(x='Injury.Severity',
y='Number.of.Engines', rot=0)
```



Conclusions

This analysis leads to showing that having less engines would have a higher and severe injury incident:

- **The more engines,the less an accident can happen.** Based on the data the more engines each plane had also had less injuries to non-fatal injuries.

Recommendations

Limitations

- **How the engine failed.** No data on how the engine failed on an individual plane.
- **Was it the plane?** limiting factor if the plane was really the cause of the accident.

Next Steps

- Further analyses could yield additional insights to further improve how many engine a plane should have to reduce accidents
- Improve the engine quality to reduce any malfunctions