First is to import pandas into the notebook

In [1]:
```
import pandas as pd
```

Importing CSV file into Pandas

In [4]:
```
#loadin the dataset in cvs format into pandas
df = pd.read_csv("AviationData.csv", encoding="latin!", low_memory=Fals
```

Then we can check the details of the dataset.

In [7]:
```
#checcking the first few rows of the data set
df.head()
```

Out[7]:

	Event.Id	Investigation.Type	Accident.Number	Event.Date	Locati
0	20001218X45444	Accident	SEA87LA080	1948-10-24	MOO CREEK,
1	20001218X45447	Accident	LAX94LA336	1962-07-19	BRIDGEPOR
2	20061025X01555	Accident	NYC07LA005	1974-08-30	Saltville,
3	20001218X45448	Accident	LAX96LA321	1977-06-19	EUREKA,
4	20041105X01764	Accident	CHI79FA064	1979-08-02	Canton,

5 rows × 31 columns

In [9]:
```
##Identifying the columns within this data set
df.columns
```

Out[9]:
```
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.Descr
iption',
       'Schedule', 'Purpose.of.flight', 'Air.carrier', 'Total.Fatal.In
juries',
       'Total.Serious.Injuries', 'Total.Minor.Injuries', 'Total.Uninju
red',
       'Weather.Condition', 'Broad.phase.of.flight', 'Report.Status',
       'Publication.Date'],
      dtype='object')
```

In [11]:
```
#in a list jformat
list(df.columns)
```

Out[11]:
```
['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']
```

In [127…

```
#other details of the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 88889 entries, 0 to 88888
Data columns (total 31 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Event.Id                88889 non-null  object 
 1   Investigation.Type      88889 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        75118 non-null  object
dtypes: float64(5), object(26)
memory usage: 21.0+ MB
```

In [13]:
```
#data types
df.dtypes
```

Out[13]:
```
Event.Id                 object
Investigation.Type       object
Accident.Number          object
Event.Date               object
Location                 object
Country                  object
Latitude                 object
Longitude                object
Airport.Code             object
Airport.Name             object
Injury.Severity          object
Aircraft.damage          object
Aircraft.Category        object
Registration.Number      object
Make                     object
Model                    object
Amateur.Built            object
Number.of.Engines       float64
Engine.Type              object
FAR.Description          object
Schedule                 object
Purpose.of.flight        object
Air.carrier              object
Total.Fatal.Injuries    float64
Total.Serious.Injuries  float64
Total.Minor.Injuries    float64
Total.Uninjured         float64
Weather.Condition        object
Broad.phase.of.flight    object
Report.Status            object
Publication.Date         object
dtype: object
```

In [131…
```
#simple statistical summaries of columns with numbers
df.describe()
```

Out[131…

	Number.of.Engines	Total.Fatal.Injuries	Total.Serious.Injuries	Total.Minor.In
count	82805.000000	77488.000000	76379.000000	76956.0(
mean	1.146585	0.647855	0.279881	0.3
std	0.446510	5.485960	1.544084	2.2
min	0.000000	0.000000	0.000000	0.0(
25%	1.000000	0.000000	0.000000	0.0(
50%	1.000000	0.000000	0.000000	0.0(
75%	1.000000	0.000000	0.000000	0.0(

max	8.000000	349.000000	161.000000	380.0

Next, we can clean the data, checking the missing values and duplicates

In [16]:
```
#checking the missing values in the columns
df.isnull()
```

Out[16]:

	Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Cou
0	False	False	False	False	False	
1	False	False	False	False	False	
2	False	False	False	False	False	
3	False	False	False	False	False	
4	False	False	False	False	False	
...	...	...	...	...	...	
88884	False	False	False	False	False	
88885	False	False	False	False	False	
88886	False	False	False	False	False	
88887	False	False	False	False	False	
88888	False	False	False	False	False	

88889 rows × 31 columns

In [18]:
```
df.isnull().sum()
```

Out[18]:
```
Event.Id                  0
Investigation.Type        0
Accident.Number           0
Event.Date                0
Location                 52
Country                 226
Latitude              54507
Longitude             54516
Airport.Code          38757
Airport.Name          36185
Injury.Severity        1000
Aircraft.damage        3194
Aircraft.Category     56602
Registration.Number    1382
Make                     63
Model                    92
Amateur.Built           102
Number.of.Engines      6084
Engine.Type            7096
FAR.Description       56866
Schedule              76307
Purpose.of.flight      6192
Air.carrier           72241
Total.Fatal.Injuries  11401
Total.Serious.Injuries12510
Total.Minor.Injuries  11933
Total.Uninjured        5912
```

```
Weather.Condition          4492
Broad.phase.of.flight     27165
Report.Status              6384
Publication.Date          13771
dtype: int64
```

In [20]:
```
#missing values as percentage
df.isnull().mean().mul(100).round(0)
```

Out[20]:
```
Event.Id                   0.0
Investigation.Type         0.0
Accident.Number            0.0
Event.Date                 0.0
Location                   0.0
Country                    0.0
Latitude                  61.0
Longitude                 61.0
Airport.Code              44.0
Airport.Name              41.0
Injury.Severity            1.0
Aircraft.damage            4.0
Aircraft.Category         64.0
Registration.Number        2.0
Make                       0.0
Model                      0.0
Amateur.Built              0.0
Number.of.Engines          7.0
Engine.Type                8.0
FAR.Description           64.0
Schedule                  86.0
Purpose.of.flight          7.0
Air.carrier               81.0
Total.Fatal.Injuries      13.0
Total.Serious.Injuries    14.0
Total.Minor.Injuries      13.0
Total.Uninjured            7.0
Weather.Condition          5.0
Broad.phase.of.flight     31.0
Report.Status              7.0
Publication.Date          15.0
dtype: float64
```

In [22]:
```
#checcking the duplicates
df.duplicated()
```

Out[22]:
```
0        False
1        False
2        False
3        False
4        False
         ...
88884    False
88885    False
88886    False
88887    False
88888    False
Length: 88889, dtype: bool
```

In [24]:
```
df.duplicated().sum()
```

Out[24]:
```
0
```

In [26]:
```
# we can drop some columns which have significant missing values
df = df.dropna(thresh=len(df)*0.6, axis=1)
```

In [28]:
```
# we can check the new list
df.isnull().mean().mul(100).round(0)
```

Out[28]:
```
Event.Id                  0.0
Investigation.Type        0.0
Accident.Number           0.0
Event.Date                0.0
Location                  0.0
Country                   0.0
Injury.Severity           1.0
Aircraft.damage           4.0
Registration.Number       2.0
Make                      0.0
Model                     0.0
Amateur.Built             0.0
Number.of.Engines         7.0
Engine.Type               8.0
Purpose.of.flight         7.0
Total.Fatal.Injuries     13.0
Total.Serious.Injuries   14.0
Total.Minor.Injuries     13.0
Total.Uninjured           7.0
Weather.Condition         5.0
Broad.phase.of.flight    31.0
Report.Status             7.0
Publication.Date         15.0
dtype: float64
```

In [30]:
```
df.isnull().sum()
```

Out[30]:
```
Event.Id                    0
Investigation.Type          0
Accident.Number             0
Event.Date                  0
Location                   52
Country                   226
Injury.Severity          1000
Aircraft.damage          3194
Registration.Number      1382
Make                       63
Model                      92
Amateur.Built             102
Number.of.Engines        6084
Engine.Type              7096
Purpose.of.flight        6192
Total.Fatal.Injuries    11401
Total.Serious.Injuries  12510
Total.Minor.Injuries    11933
Total.Uninjured          5912
Weather.Condition        4492
Broad.phase.of.flight   27165
Report.Status            6384
Publication.Date        13771
dtype: int64
```

In [32]:
```
df["Injury.Severity"].dtype
```

Out[32]:
```
dtype('O')
```

Filling in the missing values

In [57]:
```
#filling in the missing values of categrical data: replacing with unkno
df_1 = df.fillna({"Location":"Unknown",
           "Country": "Unknown",
           "Injury.Severity": "Unknown",
           "Aircraft.damage": "Unknown",
           "Registration.Number": "Unknown",
           "Make": "Unknown",
           "Model": "Unknown",
           "Amateur.Built": "Unknown",
           "Purpose.of.flight": "Unknown",
           "Purpose.of.flight": "Unknown",
           "Broad.phase.of.flight": "Unknown",
           "Report.Status": "Unknown",
           "Publication.Date": "Unknown", "Engine.Type": "Unknown", "We
```

In [59]:
```
#filling in the missing values of numerical data: replacing with 0
df_2 = df_1.fillna({"Number.of.Engines": 0,
           "Total.Fatal.Injuries": 0,
           "Total.Serious.Injuries": 0,
           "Total.Minor.Injuries": 0,
           "Total.Uninjured": 0})
```

In [61]:
```
#checking the missing values again in the columns
df_2.isnull().sum()
```

Out[61]:
```
Event.Id                0
Investigation.Type      0
Accident.Number         0
Event.Date              0
Location                0
Country                 0
Injury.Severity         0
Aircraft.damage         0
Registration.Number     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
Broad.phase.of.flight   0
Report.Status           0
Publication.Date        0
dtype: int64
```

Our new data set df_2 has no missing values and hence we can analyse it.

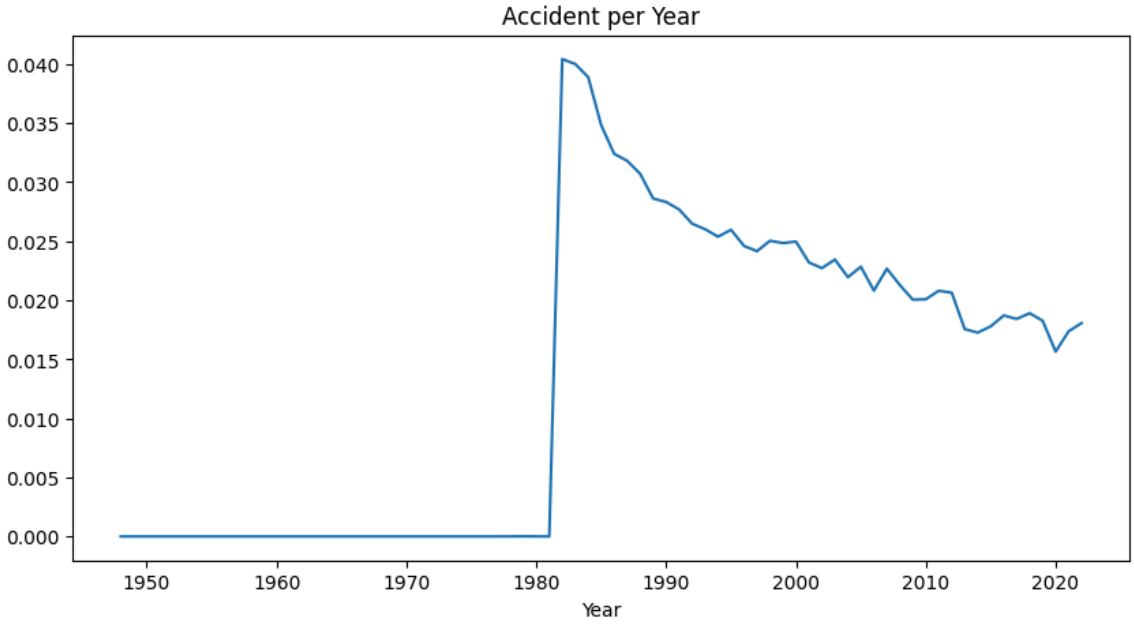
In [101…
```
#Change the event date to year

df_2["Event.Date"] = pd.to_datetime(df_2["Event.Date"], errors="coerce"
```

```
df_2["Year"] = df_2["Event.Date"].dt.year
```

1.We can analyse the number of accidencts relative to event date

In [105…
```
#accidents per period bar graph for all the years
df_2["Year"].value_counts(10).sort_index().plot(kind="line", figsize=(1
```

Out[105… <Axes: title={'center': 'Accident per Year'}, xlabel='Year'>

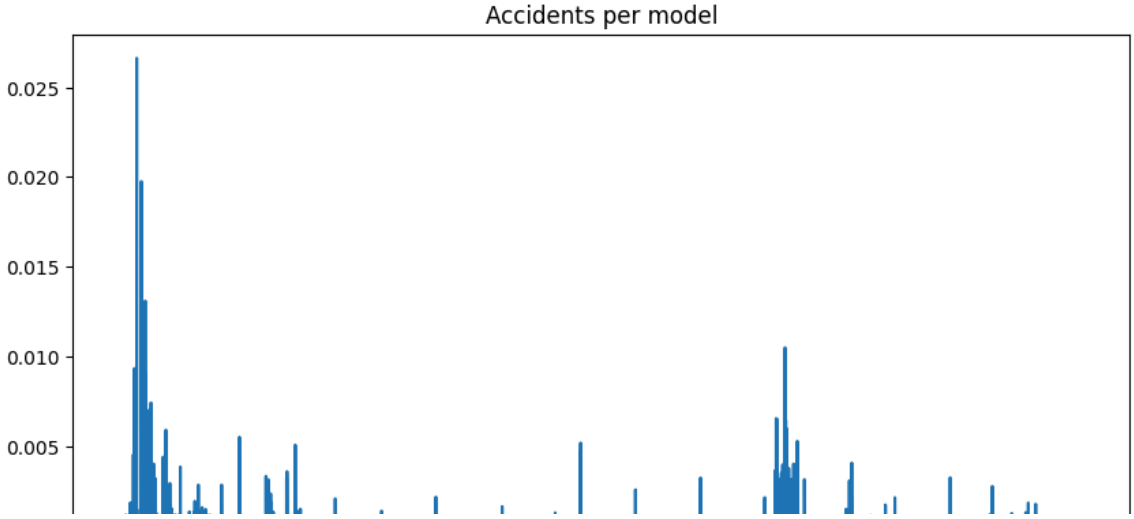


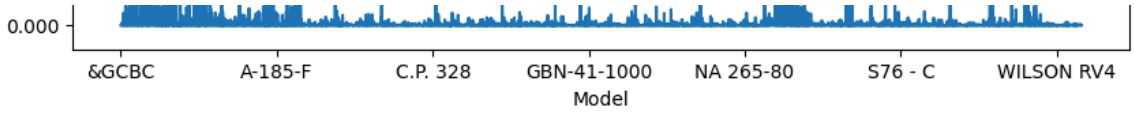
In [82]:
```
#Which model has the most accidents
df_2["Model"].value_counts().head()
```

Out[82]:
```
Model
152          2367
172          1756
172N         1164
PA-28-140     932
150           829
Name: count, dtype: int64
```

In [115…
```
df_2["Model"].value_counts(10).sort_index().plot(kind="line",figsize=(1
```

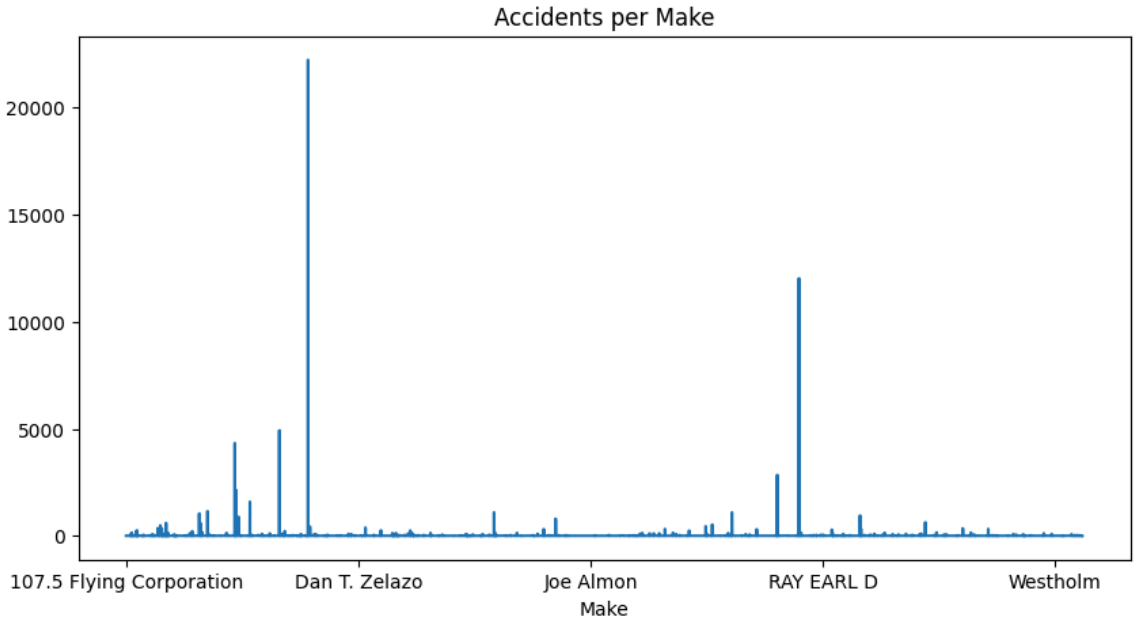
Out[115… <Axes: title={'center': 'Accidents per model'}, xlabel='Model'>





In [123…
```
# which make of airline has the most accidents
df_2["Make"].value_counts().sort_index().plot(kind="line",figsize=(10,5
```

Out[123… <Axes: title={'center': 'Accidents per Make'}, xlabel='Make'>



In [203…
```
#checking the aircraft model against the fatal injuries
df_2.groupby("Model")["Total.Fatal.Injuries"].mean().sort_values(ascend
#the higher the average fatalities the more serious the accidents
#and hence higher exposure
```

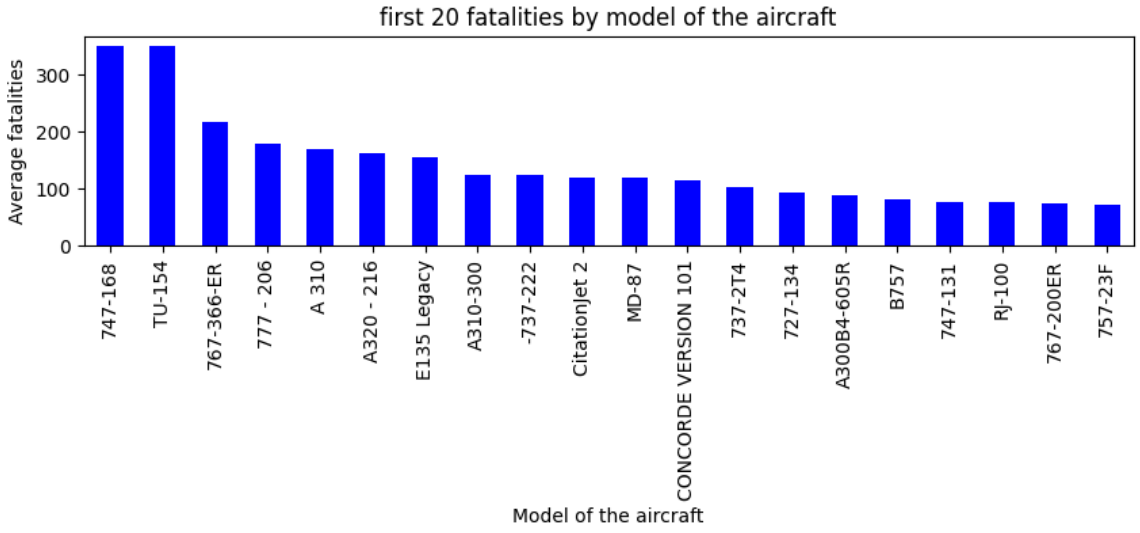
Out[203…
```
Model
747-168                 349.000000
TU-154                  349.000000
767-366-ER              217.000000
777 - 206               178.000000
A 310                   169.000000
A320 - 216              162.000000
E135 Legacy             154.000000
A310-300                124.000000
-737-222                123.000000
CitationJet 2           118.000000
MD-87                   118.000000
CONCORDE VERSION 101    113.000000
737-2T4                 102.000000
727-134                  92.000000
A300B4-605R              88.666667
B757                     80.000000
747-131                  76.666667
RJ-100                   75.000000
767-200ER                73.750000
757-23F                  71.000000
Name: Total.Fatal.Injuries, dtype: float64
```

In [145…
```
import matplotlib.pyplot as plt
```

In [205…
```
#Analysis of first 20 fatalities by Model
```

```
first_20_fatalities = df_2.groupby("Model")["Total.Fatal.Injuries"].mea

plt.figure(figsize=(10, 2))
first_20_fatalities.plot(kind="bar", color="blue")

plt.title("first 20 fatalities by model of the aircraft")

plt.xlabel("Model of the aircraft")

plt.ylabel("Average fatalities")
```

Out[205… `Text(0, 0.5, 'Average fatalities')`



In [207…

```
# Analysis of the last 20 fatalities by Model
#the lower the average fatalities the les serious the accidents
#and hence lower the exposure
df_2.groupby("Model")["Total.Fatal.Injuries"].mean().sort_values(ascend
```

Out[207…

```
Model
E-Racer                0.0
Dragonfly Mark II      0.0
Dragonfly B            0.0
Dragon Fly-B           0.0
Davis                  0.0
DX4                    0.0
DYKE DELTA JD-2        0.0
DYNAMIC                0.0
DYNAMIC WT9            0.0
Dakota Hawk            0.0
Dakota Hawk DH23       0.0
Debus-Casst-Snoshoo    0.0
Dornier 328-300        0.0
Defiant                0.0
Discus 2b              0.0
Discus B               0.0
Discus-2CT             0.0
Discus-CS              0.0
Discuss B              0.0
GV-SP                  0.0
Name: Total.Fatal.Injuries, dtype: float64
```

In [219…

```
#checking the last 20 non zero values
last_20_fatalities_greater_than_zero = df_2.groupby("Model")["Total.Fat
```

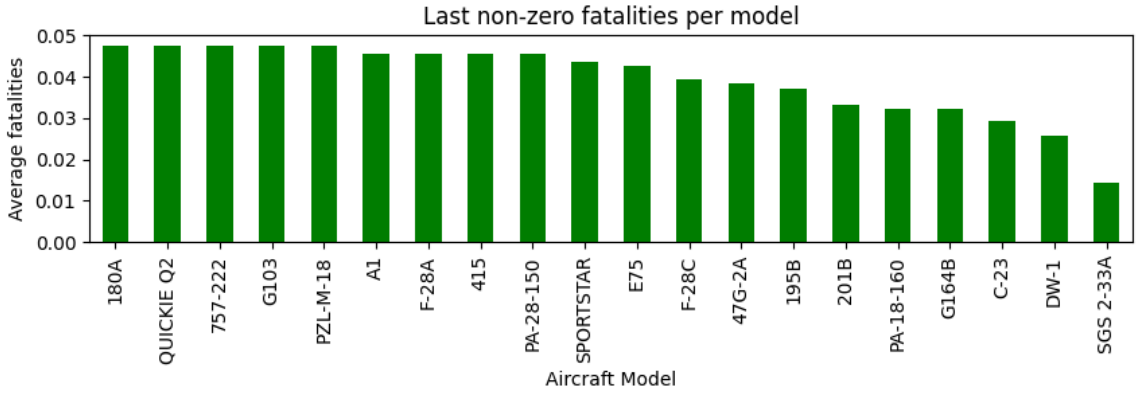
```
last_20_fatalities_greater_than_zero = last_20_fatalities_greater_than_

print(last_20_fatalities_greater_than_zero)
```

```
Model
180A           0.047619
QUICKIE Q2     0.047619
757-222        0.047619
G103           0.047619
PZL-M-18       0.047619
A1             0.045455
F-28A          0.045455
415            0.045455
PA-28-150      0.045455
SPORTSTAR      0.043478
E75            0.042553
F-28C          0.039216
47G-2A         0.038462
195B           0.037037
201B           0.033333
PA-18-160      0.032258
G164B          0.032258
C-23           0.029412
DW-1           0.025641
SGS 2-33A      0.014286
Name: Total.Fatal.Injuries, dtype: float64
```

In [233…

```
plt.figure(figsize=(10,2))

last_20_fatalities_greater_than_zero.plot(kind="bar",color="green")

plt.title("Last non-zero fatalities per model")

plt.xlabel("Aircraft Model")

plt.ylabel("Average fatalities")
```

Out[233… `Text(0, 0.5, 'Average fatalities')`



We can analyse the accidents per phase of the flight

In [243…

```
# we can do this by checking the fatalities per phase, whether at takeo

df_2["Broad.phase.of.flight"].value_counts().head()
```

Out[243…

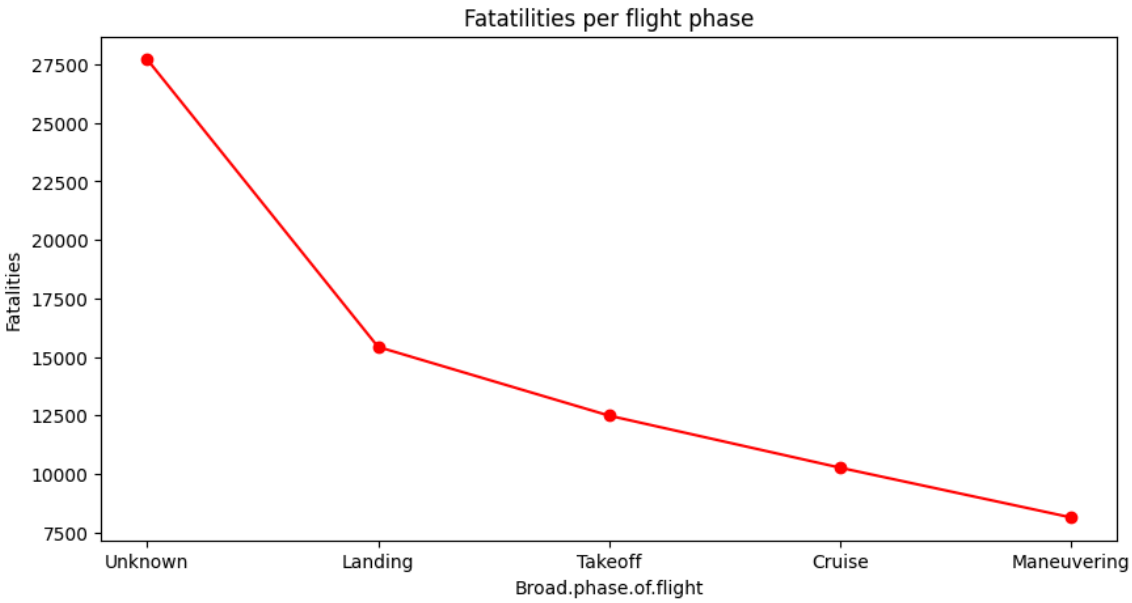
```
Broad.phase.of.flight
Unknown        27713
Landing        15428
Takeoff        12493
```

```
Cruise          10269
Maneuvering      8144
Name: count, dtype: int64
```

In [259…
```
flight_phases = df_2["Broad.phase.of.flight"].value_counts().head()
```

In [263…
```
plt.figure(figsize=(10, 5))
plt.plot(flight_phases.index, flight_phases .values, marker='o', linest

plt.title("Fatatilities per flight phase")

plt.xlabel("Broad.phase.of.flight")

plt.ylabel("Fatalities")
```

Out[263…
```
Text(0, 0.5, 'Fatalities')
```



We can now assess the impact of weather on accidents

In [275…
```
#Standardise the UNK and unk
df_2["Weather.Condition"] = df_2["Weather.Condition"].replace({"UNK": '
```

In [277…
```
#Cheecking weather conditions counts
df_2["Weather.Condition"].value_counts()
```

Out[277…
```
Weather.Condition
VMC        77303
IMC         5976
Unknown     5610
Name: count, dtype: int64
```

The results indicate that the most common wesather conditions is the VMC (Visual Meteorological conditions) This is followed by IMC(Instrument meteorological conditions)

The next assessment is then to see if there any relationships between fatality and weather

Weather conditions and fatalities

In [280…
```
df_2.groupby("Weather.Condition")["Total.Fatal.Injuries"].sum().sort_va
```

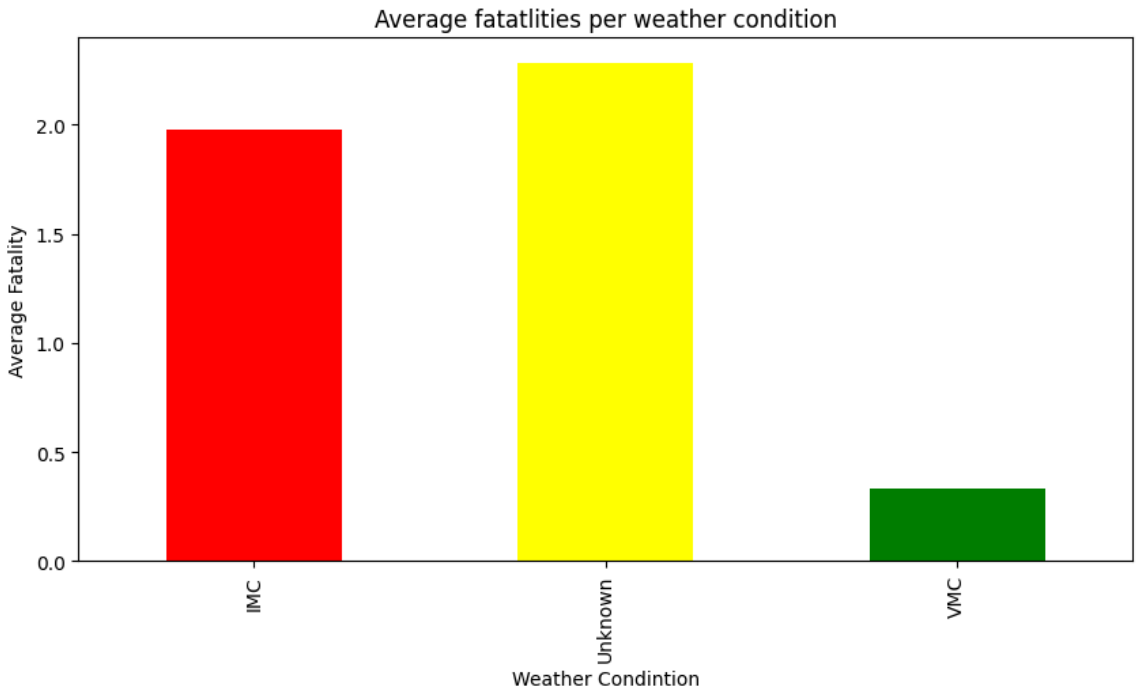
Out[280…
```
Weather.Condition
VMC        25558.0
Unknown    12819.0
IMC        11824.0
Name: Total.Fatal.Injuries, dtype: float64
```

In [284…
```
Waether_fatality = df_2.groupby("Weather.Condition")["Total.Fatal.Inju
print(Waether_fatality)
```

```
Weather.Condition
IMC        1.978581
Unknown    2.285027
VMC        0.330621
Name: Total.Fatal.Injuries, dtype: float64
```

In [290…
```
plt.figure(figsize=(10, 5))
Waether_fatality.plot(kind="bar", color=["red", "Yellow", "green"])

plt.title("Average fatatlities per weather condition")

plt.xlabel("Weather Condintion")

plt.ylabel("Average Fatality")
```

Out[290…
```
Text(0, 0.5, 'Average Fatality')
```



In [ ]: