Aviation Accidents

Traveling by plane is considered among the safest modes of transportation.However incidents in the Aviation sector do happen in the form of crash landings, crashes causing life altering events overall.This report will analyze the aviation crashes throughout the years, and how they have improved or declined with time and how to available data was processed in to order to extract useful insights for future analysis which in return provides you with a deep understanding of the aviation incidents in recent times.

Libraries Used

The following libraries were used.

- **Pandas**
- **Numpy**
- **Matplot**
- **Scipy**

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
import seaborn as sns
```

Loading the Datasets

For this task two datasets were used to get better insights on the Aviation accidents overall.

- Airplane Crashes Since 1908 (https://www.kaggle.com/datasets/saurograndi/airplane-crashes-since-1908) : This dataset contains the complete data of aviation crashes from 1908 till 2009 with 13 columns in total.
- Historical Plane Crash Data (https://www.kaggle.com/datasets/abeperez/historical-plane-crash-data) : This dataset contains the complete data of crashes overall in history, ranging from 1918-2022 respectively.

In [2]:
```
#Airplane Crashes Since 1908 Dataset
data=pd.read_csv('C:\\Users\HP\\Desktop\\Python\\Airplane_Crashes_and_Fatalities_Since_1908.csv')
```

In [3]:
```
#Historical Plane Crash Data dataset
crashes_data=pd.read_csv('C:\\Users\HP\\Desktop\\Python\\PlaneCrashes.csv')
```

In [ ]:
```
```

Cleaning and Preparing Data

We will view the first dataset first in order to understand the flow of data available which will include an Exploratry data analysis of the dataset to make the data available for further processing .

In [4]:
```
#printing the first dataset
data.head()
```

Out[4]:

	index	Date	Time	Location	Operator	Flight #	Route	Type	Registration	cn/ln	Aboard	Fatalities	Ground	Summary
0	0	09/17/1908	17:18	Fort Myer, Virginia	Military - U.S. Army	NaN	Demonstration	Wright Flyer III	NaN	1	2.0	1.0	0.0	During a demonstration flight, a U.S. Army fly...
1	1	07/12/1912	06:30	AtlantiCity, New Jersey	Military - U.S. Navy	NaN	Test flight	Dirigible	NaN	NaN	5.0	5.0	0.0	First U.S. dirigible Akron exploded just offsh...
2	2	08/06/1913	NaN	Victoria, British Columbia, Canada	Private	-	NaN	Curtiss seaplane	NaN	NaN	1.0	1.0	0.0	The first fatal airplane accident in Canada oc...
3	3	09/09/1913	18:30	Over the North Sea	Military - German Navy	NaN	NaN	Zeppelin L-1 (airship)	NaN	NaN	20.0	14.0	0.0	The airship flew into a thunderstorm and encou...
4	4	10/17/1913	10:30	Near Johannisthal, Germany	Military - German Navy	NaN	NaN	Zeppelin L-2 (airship)	NaN	NaN	30.0	30.0	0.0	Hydrogen gas which was being vented was sucked...

In [5]:
```
#printing information about first dataset
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5268 entries, 0 to 5267
Data columns (total 14 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   index         5268 non-null   int64
 1   Date          5268 non-null   object
 2   Time          3049 non-null   object
 3   Location      5248 non-null   object
 4   Operator      5250 non-null   object
 5   Flight #      1069 non-null   object
 6   Route         3562 non-null   object
 7   Type          5241 non-null   object
 8   Registration  4933 non-null   object
 9   cn/In         4040 non-null   object
 10  Aboard        5246 non-null   float64
 11  Fatalities    5256 non-null   float64
 12  Ground        5246 non-null   float64
 13  Summary       4878 non-null   object
dtypes: float64(3), int64(1), object(10)
memory usage: 576.3+ KB
```

In [6]:
```
#description of first dataset
data.describe(include='all')
```

Out[6]:

	index	Date	Time	Location	Operator	Flight #	Route	Type	Registration	cn/In	Aboard	Fatalities	Ground	Summary
count	5268.00000	5268	3049	5248	5250	1069	3562	5241	4933	4040	5246.000000	5256.000000	5246.000000	4878
unique	NaN	4753	1005	4303	2476	724	3244	2446	4905	3707	NaN	NaN	NaN	4673
top	NaN	09/11/2001	15:00	Sao Paulo, Brazil	Aeroflot	-	Training	Douglas DC-3	49	178	NaN	NaN	NaN	Crashed during takeoff.
freq	NaN	4	32	15	179	67	81	334	3	6	NaN	NaN	NaN	15
mean	2633.50000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	27.554518	20.068303	1.608845	NaN
std	1520.88494	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	43.076711	33.199952	53.987827	NaN
min	0.00000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.000000	0.000000	0.000000	NaN
25%	1316.75000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5.000000	3.000000	0.000000	NaN
50%	2633.50000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	13.000000	9.000000	0.000000	NaN
75%	3950.25000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	30.000000	23.000000	0.000000	NaN
max	5267.00000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	644.000000	583.000000	2750.000000	NaN

In [7]:
```
#Here we will find null values in our dataset
data.isnull().sum()
```

Out[7]:
```
index            0
Date             0
Time          2219
Location        20
Operator        18
Flight #      4199
Route         1706
Type            27
Registration   335
cn/In         1228
Aboard          22
Fatalities      12
Ground          22
Summary        390
dtype: int64
```

Looking above it is seen that the columns have a huge number of missing values present which need to be revised in order to make the analysis better. Keeping the tasks in mind we will remove the columns that won't be used for analysis in order to reduce the complexity of removing and fixing null data in our dataset. The following columns are removed.

- Time
- Route
- Flight #
- cn/In
- Registration
- Ground

In [8]:
```
#removing columns that won't be used and with high null values
data=data.drop(['Time','Route','Flight #','cn/In','Registration','Ground'] ,axis=1)
```

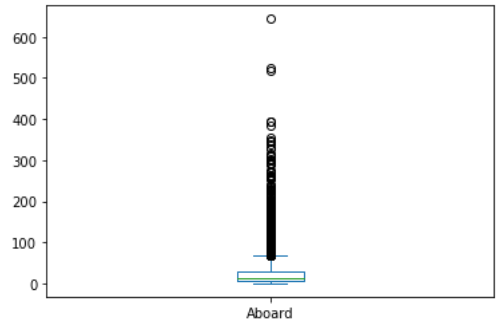
After removing the columns with the highest number of missing values, the remaining columns are left with a small amount of null values which can be adjusted by removing the rows having missing values which in return would arrage the dataset for further processing.

In [9]:
```
data.head()
```

Out[9]:

	index	Date	Location	Operator	Type	Aboard	Fatalities	Summary
0	0	09/17/1908	Fort Myer, Virginia	Military - U.S. Army	Wright Flyer III	2.0	1.0	During a demonstration flight, a U.S. Army fly...
1	1	07/12/1912	AtlantiCity, New Jersey	Military - U.S. Navy	Dirigible	5.0	5.0	First U.S. dirigible Akron exploded just offsh...
2	2	08/06/1913	Victoria, British Columbia, Canada	Private	Curtiss seaplane	1.0	1.0	The first fatal airplane accident in Canada oc...
3	3	09/09/1913	Over the North Sea	Military - German Navy	Zeppelin L-1 (airship)	20.0	14.0	The airship flew into a thunderstorm and encou...
4	4	10/17/1913	Near Johannisthal, Germany	Military - German Navy	Zeppelin L-2 (airship)	30.0	30.0	Hydrogen gas which was being vented was sucked...

In [10]:
```
#Null values exist in the remaining columns outliers exist
data['Aboard'].plot(kind='box')
plt.show()
```



In [11]:
```
#Removing the to rows with null values
data.dropna(subset=['Operator','Aboard','Location','Type','Fatalities'],inplace=True)
```

Here, we will convert the Date column into datetime

In [12]:
```
#To make it easier for further analysis we will convert the Date column
data["Date"]=pd.to_datetime(data["Date"])
```

In [13]:
```
data.head(5)
```

Out[13]:

	index	Date	Location	Operator	Type	Aboard	Fatalities	Summary
0	0	1908-09-17	Fort Myer, Virginia	Military - U.S. Army	Wright Flyer III	2.0	1.0	During a demonstration flight, a U.S. Army fly...
1	1	1912-07-12	AtlantiCity, New Jersey	Military - U.S. Navy	Dirigible	5.0	5.0	First U.S. dirigible Akron exploded just offsh...
2	2	1913-08-06	Victoria, British Columbia, Canada	Private	Curtiss seaplane	1.0	1.0	The first fatal airplane accident in Canada oc...
3	3	1913-09-09	Over the North Sea	Military - German Navy	Zeppelin L-1 (airship)	20.0	14.0	The airship flew into a thunderstorm and encou...
4	4	1913-10-17	Near Johannisthal, Germany	Military - German Navy	Zeppelin L-2 (airship)	30.0	30.0	Hydrogen gas which was being vented was sucked...

The dataset now contains almost none missing values and is cleaned and prepared for further processing.

In [14]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5191 entries, 0 to 5267
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   index       5191 non-null   int64
 1   Date        5191 non-null   datetime64[ns]
 2   Location    5191 non-null   object
 3   Operator    5191 non-null   object
 4   Type        5191 non-null   object
 5   Aboard      5191 non-null   float64
 6   Fatalities  5191 non-null   float64
 7   Summary     4823 non-null   object
dtypes: datetime64[ns](1), float64(2), int64(1), object(4)
memory usage: 365.0+ KB
```

In [15]:
```
#First Data Set cleaned
data.isnull().sum()
```

Out[15]:
```
index           0
Date            0
Location        0
Operator        0
Type            0
Aboard          0
Fatalities      0
Summary       368
dtype: int64
```

After cleaning the first dataset we move on to the second one by having a general overview of the dataset.

In [16]:
```
#Historical Plane Crash Data dataset
crashes_data.head()
```

Out[16]:

	Date	Time	Aircraft	Operator	Registration	Flight phase	Flight type	Survivors	Crash site	Schedule	...	Country	Region	Crew on board	Crew fatalities	Pax on board	PAX fatalities	fa
0	1918-05-02	NaN	De Havilland DH.4	United States Signal Corps - USSC	AS-32084	Takeoff (climb)	Test	No	Airport (less than 10 km from airport)	Dayton - Dayton	...	United States of America	North America	2.0	2.0	0.0	0.0	
1	1918-06-08	NaN	Handley Page V/1500	Handley Page Aircraft Company Ltd	E4104	Takeoff (climb)	Test	Yes	Airport (less than 10 km from airport)	Cricklewood - Cricklewood	...	United Kingdom	Europe	6.0	5.0	0.0	0.0	
2	1918-06-11	NaN	Avro 504	Royal Air Force - RAF	A8544	Flight	Training	Yes	Plain, Valley	Abukir - Abukir	...	Egypt	Africa	2.0	1.0	0.0	0.0	
3	1918-06-19	NaN	De Havilland DH.4	United States Signal Corps - USSC	AS-32098	Flight	Military	No	Airport (less than 10 km from airport)	Wright Patterson AFB-Wright Patterson AFB	...	United States of America	North America	1.0	1.0	0.0	0.0	
4	1918-06-24	NaN	Breguet 14	French Air Force - Armée de l'Air	AS-4130	Landing (descent or approach)	Military	Yes	NaN	NaN	...	France	Europe	NaN	0.0	NaN	0.0	

5 rows × 24 columns

In [17]:
```
crashes_data.describe()
```

Out[17]:

	YOM	Flight no.	Crew on board	Crew fatalities	Pax on board	PAX fatalities	Other fatalities	Total fatalities
count	23225.000000	0.0	28512.000000	28535.000000	28482.000000	28535.000000	28526.000000	28536.000000
mean	1931.942519	NaN	3.052539	1.771649	7.705393	3.679727	0.109760	5.567389
std	285.486067	NaN	11.738151	2.520554	24.066368	15.288171	2.644296	16.713203
min	0.000000	NaN	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1944.000000	NaN	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	1958.000000	NaN	2.000000	1.000000	0.000000	0.000000	0.000000	1.000000
75%	1974.000000	NaN	4.000000	3.000000	4.000000	1.000000	0.000000	5.000000
max	19567.000000	NaN	1924.000000	25.000000	509.000000	506.000000	297.000000	520.000000

In [18]:
```
crashes_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28536 entries, 0 to 28535
Data columns (total 24 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Date              28536 non-null  object 
 1   Time              13949 non-null  object 
 2   Aircraft          28535 non-null  object 
 3   Operator          28536 non-null  object 
 4   Registration      27721 non-null  object 
 5   Flight phase      27898 non-null  object 
 6   Flight type       28479 non-null  object 
 7   Survivors         27239 non-null  object 
 8   Crash site        28153 non-null  object 
 9   Schedule          19590 non-null  object 
 10  MSN               24354 non-null  object 
 11  YOM               23225 non-null  float64
 12  Flight no.        0 non-null      float64
 13  Crash location    28524 non-null  object 
 14  Country           28535 non-null  object 
 15  Region            28535 non-null  object 
 16  Crew on board     28512 non-null  float64
 17  Crew fatalities   28535 non-null  float64
 18  Pax on board      28482 non-null  float64
 19  PAX fatalities    28535 non-null  float64
 20  Other fatalities  28526 non-null  float64
 21  Total fatalities  28536 non-null  int64  
 22  Circumstances     28511 non-null  object 
 23  Crash cause       28536 non-null  object 
dtypes: float64(7), int64(1), object(16)
memory usage: 5.2+ MB
```

Keeping in mind the processing and preparing steps for our first dataset the similar tasks would be performed on the second dataset with checking our dataset for any null values, which would be removed afterwards to prepare the dataset for further processing. The columns with highest number of missing values would be removed from the dataset

In [19]:
```
crashes_data.isnull().sum()
```

Out[19]:
```
Date                   0
Time               14587
Aircraft               1
Operator               0
Registration         815
Flight phase         638
Flight type           57
Survivors           1297
Crash site           383
Schedule            8946
MSN                 4182
YOM                 5311
Flight no.         28536
Crash location        12
Country                1
Region                 1
Crew on board         24
Crew fatalities        1
Pax on board          54
PAX fatalities         1
Other fatalities      10
Total fatalities       0
Circumstances         25
Crash cause            0
dtype: int64
```

In [20]:
```
#removing columns that won't be used and with high null values
crashes_data=crashes_data.drop(['Time','Survivors','Flight no.','Schedule','MSN','YOM','Pax on board',
                                'PAX fatalities',
                                 'Other fatalities'],axis=1 )
```

In [21]:
```
crashes_data.head(5)
```

Out[21]:

	Date	Aircraft	Operator	Registration	Flight phase	Flight type	Crash site	Crash location	Country	Region	Crew on board	Crew fatalities	Total fatalities	Circumstances	Crash cause
0	1918-05-02	De Havilland DH.4	United States Signal Corps - USSC	AS-32084	Takeoff (climb)	Test	Airport (less than 10 km from airport)	Dayton-McCook Field Ohio	United States of America	North America	2.0	2.0	2	The single engine airplane departed Dayton-McC...	Technical failure
1	1918-06-08	Handley Page V/1500	Handley Page Aircraft Company Ltd	E4104	Takeoff (climb)	Test	Airport (less than 10 km from airport)	Cricklewood London Metropolis	United Kingdom	Europe	6.0	5.0	5	Assembled at Cricklewood Airfield in May 1918,...	Technical failure
2	1918-06-11	Avro 504	Royal Air Force - RAF	A8544	Flight	Training	Plain, Valley	Abukir (Abu Qir) Alexandria	Egypt	Africa	2.0	1.0	1	The single engine aircraft was completing a lo...	Unknown
3	1918-06-19	De Havilland DH.4	United States Signal Corps - USSC	AS-32098	Flight	Military	Airport (less than 10 km from airport)	Wright-Patterson AFB (Dayton) Ohio	United States of America	North America	1.0	1.0	1	Lt. Frank Stuart Patterson, son and nephew of ...	Technical failure
4	1918-06-24	Breguet 14	French Air Force - Armée de l'Air	AS-4130	Landing (descent or approach)	Military	NaN	France All France	France	Europe	NaN	0.0	0	The aircraft crashed iupon landing somewhere i...	Unknown

In [22]:
```
data.tail()
```

Out[22]:

	index	Date	Location	Operator	Type	Aboard	Fatalities	Summary
5263	5263	2009-05-20	Near Madiun, Indonesia	Military - Indonesian Air Force	Lockheed C-130 Hercules	112.0	98.0	While on approach, the military transport cras...
5264	5264	2009-05-26	Near Isiro, DemocratiRepubliCongo	Service Air	Antonov An-26	4.0	4.0	The cargo plane crashed while on approach to I...
5265	5265	2009-06-01	AtlantiOcean, 570 miles northeast of Natal, Br...	Air France	Airbus A330-203	228.0	228.0	The Airbus went missing over the AtlantiOcean ...
5266	5266	2009-06-07	Near Port Hope Simpson, Newfoundland, Canada	Strait Air	Britten-Norman BN-2A-27 Islander	1.0	1.0	The air ambulance crashed into hills while att...
5267	5267	2009-06-08	State of Arunachal Pradesh, India	Military - Indian Air Force	Antonov An-32	13.0	13.0	The military transport went missing while en r...

In [23]:
```
#Selecting the data frame for the crashes in last two decades
data_first=data[(data['Date']>'1999/01/01') & (data['Date']<='2009/06/08')]
data_second=data[(data['Date']>'1989/01/01') & (data['Date']<='1999/01/08')]
```

In [24]:
```
#Data from 1999 to 2009
data_first.head()
```

Out[24]:

	index	Date	Location	Operator	Type	Aboard	Fatalities	Summary
4572	4572	1999-02-25	Genoa, Italy	Minerva Airlines	Dornier 328-110	31.0	4.0	The aircraft touched down briefly, overran the...
4586	4586	1999-07-01	Luzamba, Angola	Savanair	Antonov 12B	5.0	1.0	The cargo plane was shot down by UNITA rebels.
4587	4587	2001-01-08	Near Silimo, Indonesia	Military - Tentara Nasional Indonesia Navy	CASA 212-MP Aviocar 200	9.0	9.0	Struck Timika Peak at 11,800 ft. shortly after...
4605	4605	1999-07-02	Sittwe, Myanmar	Myanmar Airways	Fokker F-27 Friendship 600	8.0	8.0	The aircraft, carrying freight, flew into the ...
4611	4611	1999-01-02	Near Huambo, Angola	Transafrik - United Nations Charter	Lockheed L-100-30 Hercules	9.0	9.0	he United Nations-chartered plane was shot dow...

In [25]:
```
#Data from 1989 to 1999
data_second.head()
```

Out[25]:

	index	Date	Location	Operator	Type	Aboard	Fatalities	Summary
2514	2514	1991-02-22	Cazombo, Angola	Fuerza Area Angolaise	Antonov AN-26	47.0	47.0	Shot down by surface-to-air missile launced by...
3537	3537	1989-10-18	Nasosny, Russia	Military - Russian Air Force	Ilyushin IL-76MD	57.0	57.0	Crashed into the Caspian sea after reporting a...
3817	3817	1989-03-15	West Lafayette, Indiana	Mid PacifiAir	NAMC YS-11A-300F	2.0	2.0	While landing the cargo plane pitched down and...
3818	3818	1989-01-08	Leicestershire, England	British Midland Airways	Boeing B-737-4Y0	126.0	47.0	While en route and climbing through FL 280, a ...
3819	3819	1989-01-12	Dayton, Ohio	Bradley Air Services	Hawker Siddeley Avro 748-215	2.0	2.0	After gaining altitude the cargo plane descend...

Survival Rate in Aviation accidents

To find the rate of survival percentage in avidation accidents we will make use of our first dataset which contains data columns of passengers aboard and the fatalities caused after the accident. In order to calculate the survival rate passengers we will subtract the Aboard column with the Fatalities which will be divided by the people Aboard and multiplied with 100 to return the percentage of survival, the result rate is stored in a new column Survived rate.

In [26]:
```
#We will find the survival rate overall in out dataset
data["Survived rate"]=100*(data['Aboard']-data['Fatalities'])/data['Aboard']
```

In [27]:
```
data.head()
```

Out[27]:

	index	Date	Location	Operator	Type	Aboard	Fatalities	Summary	Survived rate
0	0	1908-09-17	Fort Myer, Virginia	Military - U.S. Army	Wright Flyer III	2.0	1.0	During a demonstration flight, a U.S. Army fly...	50.0
1	1	1912-07-12	AtlantiCity, New Jersey	Military - U.S. Navy	Dirigible	5.0	5.0	First U.S. dirigible Akron exploded just offsh...	0.0
2	2	1913-08-06	Victoria, British Columbia, Canada	Private	Curtiss seaplane	1.0	1.0	The first fatal airplane accident in Canada oc...	0.0
3	3	1913-09-09	Over the North Sea	Military - German Navy	Zeppelin L-1 (airship)	20.0	14.0	The airship flew into a thunderstorm and encou...	30.0
4	4	1913-10-17	Near Johannisthal, Germany	Military - German Navy	Zeppelin L-2 (airship)	30.0	30.0	Hydrogen gas which was being vented was sucked...	0.0

In [28]:
```
#Finding the  MEDIAN MEAN, Standard Deviation and Variance of the Survival Rate
data_mean=data['Survived rate'].mean()
data_median=data['Survived rate'].median()

data_std=data['Survived rate'].std()
data_variance=data['Survived rate'].var()
```

In [29]:
```
print('The mean of survived rate is :',round(data_mean,2))
```

```
The mean of survived rate is : 16.57
```

In [30]:
```
print("The median of Survival rate is: ",round(data_median,2))
```

```
The median of Survival rate is:  0.0
```

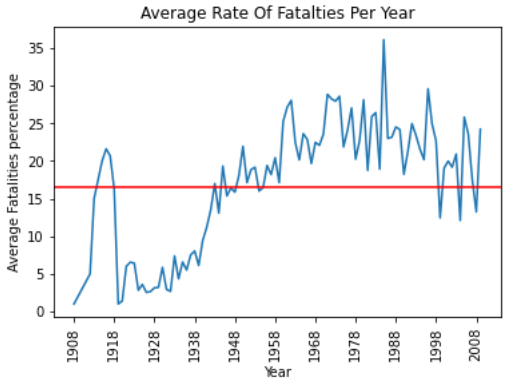
In [31]:
```
print("The standard deviation of survived rate is: ",round(data_std,2))
```

```
The standard deviation of survived rate is:  29.97
```

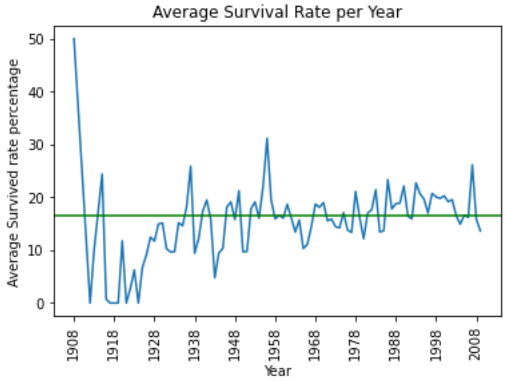
In [32]:
```
print("The variation of survived rate is: ",round(data_variance,2))
```

```
The variation of survived rate is:  897.95
```

In [33]:
```
fatalities_yearly = data[["Fatalities"]].groupby(data["Date"].dt.year).agg(["mean"])
fatalities_yearly.plot(legend=None)
plt.ylabel("Average Fatalities percentage")
plt.xlabel("Year")
plt.title('Average Rate Of Fatalties Per Year')
plt.xticks([x for x in range(1908,2009,10)], rotation='vertical') #Dataset contains reports from 1908 uptil 2009
plt.axhline(y=data_mean, color='r', linestyle='-')
plt.show()
```



In [34]:
```
yearly_survival = data[["Survived rate"]].groupby(data["Date"].dt.year).agg(["mean"])
yearly_survival.plot(legend=None)
plt.ylabel("Average Survived rate percentage")
plt.xlabel("Year")
plt.title("Average Survival Rate per Year")
plt.xticks([x for x in range(1908,2009,10)], rotation='vertical')
plt.axhline(y=data_mean, color='g', linestyle='-')
plt.show()
```



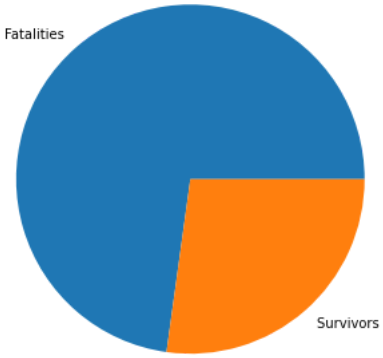
In [35]:
```
#Finding survivors based on fatalities in piechart
plt.figure(figsize=(8,6))
Aboard = data['Aboard'].sum()

Fatalities = data['Fatalities'].sum()

Survivors = Aboard - Fatalities

y = np.array([Fatalities, Survivors])
mylabels = ["Fatalities", "Survivors "]

plt.pie(y, labels = mylabels)
plt.show()
```



Seeing the output it is seen that the average rate of survival is around 16.50%. which can also varry as the dataset contains data uptill 2009.

Safety Improvements throughout the years

Keeping the rate of survival and fatalities in mind, the question that arises the most is on wheter airtravel is more safer than before with safety improvements made throughout. To answer these questions we will use our second dataset, as it contains data records uptill 2022. we will thee categorize our dataset in two time frames based on thirty years each starting from 1960-1989 and 1990-2022 respectively.

In [36]:
```
#Taking crash data based on two time frames first from 1960 till 1989 and the second from 1999 till 2022
crashes_data_60=crashes_data[(crashes_data['Date']>'1960/01/01') & (crashes_data['Date']<='1989/12/12')]
crashes_data_90=crashes_data[(crashes_data['Date']>'1990/01/01') & (crashes_data['Date']<='2022/12/12')]
```

In [37]:
```
crashes_data.head()
```

Out[37]:

	Date	Aircraft	Operator	Registration	Flight phase	Flight type	Crash site	Crash location	Country	Region	Crew on board	Crew fatalities	Total fatalities	Circumstances	Crash cause
0	1918-05-02	De Havilland DH.4	United States Signal Corps - USSC	AS-32084	Takeoff (climb)	Test	Airport (less than 10 km from airport)	Dayton-McCook Field Ohio	United States of America	North America	2.0	2.0	2	The single engine airplane departed Dayton-McC...	Technical failure
1	1918-06-08	Handley Page V/1500	Handley Page Aircraft Company Ltd	E4104	Takeoff (climb)	Test	Airport (less than 10 km from airport)	Cricklewood London Metropolis	United Kingdom	Europe	6.0	5.0	5	Assembled at Cricklewood Airfield in May 1918,...	Technical failure
2	1918-06-11	Avro 504	Royal Air Force - RAF	A8544	Flight	Training	Plain, Valley	Abukir (Abu Qir) Alexandria	Egypt	Africa	2.0	1.0	1	The single engine aircraft was completing a lo...	Unknown
3	1918-06-19	De Havilland DH.4	United States Signal Corps - USSC	AS-32098	Flight	Military	Airport (less than 10 km from airport)	Wright-Patterson AFB (Dayton) Ohio	United States of America	North America	1.0	1.0	1	Lt. Frank Stuart Patterson, son and nephew of ...	Technical failure
4	1918-06-24	Breguet 14	French Air Force - Armée de l'Air	AS-4130	Landing (descent or approach)	Military	NaN	France All France	France	Europe	NaN	0.0	0	The aircraft crashed iupon landing somewhere i...	Unknown

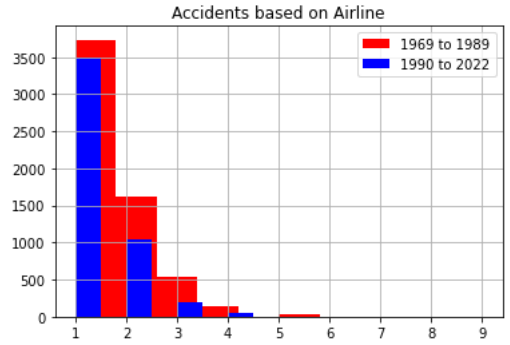
In [38]:
```
crashes_data_60.head(5)
```

Out[38]:

	Date	Aircraft	Operator	Registration	Flight phase	Flight type	Crash site	Crash location	Country	Region	Crew on board	Crew fatalities	Total fatalities	Circumstances
12553	1961-01-02	Douglas C-47 Skytrain (DC-3)	Royal Netherlands Air Force - Koninklijke Luch...	079	Flight	Training	Lake, Sea, Ocean, River	Biak Special Region of Papua	Indonesia	Asia	5.0	5.0	5	The aircraft wa involved in a night training ..
12554	1961-01-02	Avia 14	CSA Czech Airlines - Ceskoslovenské Státní Aer...	OK-MCZ	Takeoff (climb)	Training	Airport (less than 10 km from airport)	Prague-Ruzyne Prague (Hlavní mesto Praha)	Czech Republic	Europe	5.0	5.0	10	After takeo from Prague Ruzyne Airport the ..
12555	1961-01-03	Douglas C-47 Skytrain (DC-3)	Finnair	OH-LCC	Landing (descent or approach)	Scheduled Revenue Flight	Airport (less than 10 km from airport)	Koivulahti Ostrobothnia	Finland	Europe	3.0	3.0	25	Whil approachin Vaasa Airpor by night, the ..
12556	1961-01-03	Antonov AN-2	Aeroflot - Russian International Airlines	CCCP-25482	Flight	Scheduled Revenue Flight	Mountains	Santash Pass Issyk Kul Province	Kyrgyzstan	Asia	2.0	2.0	9	Seve passenger embarked a Przhevals Airpor..
12557	1961-01-04	Martin PBM Mariner	Argentinian Navy - Armada Argentina	2-P-206	Takeoff (climb)	Military	Lake, Sea, Ocean, River	Puerto Belgrano NAS Buenos Aires province	Argentina	South America	10.0	0.0	0	Crashed int the Bahí Blanca short after ta..

In [39]:
```
#airplane safety over the years
#1960-1989 1990-2022
crash_60=crashes_data_60.Date.value_counts()
crash_90=crashes_data_90.Date.value_counts()
crash_60.hist(label='1969 to 1989',alpha=1,color='r')
crash_90.hist(label='1990 to 2022',alpha=1,color='b')
plt.legend(loc="upper right")

plt.xlabel=("Accidents based on Airplane Operators")
plt.ylabel=("Frequency")
plt.title("Accidents based on Airline")
plt.show()
```



In [40]:
```
crash_60=crashes_data_60.Date.value_counts()
print("Crash Stats:")
print('Total Crashes from 1960 to 1989 :',len(crash_60))
print("The mean of total incidents from 1960 to 1989 is : ",round(crash_60.mean(),2))
print("Variance : ",round(crash_60.var(),2))
print("Standard Deviation : ",round(crash_60.std(),2))
print('Min : ',round(crash_60.min(),2))
print('Max : ',round(crash_60.max(),2))
print("Skewness ",round(crash_60.skew(),2))
```

```
Crash Stats:
Total Crashes from 1960 to 1989 : 6057
The mean of total incidents from 1960 to 1989 is :  1.53
Variance :  0.64
Standard Deviation :  0.8
Min :  1
Max :  9
Skewness  1.72
```

In [41]:
```
crash_90=crashes_data_90.Date.value_counts()
print("Crash Stats:")
print('Total Crashes from 1990 to 2022 :',len(crash_90))
print("The mean of total incidents from 1990 to 2022 is : ",round(crash_90.mean(),2))
print("Variance : ",round(crash_90.var(),2))
print("Standard Deviation : ",round(crash_90.std(),2))
print('Min : ',round(crash_90.min(),2))
print('Max : ',round(crash_90.max(),2))
print("Skewness ",round(crash_90.skew(),2))
```

```
Crash Stats:
Total Crashes from 1990 to 2022 : 4796
The mean of total incidents from 1990 to 2022 is :  1.34
Variance :  0.38
Standard Deviation :  0.62
Min :  1
Max :  6
Skewness  2.06
```

With the current stats noted, the aviation accidents averaged higher in our first category of data from 1969 to 1989, so we would need to find the difference between the two categories , as the data isn't distributed normally varrying the relationship between the two categories for which we will use the students t test

In [55]:
```
#Performing t test
t_stat,p_value=stats.ttest_ind(crash_60 ,crash_90)
print("t_stat: ",round(t_stat,3))
print("p_value: ",round(p_value,3))
```

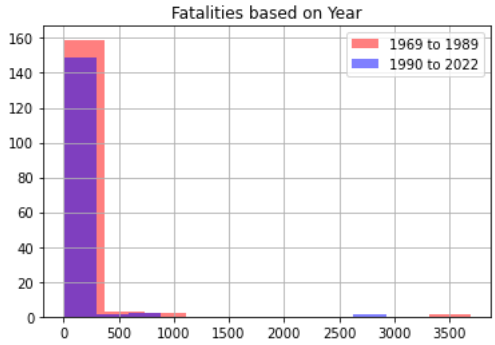
```
t_stat:  0.449
p_value:  0.654
```

The above cell shows that the t_stat value is 0.449 an the p_value is 0.654 which is greater then the threshold value of 0.05 which shows that its non significant and null hypothesis is not rejected

In [ ]:

In [43]:
```
#Fatalities Safety over the years
#1960-1989 1990-2022
crash_60=crashes_data_60['Total fatalities'].value_counts()
crash_90=crashes_data_90['Total fatalities'].value_counts()
crash_60.hist(label='1969 to 1989',alpha=0.5,color='r')
crash_90.hist(label='1990 to 2022',alpha=0.5,color='b')
plt.legend(loc="upper right")

plt.xlabel=("Fatalities based on Years ")
plt.ylabel=("Frequency")
plt.title("Fatalities based on Year")
plt.show()
```



From the above graph it is seen that the data isn't normally distributed and needs further analysis.

In [44]:
```
crash_60=crashes_data_60['Total fatalities'].value_counts()
print("Fatalities Stats:")
print('Total Fatalities from 1960 to 1989 :',sum(crash_60))
print("The mean of total Fatalities  from 1960 to 1989 is : ",round(crash_60.mean(),2))
print("Variance : ",round(crash_60.var(),2))
print("Standard Deviation : ",round(crash_60.std(),2))
print('Min : ',round(crash_60.min(),2))
print('Max : ',round(crash_60.max(),2))
print("Skewness ",round(crash_60.skew(),2))
```

```
Fatalities Stats:
Total Fatalities from 1960 to 1989 : 9286
The mean of total Fatalities  from 1960 to 1989 is :  56.28
Variance :  97199.01
Standard Deviation :  311.77
Min :  1
Max :  3693
Skewness  10.14
```

In [45]:
```
crash_90=crashes_data_90['Total fatalities'].value_counts()
print("Fatalities Stats:")
print('Total Fatalities from 1990 to 2022 :',sum(crash_90))
print("The mean of total Fatalities from 1990 to 2022 is : ",round(crash_90.mean(),2))
print("Variance : ",round(crash_90.var(),2))
print("Standard Deviation : ",round(crash_90.std(),2))
print('Min : ',round(crash_90.min(),2))
print('Max : ',round(crash_90.max(),2))
print("Skewness ",round(crash_90.skew(),2))
```

```
Fatalities Stats:
Total Fatalities from 1990 to 2022 : 6415
The mean of total Fatalities from 1990 to 2022 is :  41.93
Variance :  63950.75
Standard Deviation :  252.88
Min :  1
Max :  2924
Skewness  10.17
```

We will perform a t test as our category data is not normally distributed which requires further testing.

In [46]:
```
#Performing t test
t_stat,p_value=stats.ttest_ind(crashes_data_60['Total fatalities'], crashes_data_90['Total fatalities'])
print("t_stat: ",round(t_stat,3))
print("p_value: ",round(p_value,3))
```

```
t_stat:  3.03
p_value:  0.002
```

The above cell shows that the t_stat value is 3.03 an the p_value is 0.002 which is less then the threshold value of 0.05 which rejects the null hypothesis.

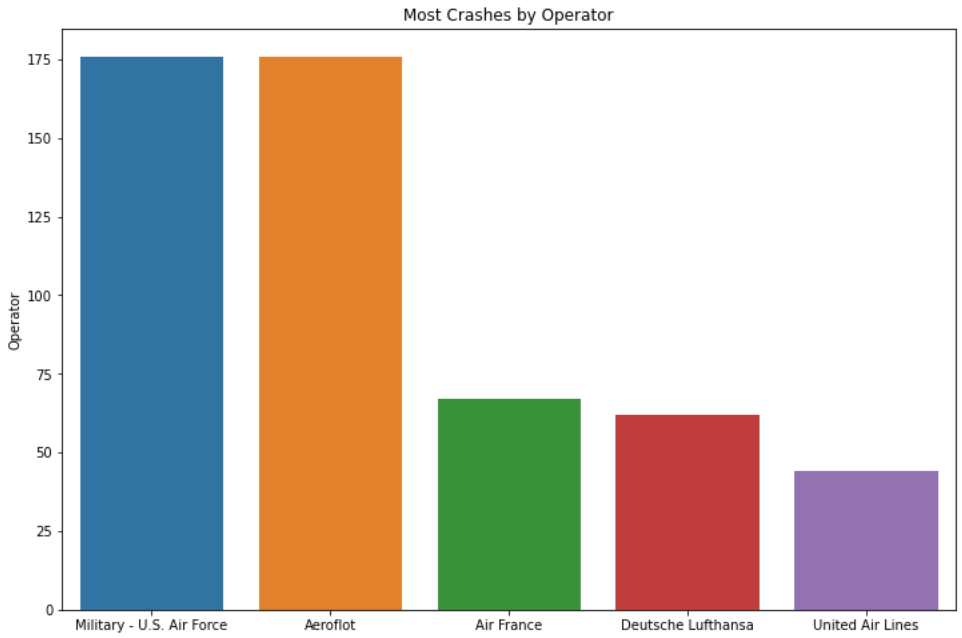
Airlines with the most incidents recorded and which ones to watch out for

We will analyze the aviation incidents based on airlines operator and which ones have had the most crashes so far and the type of plane which could help us derive the information for our analysis based on these incidents. The airline operator in our datasets could be Commercial airline, Military Force, Private, Other

In [47]:
```
#Finding the airlines with the most Accidents
data_operator=data.Operator.value_counts().head(5)
data_operator
```

Out[47]:
```
Military - U.S. Air Force    176
Aeroflot                     176
Air France                    67
Deutsche Lufthansa            62
United Air Lines              44
Name: Operator, dtype: int64
```

In [48]:
```
#Most crashes by operator
fig,ax = plt.subplots(figsize = (12,8))
sns.barplot(x =data_operator.index, y = data_operator.head(5))
plt.title("Most Crashes by Operator")
plt.ylabel="Operator"
plt.xlabel="Count"
plt.show()
```



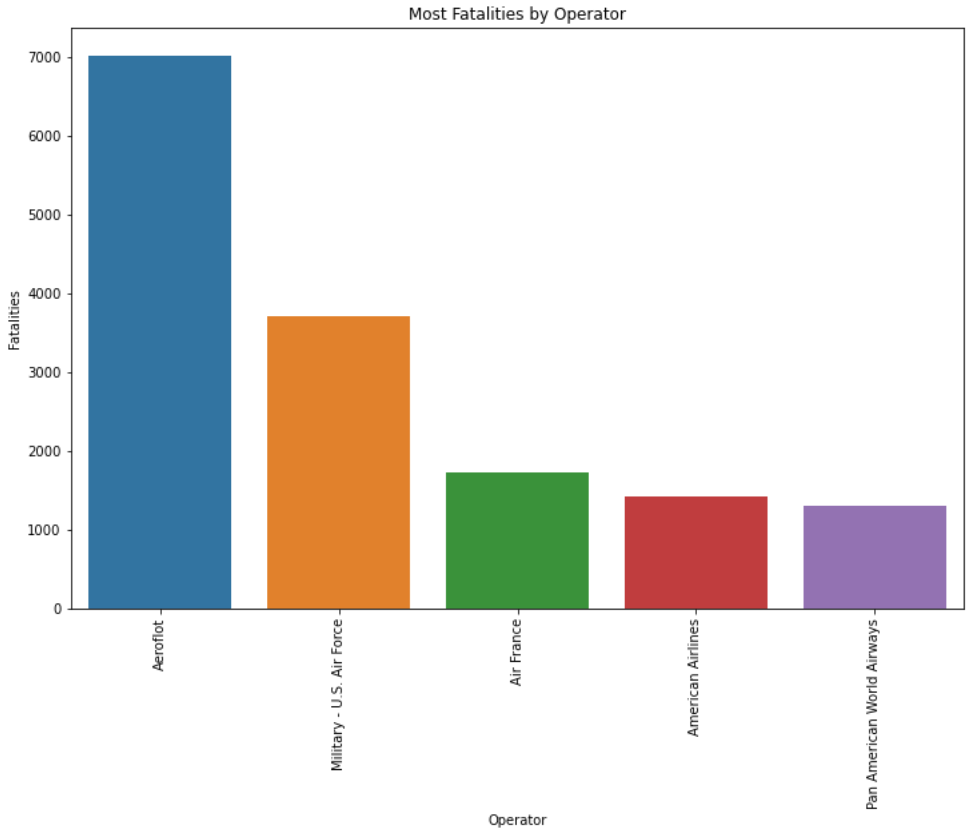
In the above cell it is seen that Aeroflot and Military-U.S Air Force are the two operators with the most incidents overall, for a more indepth analysis we will find the crashes data with the number of fatalaties overall

In [49]:
```
#Finding the airline with the highest number of fatalities
data_crash_death=data[['Fatalities']].groupby(data['Operator']).sum()
data_crash_death=data_crash_death.sort_values(by=['Fatalities'],ascending=False)
data_crash_death=data_crash_death.head(5)
data_crash_death
```

Out[49]:

Operator	Fatalities
Aeroflot	7016.0
Military - U.S. Air Force	3717.0
Air France	1729.0
American Airlines	1421.0
Pan American World Airways	1301.0

In [50]:
```
fig,ax = plt.subplots(figsize = (12,8))
sns.barplot(x =data_crash_death.index,y=data_crash_death['Fatalities'])
plt.title('Most Fatalities by Operator')
plt.ylabel="Operator"
plt.xlabel='Count'
ticks = plt.setp(ax.get_xticklabels(),rotation=90)
```

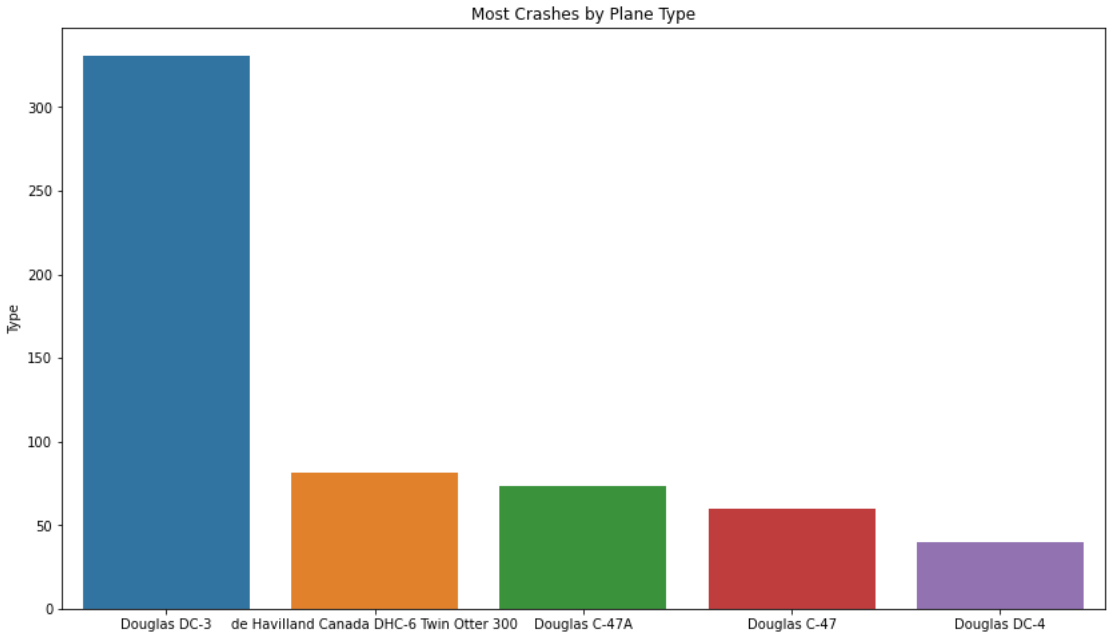


The above visualiztion shows that Aeroflot has had the highest number of fatalities with 7016 which makes it the most dangerous airline in our dataset to travel

In [51]:
```
#finding the plane type with the most incidents
data_plane_type=data.Type.value_counts().head(5)
data_plane_type
```

Out[51]:
```
Douglas DC-3                               331
de Havilland Canada DHC-6 Twin Otter 300    81
Douglas C-47A                               73
Douglas C-47                                60
Douglas DC-4                                40
Name: Type, dtype: int64
```

In [52]:
```
fig,ax = plt.subplots(figsize = (14,8))
sns.barplot(x =data_plane_type.index, y = data_plane_type,).set(title='Width = 0.4')
plt.title('Most Crashes by Plane Type')
plt.ylabel="Plane Type"
plt.xlabel='Count'
plt.show()
```



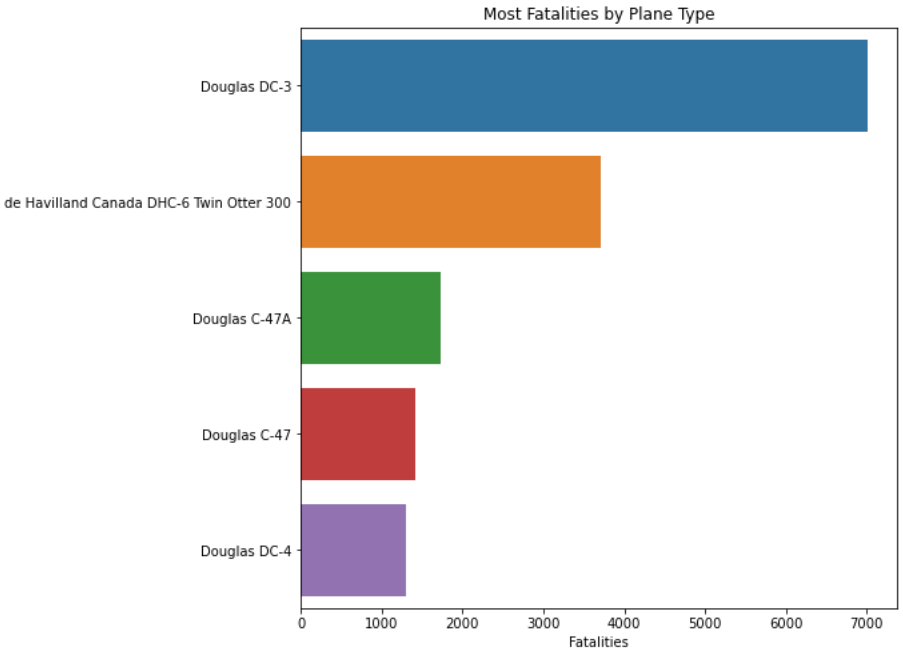
From the Above cell it is seen that the Douglas DC-3 has been the plane type with most crashes overall with almost 331 crashes so far. Now we will try to find out which plane type had the highest number of fatalities

In [53]:
```
#Finding the Airplane type with the highest number of fatalities
data_crash_death_type=data[['Fatalities']].groupby(data['Type']).sum()
data_crash_death_type=data_crash_death_type.sort_values(by=['Fatalities'],ascending=False)
data_crash_death_type=data_crash_death.head(5)
data_crash_death_type
```

Out[53]:

Operator	Fatalities
Aeroflot	7016.0
Military - U.S. Air Force	3717.0
Air France	1729.0
American Airlines	1421.0
Pan American World Airways	1301.0

In [54]:
```
fig,ax = plt.subplots(figsize = (8,8))
sns.barplot(y =data_plane_type.index, x = data_crash_death_type.Fatalities)
plt.title('Most Fatalities by Plane Type')
plt.ylabel="Plane Type"
plt.xlabel='Count'
plt.show()
```



From the figure above it is clear that the Douglas DC-3 is the plane type having the higher number of fatalities followed by the other plane types