NTSB Aviation Accident dataset

The business is interested in becoming involved in commercial aviation.

An initial overview of the risks in aviation is the purpose of this data analysis.

The dataset for the overview is the National Transportation Safety Board (NTSB) dataset that covers the years 1948 through the end of 2022.

Open the dataset called AviationData.csv

In [1]:

```
import pandas as pd

df = pd.read_csv('working-df/AviationData.csv', encoding='latin-1')

df.info()
```

```
C:\Users\Joseph Harvey\AppData\Local\Temp\ipykernel_9224\2841597984.py:3: DtypeWarning: C
olumns (6,7,28) have mixed types. Specify dtype option on import or set low_memory=False.
  df = pd.read_csv('working-df/AviationData.csv', encoding='latin-1')
```

```
<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
```

Cleaning the Data

rename columns to remove dots as they may cause errors in Python (replace dots with dashes or underscores)

In [2]:

```
df.columns = df.columns.str.replace('.', '_')

df.head()
```

Out[2]:

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

5 rows × 31 columns

In [3]:

```
# Check for duplicate rows
duplicate_rows_events = df[df.duplicated(subset=['Event_Id'], keep=False)]

duplicate_rows_events.head(10)
```

Out[3]:

	Event_Id	Investigation_Type	Accident_Number	Event_Date	Location	Country	Latitude	Longitude	Airport_
117	20020917X01908	Accident	DCA82AA012B	1982-01-19	ROCKPORT, TX	United States	NaN	NaN	
118	20020917X01908	Accident	DCA82AA012A	1982-01-19	ROCKPORT, TX	United States	NaN	NaN	
153	20020917X02259	Accident	LAX82FA049A	1982-01-23	VICTORVILLE, CA	United States	NaN	NaN	
158	20020917X02400	Accident	MIA82FA038B	1982-01-23	NEWPORT RICHEY, FL	United States	NaN	NaN	
159	20020917X02400	Accident	MIA82FA038A	1982-01-23	NEWPORT RICHEY, FL	United States	NaN	NaN	
160	20020917X02259	Accident	LAX82FA049B	1982-01-23	VICTORVILLE, CA	United States	NaN	NaN	
242	20020917X02585	Accident	SEA82DA028A	1982-02-06	MEDFORD, OR	United States	NaN	NaN	
244	20020917X02173	Accident	LAX82DA065B	1982-02-06	SAN JOSE, CA	United States	NaN	NaN	
245	20020917X02585	Accident	SEA82DA028B	1982-02-06	MEDFORD, OR	United States	NaN	NaN	
248	20020917X02173	Accident	LAX82DA065A	1982-02-06	SAN JOSE, CA	United States	NaN	NaN	

10 rows × 31 columns

I see here that though these duplicate rows do represent separate aircraft in multi-aircraft incidents, the injury and/or fatality numbers are combined. This would constitute duplicate numbers in certain columns that would render errors in the analysis.

So let's remove the duplicates from this subset.

In [4]:

```
# remove the duplicate rows using the Event_Id column
df = df.drop_duplicates(subset=['Event_Id'], keep='first')
```

In [5]:

```
# check for duplicates again in Event_Id column
duplicate_rows_events = df[df.duplicated(subset=['Event_Id'], keep=False)]

duplicate_rows_events.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 0 entries
Data columns (total 31 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Event_Id                0 non-null      object 
 1   Investigation_Type      0 non-null      object 
 2   Accident_Number         0 non-null      object 
 3   Event_Date              0 non-null      object 
 4   Location                0 non-null      object 
 5   Country                 0 non-null      object 
 6   Latitude                0 non-null      object 
 7   Longitude               0 non-null      object 
 8   Airport_Code            0 non-null      object 
 9   Airport_Name            0 non-null      object 
 10  Injury_Severity         0 non-null      object 
 11  Aircraft_damage         0 non-null      object 
 12  Aircraft_Category       0 non-null      object 
 13  Registration_Number     0 non-null      object 
 14  Make                    0 non-null      object 
 15  Model                   0 non-null      object 
 16  Amateur_Built           0 non-null      object 
 17  Number_of_Engines       0 non-null      float64
 18  Engine_Type             0 non-null      object 
 19  FAR_Description         0 non-null      object 
 20  Schedule                0 non-null      object 
 21  Purpose_of_flight       0 non-null      object 
 22  Air_carrier             0 non-null      object 
 23  Total_Fatal_Injuries    0 non-null      float64
 24  Total_Serious_Injuries  0 non-null      float64
 25  Total_Minor_Injuries    0 non-null      float64
 26  Total_Uninjured         0 non-null      float64
 27  Weather_Condition       0 non-null      object 
 28  Broad_phase_of_flight   0 non-null      object 
 29  Report_Status           0 non-null      object 
 30  Publication_Date        0 non-null      object 
dtypes: float64(5), object(26)
memory usage: 0.0+ bytes
```

In [6]:

```
# check for duplicate rows in the Accident_Number column to verify there are no more duplicates
duplicate_rows_accidents = df[df.duplicated(subset=['Accident_Number'], keep=False)]

if duplicate_rows_accidents.empty:
  print("No duplicate rows found.")
else:
  print("Duplicate rows found.")
```

```
No duplicate rows found.
```

In [7]:

In [7]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 87951 entries, 0 to 88888
Data columns (total 31 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Event_Id                87951 non-null  object 
 1   Investigation_Type      87951 non-null  object 
 2   Accident_Number         87951 non-null  object 
 3   Event_Date              87951 non-null  object 
 4   Location                87899 non-null  object 
 5   Country                 87729 non-null  object 
 6   Latitude                34212 non-null  object 
 7   Longitude               34203 non-null  object 
 8   Airport_Code            49484 non-null  object 
 9   Airport_Name            52031 non-null  object 
 10  Injury_Severity         86961 non-null  object 
 11  Aircraft_damage         84848 non-null  object 
 12  Aircraft_Category       32181 non-null  object 
 13  Registration_Number     86601 non-null  object 
 14  Make                    87888 non-null  object 
 15  Model                   87859 non-null  object 
 16  Amateur_Built           87851 non-null  object 
 17  Number_of_Engines       81924 non-null  float64
 18  Engine_Type             80908 non-null  object 
 19  FAR_Description         31915 non-null  object 
 20  Schedule                12360 non-null  object 
 21  Purpose_of_flight       81829 non-null  object 
 22  Air_carrier             16533 non-null  object 
 23  Total_Fatal_Injuries    76684 non-null  float64
 24  Total_Serious_Injuries  75629 non-null  float64
 25  Total_Minor_Injuries    76191 non-null  float64
 26  Total_Uninjured         82088 non-null  float64
 27  Weather_Condition       83478 non-null  object 
 28  Broad_phase_of_flight   60837 non-null  object 
 29  Report_Status           81587 non-null  object 
 30  Publication_Date        74352 non-null  object 
dtypes: float64(5), object(26)
memory usage: 21.5+ MB
```

So now we have 87,951 accident records to work with.

Columns that are not needed

Remove certain columns that are mostly empty and would not contain data useful to the intended analysis.

I propose removing Latitude, Longitude, Schedule, and Air_carrier as those columns are mostly empty and would not contribute to my analysis.

In [8]:

```
df = df.drop(['Latitude', 'Longitude', 'Schedule', 'Air_carrier'], axis=1)

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 87951 entries, 0 to 88888
Data columns (total 27 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Event_Id                87951 non-null  object 
 1   Investigation_Type      87951 non-null  object 
 2   Accident_Number         87951 non-null  object 
 3   Event_Date              87951 non-null  object 
 4   Location                87899 non-null  object 
 5   Country                 87729 non-null  object 
 6   Airport_Code            49484 non-null  object 
 7   Airport_Name            52031 non-null  object 
 8   Injury_Severity         86961 non-null  object 
 9   Aircraft_damage         84848 non-null  object 
 10  Aircraft_Category       32181 non-null  object 
 11  Registration_Number     86601 non-null  object 
 12  Make                    87888 non-null  object 
 13  Model                   87859 non-null  object 
 14  Amateur_Built           87851 non-null  object 
 15  Number_of_Engines       81924 non-null  float64
 16  Engine_Type             80908 non-null  object 
 17  FAR_Description         31915 non-null  object 
 18  Purpose_of_flight       81829 non-null  object 
 19  Total_Fatal_Injuries    76684 non-null  float64
 20  Total_Serious_Injuries  75629 non-null  float64
 21  Total_Minor_Injuries    76191 non-null  float64
 22  Total_Uninjured         82088 non-null  float64
 23  Weather_Condition       83478 non-null  object 
 24  Broad_phase_of_flight   60837 non-null  object 
 25  Report_Status           81587 non-null  object 
 26  Publication_Date        74352 non-null  object 
dtypes: float64(5), object(22)
memory usage: 18.8+ MB
```

Aircraft_Category

The column for Aircraft_Category is also mostly empty, but that data could be useful. The business is after all seeking data related to types of aircraft and airplanes specifically, so removing the column entirely would not work well. Simply removing all rows that do not have a category entry would greatly reduce the number of total rows available for analysis, and most of those removed would likely be airplanes.

I would like to explore the idea of filling in as many of the missing values as I can. This could be done to some extent by making use of the Make column.

In [9]:

```
# Aircraft_Category values for Cessna in the Make column
df[df['Make'] == 'Cessna']['Aircraft_Category'].unique()
```

Out[9]:

```
array([nan, 'Airplane'], dtype=object)
```

So here I see that Cessna categories are either empty or 'airplane'. Therefore, it's reasonable to fill in the empty category values for Cessnas as 'airplane'

In [10]:

```
# Show how many nan Aircraft_Category values there are for Cessna
df[df['Make'] == 'Cessna']['Aircraft_Category'].isna().sum()
```

Out[10]:

```
18344
```

In [11]:

```
# Show how many 'Airplane' Aircraft_Category values there are for Cessna
df[(df['Make'] == 'Cessna') & (df['Aircraft_Category'] == 'Airplane')]['Aircraft_Categor
y'].count()
```

Out[11]:

```
3581
```

So we can add another 18344 airplane entries to our data by filling in the missing value for Cessna in the Aircraft_Category column

In [12]:

```
# Aircraft_Category values for Sikorsky in the Make column
# Aircraft_Category values for Sikorsky in the Make column
df[df['Make'] == 'Sikorsky']['Aircraft_Category'].unique()
```

Out[12]:

```
array(['Helicopter', nan], dtype=object)
```

In [13]:

```
# Sikorsky nan values there are for
df[df['Make'] == 'Sikorsky']['Aircraft_Category'].isna().sum()
```

Out[13]:

```
128
```

And here we would be able to add 128 additional helicopters for Sikorsky.

In [14]:

```
# Fill in Aircraft_Category as 'Airplane' for Cessna
df.loc[df['Make'] == 'Cessna', 'Aircraft_Category'] = 'Airplane'
```

In [15]:

```
# Show how many 'Airplane' Aircraft_Category values there are for Cessna now
df[(df['Make'] == 'Cessna') & (df['Aircraft_Category'] == 'Airplane')]['Aircraft_Categor
y'].count()
```

Out[15]:

```
21925
```

So now, instead of only 3500 Cessna airplanes, we have almost 22000 entries, greatly increasing the verified airplane subset.

So here I will continue finding Makes that are airplanes only, and filling in the missing values.

In [16]:

```
# list of the unique values in the Make column
df['Make'].value_counts()
```

Out[16]:

```
Make
Cessna           21925
Piper            11903
CESSNA            4914
Beech             4290
PIPER             2841
                 ...  
Geertz               1
Conrad Menzel        1
Blucher              1
Gideon               1
ROYSE RALPH L        1
Name: count, Length: 8202, dtype: int64
```

I realize here that I need to do some further cleaning of the Make column so Cessna and CESSNA (and other similar issues) are not separate values.

Clean up the Make column

After going through the list of makes in a plain text document, I put together a list of make values to replace the alternative spellings, all caps, etc.

In [17]:

```
#Fill Nan values in Make first
```

```
df['Make'] = df['Make'].fillna('Unknown')

# Clean the Make column for misspellings, all caps issues, etc
make_column_name_replace = {'Ab Sportine Aviacija': 'Sportine Aviacija', 'AB SPORTINE AVI
ACIJA': 'Sportine Aviacija', 'SPORTINE AVIACIJA': 'Sportine Aviacija', 'Abrams/nuding': '
Abrams', 'ACRO': 'Acro Sport', 'Adams': 'Adams Balloon', 'ADAMS': 'Adams Balloon', 'ADAM
S BALLOONS LLC': 'Adams Balloon', 'AERO COMMANDER': 'Aero Commander', 'AERO VODOCHODY':
'Aero Vodochody', 'AEROVODOCHODY': 'Aero Vodochody', 'Aero Vodochody Aero Works': 'Aero
Vodochody', 'AEROFAB INC': 'Aerofab Inc.', 'AEROMOT': 'Aeromot', 'AERONCA': 'Aeronca', 'A
eronca Aircraft Corporation': 'Aeronca', 'AEROPRO CZ': 'Aeropro CZ', 'AEROS': 'Aeros', 'A
eros LTD': 'Aeros', 'AEROS LTD': 'Aeros', 'AEROSPATIALE': 'Aerospatiale', 'AEROSTAR': 'A
erostar', 'Aerostar International': 'Aerostar', 'AEROSTAR INTERNATIONAL': 'Aerostar', 'Ae
rostar International Inc': 'Aerostar', 'AEROSTAR INTERNATIONAL INC': 'Aerostar', 'Aerosta
r International Inc.': 'Aerostar', 'Aerostar International, Inc.': 'Aerostar', 'AEROTEK':
'Aerotek', 'Aerotek-pitts': 'Aerotek', 'AEROTEK INC': 'Aerotek', 'AGUSTA': 'Agusta', 'AGU
STA AEROSPACE CORP': 'Agusta', 'AGUSTA SPA': 'Agusta', 'Agusta Spa': 'Agusta', 'Agusta-be
ll': 'Agusta', 'Agusta/Westland': 'Agusta', 'AGUSTAWESTLAND': 'Agusta', 'AgustaWestland':
'Agusta', 'AgustadWestland': 'Agusta', 'AGUSTAWESTLAND PHILADELPHIA': 'Agusta', 'AGUSTAWE
STLAND SPA': 'Agusta', 'AIR CREATION': 'Air Creation', 'Air Creations': 'Air Creation', '
AIR TRACTOR': 'Air Tractor', 'AIR TRACTOR INC': 'Air Tractor', 'Air Tractor Inc': 'Air Tr
actor', 'Air Tractor Inc.': 'Air Tractor', 'AIR TRACTOR INC.': 'Air Tractor', 'Air Tracto
r, Inc.': 'Air Tractor', 'Air Borne Windsports Pty. Ltd.': 'Airborne', 'AIRBORNE (AUSTRAL
IA)': 'Airborne', 'AIRBORNE AUSTRALIA': 'Airborne', 'AIRBORNE EXTREME LLC': 'Airborne', '
AirBorne WindSport': 'Airborne', 'Airborne Windsports': 'Airborne', 'AIRBORNE WINDSPORTS'
: 'Airborne', 'Airborne Windsports Ltd': 'Airborne', 'AIRBORNE WINDSPORTS PTY LTD': 'Airb
orne', 'Airborne Windsports PTY LTD': 'Airborne', 'AIRBORNE': 'Airborne', 'AIRBUS': 'Airb
us', 'AIRBUS HELICOPTERS': 'Airbus Helicopters', 'AIRBUS HELICOPTER': 'Airbus Helicopter
s', 'Airbus Helicopters (Eurocopte': 'Airbus Helicopters', 'Airbus Helicopters Deutschla
nd': 'Airbus Helicopters', 'AIRBUS HELICOPTERS INC': 'Airbus Helicopters', 'Airbus Indus
trie': 'Airbus', 'AIRBUS INDUSTRIE': 'Airbus', 'AIRCRAFT MFG & DEVELOPMENT CO': 'Aircraft
Mfg & Dev. Co.', 'Aircraft Mfg & Dev. Co. (amd)': 'Aircraft Mfg & Dev. Co.', 'Aircraft Mf
g & Dev. Co. (AMD)':'Aircraft Mfg & Dev. Co.', 'Aircraft Mfg & Development Co.': 'Aircraf
t Mfg & Dev. Co.', 'AIRCRAFT MFG & DVLPMT CO': 'Aircraft Mfg & Dev. Co.', 'ALON': 'Alon'
, 'AMERICAN': 'American Aviation', 'American': 'American Aviation', 'AMERICAN AVIATION':
'American Aviation', 'American Aviation Corp. (aac)': 'American Aviation', 'AMERICAN CHAM
PION': 'American Champion', 'American Champion (acac)': 'American Champion', 'American Ch
ampion (ACAC)': 'American Champion', 'American Champion Aircraft': 'American Champion', '
AMERICAN CHAMPION AIRCRAFT': 'American Champion', 'American Champion Aircraft Cor': 'Amer
ican Champion', 'AMERICAN EUROCOPTER CORP': 'American Eurocopter', 'AMERICAN GENERAL ACFT
CORP': 'American General Aircraft', 'American Legand Aircraft': 'American Legend', 'AMERI
CAN LEGEND': 'American Legend', 'AMERICAN LEGEND AIRCRAFT CO': 'American Legend', 'Americ
an Legend Aircraft Co.': 'American Legend', 'Anderson': 'Anderson Aircraft Corp.', 'Atr':
'ATR', 'AVIAT': 'Aviat', 'AVIAT AIRCRAFT': 'Aviat', 'Aviat Aircraft Inc': 'Aviat', 'AVIA
T AIRCRAFT INC': 'Aviat', 'Aviat Aircraft Inc.': 'Aviat', 'Aviat Aircraft, Inc.': 'Aviat
', 'Aviat Inc': 'Aviat', 'AVIAT INC': 'Aviat', 'Avid': 'Avid Aircraft', 'AVID': 'Avid Ai
rcraft', 'AYERS': 'Ayres', 'Ayers': 'Ayres', 'AYRES': 'Ayres', 'AYRES CORP': 'Ayres', 'A
yres Corporation': 'Ayres', 'AYRES CORPORATION': 'Ayres', 'Bede': 'Bede Aircraft', 'BEDE
': 'Bede Aircraft', 'BEECH': 'Beech', 'BEECH AIRCRAFT': 'Beech', 'BEECH AIRCRAFT CO.': '
Beech', 'Beech Aircraft Corp': 'Beech', 'Beech Aircraft Corporation': 'Beech', 'BEECH AI
RCRAFT CORPORATION': 'Beech', 'Beechcraft': 'Beech', 'BEECHCRAFT': 'Beech', 'Beechcraft
Corporation': 'Beech', 'BOEING': 'Boeing', 'Boeing - Canada (de Havilland)': 'Boeing', '
Boeing (Stearman)': 'Boeing', 'BOEING 777-306ER': 'Boeing', 'Boeing Commercial Airplane
Gro': 'Boeing', 'BOEING COMPANY': 'Boeing', 'Boeing Company': 'Boeing', 'BOEING OF CANADA
/DEHAV DIV': 'Boeing', 'Boeing Stearman': 'Boeing', 'BOEING-STEARMAN': 'Boeing', 'Boeing-
brown': 'Boeing', 'BOMBARDIER': 'Bombardier', 'Bombardier Aerospace, Inc.': 'Bombardier'
, 'Bombardier Canadair': 'Bombardier', 'BOMBARDIER INC': 'Bombardier', 'BOMBARDIER LEARJ
ET CORP.': 'Bombardier', 'Bombardier, Inc.': 'Bombardier', 'BRITISH AEROSPACE': 'British
Aerospace', 'British Aerospace Civil Aircr': 'British Aerospace', 'BRITTEN NORMAN': 'Brit
ten Norman', 'Britten-norman': 'Britten Norman', 'BRITTEN-NORMAN': 'Britten Norman', 'CAN
ADAIR': 'Canadair', 'CANADAIR LTD': 'Canadair', 'CASA': 'Casa', 'Cassult Racer': 'Cassui
t', 'CASSUTT': 'Cassuit', 'Cesna': 'Cessna', 'CESSNA': 'Cessna', 'CESSNA AIRCRAFT': 'Ces
sna', 'CESSNA Aircraft': 'Cessna', 'CESSNA AIRCRAFT CO': 'Cessna', 'Cessna Aircraft Co.':
'Cessna', 'CESSNA AIRCRAFT COMPANY': 'Cessna', 'Cessna Aircraft Company': 'Cessna', 'Cess
na Ector': 'Cessna', 'CESSNA ECTOR': 'Cessna', 'Cessna Reems': 'Cessna', 'CESSNA REIMS':
'Cessna', 'Cessna Robertson': 'Cessna', 'Cessna Skyhawk II': 'Cessna', 'Cessna Soloy': '
Cessna', 'Cessna Wren': 'Cessna', 'CESSNA/AIR REPAIR INC': 'Cessna', 'CESSNA/WEAVER': 'C
essna', 'CHALLENGER': 'Challenger', 'Challenger Ii': 'Challenger', 'CHAMBERLAIN GERALD':
'Chamberlain', 'CHAMBERLIN VICTOR WAYNE': 'Chamberlain', 'CHAMPION': 'Champion', 'CHANCE
VOUGHT': 'Chance Vought', 'CHRISTEN INDUSTRIES INC': 'Christen Industries', 'Christen Ind
ustries, Inc.': 'Christen Industries', 'CIRRUS': 'Cirrus', 'Cirrus Design': 'Cirrus', 'CI
RRUS DESIGN': 'Cirrus', 'Cirrus Design Corp': 'Cirrus', 'CIRRUS DESIGN CORP': 'Cirrus',
'Cirrus Design Corp.': 'Cirrus', 'CIRRUS DESIGN CORP.': 'Cirrus', 'Cirrus Design Corpora
tion': 'Cirrus', 'CIRRUS DESIGN CORPORATION': 'Cirrus', 'CLASSIC AIRCRAFT CORP': 'Classic
Aircraft Corp', 'Classic Aircraft Corp.': 'Classic Aircraft Corp', 'COLUMBIA': 'Columbia'
, 'Columbia Aircraft': 'Columbia', 'Columbia Aircraft Mfg': 'Columbia', 'COLUMBIA AIRCRAF
T MFG': 'Columbia', 'Columbia Aircraft Mfg.': 'Columbia', 'COMMANDER': 'Commander', 'Com
mander Aircraft': 'Commander', 'COMMANDER AIRCRAFT CO': 'Commander', 'Commander Aircraft
Company': 'Commander', 'CONSOLIDATED AERONAUTICS': 'Consolidated Aero', 'Consolidated Aer
onautics Inc.': 'Consolidated Aero', 'CONSOLIDATED AERONAUTICS INC.': 'Consolidated Aero'
, 'Consolidated Aeronautics, Inc': 'Consolidated Aero', 'Consolidated Aeronautics, Inc.':
'Consolidated Aero', 'CONSOLIDATED VULTEE': 'Consolidated Aero', 'Consolidated-vultee': '
Consolidated Aero', 'CONVAIR': 'Convair', 'Convair Div. Of Gen. Dynamics': 'Convair', 'CO
STRUZIONI AERONAUTICHE TECNA': 'Costruzioni', 'Costruzioni Aeronautiche': 'Costruzioni',
'Costruzioni AeronauticheTecnam': 'Costruzioni', 'CUB CRAFTER': 'Cub Crafters', 'CUB CRAF
TERS': 'Cub Crafters', 'CUB CRAFTERS INC': 'Cub Crafters', 'Cub Crafters Inc': 'Cub Craft
ers', 'Cub Crafters Inc.': 'Cub Crafters', 'Cub Crafters, Inc.': 'Cub Crafters', 'Cubcraf
ter': 'Cub Crafters', 'Cubcrafters': 'Cub Crafters', 'CUBCRAFTERS': 'Cub Crafters', 'CubC
rafters': 'Cub Crafters', 'CUBCRAFTERS INC': 'Cub Crafters', 'CubCrafters Inc': 'Cub Craf
ters', 'Cubcrafters, Inc': 'Cub Crafters', 'CubCrafters, Inc': 'Cub Crafters', 'CULVER':
'Culver', 'CULVER GLENN': 'Culver', 'Culver-revolution': 'Culver', 'Culver, Aurther L.':
'Culver', 'CURTIS JOHN P': 'Curtiss Wright', 'Curtis-travel Air': 'Curtiss Wright', 'Curt
is-wright': 'Curtiss Wright', 'CURTISS': 'Curtiss Wright', 'CURTISS WRIGHT': 'Curtiss Wri
ght', 'Curtiss-wright': 'Curtiss Wright', 'Curtiss-Wright': 'Curtiss Wright', 'CZECH': 'C
zech Aircraft Works', 'CZECH AIRCRAFT WORKS': 'Czech Aircraft Works', 'CZECH AIRCRAFT WOR
KS SPOL SRO': 'Czech Aircraft Works', 'Czech Aircraft Works SPOL SRO': 'Czech Aircraft Wo
rks', 'Czech Aircraft Works Spol Sro': 'Czech Aircraft Works', 'Czech Sport Aircraft': '
Czech Sport', 'CZECH SPORT AIRCRAFT A S': 'Czech Sport', 'Czech Sport Aircraft a.s.': 'C
zech Sport', 'Czech Sport Aircraft AS': 'Czech Sport', 'CZECH SPORT AIRCRAFT AS': 'Czech
Sport', 'CZECH SPORTPLANES SRO': 'Czech Sport', 'DASSAULT': 'Dassault', 'Dassault Aviati
on': 'Dassault', 'DASSAULT AVIATION': 'Dassault', 'Dassault Falcon': 'Dassault', 'Dassaul
t-breguet': 'Dassault', 'DASSAULT-BREGUET': 'Dassault', 'Dassault-Breguet': 'Dassault', '
Dassault/sud': 'Dassault', 'DASSAULT/SUD': 'Dassault', 'DE HAVILLAND': 'de Havilland', 'D
e Havilland': 'de Havilland', 'Dehavilland': 'de Havilland', 'DEHAVILLAND': 'de Havilland
', 'DeHavilland': 'de Havilland', 'DEHAVILLAND CANADA': 'de Havilland', 'DIAMOND': 'Diamo
nd', 'Diamond Aicraft Industries Inc': 'Diamond', 'Diamond Aircraft': 'Diamond', 'DIAMOND
AIRCRAFT': 'Diamond', 'DIAMOND AIRCRAFT IND GMBH': 'Diamond', 'Diamond Aircraft Industrie
s': 'Diamond', 'DIAMOND AIRCRAFT INDUSTRIES': 'Diamond', 'DIAMOND AIRCRAFT INDUSTRIES IN'
: 'Diamond', 'Diamond Aircraft Industry Inc': 'Diamond', 'DORNIER': 'Dornier', 'DORNIER G
MBH': 'Dornier', 'DOUGLAS': 'Douglas', 'DOWNER': 'Downer', 'Downer Aircraft Industries':
'Downer', 'EAGLE': 'Eagle', 'Eagle (ultralight)': 'Eagle', 'EAGLE AIRCRAFT CO': 'Eagle',
'Eagle Aircraft Co.': 'Eagle', 'ECLIPSE AVIATION': 'Eclipse Aviation', 'ECLIPSE AVIATION
CORP': 'Eclipse Aviation', 'Eclipse Aviation Corporation': 'Eclipse Aviation', 'EDGE': 'E
dge', 'EIPPER': 'Eipper', 'Eipper Formance': 'Eipper', 'EIPPER FORMANCE INC': 'Eipper', '
Eipper Mx Ii Quicksilver': 'Eipper', 'Eipper Quicksilver': 'Eipper', 'Eipper Quicksiver E
': 'Eipper', 'Eipper-formance': 'Eipper', 'EMBRAER': 'Embraer', 'Embraer Aircraft': 'Embr
aer', 'EMBRAER EXECUTIVE AIRCRAFT INC': 'Embraer', 'EMBRAER S A': 'Embraer', 'EMBRAER-EMP
RESA BRASILEIRA DE': 'Embraer', 'ENGINEERING & RESEARCH': 'Engineering & Research', 'Eng
ineering and Research': 'Engineering & Research', 'ENSTROM': 'Enstrom', 'ENSTROM HELICOPT
ER CORP': 'Enstrom', 'ERCOUPE': 'Ercoupe', 'Ercoupe (eng & Research Corp.)': 'Ercoupe', '
EUROCOPTER': 'Eurocopter', 'Eurocopter Deutsch': 'Eurocopter', 'Eurocopter Deutschland':
'Eurocopter', 'Eurocopter Deutschland Gmbh': 'Eurocopter', 'EUROCOPTER DEUTSCHLAND GMBH'
: 'Eurocopter', 'Eurocopter France': 'Eurocopter', 'EUROCOPTER FRANCE': 'Eurocopter', 'E
UROPA': 'Europa', 'Europa Aviation Inc': 'Europa', 'EVEKTOR AEROTECHNIK': 'Evektor Aerote
chnik', 'Evektor Aerotechnik': 'Evektor Aerotechnik', 'Evektor Aerotechnik AS': 'Evektor
Aerotechnik', 'Evektor-aerotechnik': 'Evektor Aerotechnik', 'Evektor-aerotechnik A.s.': '
Evektor Aerotechnik', 'Evektor-aerotechnik As': 'Evektor Aerotechnik', 'EVEKTOR-AEROTECHN
IK AS': 'Evektor Aerotechnik', 'Evektor-Aerotechnik AS': 'Evektor Aerotechnik', 'EXTRA':
'Extra', 'Extra Flugzeugbau': 'Extra', 'EXTRA FLUGZEUGBAU': 'Extra', 'Extra Flugzeugbau
Gmbh': 'Extra', 'EXTRA FLUGZEUGBAU GMBH': 'Extra', 'EXTRA Flugzeugproduktions-GMBH': 'Ex
tra', 'Extra Flugzeugproduktions-und': 'Extra', 'EXTRA FLUGZEUGPRODUKTIONS-UND': 'Extra'
, 'Extra Flugzeugrau Gmbh': 'Extra', 'FAIRCHILD': 'Fairchild', 'Fairchild Industries': '
Fairchild', 'Fairchild Swearingen': 'Fairchild', 'Fairchild/swearingen': 'Fairchild', 'F
AIRCHILD(HOWARD)': 'Fairchild', 'Fairchild Heli-porter': 'Fairchild', 'FAIRCHILD HELI-PO
RTER': 'Fairchild', 'Fairchild-heliporter': 'Fairchild', 'FAIRCHILD HILLER': 'Fairchild
Hiller', 'FANTASY AIR': 'Fantasy Air', 'FANTASY AIR SRO': 'Fantasy Air', 'Fantasy Sky Pro
motions': 'Fantasy Air', 'FIELDS': 'Fields', 'FISHER': 'Fisher', 'Fischer': 'Fisher', 'Fi
sher Aero': 'Fisher', 'Fisher Flying Products': 'Fisher', 'FISHER FLYING PRODUCTS': 'Fis
her', 'FISHER HAROLD R': 'Fisher', 'FISHER MICHAEL E': 'Fisher', 'FISHER MICHAEL H': 'Fis
her', 'FLEET': 'Fleet', 'FLIGHT DESIGN': 'Flight Design', 'FLIGHT DESIGN GENERAL AVN GMB
H': 'Flight Design', 'Flight Design Gmbh': 'Flight Design', 'FLIGHT DESIGN GMBH': 'Fligh
t Design', 'Flight Design GMBH': 'Flight Design', 'Flight Star': 'Flightstar', 'FLIGHTST
AR': 'Flightstar', 'FLIGHTstar': 'Flightstar', 'FLIGHTSTAR INC': 'Flightstar', 'FLIGHTST
AR SPORTPLANES': 'Flightstar', 'Flightstar Sportplanes': 'Flightstar', 'FOKKER': 'Fokker'
```

```
, 'FOUND ACFT CANADA INC': 'Found Aircraft Canada', 'Found Acft': 'Found Aircraft Canada'
, 'Found Aircraft Canada Inc': 'Found Aircraft Canada', 'GARLICK': 'Garlick', 'Garlick He
licipters Inc.': 'Garlick', 'GARLICK HELICOPTERS INC': 'Garlick', 'Garlick Helicopters In
c.': 'Garlick', 'Gates Lear Jet': 'Gates Learjet', 'GATES LEAR JET': 'Gates Learjet', 'GA
TES LEAR JET CORP.': 'Gates Learjet', 'GATES LEARJET': 'Gates Learjet', 'GATES LEARJET CO
RP': 'Gates Learjet', 'GATES LEARJET CORP.': 'Gates Learjet', 'Gates Learjet Corporation'
:'Gates Learjet', 'GENERAL ATOMICS': 'General Atomics', 'GLASAIR': 'Glasair', 'Glasair Av
iation LLC': 'Glasair', 'GLASAIR AVIATION USA LLC': 'Glasair', 'Glasair Iii': 'Glasair',
'Glassair': 'Glasair', 'Glassair Iii': 'Glasair', 'GLOBE': 'Globe', 'Globe Swift': 'Glob
e', 'GREAT LAKES': 'Great Lakes', 'Great Lakes Adams': 'Great Lakes', 'Great Lakes Aircr
aft Company': 'Great Lakes', 'GRUMMAN': 'Grumman', 'Grumman Acft Eng': 'Grumman', 'GRUMMA
N ACFT ENG COR': 'Grumman', 'GRUMMAN ACFT ENG COR-SCHWEIZER': 'Grumman', 'GRUMMAN AIRCRAF
T': 'Grumman', 'GRUMMAN AIRCRAFT COR-SCHWEIZER': 'Grumman', 'GRUMMAN AIRCRAFT ENG CORP':
'Grumman', 'Grumman American': 'Grumman', 'GRUMMAN AMERICAN': 'Grumman', 'Grumman America
n Aviation': 'Grumman', 'GRUMMAN AMERICAN AVIATION CORP': 'Grumman', 'GRUMMAN AMERICAN AV
N. CORP': 'Grumman', 'Grumman American Avn. Corp.': 'Grumman', 'GRUMMAN AMERICAN AVN. COR
P.': 'Grumman', 'GRUMMAN American Corporation': 'Grumman', 'Grumman Schweizer': 'Grumman'
, 'GRUMMAN SCHWEIZER': 'Grumman', 'Grumman-schweizer': 'Grumman', 'Grumman-Schweizer': 'G
rumman', 'GULFSTREAM': 'Gulfstream', 'Gulfstream Aerospace': 'Gulfstream', 'GULFSTREAM A
EROSPACE': 'Gulfstream', 'Gulfstream Aerospace Corp': 'Gulfstream', 'Gulfstream Aerospac
e Corp.': 'Gulfstream', 'Gulfstream Aerospace LP': 'Gulfstream', 'GULFSTREAM AM CORP COM
M DIV': 'Gulfstream', 'Gulfstream American': 'Gulfstream', 'Gulfstream American Corp': '
Gulfstream', 'GULFSTREAM AMERICAN CORP': 'Gulfstream', 'Gulfstream American Corp.': 'Gul
fstream', 'GULFSTREAM AMERICAN CORP.': 'Gulfstream', 'GULFSTREAM SCHWEIZER A/C CORP': 'Gu
lfstream', 'Gulfstream-schweizer': 'Gulfstream', 'GULFSTREAM-SCHWEIZER': 'Gulfstream', '
HALL': 'Hall', 'Hall Christen Eagle': 'Hall', 'HALL DON H': 'Hall', 'HALL JEFFREY': 'Hal
l', 'HALL STEVEN C': 'Hall', 'HALL TERRENCE / HALL CATHIE': 'Hall', 'HALL THOMAS K': 'Ha
ll', 'Hall-cavalier': 'Hall', 'HALL, WENDALL W': 'Hall', 'Hamilton Stoddard': 'Hamilton',
'HAWKER': 'Beech', 'HAWKER AIRCRAFT LTD': 'Beech', 'Hawker Aircraft Ltd': 'Beech', 'Hawk
er Aircraft Ltd.': 'Beech', 'Hawker Beech': 'Beech', 'HAWKER BEECH': 'Beech', 'Hawker Be
echcraft': 'Beech', 'HAWKER BEECHCRAFT': 'Beech', 'HAWKER BEECHCRAFT CORP': 'Beech', 'Ha
wker Beechcraft Corp.': 'Beech', 'Hawker Beechcraft Corporation': 'Beech', 'HAWKER BEECH
CRAFT CORPORATION': 'Beech', 'Hawker Siddeley': 'Hawker', 'HAWKER SIDDELEY': 'Hawker', '
Hawker Siddely': 'Hawker', 'Hawker-beechcraft': 'Beech', 'Hawker-Beechcraft': 'Beech', '
Hawker-Beechcraft Corporation': 'Beech', 'Helie': 'Helio', 'HELIO': 'Helio', 'Helio Airc
raft Ltd': 'Helio', 'HILLER': 'Hiller', 'Hiller-osborn': 'Hiller', 'HILLER-ROGERSON HELIC
OPTER': 'Hiller', 'Hiller-soloy': 'Hiller', 'HILLER-TRI-PLEX IND.INC.': 'Hiller', 'HONDA
AIRCRAFT': 'Honda Aircraft', 'HONDA AIRCRAFT CO LLC': 'Honda Aircraft', 'Honda Jet': 'Hon
da Aircraft', 'HOWARD AIRCRAFT': 'Howard Aircraft', 'Howard Aircraft Corp.': 'Howard Air
craft', 'HUGHES': 'Hughes', 'HUGHES HELICOPTERS INC': 'Hughes', 'HUGHES/HELICOPTER ASSOCS
INC': 'Hughes', 'HUNTER': 'Hunter', 'HUNTER GEORGE': 'Hunter', 'ICON': 'Icon', 'ICON AIRC
RAFT INC': 'Icon', 'INTERPLANE': 'Interplane', 'Interplane Llc': 'Interplane', 'INTERPLA
NE S R O': 'Interplane', 'Interplane Sro': 'Interplane', 'INTERSTATE': 'Interstate', 'IS
RAEL AEROSPACE INDUSTRIESLTD': 'Israel Aircraft Industries', 'ISRAEL AIRCRAFT INDUSTRIES'
: 'Israel Aircraft Industries', 'JABIRU': 'Jabiru Aircraft', 'JABIRU AIRCRAFT PTY LTD':
'Jabiru Aircraft', 'JABIRU USA SPORT AIRCRAFT': 'Jabiru Aircraft', 'JABIRU USA SPORT AIR
CRAFT LLC': 'Jabiru Aircraft', 'JABIRU USA SPORT AIRCRAFT, LLC': 'Jabiru Aircraft', 'JAC
KSON': 'Jackson', 'JACKSON DENNIS': 'Jackson', 'JACKSON FRED M': 'Jackson', 'JACKSON JERE
MIAH D': 'Jackson', 'JONES': 'Jones', 'JONES KENT C': 'Jones', 'JONES PETER M': 'Jones',
'JONES RALPH D': 'Jones', 'JONES RODNEY V': 'Jones', 'JONES RONALD C': 'Jones', 'JUST':
'Just', 'JUST AIRCRAFT': 'Just', 'Just Aircraft Llc': 'Just', 'JUST AIRCRAFT LLC': 'Just
', 'Just Aircraft LLC': 'Just', 'KAMAN': 'Kaman', 'KAMAN AEROSPACE CORP': 'Kaman', 'KITF
OX': 'Kitfox', 'Kitfox IV': 'Kitfox', 'Kitfox Ten, Inc': 'Kitfox', 'KOLB': 'Kolb', 'KOLB
AIRCRAFT CO': 'Kolb', 'KOLB AIRCRAFT INC': 'Kolb', 'Kolb Company': 'Kolb', 'Kolb Twin Sta
r': 'Kolb', 'LAKE': 'Lake', 'Lake John K': 'Lake', 'LANCAIR': 'Lancair', 'Lancair Compan
y': 'Lancair', 'LANCAIR COMPANY': 'Lancair', 'Lanceair': 'Lancair', 'LARSEN': 'Larsen', '
Larsen Charles Bennett': 'Larsen', 'Larson': 'Larsen', 'LARSON': 'Larsen', 'LARSON KEN W'
: 'Larsen', 'LARSON ROGER H': 'Larsen', 'Larson Smith Miniplane': 'Larsen', 'Larson, C.h.
': 'Larsen', 'LEARJET': 'Learjet', 'LEARJET INC': 'Learjet', 'Learjet Inc': 'Learjet', 'L
ET': 'Let', 'LIBERTY AEROSPACE INCORPORATED': 'Liberty', 'LIBERTY AEROSPACE': 'Liberty',
'Liberty Aerospace': 'Liberty', 'Liberty Aerospace Inc.': 'Liberty', 'Liberty Aerospace I
ncorporate': 'Liberty', 'Liberty Aerospace, Inc.': 'Liberty', 'LINDSTRAND': 'Lindstrand'
, 'Lindstrand Balloons': 'Lindstrand', 'LINDSTRAND BALLOONS': 'Lindstrand', 'LINDSTRAND
BALLOONS USA': 'Lindstrand', 'LINSTRAND': 'Lindstrand', 'LOCKHEED': 'Lockheed', 'Lockhee
d-martin': 'Lockheed', 'LOCKWOOD': 'Lockwood', 'LOCKWOOD AIRCRAFT CORP': 'Lockwood', 'LO
NG': 'Long', 'Long E-z-e': 'Long', 'Long Pitts': 'Long', 'Long-ez': 'Long', 'Longjev': '
Long', 'LUSCOMBE': 'Luscombe', 'Luscombe Silvaire Aircraft Co.': 'Luscombe', 'MARTIN CHAR
LES A': 'Martin', 'Martin Company': 'Martin', 'MARTIN CURTIS': 'Martin', 'MARTIN EDWARD H
': 'Martin', 'Martin-pitts': 'Martin', 'Martin/harris': 'Martin', 'MAULE': 'Maule', 'Mau
le Air Inc.': 'Maule', 'MAULE AIRCRAFT CORP': 'Maule', 'MAXAIR': 'Maxair', 'Maxair Aircr
aft Corp.': 'Maxair', 'MAXAIR DRIFTER': 'Maxair', 'MCCLISH': 'McClish', 'Mcclish': 'McCli
sh', 'Mcdonald': 'Mcdonnell Douglas', 'Mcdonald Douglas': 'Mcdonnell Douglas', 'MCDONNELL
```

```
DOUGLAS': 'Mcdonnell Douglas', 'MCDONNELL DOUGLAS AIRCRAFT CO': 'Mcdonnell Douglas', 'MCD
ONNELL DOUGLAS CORPORATION': 'Mcdonnell Douglas', 'Mcdonnell-douglas': 'Mcdonnell Douglas
', 'MCDONNELL DOUGLAS HELI CO': 'Mcdonnell Douglas Helicopter', 'MCDONNELL DOUGLAS HELICO
PTER': 'Mcdonnell Douglas Helicopter', 'McDonnell Douglas Helicopter C': 'Mcdonnell Dougl
as Helicopter', 'Mcdonnell Douglas Helicopters': 'Mcdonnell Douglas Helicopter', 'McDonne
ll Douglas Helicopters': 'Mcdonnell Douglas Helicopter', 'MD HELICOPTER': 'Md Helicopter'
, 'MD Helicopter Inc': 'Md Helicopter', 'MD Helicopters Inc': 'Md Helicopter', 'MD HELICO
PTER INC': 'Md Helicopter', 'Md Helicopter Inc.': 'Md Helicopter', 'Md Helicopters': 'Md
Helicopter', 'MD HELICOPTERS': 'Md Helicopter', 'MD HELICOPTERS INC': 'Md Helicopter', 'M
d Helicopters Inc.': 'Md Helicopter', 'Md Helicopters, Inc.': 'Md Helicopter', 'MD HELICO
PTERS, INC.': 'Md Helicopter', 'Messerschmitt-boelkow-blohm': 'Messerschmitt', 'MESSERSCH
MITT-BOELKOW-BLOHM': 'Messerschmitt', 'Messerschmitt-bolkow-blohm': 'Messerschmitt', 'MES
SERSCHMITT-BOLKOW-BLOHM': 'Messerschmitt', 'MEYER': 'Meyer', 'MEYERS': 'Meyers', 'Meyers
Aircraft Co.': 'Meyers', 'MEYERS INDUSTRIES INC': 'Meyers', 'MILLER': 'Miller', 'Miller A
ir Sports': 'Miller', 'Mitchell Aircraft Corp': 'Mitchell', 'MITSUBISHI': 'Mitsubishi',
'MONOCOUPE': 'Monocoupe', 'Monocoupe Aircraft': 'Monocoupe', 'MOONEY': 'Mooney', 'Mooney
Aircraft': 'Mooney', 'Mooney Aircraft Corp': 'Mooney', 'MOONEY AIRCRAFT CORP.': 'Mooney',
'Mooney Aircraft Corporation': 'Mooney', 'MOONEY AIRPLANE CO INC': 'Mooney', 'MOONEY AIRP
LANE COMPANY, INC.': 'Mooney', 'MOONEY INTERNATIONAL CORP': 'Mooney', 'MOORE': 'Moore',
'MORAVAN': 'Moravan', 'MORRISEY': 'Morrisey', 'MORRISON': 'Morrison', 'Murphy Aircraft':
'Murphy', 'Murphy Aircraft Mfg, Ltd.': 'Murphy', 'Murphy-charles': 'Murphy', 'Murphey': '
Murphy', 'MYERS': 'Myers', 'NANCHANG': 'Nanchang', 'Nanchang China': 'Nanchang', 'NANCHA
NG CHINA': 'Nanchang', 'NAVAL AIRCRAFT FACTORY': 'Naval Aircraft Factory', 'Navy': 'Nava
l Aircraft Factory', 'NAVION': 'Navion', 'NELSON': 'Nelson', 'Nelson/nelson': 'Nelson', '
New Glasair': 'Glasair', 'NEW KOLB AIRCRAFT CO': 'Kolb', 'NEW KOLB AIRCRAFT CO LLC': 'Kol
b', 'New Piper': 'Piper', 'NEW PIPER': 'Piper', 'NEW PIPER AIRCRAFT INC': 'Piper', 'New
Piper Aircraft, Inc.': 'Piper', 'NEW STANDARD': 'New Standard', 'NORD': 'Nord', 'Nord (s
ncan)': 'Nord', 'Nord Aviation': 'Nord', 'NORTH AMERICAN': 'North American', 'North Amer
ican Aviation Div.': 'North American', 'North American Rockwell': 'North American', 'Nort
h American Rockwell Corp.': 'North American', 'NORTH AMERICAN-MEDORE': 'North American',
'North American/aero Classics': 'North American', 'North American-aero Classics': 'North
American', 'North American-barene': 'North American', 'North American-kenney': 'North Am
erican', 'North American-maslon': 'North American', 'NORTH AMERICAN/AERO CLASSICS': 'Nor
th American', 'NORTH AMERICAN/SCHWAMM': 'North American', 'NORTH AMERICAN/VICTORIA MNT LT
': 'North American', 'NORTH WING': 'North Wing', 'NORTH WING DESIGN': 'North Wing', 'NOR
TH WING UUM INC': 'North Wing', 'North Wing UUM INC': 'North Wing', 'NORTH WING UUM INC.
': 'North Wing', 'NORTHROP': 'Northrop', 'NORTHWING': 'Northwing', 'Northwing Design': '
Northwing', 'NORTHWING DESIGN': 'Northwing', 'PACIFIC AEROSPACE CORP LTD': 'Pacific Aeros
pace', 'PIAGGIO': 'Piaggio', 'Piaggio Aero Industries S.p.a.': 'Piaggio', 'PIAGGIO AERO I
NDUSTRIES SPA': 'Piaggio', 'Piaggio Industrie': 'Piaggio', 'PIETENPOL': 'Pietenpol', 'Pie
tenpol-grega': 'Pietenpol', 'PILATUS': 'Pilatus', 'Pilatus Aircraft': 'Pilatus', 'PILATUS
AIRCRAFT LTD': 'Pilatus', 'Pilatus Britten-norman': 'Pilatus', 'PILATUS BRITTEN-NORMAN':
'Pilatus', 'PIPER': 'Piper', 'PIPER / LAUDEMAN': 'Piper', 'Piper Aerostar': 'Piper', 'Pi
per Aircraft': 'Piper', 'PIPER AIRCRAFT': 'Piper', 'PIPER AIRCRAFT CORPORATION': 'Piper'
, 'Piper Aircraft Corporation': 'Piper', 'PIPER AIRCRAFT INC': 'Piper', 'PIPER AIRCRAFT,
INC.': 'Piper', 'Piper Aircraft, Inc.': 'Piper', 'Piper Cub Crafters': 'Piper', 'Piper P
awnee': 'Piper', 'Piper-aerostar': 'Piper', 'PIPER-HARRIS': 'Piper', 'Piper/cub Crafters
': 'Piper', 'PIPER/CUB CRAFTERS': 'Piper', 'Piper/Cub Crafters': 'Piper', 'Piper/stevens
': 'Piper', 'PIPER/WALLY\'S FLYERS INC': 'Piper', 'PIPISTREL': 'Pipistrel', 'PIPISTREL D
O O': 'Pipistrel', 'PIPISTREL DOO AJDOVSCINA': 'Pipistrel', 'PIPISTREL ITALIA S R L': 'P
ipistrel', 'Pipistrel Italia SRL': 'Pipistrel', 'PITCAIRN': 'Pitcairn', 'PITTS': 'Pitts'
, 'PITTS AEROBATICS': 'Pitts', 'Pitts Special': 'Pitts', 'PITTS SPECIAL': 'Pitts', 'Pitt
s Spl.': 'Pitts', 'PROGRESSIVE AERODYNE INC': 'Progressive Aerodyne', 'Progressive Aerod
yne Inc.': 'Progressive Aerodyne', 'PZL BIELSKO': 'PZL', 'PZL MIELEC': 'PZL', 'PZL OKECIE
': 'PZL', 'Pzl Okecie': 'PZL', 'Pzl Swidnik': 'PZL', 'PZL SWIDNIK': 'PZL', 'PZL Warszawa-
Okecie': 'PZL', 'Pzl Warzawa-cnpsl': 'PZL', 'Pzl Warzawa-okecie': 'PZL', 'Pzl-bielsko':
'PZL', 'PZL-BIELSKO': 'PZL', 'Pzl-mielec': 'PZL', 'Pzl-okecie': 'PZL', 'Pzl-swidnik': 'P
ZL', 'PZL-Swidnik': 'PZL', 'PZL-SWIDNIK': 'PZL', 'QUAD CITY': 'Quad City', 'Quad City Air
craft': 'Quad City', 'Quad City Aircraft Corp': 'Quad City', 'Quad City Ultralight': 'Qu
ad City', 'QUAD CITY ULTRALIGHT ACFT CORP': 'Quad City', 'Quad City Ultralight Aircraft'
: 'Quad City', 'QUAD CITY ULTRALIGHT CORP': 'Quad City', 'Quad City Ultralight, Llc': 'Q
uad City', 'QUAD CITY ULTRALIGHTS': 'Quad City', 'Quadcity': 'Quad City', 'QUEST': 'Ques
t', 'QUEST Aircraft Company': 'Quest', 'QUEST AIRCRAFT COMPANY LLC': 'Quest', 'QUESTAIR
INC': 'Questair', 'Questair, Inc.': 'Questair', 'Questaire': 'Questair', 'QUICKIE': 'Qui
ckie', 'Quickie Aircraft': 'Quickie', 'Quickie-myers': 'Quickie', 'Quick Silver': 'Quicks
ilver', 'QUICKSILVER': 'Quicksilver', 'Quicksilver Aircraft': 'Quicksilver', 'QUICKSILVER
AIRCRAFT': 'Quicksilver', 'QUICKSILVER AIRCRAFT CO': 'Quicksilver', 'Quicksilver Aircraft
Northeast': 'Quicksilver', 'QUICKSILVER EIPPER ACFT INC': 'Quicksilver', 'QUICKSILVER ENT
ERPRISES INC': 'Quicksilver', 'Quicksilver II': 'Quicksilver', 'Quicksilver Manufacturing
': 'Quicksilver', 'QUICKSILVER MANUFACTURING INC': 'Quicksilver', 'QUICKSILVER MFG': 'Qui
cksilver', 'Quiksilver': 'Quicksilver', 'RANS': 'Rans', 'RANS AIRCRAFT': 'Rans', 'Rans C
ompany': 'Rans', 'RANS DESIGNS INC': 'Rans', 'Rans Employee Flying Club': 'Rans', 'RANS
```

```
EMPLOYEE FLYING CLUB': 'Rans', 'RANS INC': 'Rans', 'Rans Inc.': 'Rans', 'RANS S-12': 'Ra
ns', 'Rans S-12 Airaile': 'Rans', 'Rans, Inc.': 'Rans', 'Rans/hine': 'Rans', 'Raven Indu
stries': 'Raven', 'RAVEN INDUSTRIES INC': 'Raven', 'RAYTHEON': 'Raytheon', 'Raytheon Air
craft Company': 'Raytheon', 'RAYTHEON AIRCRAFT COMPANY': 'Raytheon', 'Raytheon Co': 'Rayt
heon', 'RAYTHEON COMPANY': 'Raytheon', 'Raytheon Corporate Jets': 'Raytheon', 'RAYTHEON C
ORPORATE JETS INC': 'Raytheon', 'REIMS': 'Reims', 'REims': 'Reims', 'Reims Aviation': 'R
eims', 'Reims Aviation Cessna': 'Reims', 'REIMS AVIATION S.A.': 'Reims', 'REIMS AVIATION 
SA': 'Reims', 'Reims-Cessna': 'Reims', 'REIMS-CESSNA': 'Reims', 'REMOS ACFT GMBH FLUGZEU
GBAU': 'Remos', 'Remos Aircraft GmbH': 'Remos', 'REMOS AIRCRAFT GMBH': 'Remos', 'Remos A
ircraft GMBH': 'Remos', 'Remos Aircraft GMBH Flugzeugba': 'Remos', 'REMOS AIRCRAFT GMBH 
FLUGZEUGBA': 'Remos', 'REPUBLIC': 'Republic', 'ROBIN': 'Robin', 'ROBINSON': 'Robinson', 
'Robinson Helicopter': 'Robinson', 'ROBINSON HELICOPTER': 'Robinson', 'ROBINSON HELICOPTE
R CO': 'Robinson', 'ROBINSON HELICOPTER CO INC': 'Robinson', 'Robinson Helicopter Co.': '
Robinson', 'Robinson Helicopter Company': 'Robinson', 'ROBINSON HELICOPTER COMPANY': 'Rob
inson', 'Robinson Helicopters': 'Robinson', 'ROCKWELL': 'Rockwell', 'Rockwell Comdr': 'Ro
ckwell', 'Rockwell Commander': 'Rockwell', 'ROCKWELL COMMANDER': 'Rockwell', 'Rockwell In
t\'t': 'Rockwell', 'Rockwell International': 'Rockwell', 'ROCKWELL INTERNATIONAL': 'Rock
well', 'Rockwell Intl': 'Rockwell', 'ROGERS': 'Rogers', 'ROLLADEN-SCHNEIDER': 'Rolladen S
chneider', 'ROLLADEN SCHNEIDER OHG': 'Rolladen Schneider', 'Rolladen-schneider': 'Rollade
n Schneider', 'Rolladen-schneider Gmbh': 'Rolladen Schneider', 'ROLLADEN-SCHNEIDER GMBH': 
'Rolladen Schneider', 'ROLLADEN-SCHNEIDER OHG': 'Rolladen Schneider', 'Roof': 'Root', 'Ro
ot, Arthur T.': 'Root', 'ROSE': 'Rose', 'ROSE HERBERT D': 'Rose', 'Rose Rhinehart': 'Ros
e', 'Rose Wesley': 'Rose', 'Rose-Rhinehart': 'Rose', 'ROSS ALFRED K/ONEILL TERRENCE': 'R
oss', 'ROSS H/HERRIOTT M': 'Ross', 'ROSS JONATHAN': 'Ross', 'Ross/stonecipher': 'Ross', 
'ROTORWAY': 'Rotorway', 'Rotorway Aircraft, Inc.': 'Rotorway', 'Rotorway Executive': 'Rot
orway', 'Rotoway International': 'Rotorway', 'RUTAN': 'Rutan', 'Rutan Aircraft Factory': 
'Rutan', 'RYAN': 'Ryan', 'Ryan Aeronautical': 'Ryan', 'RYAN AERONAUTICAL': 'Ryan', 'Ryan 
Aeronautics': 'Ryan', 'RYAN JOHN STEFFEY': 'Ryan', 'RYAN W Gross': 'Ryan', 'Ryan-navion'
: 'Ryan', 'SAAB': 'Saab', 'Saab-fairchild': 'Saab', 'Saab-scania': 'Saab', 'SAAB-SCANIA'
: 'Saab', 'SAAB-SCANIA AB': 'Saab', 'Saab-scania Ab (saab)': 'Saab', 'Saab-Scania AB (Sa
ab)': 'Saab', 'SCHWEIZER': 'Schweizer', 'Schweizer 300CBi': 'Schweizer', 'Schweizer Airc
raft Corp': 'Schweizer', 'SCHWEIZER AIRCRAFT CORP': 'Schweizer', 'Schweizer Aircraft Corp
.': 'Schweizer', 'Schweizer Sgs': 'Schweizer', 'Schweizer, N36289': 'Schweizer', 'SCHWEI
ZER(HUGHES)': 'Schweizer', 'SCHWEIZER(HUGHES)AIRCRAFT CORP': 'Schweizer', 'Scottish': 'S
cottish Aviation', 'SCOTTISH AVIATION': 'Scottish Aviation', 'SHORT': 'Short', 'SHORT BR
OS': 'Short', 'Short Bros.': 'Short', 'SHORT BROS. & HARLAND': 'Short', 'Short Brothers'
: 'Short', 'SHORT BROTHERS & HARLAND LTD.': 'Short', 'SHORT BROTHERS PLC': 'Short', 'SIA
I MARCHETTI': 'Siai Marchetti', 'Siai-marchetti': 'Siai Marchetti', 'SIAI-MARCHETTI': 'S
iai Marchetti', 'Siai-Marchetti': 'Siai Marchetti', 'SIKORSKY': 'Sikorsky', 'SIKORSKY AIR
CRAFT CORP': 'Sikorsky', 'Sikorsky/orlando': 'Sikorsky', 'SILVAIRE': 'Silvaire', 'SKYKITS
': 'Skykits', 'SKYKITS CORP': 'Skykits', 'Skykits Corporation': 'Skykits', 'SKYKITS USA C
ORP': 'Skykits', 'SMITH': 'Smith', 'Smith & R. Mathews': 'Smith', 'Smith Aerostar': 'Smi
th', 'SMITH ALBERT F': 'Smith', 'SMITH ALLEN': 'Smith', 'Smith Arthur Fox': 'Smith', 'SM
ITH BRET B': 'Smith', 'SMITH Capella': 'Smith', 'Smith Carter A': 'Smith', 'Smith Dougla
s J.': 'Smith', 'SMITH EDWARD I': 'Smith', 'Smith Mini': 'Smith', 'Smith Miniplane': 'Sm
ith', 'SMITH MINIPLANE': 'Smith', 'SMITH VILAS': 'Smith', 'Smith Wylie Jay': 'Smith', 'S
mith, Ted Aerostar': 'Smith', 'Smith/davis': 'Smith', 'SNOW': 'Snow', 'SOCATA': 'Socata'
, 'Socata-Groupe Aerospatiale': 'Socata', 'SONEX': 'Sonex', 'Sonex / John D. McCarter': 
'Sonex', 'SONEX AIRCRAFT': 'Sonex', 'SONEX LIMITED': 'Sonex', 'SORENSEN': 'Sorensen', 'S
ORENSEN DANNY': 'Sorensen', 'SORENSEN DANNY S': 'Sorensen', 'SORENSON': 'Sorensen', 'Sor
rel': 'Sorrell', 'Sorrell Aircraft': 'Sorrell', 'STANLEY': 'Stanley', 'STANLEY ARTHUR FRE
EMAN': 'Stanley', 'STANLEY B E': 'Stanley', 'STANLEY ERNIE SIGURD': 'Stanley', 'Stanley, 
Davey L': 'Stanley', 'STANTON': 'Stanton', 'Star Duster': 'Starduster', 'Star Duster Too'
: 'Starduster', 'Starduster Ii': 'Starduster', 'STARDUSTER II': 'Starduster', 'Starduste
r Too': 'Starduster', 'STAUDACHER AIRCRAFT INC': 'Staudacher', 'Staudacher Aircraft, Inc
.': 'Staudacher', 'STAUDACHER HYDROPLANES': 'Staudacher', 'STAUDACHER JON': 'Staudacher'
, 'Staudaucher': 'Staudacher', 'STEARMAN': 'Stearman', 'STEARMAN AIRCRAFT': 'Stearman', 
'STEELE': 'Steele', 'STEELE  JOHN J': 'Steele', 'STEELE RALPH BRUCE': 'Steele', 'STEELE 
SAMUEL D': 'Steele', 'STEEN': 'Steen', 'Steen Aero Lab': 'Steen', 'Steen Skybolt': 'Stee
n', 'STINSON': 'Stinson', 'Stits Aircraft': 'Stits', 'Stits Flut-r-bug': 'Stits', 'STITS 
FLUT-R-BUG': 'Stits', 'Stits Playboy': 'Stits', 'Stits-itrich': 'Stits', 'Stitts': 'Stit
s', 'STODDARD HAMILTON': 'Stoddard Hamilton', 'Stoddard-Hamilton': 'Stoddard Hamilton', '
STOL': 'Stol', 'Stol Aircraft': 'Stol', 'STOL Aircraft Corp': 'Stol', 'STOL LLC': 'Stol'
, 'STOLP STARDUSTER': 'Stolp Starduster', 'Stolp Starduster Corp.': 'Stolp Starduster', 
'Stolp-adams': 'Stolp Starduster', 'Stolp-starduster Too': 'Stolp Starduster', 'SUKHOI': 
'Sukhoi', 'SUTTON': 'Sutton', 'Sutton Tailwind': 'Sutton', 'SUTTON WILLIAM J': 'Sutton', 
'SWANSON': 'Swanson', 'Swanson/bensen': 'Swanson', 'SWEARINGEN': 'Swearingen', 'Swearing
en T R/masters W': 'Swearingen', 'TAYLOR': 'Taylor', 'Taylor Air Command': 'Taylor', 'Tay
lor Lonsdale': 'Taylor', 'Taylor Smith': 'Taylor', 'Taylor Titch': 'Taylor', 'TAYLORCRAF
T': 'Taylorcraft', 'Taylorcraft Aviation': 'Taylorcraft', 'TAYLORCRAFT AVIATION CORP': '
Taylorcraft', 'TAYLORCRAFT CORP': 'Taylorcraft', 'Taylorcraft Corporation': 'Taylorcraft'
, 'TECNAM': 'Tecnam', 'TEMCO': 'Temco', 'Temco Luscombe': 'Temco', 'TERATORN': 'Teratorn
```

```
', 'Teratorn Acft Inc.': 'Teratorn', 'Teratorn Aircraft, Inc.': 'Teratorn', 'Teratron':
'Teratorn', 'TEXAS HELICOPTER CORP': 'Texas Helicopter', 'Texas Helicopter Corp.':'Texas
Helicopter', 'Texas Helicopter Corporation': 'Texas Helicopter', 'TEXTRON AVIATION INC':
'Textron Aviation', 'Textron Aviation Inc': 'Textron Aviation', 'THE BOEING COMPANY': 'Bo
eing', 'THOMPSON': 'Thompson', 'THORP': 'Thorp', 'Thorp Aero, Inc.': 'Thorp', 'Thorpe':
'Thorp', 'THRUSH': 'Thrush', 'THRUSH AIRCRAFT INC': 'Thrush', 'Thrush Aircraft Inc.': 'Th
rush', 'THRUSH AIRCRAFT LLC': 'Thrush', 'Thrush Aircraft, Inc.': 'Thrush', 'TITAN': 'Tit
an', 'TITAN AEROSPACE HOLDINGS INC': 'Titan', 'Titan Aircraft': 'Titan', 'TRAVEL AIR': '
Travel Air', 'TUPOLEV': 'Tupolev', 'Univair Aircraft Corporation': 'Univair', 'UNIVAIR AI
RCRAFT CORPORATION': 'Univair', 'Univar': 'Univair', 'UNIVERSAL': 'Universal', 'Universal
Aircraft Industries': 'Universal', 'Universal Globe': 'Universal', 'Universal Moulded Pdt
s.': 'Universal', 'Universal Stinson': 'Universal', 'UNIVERSAL STINSON': 'Universal', 'VA
NS': 'Vans', 'Vans Aircraft': 'Vans', 'Vans Aircraft Inc': 'Vans', 'VANS AIRCRAFT INC':
'Vans', 'Vans Aircraft, Inc.': 'Vans', 'VARGA AIRCRAFT CORP.': 'Varga', 'VELOCITY INC':
'Velocity', 'VICKERS': 'Vickers', 'WACO': 'Waco', 'Waco Classic Aircraft': 'Waco', 'WACO
CLASSIC AIRCRAFT': 'Waco', 'Waco Classic Aircraft Corp': 'Waco', 'WACO CLASSIC AIRCRAFT C
ORP': 'Waco', 'Waco Classic Aircraft Corp.': 'Waco', 'WEATHERLY': 'Weatherly', 'WEATHERL
Y AVIATION CO INC': 'Weatherly', 'Weatherly Aviation Company Inc': 'Weatherly', 'WEBER':
'Weber', 'WHEELER': 'Wheeler', 'Wheeler Acft. Co.': 'Wheeler', 'WHEELER C / WHEELER K': '
Wheeler', 'Wheeler Technology, Inc.': 'Wheeler', 'WHITTMAN': 'Whittman', 'Whittman Tailwi
nd': 'Whittman', 'WILSON': 'Wilson', 'Wing Aircraft': 'Wing', 'Wing Aircraft Co.': 'Wing'
, 'WOOD': 'Wood', 'WSK PZL MIELEC': 'Wsk Pzl Mielec', 'Wsk Pzl Swidnik': 'Wsk Pzl Mielec'
, 'Wsk Pzl Warzawa-okecie': 'Wsk Pzl Mielec', 'Wsk Pzl-krosno': 'Wsk Pzl Mielec', 'WSK-MI
ELEC': 'Wsk Pzl Mielec', 'WSK-PZL MEILEC': 'Wsk Pzl Mielec', 'Wsk-pzl Mielec': 'Wsk Pzl M
ielec', 'WSK-PZL WARZAWA-OKECIE': 'Wsk Pzl Mielec', 'Wsk-pzl Warzawaokecie': 'Wsk Pzl Mie
lec', 'WSL PZL': 'Wsk Pzl Mielec', 'XTREMEAIR GMBH': 'Xtremeair GMBH', 'YAKOVLEV': 'Yakov
lev', 'YAKOVLEV/CHINNERY': 'Yakovlev', 'YAKOVLEV/DAY': 'Yakovlev', 'ZENAIR': 'Zenair', 'Z
ENAIR LTD': 'Zenair', 'Zenair Zodiac': 'Zenair', 'ZENITH': 'Zenith', 'ZENITH ACFT CO': 'Z
enith', 'ZENITH AIRCRAFT CO': 'Zenith', 'ZIMMERMAN': 'Zimmerman', 'ZIVKO AERONAUTICS INC
': 'Zivko', 'Zivko Aeronautics': 'Zivko', 'Zivko Aeronautics Inc.': 'Zivko', 'ZLIN': 'Zl
in', 'Zlin Aviation': 'Zlin', 'Zlin Aviation S.r.o.': 'Zlin'}

df['Make'] = df['Make'].replace(make_column_name_replace)
```

Combine the MD Helicopters and McDonnell Douglas into McDonnell Douglas Helicopters Combine the BELL variations into Bell

In [18]:

```
df[df['Make'].isin(['Md Helicopter', 'Mcdonnell Douglas Helicopter', 'McDonnell Douglas
Helicopter', 'BELL',
                    'BELL HELICOPTER TEXTRON CANADA', 'BELL HELICOPTER TEXTRON'])][
    'Aircraft_Category'].value_counts(dropna=False)
```

Out[18]:

```
Aircraft_Category
Helicopter    719
NaN            21
Unknown         2
Airplane        1
Name: count, dtype: int64
```

In [19]:

```
# Clean up make column for McDonnell Douglas Helicopters and Bell
make_column_name_replace = {'Md Helicopter': 'McDonnell Douglas Helicopters',
                            'Mcdonnell Douglas Helicopter': 'McDonnell Douglas Helicopte
rs',
                            'McDonnell Douglas Helicopter': 'McDonnell Douglas Helicopte
rs',
                            'BELL': 'Bell',
                            'BELL HELICOPTER TEXTRON CANADA': 'Bell',
                            'BELL HELICOPTER TEXTRON': 'Bell'}

df['Make'] = df['Make'].replace(make_column_name_replace)
```

In [20]:

```
# Show Make value_counts over 10
makes_value_10 = df['Make'].value_counts()[df['Make'].value_counts() > 10]
```

```
makes_value_10
```

Out[20]:

```
Make
Cessna              26903
Piper               14818
Beech                5431
Bell                 2750
Boeing               2726
                    ...  
Bushby                 11
Hispano Aviacion       11
Flightstar             11
BURKHART GROB          11
Univair                11
Name: count, Length: 234, dtype: int64
```

Now we look at the category column to see how we can fill it in using cleaned make column

In [21]:

```
# Show the Aircraft_Category value_counts for makes_value_10 including NaN
df[df['Make'].isin(makes_value_10.index)]['Aircraft_Category'].value_counts(dropna=False)
```

Out[21]:

```
Aircraft_Category
Airplane             42498
NaN                  32784
Helicopter            3164
Glider                 326
Balloon                128
Weight-Shift            60
Gyrocraft               14
Ultralight              10
Unknown                 10
Powered-Lift             4
WSFT                     4
Powered Parachute        1
Name: count, dtype: int64
```

In [22]:

```
# Show Aircraft Category value_counts for Cessna, Piper, Beech, Boeing, and Mooney, including Nan
df[df['Make'].isin(['Cessna', 'Piper', 'Beech', 'Boeing', 'Mooney'])][
    'Aircraft_Category'].value_counts(dropna=False)
```

Out[22]:

```
Aircraft_Category
Airplane        35183
NaN             16062
Helicopter          2
Powered-Lift        1
Unknown             1
Name: count, dtype: int64
```

For these 5 makes, I feel it's reasonable to make them all airplane

In [23]:

```
# make all category entries for particular makes 'Airplane'
df.loc[df['Make'].isin(['Cessna', 'Piper', 'Beech', 'Boeing', 'Mooney']), 'Aircraft_Category'] = 'Airplane'
```

In [24]:

```
# Show Aircraft Category value_counts for Cessna, Piper, Beech, Boeing, and Mooney, inclu
```

```
ding Nan
df[df['Make'].isin(['Cessna', 'Piper', 'Beech', 'Boeing', 'Mooney'])][
    'Aircraft_Category'].value_counts(dropna=False)
```

Out[24]:

```
Aircraft_Category
Airplane    51249
Name: count, dtype: int64
```

In [25]:

```
# Show the Aircraft_Category value_counts for makes_value_10 including NaN
df[df['Make'].isin(makes_value_10.index)]['Aircraft_Category'].value_counts(dropna=False
)
```

Out[25]:

```
Aircraft_Category
Airplane             58564
NaN                  16722
Helicopter            3162
Glider                 326
Balloon                128
Weight-Shift            60
Gyrocraft               14
Ultralight              10
Unknown                  9
WSFT                     4
Powered-Lift             3
Powered Parachute        1
Name: count, dtype: int64
```

At this point, we still have over 16000 empty values in category. Let's see about helicopters and filling in some missing values there.

In [26]:

```
# Helicopter value_counts
df[df['Aircraft_Category'] == 'Helicopter']['Make'].value_counts()
```

Out[26]:

```
Make
Robinson                981
Bell                    948
Hughes                  248
Eurocopter              190
Schweizer               115
                       ... 
BOYKIN STEPHEN VANCE      1
PIASECKI/PIKE             1
Embraer                   1
SMITH RICHARD D JR        1
CHILDS MICHAEL  A         1
Name: count, Length: 238, dtype: int64
```

In [27]:

```
# Show Aircraft Category value_counts for Robinson, Bell, Hughes, Eurocopter, Schweizer,
including Nan
df[df['Make'].isin(['Robinson', 'Bell', 'Hughes', 'Eurocopter', 'Schweizer'])][
    'Aircraft_Category'].value_counts(dropna=False)
```

Out[27]:

```
Aircraft_Category
NaN           3821
Helicopter    2482
Glider         111
Airplane        34
Unknown          4
Name: count, dtype: int64
```

```
Name: count, dtype: int64
```

In [28]:

```
# Again, there's an overwhelming number that are helicopter, so let's change these
# Function to make all category entries for particular makes 'Helicopter'
df.loc[df['Make'].isin(['Robinson', 'Bell', 'Hughes', 'Eurocopter', 'Schweizer']), 'Aircraft_Category'] = 'Helicopter'
```

In [29]:

```
# Show Helicopter value_counts
df[df['Aircraft_Category'] == 'Helicopter']['Make'].value_counts()
```

Out[29]:

```
Make
Bell                    2750
Robinson                1672
Hughes                   935
Schweizer                800
Eurocopter               295
                        ... 
BOYKIN STEPHEN VANCE       1
PIASECKI/PIKE              1
Embraer                    1
SMITH RICHARD D JR         1
CHILDS MICHAEL  A          1
Name: count, Length: 238, dtype: int64
```

In [30]:

```
# Show the Aircraft_Category value_counts for makes_value_10 including NaN
df[df['Make'].isin(makes_value_10.index)]['Aircraft_Category'].value_counts(dropna=False)
```

Out[30]:

```
Aircraft_Category
Airplane             58530
NaN                  12901
Helicopter            7132
Glider                 215
Balloon                128
Weight-Shift            60
Gyrocraft               14
Ultralight              10
Unknown                  5
WSFT                     4
Powered-Lift             3
Powered Parachute        1
Name: count, dtype: int64
```

In [31]:

```
# Now we're down to 13,000 empty category values. Let's look at the makes.
# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[31]:

```
Make
Grumman               1207
Bellanca               750
Air Tractor            462
Mcdonnell Douglas      453
Aeronca                399
                      ... 
Dewitt, Richard A.       1
Greth                    1
Robert D. Waldron        1
Alkire                   1
BELLANCA                 1
Name: count, Length: 3799, dtype: int64
```

```
Name: count, Length: 3799, dtype: int64
```

In [32]:

```
# Show category value_counts for Grumman, Bellanca, Air Tractor, and Aeronca, including NaN
df[df['Make'].isin(['Grumman', 'Bellanca', 'Air Tractor', 'Aeronca'])][
    'Aircraft_Category'].value_counts(dropna=False)
```

Out[32]:

```
Aircraft_Category
NaN         2818
Airplane    1226
Name: count, dtype: int64
```

In [33]:

```
# Fill in 'Airplane' for Grumman, Bellanca, Air Tractor, and Aeronca
df.loc[df['Make'].isin(['Grumman', 'Bellanca', 'Air Tractor', 'Aeronca']), 'Aircraft_Category'] = 'Airplane'
```

In [34]:

```
# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[34]:

```
Make
Mcdonnell Douglas     453
Maule                 356
Champion              347
de Havilland          328
Aero Commander        318
                     ...
Dewitt, Richard A.      1
Greth                   1
Robert D. Waldron       1
Alkire                  1
BELLANCA                1
Name: count, Length: 3795, dtype: int64
```

In [35]:

```
# Show category value_counts for Maule, Champion, de Havilland, Aero Commander, including NaN
df[df['Make'].isin(['Maule', 'Champion', 'de Havilland', 'Aero Commander'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[35]:

```
Aircraft_Category
NaN         1349
Airplane     694
Name: count, dtype: int64
```

In [36]:

```
# Fill in 'Airplane' for Maule, Champion, de Havilland, Aero Commander
df.loc[df['Make'].isin(['Maule', 'Champion', 'de Havilland', 'Aero Commander']), 'Aircraft_Category'] = 'Airplane'
```

In [37]:

```
# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[37]:

```
Make
Mcdonnell Douglas     453
Rockwell              311
Hiller                200
Hiller                    290
Stinson                   287
Aerospatiale              283
                         ...
Dewitt, Richard A.          1
Greth                       1
Robert D. Waldron           1
Alkire                      1
BELLANCA                    1
Name: count, Length: 3791, dtype: int64
```

In [38]:

```
# Show category value_counts for Rockwell, Hiller, Stinson, Aerospatiale, including NaN
df[df['Make'].isin(['Rockwell', 'Stinson'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[38]:

```
Aircraft_Category
NaN         598
Airplane    270
Name: count, dtype: int64
```

In [39]:

```
# Fill in 'Airplane' for Rockwell and Stinson
df.loc[df['Make'].isin(['Rockwell', 'Stinson']), 'Aircraft_Category'] = 'Airplane'
```

In [40]:

```
# Deal with Hiller and Aerospatiale
df[df['Make'].isin(['Aerospatiale', 'Hiller'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[40]:

```
Aircraft_Category
NaN                  573
Helicopter           163
Airplane               3
Powered Parachute      1
Name: count, dtype: int64
```

In [41]:

```
# Since Hiller and Aerospatiale are overwhelmly Helicopter, let's fill those in
df.loc[df['Make'].isin(['Aerospatiale', 'Hiller']), 'Aircraft_Category'] = 'Helicopter'
```

In [42]:

```
# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[42]:

```
Make
Mcdonnell Douglas     453
Taylorcraft           269
North American        265
Luscombe              248
Douglas               222
                     ...
Dewitt, Richard A.      1
Greth                   1
Robert D. Waldron       1
Alkire                  1
BELLANCA                1
Name: count, Length: 3787, dtype: int64
```

In [43]:

```
# deal with Mcdonnell Douglas
```

```
df[df['Make'].isin(['Mcdonnell Douglas'])]['Aircraft_Category'].value_counts(dropna=Fals
e)
```

Out[43]:

```
Aircraft_Category
NaN           453
Airplane      121
Helicopter     15
Name: count, dtype: int64
```

In [44]:

```
# If Make is Mcdonnell Douglas and category is Helicopter, change make to McDonnell Dougl
as Helicopters
df.loc[(df['Make'] == 'Mcdonnell Douglas') & (df['Aircraft_Category'] == 'Helicopter'),
'Make'] = 'McDonnell Douglas Helicopters'

df[df['Make'].isin(['Mcdonnell Douglas'])]['Aircraft_Category'].value_counts(dropna=Fals
e)
```

Out[44]:

```
Aircraft_Category
NaN         453
Airplane    121
Name: count, dtype: int64
```

In [45]:

```
df.loc[df['Make'].isin(['Mcdonnell Douglas']), 'Aircraft_Category'] = 'Airplane'
```

In [46]:

```
# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[46]:

```
Make
Taylorcraft           269
North American        265
Luscombe              248
Douglas               222
Enstrom               212
                     ...
Dewitt, Richard A.      1
Greth                   1
Robert D. Waldron       1
Alkire                  1
BELLANCA                1
Name: count, Length: 3786, dtype: int64
```

In [47]:

```
df[df['Make'].isin(['Taylorcraft', 'North American', 'Luscombe', 'Douglas'])]['Aircraft_
Category'].value_counts(dropna=False)
```

Out[47]:

```
Aircraft_Category
NaN         1004
Airplane     462
Name: count, dtype: int64
```

In [48]:

```
df.loc[df['Make'].isin(['Taylorcraft', 'North American', 'Luscombe', 'Douglas']), 'Aircr
aft_Category'] = 'Airplane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[48]:

```
Make
Enstrom              212
Ayres                181
Ercoupe              168
Airbus               140
Sikorsky             136
                    ... 
Greth                  1
Robert D. Waldron      1
Alkire                 1
Donald L. Betchan      1
BELLANCA               1
Name: count, Length: 3782, dtype: int64
```

In [49]:

```
df[df['Make'].isin(['Enstrom'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[49]:

```
Aircraft_Category
NaN           212
Helicopter     91
Name: count, dtype: int64
```

In [50]:

```
df.loc[df['Make'].isin(['Enstrom']), 'Aircraft_Category'] = 'Helicopter'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[50]:

```
Make
Ayres                181
Ercoupe              168
Airbus               140
Sikorsky             136
Gulfstream           135
                    ... 
Greth                  1
Robert D. Waldron      1
Alkire                 1
Donald L. Betchan      1
BELLANCA               1
Name: count, Length: 3781, dtype: int64
```

In [51]:

```
df[df['Make'].isin(['Ayres', 'Ercoupe', 'Gulfstream'])]['Aircraft_Category'].value_count
s(dropna=False)
```

Out[51]:

```
Aircraft_Category
NaN         484
Airplane    257
Name: count, dtype: int64
```

In [52]:

```
df.loc[df['Make'].isin(['Ayres', 'Ercoupe', 'Gulfstream']), 'Aircraft_Category'] = 'Airp
lane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[52]:

```
Make
Airbus               140
```

```
Sikorsky              136
Fairchild             131
Pitts                 125
Balloon Works         120
                     ...
Greth                   1
Robert D. Waldron       1
Alkire                  1
Donald L. Betchan       1
BELLANCA                1
Name: count, Length: 3778, dtype: int64
```

In [53]:

```
df[df['Make'].isin(['Airbus'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[53]:

```
Aircraft_Category
Airplane        284
NaN             140
Helicopter       20
Powered-Lift      1
Name: count, dtype: int64
```

In [54]:

```
#Deal with Airbus name for helicopters
df.loc[(df['Make'] == 'Airbus') & (df['Aircraft_Category'] == 'Helicopter'), 'Make'] = 'Airbus Helicopters'

df[df['Make'].isin(['Airbus'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[54]:

```
Aircraft_Category
Airplane        284
NaN             140
Powered-Lift      1
Name: count, dtype: int64
```

In [55]:

```
# Fill in the rest for Airbus
df.loc[df['Make'].isin(['Airbus']), 'Aircraft_Category'] = 'Airplane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[55]:

```
Make
Sikorsky              136
Fairchild             131
Pitts                 125
Balloon Works         120
Swearingen            120
                     ...
Greth                   1
Robert D. Waldron       1
Alkire                  1
Donald L. Betchan       1
BELLANCA                1
Name: count, Length: 3777, dtype: int64
```

In [56]:

```
df[df['Make'].isin(['Sikorsky'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[56]:

```
Aircraft_Category
NaN            136
```

```
Helicopter     95
Name: count, dtype: int64
```

In [57]:

```
# Fill in the rest for Sikorsky
df.loc[df['Make'].isin(['Sikorsky']), 'Aircraft_Category'] = 'Helicopter'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[57]:

```
Make
Fairchild             131
Pitts                 125
Balloon Works         120
Swearingen            120
Lake                  114
                     ...
Greth                   1
Robert D. Waldron       1
Alkire                  1
Donald L. Betchan       1
BELLANCA                1
Name: count, Length: 3776, dtype: int64
```

In [58]:

```
df[df['Make'].isin(['Fairchild', 'Pitts', 'Swearingen', 'Lake'])]['Aircraft_Category'].v
alue_counts(dropna=False)
```

Out[58]:

```
Aircraft_Category
NaN         490
Airplane    170
Name: count, dtype: int64
```

In [59]:

```
# Fill in the rest for previous
df.loc[df['Make'].isin(['Fairchild', 'Pitts', 'Swearingen', 'Lake']), 'Aircraft_Category
'] = 'Airplane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[59]:

```
Make
Balloon Works         120
Mitsubishi            109
Burkhart Grob         100
Let                    96
Waco                   95
                     ...
Greth                   1
Robert D. Waldron       1
Alkire                  1
Donald L. Betchan       1
BELLANCA                1
Name: count, Length: 3772, dtype: int64
```

In [60]:

```
df[df['Make'].isin(['Mitsubishi', 'Waco'])]['Aircraft_Category'].value_counts(dropna=Fal
se)
```

Out[60]:

```
Aircraft_Category
NaN         204
Airplane     89
Airplane      89
Unknown        1
Name: count, dtype: int64
```

In [61]:

```
# Fill in the rest for previous
df.loc[df['Make'].isin(['Mitsubishi', 'Waco']), 'Aircraft_Category'] = 'Airplane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[61]:

```
Make
Balloon Works        120
Burkhart Grob        100
Let                   96
Lockheed              94
Ryan                  88
                    ...
Greth                  1
Robert D. Waldron      1
Alkire                 1
Donald L. Betchan      1
BELLANCA               1
Name: count, Length: 3770, dtype: int64
```

In [62]:

```
df[df['Make'].isin(['Ryan', 'Lockheed'])]['Aircraft_Category'].value_counts(dropna=False
)
```

Out[62]:

```
Aircraft_Category
NaN             182
Airplane         63
Powered-Lift      1
Name: count, dtype: int64
```

In [63]:

```
# Fill in the rest for previous
df.loc[df['Make'].isin(['Ryan', 'Lockheed']), 'Aircraft_Category'] = 'Airplane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[63]:

```
Make
Balloon Works        120
Burkhart Grob        100
Let                   96
Aerostar              87
Learjet               86
                    ...
Greth                  1
Robert D. Waldron      1
Alkire                 1
Donald L. Betchan      1
BELLANCA               1
Name: count, Length: 3768, dtype: int64
```

In [64]:

```
df[df['Make'].isin(['Learjet'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[64]:

```
Aircraft_Category
NaN         86
Airplane    59
```

```
Name: count, dtype: int64
```

In [65]:

```
# Fill in the rest for previous
df.loc[df['Make'].isin(['Balloon Works', 'Aerostar']), 'Aircraft_Category'] = 'Balloon'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[65]:

```
Make
Burkhart Grob        100
Let                   96
Learjet               86
Helio                 85
Smith                 83
                    ...
Greth                  1
Robert D. Waldron      1
Alkire                 1
Donald L. Betchan      1
BELLANCA               1
Name: count, Length: 3766, dtype: int64
```

In [66]:

```
# Fill in the rest for previous
df.loc[df['Make'].isin(['Burkhart Grob', 'Let']), 'Aircraft_Category'] = 'Glider'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[66]:

```
Make
Learjet              86
Helio                85
Smith                83
Embraer              81
Raven                79
                     ..
Greth                 1
Robert D. Waldron     1
Alkire                1
Donald L. Betchan     1
BELLANCA              1
Name: count, Length: 3764, dtype: int64
```

In [67]:

```
df[df['Make'].isin(['Raven'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[67]:

```
Aircraft_Category
NaN        79
Balloon     9
Name: count, dtype: int64
```

In [68]:

```
# Fill in the rest for previous
df.loc[df['Make'].isin(['Learjet', 'Helio', 'Smith', 'Embraer']), 'Aircraft_Category'] =
'Airplane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[68]:

```
Make
```

```
Raven                79
Wsk Pzl Mielec       79
British Aerospace    79
Aviat                72
American Aviation    72
                     ..
Greth                 1
Robert D. Waldron     1
Alkire                1
Donald L. Betchan     1
BELLANCA              1
Name: count, Length: 3760, dtype: int64
```

In [69]:

```
df[df['Make'].isin(['Aviat'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[69]:

```
Aircraft_Category
Airplane    159
NaN          72
Name: count, dtype: int64
```

In [70]:

```
# Fill in the rest for previous
df.loc[df['Make'].isin(['Wsk Pzl Mielec', 'British Aerospace', 'American Aviation', 'Aviat']), 'Aircraft_Category'] = 'Airplane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[70]:

```
Make
Raven                79
Globe                71
Schleicher           69
Weatherly            69
Unknown              68
                     ..
Greth                 1
Robert D. Waldron     1
Alkire                1
Donald L. Betchan     1
BELLANCA              1
Name: count, Length: 3756, dtype: int64
```

In [71]:

```
df[df['Make'].isin(['Cirrus'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[71]:

```
Aircraft_Category
Airplane    401
NaN          65
Name: count, dtype: int64
```

In [72]:

```
# Fill in the rest for previous
df.loc[df['Make'].isin(['Globe', 'Weatherly', 'Cirrus']), 'Aircraft_Category'] = 'Airplane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[72]:

```
Make
Raven                79
Schleicher           69
Schleicher            69
Unknown               68
Mbb                   62
Gates Learjet         58
                      ..
Greth                  1
Robert D. Waldron      1
Alkire                 1
Donald L. Betchan      1
BELLANCA               1
Name: count, Length: 3753, dtype: int64
```

In [73]:

```
# Fill in the rest for previous
df.loc[df['Make'].isin(['Raven']), 'Aircraft_Category'] = 'Balloon'
```

In [74]:

```
# Fill in the rest for previous
df.loc[df['Make'].isin(['Schleicher']), 'Aircraft_Category'] = 'Glider'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[74]:

```
Make
Unknown               68
Mbb                   62
Gates Learjet         58
Schempp-hirth         57
Saab                  55
                      ..
Greth                  1
Robert D. Waldron      1
Alkire                 1
Donald L. Betchan      1
BELLANCA               1
Name: count, Length: 3751, dtype: int64
```

In [75]:

```
df[df['Make'].isin(['Navion'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[75]:

```
Aircraft_Category
NaN         53
Airplane    26
Name: count, dtype: int64
```

In [76]:

```
# Fill in the rest for previous
df.loc[df['Make'].isin(['Mbb']), 'Aircraft_Category'] = 'Helicopter'
df.loc[df['Make'].isin(['Schempp-hirth']), 'Aircraft_Category'] = 'Glider'
df.loc[df['Make'].isin(['Gates Learjet', 'Saab', 'Navion']), 'Aircraft_Category'] = 'Air
plane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[76]:

```
Make
Unknown               68
Canadair              53
Socata                50
Dassault              50
Cameron               47
                      ..
Greth                  1
Robert D. Waldron      1
Robert D. Waldron     1
Alkire                1
Donald L. Betchan     1
BELLANCA              1
Name: count, Length: 3746, dtype: int64
```

In [77]:

```
df[df['Make'].isin(['Cameron'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[77]:

```
Aircraft_Category
NaN        47
Balloon     7
Name: count, dtype: int64
```

In [78]:

```
df.loc[df['Make'].isin(['Balloon Works', 'Aerostar', 'Raven', 'Cameron']), 'Aircraft_Category'] = 'Balloon'
df.loc[df['Make'].isin(['Burkhart Grob', 'Let', 'Schleicher', 'Schempp-hirth']), 'Aircraft_Category'] = 'Glider'
df.loc[df['Make'].isin(['Robinson', 'Bell', 'Hughes', 'Eurocopter', 'Schweizer', 'Aerospatiale', 'Hiller', 'Enstrom', 'Sikorsky', 'Mbb']), 'Aircraft_Category'] = 'Helicopter'
df.loc[df['Make'].isin(['Cessna', 'Piper', 'Beech', 'Boeing', 'Mooney', 'Grumman', 'Bellanca', 'Air Tractor', 'Aeronca', 'Maule', 'Champion', 'de Havilland', 'Aero Commander', 'Rockwell', 'Stinson', 'Mcdonnell Douglas', 'Taylorcraft', 'North American', 'Luscombe', 'Douglas', 'Ayres', 'Ercoupe', 'Gulfstream', 'Airbus', 'Fairchild', 'Pitts', 'Swearingen', 'Lake', 'Mitsubishi', 'Waco', 'Ryan', 'Lockheed', 'Learjet', 'Helio', 'Smith', 'Embraer', 'Wsk Pzl Mielec', 'British Aerospace', 'American Aviation', 'Aviat', 'Globe', 'Weatherly', 'Cirrus', 'Gates Learjet', 'Saab', 'Navion', 'Canadair', 'Dassault', 'Socata']), 'Aircraft_Category'] = 'Airplane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[78]:

```
Make
Unknown                68
Fokker                 47
Bombardier             45
Christen Industries    45
Rotorway               45
                       ..
Greth                   1
Robert D. Waldron       1
Alkire                  1
Donald L. Betchan       1
BELLANCA                1
Name: count, Length: 3742, dtype: int64
```

In [79]:

```
df[df['Make'].isin(['Christen Industries'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[79]:

```
Aircraft_Category
NaN         45
Airplane    25
Name: count, dtype: int64
```

In [80]:

```
df.loc[df['Make'].isin(['Rotorway']), 'Aircraft_Category'] = 'Helicopter'
df.loc[df['Make'].isin(['Fokker', 'Bombardier', 'Christen Industries']), 'Aircraft_Category'] = 'Airplane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[80]:

```
Make
Unknown                      68
Great Lakes                  42
Eagle                        42
Eipper                       42
Convair                      40
                             ..
Gera                          1
Kit Fox                       1
Darst                         1
Aero Falcon Intl., Inc.       1
BELLANCA                      1
Name: count, Length: 3738, dtype: int64
```

In [81]:

```
df[df['Make'].isin(['Convair'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[81]:

```
Aircraft_Category
NaN         40
Airplane     9
Name: count, dtype: int64
```

In [82]:

```
df.loc[df['Make'].isin(['Great Lakes', 'Eagle', 'Eipper', 'Convair']), 'Aircraft_Categor
y'] = 'Airplane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[82]:

```
Make
Unknown                 68
Consolidated Aero       37
PZL                     36
Brantly Helicopter      35
Aerotek                 35
                        ..
Greth                    1
Robert D. Waldron        1
Alkire                   1
Donald L. Betchan        1
BELLANCA                 1
Name: count, Length: 3734, dtype: int64
```

In [83]:

```
df[df['Make'].isin(['Brantly Helicopter'])]['Aircraft_Category'].value_counts(dropna=Fal
se)
```

Out[83]:

```
Aircraft_Category
NaN           35
Helicopter     4
Name: count, dtype: int64
```

In [84]:

```
df.loc[df['Make'].isin(['Brantly Helicopter']), 'Aircraft_Category'] = 'Helicopter'
df.loc[df['Make'].isin(['Consolidated Aero', 'Short']), 'Aircraft_Category'] = 'Airplane
'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[84]:

```
Make
Unknown                68
PZL                    36
Aerotek                35
Kaman                  33
Britten Norman         33
                       ..
Greth                   1
Robert D. Waldron       1
Alkire                  1
Donald L. Betchan       1
BELLANCA                1
Name: count, Length: 3731, dtype: int64
```

In [85]:

```
df[df['Make'].isin(['Rolladen Schneider'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[85]:

```
Aircraft_Category
NaN         33
Glider      14
Airplane     1
Name: count, dtype: int64
```

In [86]:

```
df.loc[df['Make'].isin(['Rolladen Schneider']), 'Aircraft_Category'] = 'Glider'
df.loc[df['Make'].isin(['Kaman']), 'Aircraft_Category'] = 'Helicopter'
df.loc[df['Make'].isin(['Aerotek']), 'Aircraft_Category'] = 'Airplane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[86]:

```
Make
Unknown                68
PZL                    36
Britten Norman         33
Fairchild Hiller       33
Pilatus                32
                       ..
Greth                   1
Robert D. Waldron       1
Alkire                  1
Donald L. Betchan       1
BELLANCA                1
Name: count, Length: 3728, dtype: int64
```

In [87]:

```
df[df['Make'].isin(['Raytheon'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[87]:

```
Aircraft_Category
Airplane    88
NaN         32
Name: count, dtype: int64
```

In [88]:

```
df.loc[df['Make'].isin(['Fairchild Hiller']), 'Aircraft_Category'] = 'Helicopter'
df.loc[df['Make'].isin(['Britten Norman', 'Raytheon']), 'Aircraft_Category'] = 'Airplane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[88]:

```
Make
Unknown              68
PZL                  36
Pilatus              32
Agusta               31
Diamond              30
                     ..
Greth                 1
Robert D. Waldron     1
Alkire                1
Donald L. Betchan     1
BELLANCA              1
Name: count, Length: 3725, dtype: int64
```

In [89]:

```
df[df['Make'].isin(['Diamond'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[89]:

```
Aircraft_Category
Airplane        76
NaN             30
Powered-Lift     1
Glider           1
Name: count, dtype: int64
```

In [90]:

```
df.loc[df['Make'].isin(['Agusta']), 'Aircraft_Category'] = 'Helicopter'
df.loc[df['Make'].isin(['Pilatus', 'Diamond']), 'Aircraft_Category'] = 'Airplane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[90]:

```
Make
Unknown              68
PZL                  36
Alon                 29
Hawker               28
Texas Helicopter     28
                     ..
Greth                 1
Robert D. Waldron     1
Alkire                1
Donald L. Betchan     1
BELLANCA              1
Name: count, Length: 3722, dtype: int64
```

In [91]:

```
df[df['Make'].isin(['Continental Copters'])]['Aircraft_Category'].value_counts(dropna=Fa
lse)
```

Out[91]:

```
Aircraft_Category
NaN    28
Name: count, dtype: int64
```

In [92]:

```
df.loc[df['Make'].isin(['Continental Copters']), 'Aircraft_Category'] = 'Helicopter'
df.loc[df['Make'].isin(['Alon', 'Hawker']), 'Aircraft_Category'] = 'Airplane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[92]:

```
Make
```

```
Make
Unknown              68
PZL                  36
Texas Helicopter     28
Republic             27
Rans                 27
                     ..
Greth                 1
Robert D. Waldron     1
Alkire                1
Donald L. Betchan     1
BELLANCA              1
Name: count, Length: 3719, dtype: int64
```

In [93]:

```
df[df['Make'].isin(['Republic'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[93]:

```
Aircraft_Category
NaN         27
Airplane     9
Name: count, dtype: int64
```

In [94]:

```
df.loc[df['Make'].isin(['Texas Helicopter']), 'Aircraft_Category'] = 'Helicopter'
df.loc[df['Make'].isin(['American Champion', 'Republic']), 'Aircraft_Category'] = 'Airpl
ane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[94]:

```
Make
Unknown              68
PZL                  36
Rans                 27
Homebuilt            27
Siai Marchetti       25
                     ..
Greth                 1
Robert D. Waldron     1
Alkire                1
Donald L. Betchan     1
BELLANCA              1
Name: count, Length: 3716, dtype: int64
```

In [95]:

```
df[df['Make'].isin(['Dornier'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[95]:

```
Aircraft_Category
NaN         25
Airplane     9
Name: count, dtype: int64
```

In [96]:

```
df.loc[df['Make'].isin(['Homebuilt', 'Rans', 'Dornier']), 'Aircraft_Category'] = 'Airpla
ne'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[96]:

```
Make
Unknown                       68
PZL                           36
```

```
Siai Marchetti                25
I.c.a. Brasov                 25
Israel Aircraft Industries    24
                              ..
Greth                          1
Robert D. Waldron              1
Alkire                         1
Donald L. Betchan              1
BELLANCA                       1
Name: count, Length: 3713, dtype: int64
```

In [97]:

```
df[df['Make'].isin(['Israel Aircraft Industries'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[97]:

```
Aircraft_Category
NaN         24
Airplane    13
Name: count, dtype: int64
```

In [98]:

```
df.loc[df['Make'].isin(['I.c.a. Brasov']), 'Aircraft_Category'] = 'Glider'
df.loc[df['Make'].isin(['Siai Marchetti', 'Israel Aircraft Industries']), 'Aircraft_Category'] = 'Airplane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[98]:

```
Make
Unknown              68
PZL                  36
Snow                 23
Yakovlev             23
Quicksilver          22
                     ..
Greth                 1
Robert D. Waldron     1
Alkire                1
Donald L. Betchan     1
BELLANCA              1
Name: count, Length: 3710, dtype: int64
```

In [99]:

```
df[df['Make'].isin(['Quicksilver'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[99]:

```
Aircraft_Category
Airplane      53
NaN           22
Ultralight     3
Unknown        1
Name: count, dtype: int64
```

In [100]:

```
df.loc[df['Make'].isin(['Snow', 'Yakovlev', 'Quicksilver']), 'Aircraft_Category'] = 'Airplane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[100]:

```
Make
Unknown              68
```

```
PZL                  36
Lancair              22
Adams Balloon        22
Varga                21
                     ..
Greth                 1
Robert D. Waldron     1
Alkire                1
Donald L. Betchan     1
BELLANCA              1
Name: count, Length: 3707, dtype: int64
```

In [101]:

```
df[df['Make'].isin(['Callair'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[101]:

```
Aircraft_Category
NaN         21
Airplane     5
Name: count, dtype: int64
```

In [102]:

```
df.loc[df['Make'].isin(['Adams Balloon']), 'Aircraft_Category'] = 'Balloon'
df.loc[df['Make'].isin(['Lancair', 'Callair']), 'Aircraft_Category'] = 'Airplane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[102]:

```
Make
Unknown              68
PZL                  36
Varga                21
Quickie              21
Thunder And Colt     21
                     ..
Greth                 1
Robert D. Waldron     1
Alkire                1
Donald L. Betchan     1
BELLANCA              1
Name: count, Length: 3704, dtype: int64
```

In [103]:

```
df[df['Make'].isin(['Quickie'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[103]:

```
Aircraft_Category
NaN         21
Airplane    14
Name: count, dtype: int64
```

In [104]:

```
df.loc[df['Make'].isin(['Thunder And Colt']), 'Aircraft_Category'] = 'Balloon'
df.loc[df['Make'].isin(['Varga', 'Quickie']), 'Aircraft_Category'] = 'Airplane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[104]:

```
Make
Unknown              68
PZL                  36
Garlick              21
Extra                20
Glasflugel           19
Glasflugel           19
                     ..
Greth                 1
Robert D. Waldron     1
Alkire                1
Donald L. Betchan     1
BELLANCA              1
Name: count, Length: 3701, dtype: int64
```

In [105]:

```
# Show Make values whose Aircraft_Category value is NaN if there are over 15
df[df['Aircraft_Category'].isna()]['Make'].value_counts()[df[df['Aircraft_Category'].isn
a()]['Make'].value_counts() > 15]
```

Out[105]:

```
Make
Unknown                  68
PZL                      36
Garlick                  21
Extra                    20
Glasflugel               19
Kolb                     18
Curtiss Wright           18
Glasair                  18
Casa                     18
ATR                      18
Johnson                  17
Classic Aircraft Corp    17
Temco                    17
Davis                    17
Barnes                   17
Air Command              17
Glaser-dirks             16
Forney                   16
Miller                   16
Name: count, dtype: int64
```

In [106]:

```
df[df['Make'].isin(['Glaser-dirks'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[106]:

```
Aircraft_Category
NaN       16
Glider     1
Name: count, dtype: int64
```

In [107]:

```
df.loc[df['Make'].isin(['Barnes']), 'Aircraft_Category'] = 'Balloon'
df.loc[df['Make'].isin(['Glasflugel', 'Glaser-dirks']), 'Aircraft_Category'] = 'Glider'
df.loc[df['Make'].isin(['Garlick']), 'Aircraft_Category'] = 'Helicopter'
df.loc[df['Make'].isin(['Extra', 'Curtiss Wright', 'Kolb', 'Glasair', 'ATR', 'Casa', 'Te
mco', 'Johnson', 'Classic Aircraft Corp', 'Davis', 'Miller', 'Forney']), 'Aircraft_Catego
ry'] = 'Airplane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[107]:

```
Make
Unknown              68
PZL                  36
Air Command          17
Vans                 15
Sukhoi               15
                     ..
Greth                 1
Robert D. Waldron     1
Alkire                1
```

```
Donald L. Betchan      1
BELLANCA               1
Name: count, Length: 3685, dtype: int64
```

In [108]:

```
# Show Make values whose Aircraft_Category value is NaN if there are over 10
df[df['Aircraft_Category'].isna()]['Make'].value_counts()[df[df['Aircraft_Category'].isn
a()]['Make'].value_counts() > 10]
```

Out[108]:

```
Make
Unknown                      68
PZL                          36
Air Command                  17
Vans                         15
Sukhoi                       15
Artic Aircraft Corp.         15
Eiriavion Oy                 15
Anderson Aircraft Corp.      15
Interstate                   15
Thorp                        15
Rotec                        15
American General Aircraft    15
Culver                       14
Stearman                     13
Bensen                       13
Mitchell                     13
Taylor                       12
Pterodactyl                  12
Aerofab Inc.                 12
Weedhopper                   12
Hall                         12
Air & Space                  12
Naval Aircraft Factory       11
Nord                         11
Meyers                       11
Jones                        11
Starduster                   11
Rutan                        11
Teratorn                     11
Howard Aircraft              11
Hispano Aviacion             11
Young                        11
Steen                        11
Partenavia                   11
Name: count, dtype: int64
```

In [109]:

```
df[df['Make'].isin(['Partenavia'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[109]:

```
Aircraft_Category
NaN         11
Airplane     1
Name: count, dtype: int64
```

In [110]:

```
df.loc[df['Make'].isin(['Bensen', 'Air & Space']), 'Aircraft_Category'] = 'Gyrocraft'
df.loc[df['Make'].isin(['Pterodactyl', 'Weedhopper']), 'Aircraft_Category'] = 'Ultraligh
t'
df.loc[df['Make'].isin(['Eiriavion Oy']), 'Aircraft_Category'] = 'Glider'
df.loc[df['Make'].isin(['Interstate', 'Sukhoi', 'Artic Aircraft Corp.', 'Vans', 'Rotec',
'Thorp', 'Anderson Aircraft Corp.', 'American General Aircraft', 'Culver', 'Mitchell', 'S
tearman', 'Aerofab Inc.', 'Hall', 'Taylor', 'Nord', 'Jones', 'Hispano Aviacion', 'Young'
, 'Rutan', 'Naval Aircraft Factory', 'Howard Aircraft', 'Steen', 'Teratorn', 'Meyers', '
Starduster', 'Partenavia']), 'Aircraft_Category'] = 'Airplane'
```

```
# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[110]:

```
Make
Unknown              68
PZL                  36
Air Command          17
Brown                10
Benson               10
                     ..
Greth                 1
Robert D. Waldron     1
Alkire                1
Donald L. Betchan     1
BELLANCA              1
Name: count, Length: 3654, dtype: int64
```

In [111]:

```
# Show Make values whose Aircraft_Category value is NaN if there are over 5 and under 11
blanks_over_5 = df[df['Aircraft_Category'].isna()]['Make'].value_counts()[(df[df['Aircra
ft_Category'].isna()]['Make'].value_counts() > 5)]

# Show blank_over_5 below 11
blanks_over_5[blanks_over_5 < 11]
```

Out[111]:

```
Make
Brown                  10
Benson                 10
Fleet                  10
Piccard                10
American Aerolights    10
                       ..
Scheibe Flugzeugbau     6
Silvaire                6
Aircoupe                6
Weber                   6
Thompson                6
Name: count, Length: 77, dtype: int64
```

In [112]:

```
df[df['Make'].isin(['Hayes'])]['Aircraft_Category'].value_counts(dropna=False)
```

Out[112]:

```
Aircraft_Category
NaN    6
Name: count, dtype: int64
```

In [113]:

```
df.loc[df['Make'].isin(['Benson']), 'Aircraft_Category'] = 'Gyrocraft'
df.loc[df['Make'].isin(['American Aerolights']), 'Aircraft_Category'] = 'Ultralight'
df.loc[df['Make'].isin(['Maxair', 'Bede Aircraft', 'Martin']), 'Aircraft_Category'] = 'Ai
rplane'

# Show Make values whose Aircraft_Category value is NaN
df[df['Aircraft_Category'].isna()]['Make'].value_counts()
```

Out[113]:

```
Make
Unknown              68
PZL                  36
Air Command          17
Piccard              10
Nanchang             10
                     ..
Greth                 1
```

```
Grech                   1
Robert D. Waldron       1
Alkire                  1
Donald L. Betchan       1
BELLANCA                1
Name: count, Length: 3649, dtype: int64
```

In [114]:

```
# Show Make value_counts over 10
makes_value_10 = df['Make'].value_counts()[df['Make'].value_counts() > 10]
makes_value_10
```

Out[114]:

```
Make
Cessna                   26903
Piper                    14818
Beech                     5431
Bell                      2750
Boeing                    2726
                         ...  
Bushby                      11
Hispano Aviacion            11
Flightstar                  11
BURKHART GROB               11
Univair                     11
Name: count, Length: 234, dtype: int64
```

In [115]:

```
# Show the Aircraft_Category value_counts for makes_value_10 including NaN
df[df['Make'].isin(makes_value_10.index)]['Aircraft_Category'].value_counts(dropna=False
)
```

Out[115]:

```
Aircraft_Category
Airplane         68918
Helicopter        8374
Glider             645
Balloon            521
NaN                402
Weight-Shift        60
Gyrocraft           49
Ultralight          29
WSFT                 4
Unknown              1
Name: count, dtype: int64
```

Go ahead here and combine NaN and Unknown in the category column.

In [116]:

```
# Make NaN category 'Unknown'
df.loc[df['Aircraft_Category'].isna(), 'Aircraft_Category'] = 'Unknown'
```

In [117]:

```
# Show the Aircraft_Category value_counts for makes_value_10 including NaN
df[df['Make'].isin(makes_value_10.index)]['Aircraft_Category'].value_counts(dropna=False
)
```

Out[117]:

```
Aircraft_Category
Airplane         68918
Helicopter        8374
Glider             645
Balloon            521
Unknown            403
Weight-Shift        60
Gyrocraft           49
```

```
Ultralight          29
WSFT                 4
Name: count, dtype: int64
```

In [118]:

```
# Make WSFT category 'Weight-Shift'
df.loc[df['Aircraft_Category'] == 'WSFT', 'Aircraft_Category'] = 'Weight-Shift'
```

At this point, the category column is filled in enough

Continue column cleaning

In [119]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 87951 entries, 0 to 88888
Data columns (total 27 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Event_Id                87951 non-null  object 
 1   Investigation_Type      87951 non-null  object 
 2   Accident_Number         87951 non-null  object 
 3   Event_Date              87951 non-null  object 
 4   Location                87899 non-null  object 
 5   Country                 87729 non-null  object 
 6   Airport_Code            49484 non-null  object 
 7   Airport_Name            52031 non-null  object 
 8   Injury_Severity         86961 non-null  object 
 9   Aircraft_damage         84848 non-null  object 
 10  Aircraft_Category       87951 non-null  object 
 11  Registration_Number     86601 non-null  object 
 12  Make                    87951 non-null  object 
 13  Model                   87859 non-null  object 
 14  Amateur_Built           87851 non-null  object 
 15  Number_of_Engines       81924 non-null  float64
 16  Engine_Type             80908 non-null  object 
 17  FAR_Description         31915 non-null  object 
 18  Purpose_of_flight       81829 non-null  object 
 19  Total_Fatal_Injuries    76684 non-null  float64
 20  Total_Serious_Injuries  75629 non-null  float64
 21  Total_Minor_Injuries    76191 non-null  float64
 22  Total_Uninjured         82088 non-null  float64
 23  Weather_Condition       83478 non-null  object 
 24  Broad_phase_of_flight   60837 non-null  object 
 25  Report_Status           81587 non-null  object 
 26  Publication_Date        74352 non-null  object 
dtypes: float64(5), object(22)
memory usage: 18.8+ MB
```

In [120]:

```
# Show FAR_Description values including NaN
df['Weather_Condition'].value_counts(dropna=False)
```

Out[120]:

```
Weather_Condition
VMC    76417
IMC     5949
NaN     4473
UNK      850
Unk      262
Name: count, dtype: int64
```

In [121]:

```
# Change UNK and Unk to Unknown
df.loc[df['Weather_Condition'] == 'Unk', 'Weather_Condition'] = 'Unknown'
df.loc[df['Weather_Condition'] == 'UNK', 'Weather_Condition'] = 'Unknown'

# Change NaN to Unknown
df.loc[df['Weather_Condition'].isna(), 'Weather_Condition'] = 'Unknown'
```

In [122]:

```
# Show FAR_Description values including NaN
df['Weather_Condition'].value_counts(dropna=False)
```

Out[122]:

```
Weather_Condition
VMC         76417
IMC          5949
Unknown      5585
Name: count, dtype: int64
```

In [123]:

```
df['Purpose_of_flight'].value_counts(dropna=False)
```

Out[123]:

```
Purpose_of_flight
Personal                     49076
Instructional                10442
Unknown                       6609
NaN                           6122
Aerial Application            4686
Business                      3971
Positioning                   1632
Other Work Use                1250
Ferry                          806
Aerial Observation             787
Public Aircraft                710
Executive/corporate            542
Flight Test                    405
Skydiving                      181
External Load                  123
Public Aircraft - Federal      104
Banner Tow                     101
Air Race show                   99
Public Aircraft - Local         74
Public Aircraft - State         64
Air Race/show                   53
Glider Tow                      53
Firefighting                    40
Air Drop                        11
ASHO                             5
PUBS                             4
PUBL                             1
Name: count, dtype: int64
```

In [124]:

```
# Change NaN to Unknown
df.loc[df['Purpose_of_flight'].isna(), 'Purpose_of_flight'] = 'Unknown'
```

In [125]:

```
df['Broad_phase_of_flight'].value_counts(dropna=False)
```

Out[125]:

```
Broad_phase_of_flight
NaN             27114
Landing         15320
Takeoff         12404
Cruise          10141
Maneuvering      8052
```

```
Approach        6389
Climb           1995
Descent         1870
Taxi            1786
Go-around       1345
Standing         872
Unknown          547
Other            116
Name: count, dtype: int64
```

In [126]:

```
# Change NaN to Unknown
df.loc[df['Broad_phase_of_flight'].isna(), 'Broad_phase_of_flight'] = 'Unknown'

# Change Other to Unknown
df.loc[df['Broad_phase_of_flight'] == 'Other', 'Broad_phase_of_flight'] = 'Unknown'
```

In [127]:

```
df['Broad_phase_of_flight'].value_counts(dropna=False)
```

Out[127]:

```
Broad_phase_of_flight
Unknown        27777
Landing        15320
Takeoff        12404
Cruise         10141
Maneuvering     8052
Approach        6389
Climb           1995
Descent         1870
Taxi            1786
Go-around       1345
Standing         872
Name: count, dtype: int64
```

In [128]:

```
df['Engine_Type'].value_counts(dropna=False)
```

Out[128]:

```
Engine_Type
Reciprocating      68885
NaN                 7043
Turbo Shaft         3583
Turbo Prop          3324
Turbo Fan           2387
Unknown             2017
Turbo Jet            684
Geared Turbofan       12
Electric              10
LR                     2
NONE                   2
Hybrid Rocket          1
UNK                    1
Name: count, dtype: int64
```

In [129]:

```
# Change NaN to Unknown
df.loc[df['Engine_Type'].isna(), 'Engine_Type'] = 'Unknown'

# Change UNK to Unknown
df.loc[df['Engine_Type'] == 'UNK', 'Engine_Type'] = 'Unknown'
```

In [130]:

```
df['Engine_Type'].value_counts(dropna=False)
```

Out[130]:

Out[130]:

```
Engine_Type
Reciprocating      68885
Unknown             9061
Turbo Shaft         3583
Turbo Prop          3324
Turbo Fan           2387
Turbo Jet            684
Geared Turbofan       12
Electric              10
LR                     2
NONE                   2
Hybrid Rocket          1
Name: count, dtype: int64
```

In [131]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 87951 entries, 0 to 88888
Data columns (total 27 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Event_Id                87951 non-null  object 
 1   Investigation_Type      87951 non-null  object 
 2   Accident_Number         87951 non-null  object 
 3   Event_Date              87951 non-null  object 
 4   Location                87899 non-null  object 
 5   Country                 87729 non-null  object 
 6   Airport_Code            49484 non-null  object 
 7   Airport_Name            52031 non-null  object 
 8   Injury_Severity         86961 non-null  object 
 9   Aircraft_damage         84848 non-null  object 
 10  Aircraft_Category       87951 non-null  object 
 11  Registration_Number     86601 non-null  object 
 12  Make                    87951 non-null  object 
 13  Model                   87859 non-null  object 
 14  Amateur_Built           87851 non-null  object 
 15  Number_of_Engines       81924 non-null  float64
 16  Engine_Type             87951 non-null  object 
 17  FAR_Description         31915 non-null  object 
 18  Purpose_of_flight       87951 non-null  object 
 19  Total_Fatal_Injuries    76684 non-null  float64
 20  Total_Serious_Injuries  75629 non-null  float64
 21  Total_Minor_Injuries    76191 non-null  float64
 22  Total_Uninjured         82088 non-null  float64
 23  Weather_Condition       87951 non-null  object 
 24  Broad_phase_of_flight   87951 non-null  object 
 25  Report_Status           81587 non-null  object 
 26  Publication_Date        74352 non-null  object 
dtypes: float64(5), object(22)
memory usage: 18.8+ MB
```

In [132]:

```
df['Total_Fatal_Injuries'].value_counts(dropna=False)
```

Out[132]:

```
Total_Fatal_Injuries
0.0      59157
NaN      11267
1.0       8801
2.0       5094
3.0       1544
         ...  
83.0         1
143.0        1
144.0        1
60.0         1
176.0        1
```

```
Name: count, Length: 126, dtype: int64
```

In [133]:

```
# Change NaN to 0 in Injury columns
df.loc[df['Total_Fatal_Injuries'].isna(), 'Total_Fatal_Injuries'] = 0
df.loc[df['Total_Serious_Injuries'].isna(), 'Total_Serious_Injuries'] = 0
df.loc[df['Total_Minor_Injuries'].isna(), 'Total_Minor_Injuries'] = 0
df.loc[df['Total_Uninjured'].isna(), 'Total_Uninjured'] = 0
```

In [134]:

```
df['Location'].value_counts(dropna=False)
```

Out[134]:

```
Location
ANCHORAGE, AK          423
MIAMI, FL              197
ALBUQUERQUE, NM        191
HOUSTON, TX            185
FAIRBANKS, AK          171
                      ...
LYNCHBURGH, VA           1
TATITNA, AK              1
LA PUNTILLA, Chile       1
LANGLEY AFB, VA          1
Brasnorte,               1
Name: count, Length: 27746, dtype: int64
```

In [135]:

```
# Change NaN to Unknown
df.loc[df['Location'].isna(), 'Location'] = 'Unknown'
```

In [136]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 87951 entries, 0 to 88888
Data columns (total 27 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Event_Id                87951 non-null  object 
 1   Investigation_Type      87951 non-null  object 
 2   Accident_Number         87951 non-null  object 
 3   Event_Date              87951 non-null  object 
 4   Location                87951 non-null  object 
 5   Country                 87729 non-null  object 
 6   Airport_Code            49484 non-null  object 
 7   Airport_Name            52031 non-null  object 
 8   Injury_Severity         86961 non-null  object 
 9   Aircraft_damage         84848 non-null  object 
 10  Aircraft_Category       87951 non-null  object 
 11  Registration_Number     86601 non-null  object 
 12  Make                    87951 non-null  object 
 13  Model                   87859 non-null  object 
 14  Amateur_Built           87851 non-null  object 
 15  Number_of_Engines       81924 non-null  float64
 16  Engine_Type             87951 non-null  object 
 17  FAR_Description         31915 non-null  object 
 18  Purpose_of_flight       87951 non-null  object 
 19  Total_Fatal_Injuries    87951 non-null  float64
 20  Total_Serious_Injuries  87951 non-null  float64
 21  Total_Minor_Injuries    87951 non-null  float64
 22  Total_Uninjured         87951 non-null  float64
 23  Weather_Condition       87951 non-null  object 
 24  Broad_phase_of_flight   87951 non-null  object 
 25  Report_Status           81587 non-null  object 
 26  Publication_Date        74352 non-null  object 
dtypes: float64(5), object(22)
```

```
memory usage: 18.8+ MB
```

Looking through the values for Make, I see that it may need more attention, and would like to try another method by going through the makes alphabetically.

In [137]:

```
# Let's go back to the Make column and clean further
# Show most popular makes beginning with 'A'
df[df['Make'].str.lower().str.startswith('a', na=False)].value_counts('Make').head(20)
```

Out[137]:

```
Make
Air Tractor            910
Aeronca                629
Airbus                 425
Aero Commander         425
Aerospatiale           380
Ayres                  284
Aviat                  231
Aerostar               122
American Aviation      119
American Champion      101
Agusta                  86
Aerotek                 48
Alon                    41
Airborne                40
Airbus Helicopters      38
ATR                     35
American Legend         29
Adams Balloon           26
Aero Vodochody          23
Avid Aircraft           20
Name: count, dtype: int64
```

In [138]:

```
df.loc[df['Make'].str.lower().str.startswith('aeron', na=False), 'Make'] = 'Aeronca'
df.loc[df['Make'].str.lower().str.startswith('air tra', na=False), 'Make'] = 'Air Tracto
r'
df.loc[df['Make'].str.lower().str.startswith('aero comm', na=False), 'Make'] = 'Aero Com
mander'
df.loc[df['Make'].str.lower().str.startswith('ayre', na=False), 'Make'] = 'Ayres'
df.loc[df['Make'].str.lower().str.startswith('aerosp', na=False), 'Make'] = 'Aerospatial
e'
df.loc[df['Make'].str.lower().str.startswith('airb', na=False), 'Make'] = 'Airbus'
df.loc[df['Make'].str.lower().str.startswith('avia', na=False), 'Make'] = 'Aviat'
```

In [139]:

```
# Show most popular makes beginning with 'A'
df[df['Make'].str.lower().str.startswith('a', na=False)].value_counts('Make').head(20)
```

Out[139]:

```
Make
Air Tractor                   911
Aeronca                       633
Airbus                        507
Aero Commander                425
Aerospatiale                  395
Ayres                         286
Aviat                         259
Aerostar                      122
American Aviation             119
American Champion             101
Agusta                         86
Aerotek                        48
Alon                           41
ATR                            35
American Legend                29
American Legend               29
Adams Balloon                 26
Aero Vodochody                23
Avid Aircraft                 20
American General Aircraft     19
Air Creation                  19
Name: count, dtype: int64
```

In [140]:

```
df.loc[df['Make'].str.lower().str.startswith('agus', na=False), 'Make'] = 'Agusta'
df.loc[df['Make'].str.lower().str.startswith('american cha', na=False), 'Make'] = 'American Champion'
df.loc[df['Make'].str.lower().str.startswith('american av', na=False), 'Make'] = 'American'
df.loc[df['Make'].str.lower().str.startswith('american leg', na=False), 'Make'] = 'American Legend'
```

In [141]:

```
# Show me most popular makes beginning with 'A'
df[df['Make'].str.lower().str.startswith('american g', na=False)].value_counts('Make').head(20)
```

Out[141]:

```
Make
American General Aircraft    19
Name: count, dtype: int64
```

In [142]:

```
# Show me most popular makes beginning with 'A'
df[df['Make'].str.lower().str.startswith('american', na=False)].value_counts('Make').head(20)
```

Out[142]:

```
Make
American                      121
American Champion             102
American Legend                30
American General Aircraft      19
American Eurocopter            12
American Aerolights            10
American Blimp Corp.            5
AMERICAN EUROCOPTER LLC         4
American Autogyro               2
AMERICAN EUROCOPTER             2
American Eagle                  2
AMERICAN AIR RACING LTD         1
American Blimp Corporation      1
AMERICAN BLIMP                  1
American Aircraft               1
American Air Racing             1
AMERICAN LONGEVITY CORP         1
American Yankee                 1
Name: count, dtype: int64
```

In [143]:

```
# Show me most popular makes beginning with 'A'
df[df['Make'].str.lower().str.startswith('american b', na=False)].value_counts('Make').head(20)
```

Out[143]:

```
Make
American Blimp Corp.          5
AMERICAN BLIMP                1
American Blimp Corporation    1
Name: count, dtype: int64
```

In [144]:

```
df.loc[df['Make'].str.lower().str.startswith('american b', na=False), 'Make'] = 'American Blimp'
```

In [145]:

```
# Show me most popular makes beginning with 'A'
df[df['Make'].str.lower().str.startswith('american', na=False)].value_counts('Make').head(20)
```

Out[145]:

```
Make
American                     121
American Champion            102
American Legend               30
American General Aircraft     19
American Eurocopter           12
American Aerolights           10
American Blimp                 7
AMERICAN EUROCOPTER LLC        4
AMERICAN EUROCOPTER            2
American Autogyro              2
American Eagle                 2
AMERICAN AIR RACING LTD        1
AMERICAN LONGEVITY CORP        1
American Air Racing            1
American Aircraft              1
American Yankee                1
Name: count, dtype: int64
```

In [146]:

```
# Show me most popular makes beginning with 'A'
df[df['Make'].str.lower().str.startswith('american eu', na=False)].value_counts('Make').head(20)
```

Out[146]:

```
Make
American Eurocopter        12
AMERICAN EUROCOPTER LLC     4
AMERICAN EUROCOPTER         2
Name: count, dtype: int64
```

In [147]:

```
df.loc[df['Make'].str.lower().str.startswith('american eu', na=False), 'Make'] = 'American Eurocopter'
```

In [148]:

```
# Show me most popular makes beginning with 'B'
df[df['Make'].str.lower().str.startswith('b', na=False)].value_counts('Make').head(20)
```

Out[148]:

```
Make
Beech                 5431
Bell                  2750
Boeing                2726
Bellanca               874
Bombardier             169
BELLANCA               159
Balloon Works          135
Burkhart Grob          109
British Aerospace       90
Britten Norman          55
Brantly Helicopter      39
Barnes                  22
BAE                     21
Benson                  18
```

```
Bensen                17
Bede Aircraft         14
BRANTLY               13
Blanik                11
BURKHART GROB         11
Bushby                11
Name: count, dtype: int64
```

In [149]:

```
# Show me most popular makes beginning with 'B'
df[df['Make'].str.lower().str.startswith('bell', na=False)].value_counts('Make').head(20
)
```

Out[149]:

```
Make
Bell                                 2750
Bellanca                              874
BELLANCA                              159
BELL HELICOPTER                         4
Bell-transworld                         3
Bell Helicopter                         3
Bell-k Copter                           2
BELL HELICOPTER CO                      2
BELL TEXTRON CANADA LTD                 2
Bell-carson                             2
Bell Helicopter Textron                 2
Bell-moore                              1
Bell-olympic Helicopters, Inc.          1
Bell-world                              1
BELL BILL                               1
Bell/garlick                            1
Bell-kitz Kopters                       1
Bell/soloy                              1
Bell/textron                            1
Bell/tsirah                             1
Name: count, dtype: int64
```

In [150]:

```
df.loc[df['Make'].str.lower().str.startswith('bell h', na=False), 'Make'] = 'Bell'
```

In [151]:

```
df[df['Make'].str.lower().str.startswith('bell', na=False)].value_counts('Make').head(20
)
```

Out[151]:

```
Make
Bell                                 2762
Bellanca                              874
BELLANCA                              159
Bell-transworld                         3
Bell-k Copter                           2
BELL TEXTRON CANADA LTD                 2
Bell-carson                             2
Bell-olympic Helicopters, Inc.          1
Bell-world                              1
Bell/garlick                            1
BELL BILL                               1
Bell/mason                              1
Bell-moore                              1
Bell/textron                            1
Bell/tsirah                             1
Bellah                                  1
Bellanca Aircraft Corporation           1
Bellanca Citabria                       1
Bell/soloy                              1
Bell-continental Copters                1
Name: count, dtype: int64
```

In [152]:

```
df.loc[df['Make'].str.lower().str.startswith('bella', na=False), 'Make'] = 'Bellanca'
df.loc[df['Make'].str.lower().str.startswith('bell-', na=False), 'Make'] = 'Bell'
df.loc[df['Make'].str.lower().str.startswith('bell/', na=False), 'Make'] = 'Bell'
df.loc[df['Make'].str.lower().str.startswith('bell t', na=False), 'Make'] = 'Bell'
df.loc[df['Make'].str.lower().str.startswith('bell b', na=False), 'Make'] = 'Bell'
df.loc[df['Make'].str.lower().str.startswith('bell 4', na=False), 'Make'] = 'Bell'
df.loc[df['Make'].str.lower().str.startswith('bell s', na=False), 'Make'] = 'Bell'
```

In [153]:

```
df[df['Make'].str.lower().str.startswith('bell', na=False)].value_counts('Make').head(20
)
```

Out[153]:

```
Make
Bell                  2792
Bellanca              1037
BELLER                   1
BELLET JAMES J           1
Name: count, dtype: int64
```

In [154]:

```
# Show me most popular makes beginning with 'B'
df[df['Make'].str.lower().str.startswith('boe', na=False)].value_counts('Make').head(20)
```

Out[154]:

```
Make
Boeing                              2726
Boeing Vertol                          6
Boeing Helicopters Div.                3
BOEHLKE KEVIN P                        1
BOEING COMPANY, LONG BEACH DIV         1
BOEING-VERTOL                          1
BOEVE EARL                             1
Name: count, dtype: int64
```

In [155]:

```
df.loc[df['Make'].str.lower().str.startswith('boeing h', na=False), 'Make'] = 'Boeing He
licopters'
df.loc[df['Make'].str.lower().str.startswith('boeing c', na=False), 'Make'] = 'Boeing'
df.loc[df['Make'].str.lower().str.startswith('boeing v', na=False), 'Make'] = 'Boeing'
```

In [156]:

```
df[df['Make'].str.lower().str.startswith('b', na=False)].value_counts('Make').head(20)
```

Out[156]:

```
Make
Beech                     5431
Bell                      2792
Boeing                    2733
Bellanca                  1037
Bombardier                 169
Balloon Works              135
Burkhart Grob              109
British Aerospace           90
Britten Norman              55
Brantly Helicopter          39
Barnes                      22
BAE                         21
Benson                      18
Bensen                      17
Bede Aircraft               14
BRANTLY                     13
BURKHART GROB               11
```

```
Bushby                  11
Blanik                  11
Brown                   10
Name: count, dtype: int64
```

In [157]:

```
df.loc[df['Make'].str.lower().str.startswith('bens', na=False), 'Make'] = 'Benson'
```

In [158]:

```
df[df['Make'].str.lower().str.startswith('c', na=False)].value_counts('Make').head(40)
```

Out[158]:

```
Make
Cessna                         26903
Champion                         517
Cirrus                           466
Cub Crafters                      81
Canadair                          72
Christen Industries               70
Consolidated Aero                 61
Cameron                           54
Convair                           49
Curtiss Wright                    38
Costruzioni                       34
Continental Copters               28
Callair                           26
Czech Aircraft Works              24
Classic Aircraft Corp             24
Czech Sport                       23
Casa                              22
Columbia                          22
Cameron Balloons                  21
Culver                            20
Commander                         18
CAMERON BALLOONS US               12
Cassutt                           10
Clark                              9
Chance Vought                      9
Centrair                           9
Challenger                         9
CAMERON                            8
CONTINENTAL COPTERS INC.           6
Colonial                           6
Commonwealth                       6
Carlson                            6
Chamberlain                        6
Carter                             5
Cassuit                            5
Curtiss                            5
Cgs Aviation                       5
COSMOS                             5
Canadian Car & Foundry             4
Curtis                             4
Name: count, dtype: int64
```

In [159]:

```
df.loc[df['Make'].str.lower().str.startswith('cub', na=False), 'Make'] = 'Cubcrafters'
df.loc[df['Make'].str.lower().str.startswith('cirrus', na=False), 'Make'] = 'Cirrus Desi
gn'
df.loc[df['Make'].str.lower().str.startswith('champ', na=False), 'Make'] = 'Champion'
df.loc[df['Make'].str.lower().str.startswith('christ', na=False), 'Make'] = 'Christen In
dustries'
df.loc[df['Make'].str.lower().str.startswith('consol', na=False), 'Make'] = 'Consolidate
d Aeronautics'
```

In [160]:

```
# Show category value counts for Cameron
# Show category value_counts for Cameron
df[df['Make'] == 'Cameron'].value_counts('Aircraft_Category', dropna=False)
```

Out[160]:

```
Aircraft_Category
Balloon    54
Name: count, dtype: int64
```

In [161]:

```
df.loc[df['Make'].str.lower().str.startswith('camer', na=False), 'Make'] = 'Cameron Balloons'
```

In [162]:

```
df[df['Make'].str.lower().str.startswith('c', na=False)].value_counts('Make').head(40)
```

Out[162]:

```
Make
Cessna                           26903
Champion                           517
Cirrus Design                      467
Cameron Balloons                   105
Christen Industries                 93
Cubcrafters                         81
Canadair                            72
Consolidated Aeronautics            63
Convair                             49
Curtiss Wright                      38
Costruzioni                         34
Continental Copters                 28
Callair                             26
Czech Aircraft Works                24
Classic Aircraft Corp               24
Czech Sport                         23
Casa                                22
Columbia                            22
Culver                              20
Commander                           18
Cassutt                             10
Chance Vought                        9
Challenger                           9
Clark                                9
Centrair                             9
Chamberlain                          6
Commonwealth                         6
Carlson                              6
CONTINENTAL COPTERS INC.             6
Colonial                             6
Cassuit                              5
Carter                               5
COSMOS                               5
Curtiss                              5
Cgs Aviation                         5
Cook                                 4
Cosmos                               4
Corben                               4
Canadian Car & Foundry               4
Curtis                               4
Name: count, dtype: int64
```

In [163]:

```
df.loc[df['Make'].str.lower().str.startswith('continental c', na=False), 'Make'] = 'Continental Copters'
df.loc[df['Make'].str.lower().str.startswith('cosmos', na=False), 'Make'] = 'Cosmos'
df.loc[df['Make'].str.lower().str.startswith('curtis', na=False), 'Make'] = 'Curtiss Wright'
```

In [164]:

```
df[df['Make'].str.lower().str.startswith('c', na=False)].value_counts('Make').head(40)
```

Out[164]:

```
Make
Cessna                      26903
Champion                      517
Cirrus Design                 467
Cameron Balloons              105
Christen Industries            93
Cubcrafters                    81
Canadair                       72
Consolidated Aeronautics       63
Curtiss Wright                 49
Convair                        49
Continental Copters            37
Costruzioni                    34
Callair                        26
Czech Aircraft Works           24
Classic Aircraft Corp          24
Czech Sport                    23
Columbia                       22
Casa                           22
Culver                         20
Commander                      18
Cosmos                         10
Cassutt                        10
Challenger                      9
Chance Vought                   9
Clark                           9
Centrair                        9
Colonial                        6
Commonwealth                    6
Carlson                         6
Chamberlain                     6
Carter                          5
Cgs Aviation                    5
Cassuit                         5
Corben                          4
Cook                            4
Condor                          4
Cunningham                      4
Canadian Car & Foundry          4
Continental                     3
Cgs                             3
Name: count, dtype: int64
```

In [165]:

```
df.loc[df['Make'].str.lower().str.startswith('cassu', na=False), 'Make'] = 'Cassutt'
df.loc[df['Make'].str.lower().str.startswith('cgs', na=False), 'Make'] = 'CGS Aviation'
df.loc[df['Make'].str.lower().str.startswith('continental', na=False), 'Make'] = 'Contin
ental Copters'
```

In [166]:

```
df[df['Make'].str.lower().str.startswith('d', na=False)].value_counts('Make').head(20)
```

Out[166]:

```
Make
de Havilland                512
Douglas                     276
Diamond                     108
DIAMOND AIRCRAFT IND INC     74
Dassault                     72
Dornier                      34
Davis                        19
Downer                        9
Denney                        6
DESTINY                       5
DG FLUGZEUGBAU GMBH           4
Drake                         3
```

```
Day                           3
DJI                           3
Davenport                     3
DAVIS MICHAEL J               2
Donegan Benton                2
Dodd                          2
DAY WILLIAM L                 2
DAYTON A BABCOCK              2
Name: count, dtype: int64
```

In [167]:

```
df.loc[df['Make'].str.lower().str.startswith('doug', na=False), 'Make'] = 'Douglas'
df.loc[df['Make'].str.lower().str.startswith('dorn', na=False), 'Make'] = 'Dornier'
```

In [168]:

```
df[df['Make'].str.lower().str.startswith('e', na=False)].value_counts('Make').head(30)
```

Out[168]:

```
Make
Enstrom                           303
Eurocopter                        295
Embraer                           253
Ercoupe                           243
Eagle                              59
Extra                              52
Eipper                             52
Evektor Aerotechnik                40
Engineering & Research             21
Eiriavion Oy                       17
Eclipse Aviation                   11
Erco                               11
Evans Aircraft                     10
Experimental                        9
EVOLUTION AIRCRAFT INC              9
EVOLUTION TRIKES                    5
Emair                               5
Ercoupe (Eng & Research Corp.)      4
Erickson                            4
Evolution                           4
Eaa Biplane                         4
Elliott                             4
Edge                                3
Eaa                                 3
Emroth Emair                        3
Ellis                               3
Europa                              3
EIRIAVION OY                        3
Elliot                              2
Eldredge                            2
Name: count, dtype: int64
```

In [169]:

```
df.loc[df['Make'].str.lower().str.startswith('eagl', na=False), 'Make'] = 'Eagle Aircraf
t'
df.loc[df['Make'].str.lower().str.startswith('embr', na=False), 'Make'] = 'Embraer'
df.loc[df['Make'].str.lower().str.startswith('enstrom', na=False), 'Make'] = 'Enstrom'
df.loc[df['Make'].str.lower().str.startswith('ercou', na=False), 'Make'] = 'Ercoupe'
df.loc[df['Make'].str.lower().str.startswith('euroc', na=False), 'Make'] = 'Eurocopter'
df.loc[df['Make'].str.lower().str.startswith('evek', na=False), 'Make'] = 'Evektor Aerot
echnik'
df.loc[df['Make'].str.lower().str.startswith('extra', na=False), 'Make'] = 'Extra'
```

In [170]:

```
df[df['Make'].str.lower().str.startswith('e', na=False)].value_counts('Make').head(30)
```

Out[170]:

```
Make
Enstrom                   303
Eurocopter                295
Embraer                   257
Ercoupe                   247
Eagle Aircraft             65
Extra                      52
Eipper                     52
Evektor Aerotechnik        44
Engineering & Research     21
Eiriavion Oy               17
Erco                       11
Eclipse Aviation           11
Evans Aircraft             10
Experimental                9
EVOLUTION AIRCRAFT INC      9
EVOLUTION TRIKES            5
Emair                       5
Eaa Biplane                 4
Elliott                     4
Evolution                   4
Erickson                    4
Eaa                         3
Ellis                       3
EIRIAVION OY                3
Europa                      3
Emroth Emair                3
Edge                        3
Eldredge                    2
Edwards                     2
Eames                       2
Name: count, dtype: int64
```

In [171]:

```
df.loc[df['Make'].str.lower().str.startswith('eaa', na=False), 'Make'] = 'Eaa'
df.loc[df['Make'].str.lower().str.startswith('ecli', na=False), 'Make'] = 'Eclipse Aviation'
df.loc[df['Make'].str.lower().str.startswith('eip', na=False), 'Make'] = 'Eipper'
df.loc[df['Make'].str.lower().str.startswith('eiri', na=False), 'Make'] = 'Eiriavion Oy'
df.loc[df['Make'].str.lower().str.startswith('eng', na=False), 'Make'] = 'Engineering & Research'
df.loc[df['Make'].str.lower().str.startswith('evol', na=False), 'Make'] = 'Evolution'
```

In [172]:

```
df[df['Make'].str.lower().str.startswith('f', na=False)].value_counts('Make').head(30)
```

Out[172]:

```
Make
Fairchild                       177
Flight Design                    79
Fokker                           64
Fairchild Hiller                 39
Fisher                           22
Fleet                            17
Forney                           17
Fantasy Air                      13
Flightstar                       11
Found Aircraft Canada             8
Firefly                           5
Funk, D.d. Aviation Co.           4
Flagg                             4
Firefly Balloons                  4
Franklin                          3
Fetherolf                         3
Fields                            3
FPNA LLC                          3
Ferguson                          3
Fairchild Dornier                 3
Fouga                             3
```

```
Fowler                            2
Focke-wulf                        2
FOUGA                             2
Folland                           2
Farrington                        2
Fagan                             2
Flugzeugbau                       2
Flug-und Fahrzeugwerke (ffa)      2
FUNK                              2
Name: count, dtype: int64
```

In [173]:

```
df.loc[df['Make'].str.lower().str.startswith('fairch', na=False), 'Make'] = 'Fairchild'
df.loc[df['Make'].str.lower().str.startswith('firef', na=False), 'Make'] = 'Firefly'
df.loc[df['Make'].str.lower().str.startswith('fish', na=False), 'Make'] = 'Fisher'
df.loc[df['Make'].str.lower().str.startswith('fleet', na=False), 'Make'] = 'Fleet'
df.loc[df['Make'].str.lower().str.startswith('flight d', na=False), 'Make'] = 'Flight Design'
df.loc[df['Make'].str.lower().str.startswith('flights', na=False), 'Make'] = 'Flightstar'
df.loc[df['Make'].str.lower().str.startswith('fokk', na=False), 'Make'] = 'Fokker'
```

In [174]:

```
df[df['Make'].str.lower().str.startswith('f', na=False)].value_counts('Make').head(30)
```

Out[174]:

```
Make
Fairchild                       221
Flight Design                    81
Fokker                           64
Fisher                           24
Fleet                            18
Forney                           17
Firefly                          15
Fantasy Air                      13
Flightstar                       11
Found Aircraft Canada             8
Funk, D.d. Aviation Co.           4
Flagg                             4
Fouga                             3
Fields                            3
Ferguson                          3
FPNA LLC                          3
Franklin                          3
Fetherolf                         3
FUNK                              2
Fagan                             2
Fuji                              2
Frost                             2
Funk Aircraft Co.                 2
Focke-wulf                        2
Folland                           2
Flug-und Fahrzeugwerke (ffa)      2
Flugzeugbau                       2
Freeman                           2
Farrington                        2
FOUGA                             2
Name: count, dtype: int64
```

In [175]:

```
df.loc[df['Make'].str.lower().str.startswith('foug', na=False), 'Make'] = 'Fouga'
df.loc[df['Make'].str.lower().str.startswith('found', na=False), 'Make'] = 'Found Aircraft'
df.loc[df['Make'].str.lower().str.startswith('funk', na=False), 'Make'] = 'Funk'
```

In [176]:

```
df[df['Make'].str.lower().str.startswith('f', na=False)].value_counts('Make').head(30)
```

Out[176]:

```
Make
Fairchild                       221
Flight Design                    81
Fokker                           64
Fisher                           24
Fleet                            18
Forney                           17
Firefly                          15
Fantasy Air                      13
Flightstar                       11
Funk                              9
Found Aircraft                    8
Fouga                             5
Flagg                             4
Franklin                          3
Ferguson                          3
Fields                            3
FPNA LLC                          3
Fetherolf                         3
Flug-und Fahrzeugwerke (ffa)      2
Freeman                           2
Fowler                            2
Folland                           2
Flugzeugbau                       2
Fuji                              2
Farrington                        2
Fagan                             2
Frost                             2
Focke-wulf                        2
Folke Wulf                        1
Foley James                       1
Name: count, dtype: int64
```

In [177]:

```
df.loc[df['Make'].str.lower().str.startswith('fant', na=False), 'Make'] = 'Fantasy'
```

In [178]:

```
df[df['Make'].str.lower().str.startswith('g', na=False)].value_counts('Make').head(30)
```

Out[178]:

```
Make
Grumman                           1631
Gulfstream                         214
Globe                               97
Gates Learjet                       82
Great Lakes                         67
Garlick                             42
Glasair                             35
Glasflugel                          25
Glaser-dirks                        17
Gardner                              9
Grob                                 9
GULFSTREAM-SCHWEIZER A/C CORP        9
General Atomics                      8
Gray                                 7
Guimbal                              6
Government Aircraft Fact (gaf)       6
General Balloon                      6
Green                                4
Griffin                              4
Goodyear                             4
Gordon                               4
GLASFLUGEL                           4
Graham                               3
Glaser Dirks                         3
GROB                                 3
Giles                                3
Giles                                3
Garrett                              3
Gulfstream-Schweizer                 3
Galloway                             2
Glover                               2
Name: count, dtype: int64
```

In [179]:

```python
df.loc[df['Make'].str.lower().str.startswith('garl', na=False), 'Make'] = 'Garlick'
df.loc[df['Make'].str.lower().str.startswith('gates', na=False), 'Make'] = 'Gates Learjet'
df.loc[df['Make'].str.lower().str.startswith('general atom', na=False), 'Make'] = 'General Atomics'
df.loc[df['Make'].str.lower().str.startswith('glasa', na=False), 'Make'] = 'Glasair'
df.loc[df['Make'].str.lower().str.startswith('glassa', na=False), 'Make'] = 'Glasair'
df.loc[df['Make'].str.lower().str.startswith('glase', na=False), 'Make'] = 'Glaser Dirks'
df.loc[df['Make'].str.lower().str.startswith('glasf', na=False), 'Make'] = 'Glasflugel'
df.loc[df['Make'].str.lower().str.startswith('globe', na=False), 'Make'] = 'Globe'
df.loc[df['Make'].str.lower().str.startswith('great l', na=False), 'Make'] = 'Great Lakes'
df.loc[df['Make'].str.lower().str.startswith('grob', na=False), 'Make'] = 'Grob'
df.loc[df['Make'].str.lower().str.startswith('grum', na=False), 'Make'] = 'Grumman'
df.loc[df['Make'].str.lower().str.startswith('gulfstr', na=False), 'Make'] = 'Gulfstream'
df.loc[df['Make'].str.lower().str.startswith('golden c', na=False), 'Make'] = 'Golden Circle Air'
df.loc[df['Make'].str.lower().str.startswith('gren', na=False), 'Make'] = 'Grenier'
```

In [180]:

```python
df[df['Make'].str.lower().str.startswith('h', na=False)].value_counts('Make').head(50)
```

Out[180]:

```
Make
Hughes                        935
Hiller                        360
Helio                         115
Hawker                         36
Homebuilt                      30
Hall                           19
Howard Aircraft                19
Hispano Aviacion               11
Head Balloons, Inc.            10
Hunter                          9
Hayes                           6
Harris                          6
Honda Aircraft                  6
HEAD BALLOONS INC               5
Hamilton                        5
Huff                            5
Holmes                          5
HELICOPTERES GUIMBAL            4
Hill                            4
Hardy                           4
Hanson                          4
Hudson                          4
HEAD                            3
Hammond                         3
Harmon                          3
Hackney                         3
Henderson                       3
Higher Class Aviation           3
Head                            3
Hatz                            3
Highcraft                       3
Hoffman, Wolf, Flugzeugbau      2
Hartman                         2
Hodges                          2
Hardie                          2
Hahn                            2
```

```
HEINLEIN GEORGE                 2
Hard                            2
Hammer                          2
Haines                          2
Howard                          2
Hutton                          2
HEMP TIMOTHY                    2
HAL                             2
Hannah                          2
Hanks                           2
HPH LTD                         2
HUGHES AERO CORP                2
HOFFMAN                         2
Harding                         2
Name: count, dtype: int64
```

In [181]:

```
df.loc[df['Make'].str.lower().str.startswith('hawk', na=False), 'Make'] = 'Hawker'
df.loc[df['Make'].str.lower().str.startswith('head', na=False), 'Make'] = 'Head Balloons'
df.loc[df['Make'].str.lower().str.startswith('helio', na=False), 'Make'] = 'Helio'
df.loc[df['Make'].str.lower().str.startswith('hiller', na=False), 'Make'] = 'Hiller'
df.loc[df['Make'].str.lower().str.startswith('howard', na=False), 'Make'] = 'Howard Aircraft'
df.loc[df['Make'].str.lower().str.startswith('hughes', na=False), 'Make'] = 'Hughes Helicopters'
```

In [182]:

```
df[df['Make'].str.lower().str.startswith('i', na=False)].value_counts('Make').head(50)
```

Out[182]:

```
Make
Israel Aircraft Industries        37
I.c.a. Brasov                     27
Interstate                        20
Icon                               7
Interplane                         7
Intermountain Mfg. (imco)          6
IAI                                4
INFINITY                           4
Issoire-aviation                   3
INIZIATIVE INDUSTRIALI ITALIAN     3
Infinity                           3
Interavia                          3
I.C.A.-BRASOV (ROMANIA)            2
Irwin                              2
ICP                                2
Iniziative Industriali Italian     2
Insua                              1
Inman                              1
Intl Ultralight                    1
Ingraham                           1
Ireneusz                           1
Irving Siewert                     1
Isgrigg                            1
Isaacson                           1
Iseman                             1
Ismari                             1
Ison Aircraft                      1
Issitt                             1
Istenes                            1
Istvanick                          1
Iv Inc.                            1
Ivan Langston                      1
Iversen                            1
Iverslie                           1
Ives                               1
Infinity Power Parachutes          1
Inav                               1
```

```
Indus Aviation                     1
Indus                              1
I.c.a. Brasov - Romania            1
I.c.a.-brasov                      1
IAR BRASOV                         1
ICA BRASOV                         1
ICP SRL                            1
ILYUSHIN                           1
INDEPENDENT TECHNOLOGIES INC       1
INDUS AVIATION INC                 1
INDUS AVIATION INC.                1
INDY AIRCRAFT LTD                  1
INFINITY POWER PARACHUTES LLC      1
Name: count, dtype: int64
```

In [183]:

```
df.loc[df['Make'].str.lower().str.startswith('i.c.a', na=False), 'Make'] = 'I.c.a. Brasov'
df.loc[df['Make'].str.lower().str.startswith('ica', na=False), 'Make'] = 'I.c.a. Brasov'
df.loc[df['Make'].str.lower().str.startswith('icon', na=False), 'Make'] = 'Icon'
df.loc[df['Make'].str.lower().str.startswith('indu', na=False), 'Make'] = 'Indus'
df.loc[df['Make'].str.lower().str.startswith('infini', na=False), 'Make'] = 'Infinity'
df.loc[df['Make'].str.lower().str.startswith('iniz', na=False), 'Make'] = 'Iniziative Industriali Italian'
df.loc[df['Make'].str.lower().str.startswith('interp', na=False), 'Make'] = 'Interplane'
df.loc[df['Make'].str.lower().str.startswith('intersta', na=False), 'Make'] = 'Interstate'
df.loc[df['Make'].str.lower().str.startswith('israel', na=False), 'Make'] = 'Israel Aircraft Industries'
```

In [184]:

```
df[df['Make'].str.lower().str.startswith('j', na=False)].value_counts('Make').head(50)
```

Out[184]:

```
Make
Jabiru Aircraft                21
Jones                          21
Johnson                        20
Just                           14
Jackson                         9
Jenkins                         5
JOHNSON                         5
Jodel                           4
Jihlavan                        4
Jonas                           4
Jetstream                       3
Jordan                          3
JONKER SAILPLANES (PTY) LTD     3
Jacobs                          2
JDT                             2
Jauch                           2
JOE SALOMONE                    2
JIHLAVAN AIRPLANES SRO          2
J&J Ultralights                 2
Jack Mcdaniel                   2
Jahns                           2
Jurca                           2
Jackman                         2
Jobe's                          1
Jodell                          1
Joel K. Senter                  1
Joel H Johnson                  1
Jim Elliott                     1
Joe Underwood                   1
Jim Garrison                    1
Joe L. Harr                     1
Joe Almon                       1
Jim Mckinstry                   1
Jim Weseman                     1
```

```
Job                              1
Jimenez Reymundo                 1
Jodel-bernier                    1
Jimi Youngblood                  1
Jarrell                          1
Jilek-smith Miniplane            1
Jiran                            1
Jaquish                          1
Joachim Hoehne                   1
Jimmy Hill                       1
Jeffs                            1
Jasper                           1
Jerry Berry                      1
Jennings                         1
Jennings John C                  1
Jerald F. Huffman                1
Name: count, dtype: int64
```

In [185]:

```
df.loc[df['Make'].str.lower().str.startswith('jabi', na=False), 'Make'] = 'Jabiru'
df.loc[df['Make'].str.lower().str.startswith('jihl', na=False), 'Make'] = 'Jihlavan'
df.loc[df['Make'].str.lower().str.startswith('jode', na=False), 'Make'] = 'Jodel'
df.loc[df['Make'].str.lower().str.startswith('johns', na=False), 'Make'] = 'Johnson'
df.loc[df['Make'].str.lower().str.startswith('jones', na=False), 'Make'] = 'Jones'
df.loc[df['Make'].str.lower().str.startswith('just', na=False), 'Make'] = 'Just Aircraft
'
```

In [186]:

```
df[df['Make'].str.lower().str.startswith('k', na=False)].value_counts('Make').head(50)
```

Out[186]:

```
Make
Kaman                       40
Kolb                        36
Kitfox                      19
KUBICEK                      6
King                         6
Kelly                        4
Keith                        3
Keller                       3
Kirkpatrick                  3
Keuthan                      3
Kennedy                      3
Knapp                        2
Kubicek                      2
Kawasaki                     2
Kauffman                     2
Kinner                       2
Keesler                      2
Kelley                       2
Kucklick                     2
Krotje                       2
KAWASAKI                     2
Kimbal                       1
Kit Built (rotorway)         1
Kindig                       1
Kitchen                      1
Klasing                      1
Kitty Hawk                   1
Kinkade E.5                  1
Kit Fox                      1
Kinnson                      1
Kirner                       1
Kittleson                    1
Kirby                        1
Kircher                      1
Kirchner                     1
Kitchens                     1
Kirst Allen J                1
```

```
Kite                        1
Kimball Enterprises Inc     1
K COPTERS                   1
Kilpatrick                  1
Ketonen                     1
Kenney                      1
Kenny                       1
Kenny Deward                1
Kenoyer                     1
Kepple                      1
Kerlin                      1
Kermit Weeks                1
Kerner                      1
Name: count, dtype: int64
```

In [187]:

```
df.loc[df['Make'].str.lower().str.startswith('kama', na=False), 'Make'] = 'Kaman'
df.loc[df['Make'].str.lower().str.startswith('kawa', na=False), 'Make'] = 'Kawasaki'
df.loc[df['Make'].str.lower().str.startswith('kitf', na=False), 'Make'] = 'Kitfox'
df.loc[df['Make'].str.lower().str.startswith('kolb', na=False), 'Make'] = 'Kolb'
df.loc[df['Make'].str.lower().str.startswith('kubic', na=False), 'Make'] = 'Kubicek'
```

In [188]:

```
df[df['Make'].str.lower().str.startswith('l', na=False)].value_counts('Make').head(50)
```

Out[188]:

```
Make
Luscombe                          412
Lake                              151
Learjet                           145
Let                               135
Lockheed                          122
Lancair                            62
Lindstrand                         30
Liberty                            18
Larsen                             13
Long                                9
Lewis                               8
Laister                             5
Lockwood                            4
Lee                                 4
Lutz                                3
Logan                               3
Levick                              2
L-BIRD LLC                          2
Leone                               2
Leblanc                             2
Lowe                                2
Lithuanian Factory Of Aviation      2
Lampert                             2
Lawrence                            2
Lamb                                2
Lett R/campbell K                   1
Letecky Zavody                      1
Let Np Kinovice                     1
Lett                                1
Lien                                1
Lester F.w. West                    1
Lidster                             1
Libersat                            1
Lidgard                             1
Leveck                              1
Levitsky                            1
Lessel                              1
Lewis Jennings                      1
Lewis-pexton                        1
Lewis-starduster                    1
Leza Lockwood                       1
Lgs Aviation, Inc.                  1
```

```
L GOLDNER                          1
Lerstang                           1
Leslie J. Royal                    1
Leslie Briggs                      1
Lazair                             1
Lazarini                           1
Lciv Llc                           1
Lee Harold Swarthout               1
Name: count, dtype: int64
```

In [189]:

```
df.loc[df['Make'].str.lower().str.startswith('lake', na=False), 'Make'] = 'Lake'
df.loc[df['Make'].str.lower().str.startswith('lanc', na=False), 'Make'] = 'Lancair'
df.loc[df['Make'].str.lower().str.startswith('lars', na=False), 'Make'] = 'Larson'
df.loc[df['Make'].str.lower().str.startswith('lear', na=False), 'Make'] = 'Learjet'
df.loc[df['Make'].str.lower().str.startswith('let', na=False), 'Make'] = 'Let'
df.loc[df['Make'].str.lower().str.startswith('liberty', na=False), 'Make'] = 'Liberty'
df.loc[df['Make'].str.lower().str.startswith('lindst', na=False), 'Make'] = 'Lindstrand Balloons'
df.loc[df['Make'].str.lower().str.startswith('lockh', na=False), 'Make'] = 'Lockheed'
df.loc[df['Make'].str.lower().str.startswith('long', na=False), 'Make'] = 'Long'
df.loc[df['Make'].str.lower().str.startswith('lusc', na=False), 'Make'] = 'Luscombe'
```

In [190]:

```
df[df['Make'].str.lower().str.startswith('m', na=False)].value_counts('Make').head(50)
```

Out[190]:

```
Make
Mooney                          1371
Maule                            589
Mcdonnell Douglas                574
Mitsubishi                       142
McDonnell Douglas Helicopters    126
Mbb                               70
Meyers                            22
Miller                            21
Martin                            16
Monocoupe                         16
Murphy                            15
Mitchell                          14
Maxair                            14
Morrisey                          12
Messerschmitt                     11
Mcculloch                         11
Monnett                            9
Moore                              9
Morrison                           7
Molino Oy                          7
Myers                              7
Meyer                              6
Mason                              5
Morgan                             5
McDonnell Douglas                  4
Mikoyan Mig                        4
Mccoy                              4
Mattison                           4
Morane-saulnier                    4
Moravan                            4
Mccall                             3
MBB                                3
MOSQUITO                           3
Mcilraith                          3
Mccarty                            3
MX AIRCRAFT LLC                    3
Mcfarland                          3
Mcclung                            3
Murray                             3
Morton                             3
Mohr                               3
```

```
McClish                                 3
Mueller                                 3
MIKOYAN GUREVICH                        2
Manville                                2
Midget                                  2
Mancini                                 2
Mclaughlin                              2
Micco Aircraft Company                  2
Mankovich                               2
Name: count, dtype: int64
```

In [191]:

```
df.loc[df['Make'].str.lower().str.startswith('martin', na=False), 'Make'] = 'Martin'
df.loc[df['Make'].str.lower().str.startswith('maul', na=False), 'Make'] = 'Maule'
df.loc[df['Make'].str.lower().str.startswith('MCDONNELL DOUGLAS H', na=False), 'Make'] = 'Mcdonnell Douglas Helicopters'
df.loc[df['Make'].str.lower().str.startswith('MCDONNELL DOUGLAS A', na=False), 'Make'] = 'Mcdonnell Douglas'
```

In [192]:

```
df[df['Make'].str.lower().str.startswith('mcdonn', na=False)].value_counts('Make').head(50)
```

Out[192]:

```
Make
Mcdonnell Douglas                    574
McDonnell Douglas Helicopters        126
McDonnell Douglas                      4
MCDONNELL-DOUGLAS                      1
Name: count, dtype: int64
```

In [193]:

```
df.loc[df['Make'].str.lower().str.startswith('mcdonnell-douglas', na=False), 'Make'] = 'Mcdonnell Douglas'
```

In [194]:

```
df[df['Make'].str.lower().str.startswith('mcdonn', na=False)].value_counts('Make').head(50)
```

Out[194]:

```
Make
Mcdonnell Douglas                    575
McDonnell Douglas Helicopters        126
McDonnell Douglas                      4
Name: count, dtype: int64
```

In [195]:

```
df.loc[(df['Make'] == 'MCDONNELL DOUGLAS') | (df['Make'] == 'McDonnell Douglas'), 'Make'] = 'Mcdonnell Douglas'
```

In [196]:

```
df[df['Make'].str.lower().str.startswith('mcdonn', na=False)].value_counts('Make').head(50)
```

Out[196]:

```
Make
Mcdonnell Douglas                    579
McDonnell Douglas Helicopters        126
Name: count, dtype: int64
```

In [197]:

```
df.loc[df['Make'].str.lower().str.startswith('maxair', na=False), 'Make'] = 'Maxair'
df.loc[df['Make'].str.lower().str.startswith('mbb', na=False), 'Make'] = 'MBB'
df.loc[df['Make'].str.lower().str.startswith('md helicopter', na=False), 'Make'] = 'Md Helicopter'
df.loc[df['Make'].str.lower().str.startswith('meyer', na=False), 'Make'] = 'Meyers'
df.loc[df['Make'].str.lower().str.startswith('miller', na=False), 'Make'] = 'Miller'
df.loc[df['Make'].str.lower().str.startswith('mitsub', na=False), 'Make'] = 'Mitsubishi'
df.loc[df['Make'].str.lower().str.startswith('monoco', na=False), 'Make'] = 'Monocoupe'
df.loc[df['Make'].str.lower().str.startswith('moone', na=False), 'Make'] = 'Mooney'
df.loc[df['Make'].str.lower().str.startswith('morris', na=False), 'Make'] = 'Morrisey'
df.loc[df['Make'].str.lower().str.startswith('murph', na=False), 'Make'] = 'Murphy'
df.loc[df['Make'].str.lower().str.startswith('messersch', na=False), 'Make'] = 'Messerschmitt'
df.loc[df['Make'].str.lower().str.startswith('mikoya', na=False), 'Make'] = 'Mikoyan'
df.loc[df['Make'].str.lower().str.startswith('moor', na=False), 'Make'] = 'Moore'
df.loc[df['Make'].str.lower().str.startswith('mong', na=False), 'Make'] = 'Mong'
```

In [198]:

```
df[df['Make'].str.lower().str.startswith('n', na=False)].value_counts('Make').head(50)
```

Out[198]:

```
Make
North American            392
Navion                     79
Nanchang                   22
North Wing                 18
Naval Aircraft Factory     14
Nord                       14
Northwing                  11
Nihon                       8
Nelson                      8
Nicholson                   5
Northrop                    5
New Standard                4
National Balloon            3
Noble                       3
NEIVA                       2
New                         2
Nolley                      2
Nolen                       2
Nolan                       2
Needham                     2
Nesmith                     1
Nichols Rans, Inc.          1
Nelson Miles                1
Nicely                      1
Noteman                     1
Neyman-pietenpol            1
Newgard                     1
Newell Thomas               1
Newell                      1
Newberg                     1
Newcomer                    1
Newbold                     1
Nerstrom-tailwind           1
Nickel                      1
Nunley                      1
Nunn                        1
Neumann-everett             1
Netz                        1
Northam                     1
Nicolas Beasely             1
Nickelson Martin            1
Noakes B J                  1
Norton                      1
Norris                      1
Norman Negus                1
Nord (SNCAN)                1
Norton/cloeren              1
Norcutt                     1
Noorduyn Aviation           1
```

```
Noonan                      1
Name: count, dtype: int64
```

In [199]:

```
df.loc[df['Make'].str.lower().str.startswith('nanch', na=False), 'Make'] = 'Nanchang'
df.loc[df['Make'].str.lower().str.startswith('navio', na=False), 'Make'] = 'Navion'
df.loc[df['Make'].str.lower().str.startswith('nelso', na=False), 'Make'] = 'Nelson'
df.loc[df['Make'].str.lower().str.startswith('new pip', na=False), 'Make'] = 'New Piper'
df.loc[df['Make'].str.lower().str.startswith('newel', na=False), 'Make'] = 'Newell'
df.loc[df['Make'].str.lower().str.startswith('nord', na=False), 'Make'] = 'Nord'
df.loc[df['Make'].str.lower().str.startswith('north ame', na=False), 'Make'] = 'North American'
df.loc[df['Make'].str.lower().str.startswith('north w', na=False), 'Make'] = 'North Wing'
df.loc[df['Make'].str.lower().str.startswith('northw', na=False), 'Make'] = 'North Wing'
```

In [200]:

```
df[df['Make'].str.lower().str.startswith('o', na=False)].value_counts('Make').head(50)
```

Out[200]:

```
Make
Olson                       4
Omf                         3
O'connor                    2
Olsen                       2
Osprey                      2
O Loughlin                  1
Omac                        1
Oostdik                     1
Oo-culley                   1
Ontario Avia. Hist. Soc.    1
Oneil                       1
Olsen-gordon                1
Opus Motorsports Llc        1
Olree Robert                1
Oldfield                    1
Ohlemeier/stratton          1
Ogden Aubrey D              1
Ofria                       1
Opperman                    1
Orlando Heli Air Inc.       1
Oregon Helicopters          1
Otis                        1
Owens/richburg              1
Owen Robert R               1
Overton                     1
Oveross                     1
Ouzts Rodney                1
Otis G. Lyons               1
Otero-pitts                 1
Oestreich                   1
Osprey 2                    1
Osborne                     1
Ortmayer/parson             1
Ortiz G/saint J             1
Orr                         1
Orlican                     1
Offord                      1
Oehling                     1
O'DELL                      1
Obryon                      1
OLIVER A BRUCE              1
OLIVER                      1
OHLGREN BRENT E             1
OHLGREN                     1
OGG RICHARD A               1
OFFCHISS EDWARD R           1
ODEN WELDON PAT             1
OCONNOR PAUL A              1
OAS                         1
```

```
UAS                         1
O'neil                      1
Name: count, dtype: int64
```

In [201]:

```
df[df['Make'].str.lower().str.startswith('p', na=False)].value_counts('Make').head(50)
```

Out[201]:

```
Make
Piper                   14818
Pitts                     161
Pilatus                    69
PZL                        57
Pipistrel                  25
Pietenpol                  14
Piaggio                    14
Pterodactyl                13
Progressive Aerodyne       12
Partenavia                 12
Piccard                    10
Porterfield                 8
Pioneer                     7
Parsons                     7
Pezetel                     7
Pzl                         7
Parker                      5
Powrachute                  5
Price                       4
Palen                       4
Pierce                      4
PARTENAVIA                  4
PDPS PZL-BIELSKO            4
Peterson                    4
POWRACHUTE                  4
P&M AVIATION LTD            3
Pdps Pzl-bielsko            3
Pereyra                     3
PARKER                      3
Pacific Aerospace           3
Peck                        3
Perth Amboy                 3
Perkins                     3
Pitcairn                    3
Porter                      3
Pratt                       3
PHILLIPS                    2
Park                        2
PIEL                        2
PIGGOTT JOHN H              2
PORTER JAMES GRANT          2
Palmer                      2
Page                        2
Pearson                     2
Parks                       2
Pilgrim                     2
Poberezny                   2
POWRACHUTE LLC              2
PPHU EKOLOT                 2
Playmate                    2
Name: count, dtype: int64
```

In [202]:

```
df.loc[df['Make'].str.lower().str.startswith('p z', na=False), 'Make'] = 'PZL'
df.loc[df['Make'].str.lower().str.startswith('pz', na=False), 'Make'] = 'PZL'
df.loc[df['Make'].str.lower().str.startswith('parke', na=False), 'Make'] = 'Parker'
df.loc[df['Make'].str.lower().str.startswith('partenav', na=False), 'Make'] = 'Partenavi
a'
df.loc[df['Make'].str.lower().str.startswith('pdp', na=False), 'Make'] = 'PDPS'
df.loc[df['Make'].str.lower().str.startswith('perth', na=False), 'Make'] = 'Perth Amboy'
df.loc[df['Make'].str.lower().str.startswith('phanto', na=False), 'Make'] = 'Phantom'
df.loc[df['Make'].str.lower().str.startswith('philli', na=False), 'Make'] = 'Phillips'
df.loc[df['Make'].str.lower().str.startswith('piagg', na=False), 'Make'] = 'Piaggio'
df.loc[df['Make'].str.lower().str.startswith('piel', na=False), 'Make'] = 'Piel'
df.loc[df['Make'].str.lower().str.startswith('piet', na=False), 'Make'] = 'Pietenpol'
df.loc[df['Make'].str.lower().str.startswith('pilat', na=False), 'Make'] = 'Pilatus'
df.loc[df['Make'].str.lower().str.startswith('piper', na=False), 'Make'] = 'Piper'
df.loc[df['Make'].str.lower().str.startswith('pipest', na=False), 'Make'] = 'Pipestrel'
df.loc[df['Make'].str.lower().str.startswith('pitts', na=False), 'Make'] = 'Pitts'
df.loc[df['Make'].str.lower().str.startswith('powr', na=False), 'Make'] = 'Powrachute'
df.loc[df['Make'].str.lower().str.startswith('progress', na=False), 'Make'] = 'Progressive'
```

In [203]:

```
df[df['Make'].str.lower().str.startswith('q', na=False)].value_counts('Make').head(50)
```

Out[203]:

```
Make
Quicksilver                      79
Quad City                        35
Quickie                          35
Quest                            14
Questair                          8
Q-berry                           1
Quist                             1
Quinn Aviation Inc.               1
Quinn Aviation                    1
Quinn                             1
Quest Ii Walsh Ltd.               1
QUARNOCCIO                        1
Qualline                          1
Qac                               1
QUASAR ACFT CO LLC                1
QUARTZ MOUNTAIN AEROSPACE INC     1
QUARTZ MOUNTAIN AEROSPACE         1
Quitzau, Carl                     1
Name: count, dtype: int64
```

In [204]:

```
df.loc[df['Make'].str.lower().str.startswith('quartz', na=False), 'Make'] = 'Quartz Mountain'
df.loc[df['Make'].str.lower().str.startswith('quad', na=False), 'Make'] = 'Quad City'
df.loc[df['Make'].str.lower().str.startswith('quest a', na=False), 'Make'] = 'Quest Aircraft'
df.loc[(df['Make'] == 'QUEST') | (df['Make'] == 'Quest'), 'Make'] = 'Quest Aircraft'
df.loc[df['Make'].str.lower().str.startswith('questa', na=False), 'Make'] = 'Questair'
df.loc[df['Make'].str.lower().str.startswith('quickie', na=False), 'Make'] = 'Quickie'
df.loc[df['Make'].str.lower().str.startswith('quick s', na=False), 'Make'] = 'Quicksilver'
df.loc[df['Make'].str.lower().str.startswith('quicksil', na=False), 'Make'] = 'Quicksilver'
df.loc[df['Make'].str.lower().str.startswith('quiks', na=False), 'Make'] = 'Quicksilver'
df.loc[df['Make'].str.lower().str.startswith('quinn', na=False), 'Make'] = 'Quinn'
```

In [205]:

```
df[df['Make'].str.lower().str.startswith('r', na=False)].value_counts('Make').head(50)
```

Out[205]:

```
Make
Robinson                       1672
Rockwell                        435
Ryan                            124
Raytheon                        120
Rans                             96
Raven                            88
Rotorway                         71
Rolladen Schneider               48
Republic                         36
Republic                         36
Reims                            25
Remos                            24
Rutan                            21
Rotec                            15
Rose                             14
Ross                              8
Robin                             8
Rotec Engineering, Inc.           7
Rand                              7
Robertson                         6
Rogers                            6
Root                              5
Rich                              5
Russell                           5
Robbins                           4
Rodgers                           3
RAINBOW SKY REACH (PTY) LTD       3
Rearwin                           2
Rominger                          2
Rainey                            2
RICE                              2
ROBINSON MICHAEL E                2
Rocket Flyers                     2
Rawdon Bros. Aircraft             2
Richardson                        2
ROTORSPORT UK LTD                 2
Rockwell Intl.                    2
Rhoades                           2
Reid                              2
Reindl                            2
Revolution                        2
Robert D. Campbell                1
Robert Frisbey                    1
Robert J. Jackson                 1
Robert J. Goodyear                1
Robert H. Low                     1
Robert Griffiths                  1
Reynolds                          1
Robert D Carr Jr                  1
Robert Eldon Idler                1
Robert D. Waldron                 1
Name: count, dtype: int64
```

In [206]:

```
df.loc[df['Make'].str.lower().str.startswith('rans', na=False), 'Make'] = 'Rans'
df.loc[df['Make'].str.lower().str.startswith('raven', na=False), 'Make'] = 'Raven'
df.loc[df['Make'].str.lower().str.startswith('raythe', na=False), 'Make'] = 'Raytheon'
df.loc[df['Make'].str.lower().str.startswith('reims', na=False), 'Make'] = 'Reims Aviation'
df.loc[df['Make'].str.lower().str.startswith('remos', na=False), 'Make'] = 'Remos'
df.loc[df['Make'].str.lower().str.startswith('republ', na=False), 'Make'] = 'Republic'
df.loc[df['Make'].str.lower().str.startswith('revolut', na=False), 'Make'] = 'Revolution Helicopters'
df.loc[df['Make'].str.lower().str.startswith('riddel', na=False), 'Make'] = 'Riddell'
df.loc[df['Make'].str.lower().str.startswith('robinson', na=False), 'Make'] = 'Robinson Helicopter'
df.loc[(df['Make'] == 'ROBIN') | (df['Make'] == 'Robin'), 'Make'] = 'Robin'
df.loc[df['Make'].str.lower().str.startswith('rockwell', na=False), 'Make'] = 'Rockwell'
df.loc[df['Make'].str.lower().str.startswith('rolladen', na=False), 'Make'] = 'Rolladen-Schneider'
df.loc[df['Make'].str.lower().str.startswith('rose', na=False), 'Make'] = 'Rose'
df.loc[df['Make'].str.lower().str.startswith('rotec', na=False), 'Make'] = 'Rotec'
df.loc[df['Make'].str.lower().str.startswith('rotorw', na=False), 'Make'] = 'Rotorway'
df.loc[df['Make'].str.lower().str.startswith('rutan', na=False), 'Make'] = 'Rutan'
df.loc[df['Make'].str.lower().str.startswith('ryan', na=False), 'Make'] = 'Ryan'
```

In [207]:

```
df[df['Make'].str.lower().str.startswith('s', na=False)].value_counts('Make').loc[lambda x : x<3].head(60)
```

Out[207]:

```
Make
S C AEROSTAR S A                  2
Sproul                            2
SLINGSBY                          2
Sanders                           2
Simpson                           2
Stafford                          2
Scorpion Too                      2
Schrack                           2
Schweitzer                        2
Sturges                           2
Sinclair                          2
Sawyer                            2
SILVERLIGHT AVIATION LLC          2
Stump                             2
Stugart                           2
Scheibe                           2
Springer                          2
Sprague                           2
Steinke                           2
Sportavia-putzer                  2
Smyth Sidewinder                  2
Snider                            2
Snobird                           2
Sellors                           2
Syracuse                          2
Senior Aero Sport                 2
Sloan                             2
S.C. Aerostar S.A.                2
S.N.I.A.S.                        2
STEPHENSON                        2
Skystar                           2
SABRE                             2
SPORT PLANE DYNAMICS LLC          2
Sadler                            2
Skov-papworth                     2
Shannon                           2
Sport Flight                      2
Schumacher                        2
Stampe                            2
Smyth                             2
SCHUMACHER                        2
Sheppard                          2
SCHEMPP-HIRTH FLUGZEUGBAU         2
Shook                             2
Stephenson                        2
Steven W. Jones                   2
Stephen                           2
SEA & SKY INC                     2
Stevenson                         2
SCHLEICHER ALEXANDER GMBH & CO    2
Storey                            2
Schramm                           2
Strissel                          2
Shiner                            1
Shirlan Dickey                    1
Sessi Midget Mustang              1
Sevdy Pitts Special               1
Skystar Aircraft Corp             1
Shepherd                          1
Shanks/becker                     1
Name: count, dtype: int64
```

In [208]:

```
df.loc[df['Make'].str.lower().str.startswith('saab', na=False), 'Make'] = 'Saab'
df.loc[df['Make'].str.lower().str.startswith('scheibe', na=False), 'Make'] = 'Scheibe'
df.loc[df['Make'].str.lower().str.startswith('schempp', na=False), 'Make'] = 'Schempp Hirth'
df.loc[df['Make'].str.lower().str.startswith('schleich', na=False), 'Make'] = 'Schleicher'
df.loc[df['Make'].str.lower().str.startswith('schwei', na=False), 'Make'] = 'Schweizer'
df.loc[df['Make'].str.lower().str.startswith('scottish', na=False), 'Make'] = 'Scottish Aviation'
df.loc[df['Make'].str.lower().str.startswith('short bro', na=False), 'Make'] = 'Short Brothers'
df.loc[df['Make'].str.lower().str.startswith('siai', na=False), 'Make'] = 'Siai Marchetti'
df.loc[df['Make'].str.lower().str.startswith('sikors', na=False), 'Make'] = 'Sikorsky'
df.loc[df['Make'].str.lower().str.startswith('silva', na=False), 'Make'] = 'Silvaire'
df.loc[df['Make'].str.lower().str.startswith('six ch', na=False), 'Make'] = 'Six Chuter'
df.loc[df['Make'].str.lower().str.startswith('skykit', na=False), 'Make'] = 'Skykits Corp'
df.loc[df['Make'].str.lower().str.startswith('slings', na=False), 'Make'] = 'Slingsby'
```

In [209]:

```
df[df['Make'].str.lower().str.startswith('t', na=False)].value_counts('Make').loc[lambda x : x<2].head(60)
```

Out[209]:

```
Make
Tingle                         1
Tinsman                        1
Timothy J Brown                1
Tibert                         1
Timothy Wingate                1
Tichacek                       1
Tidd Wesley                    1
Timm                           1
Tierra Ii                      1
Tilbert                        1
Tifft-v                        1
T BIRD                         1
Thomsen, Horst                 1
Thurmond                       1
Thomas C. Piper                1
Tetrault Ronad                 1
Thacker                        1
Thalman                        1
Tharp S Richard                1
The Old Hen Crow               1
Theis                          1
Theordore Jankowski            1
Therrien Roger                 1
Thiessen                       1
Thomas D. Parkes               1
Thunder Mustang                1
Thomas E. Georges              1
Thomas J. Dorsey               1
Thomas Long                    1
Thomas Wild                    1
Thompson Howard                1
Tocholke                       1
Thornhill                      1
Thornley                       1
Thunder & Colt Ltd             1
Tjerrild                       1
Tom Kilgore                    1
Todd                           1
Trohoski                       1
Troy                           1
Troy A. Woodland               1
Troy Bellah                    1
Truckee Meadows                1
Trudel                         1
True Flight Holdings LLC       1
Truitt Peter                   1
Truthan                        1
Tschida                        1
```

```
Tubb                           1
Tubbs                          1
Tubbs S/Performance            1
Tubbs S/Performance Air Inc    1
Tuk Gregory                    1
Tukan                          1
Tullis                         1
Tundermann Venne               1
Tupta-smith Miniplane          1
Turkan                         1
Turnbloom                      1
Turner                         1
Name: count, dtype: int64
```

In [210]:

```
df.loc[df['Make'].str.lower().str.startswith('taylorcr', na=False), 'Make'] = 'Taylorcraft'
df.loc[df['Make'].str.lower().str.startswith('tecn', na=False), 'Make'] = 'Tecnam'
df.loc[df['Make'].str.lower().str.startswith('temco', na=False), 'Make'] = 'Temco'
df.loc[df['Make'].str.lower().str.startswith('terato', na=False), 'Make'] = 'Teratorn'
df.loc[df['Make'].str.lower().str.startswith('texas h', na=False), 'Make'] = 'Texas Helicopter'
df.loc[df['Make'].str.lower().str.startswith('textro', na=False), 'Make'] = 'Textron Aviation'
df.loc[df['Make'].str.lower().str.startswith('thorp', na=False), 'Make'] = 'Thorp'
df.loc[df['Make'].str.lower().str.startswith('thrush', na=False), 'Make'] = 'Thrush Aircraft'
df.loc[df['Make'].str.lower().str.startswith('thunder', na=False), 'Make'] = 'Thunder And Colt'
df.loc[df['Make'].str.lower().str.startswith('titan', na=False), 'Make'] = 'Titan'
df.loc[df['Make'].str.lower().str.startswith('tl u', na=False), 'Make'] = 'TL Ultralight'
df.loc[df['Make'].str.lower().str.startswith('travel', na=False), 'Make'] = 'Travel Air'
df.loc[df['Make'].str.lower().str.startswith('trick', na=False), 'Make'] = 'Trick Trikes'
df.loc[df['Make'].str.lower().str.startswith('tubb', na=False), 'Make'] = 'Tubbs'
df.loc[df['Make'].str.lower().str.startswith('tupole', na=False), 'Make'] = 'Tupolev'
df.loc[df['Make'].str.lower().str.startswith('the boei', na=False), 'Make'] = 'Boeing'
```

In [211]:

```
df[df['Make'].str.lower().str.startswith('u', na=False)].value_counts('Make').loc[lambda x : x>0].head(60)
```

Out[211]:

```
Make
Unknown                      84
Universal                    16
Univair                      11
ULTRAMAGIC                    9
Ultramagic                    5
ULTRAMAGIC SA                 5
Upton                         3
ULTRALIGHT AMERICA            3
UNKNOWN                       3
Ultralight Flight             2
Unrein                        2
Ultralight Flight, Inc.       2
Us/lta                        2
Urban Air                     1
United Consultant Corp.       1
Uvify                         1
Ultravia Aero Int', Inc.      1
Ultravia Aero Int'l           1
Unander                       1
Underwood                     1
Ungerecht                     1
Utva                          1
Urban Air SRO                 1
Utterback                     1
University Of Alabama         1
University Of Alabama         1
Uskoski                       1
Unregistered                  1
Ultralight Ufm                1
Upright                       1
Upchurch                      1
U-FLY-IT                      1
Ultralight Soaring            1
Ufm                           1
ULTRALITE SRO                 1
ULTRAMAGIC S A                1
UNDERLAND/GROTHE/FINSTROM     1
UNITED CONSULTANTS            1
UNIVERSITY OF KANSAS          1
UNknown                       1
URBAN AIR SRO                 1
URBANCZYK MIROSLAW            1
US LIGHT AIRCRAFT CORP        1
Uetz, Walter                  1
Uhley-mason                   1
ULTRA FLIGHT LLC              1
Ulfeldt                       1
Ullman                        1
Ullrich                       1
Ulrich Christen               1
Ultimate                      1
Ultra Magic                   1
Ultraclassics                 1
Ultraflight                   1
Ultralight                    1
Ultralight Engineering        1
unknown                       1
Name: count, dtype: int64
```

In [212]:

```
df.loc[df['Make'].str.lower().str.startswith('ultrali', na=False), 'Make'] = 'Ultralight Flight'
df.loc[df['Make'].str.lower().str.startswith('ultramag', na=False), 'Make'] = 'Ultramagic'
df.loc[df['Make'].str.lower().str.startswith('ultravia', na=False), 'Make'] = 'Ultravia Aero'
df.loc[df['Make'].str.lower().str.startswith('united cons', na=False), 'Make'] = 'United Consultant Corp.'
df.loc[df['Make'].str.lower().str.startswith('univa', na=False), 'Make'] = 'Univair'
df.loc[df['Make'].str.lower().str.startswith('universal s', na=False), 'Make'] = 'Universal'
df.loc[df['Make'].str.lower().str.startswith('unknow', na=False), 'Make'] = 'Unknown'
df.loc[df['Make'].str.lower().str.startswith('unregis', na=False), 'Make'] = 'Unknown'
df.loc[df['Make'].str.lower().str.startswith('urban a', na=False), 'Make'] = 'Urban Air'
```

In [213]:

```
df[df['Make'].str.lower().str.startswith('v', na=False)].value_counts('Make').loc[lambda x : x>0].head(60)
```

Out[213]:

```
Make
Vans                    166
Varga                    26
Velocity                 12
Varieze                   8
Vickers                   7
Volksplane                5
Valentin                  4
Volmer Aircraft           3
VASHON AIRCRAFT           2
Violet                    2
VAUGHN                    2
Van's Aircraft            2
Viking                    2
Vultee                    2
```

```
Vangrunsven               2
Vector                    2
Veltman, Robert           1
Ventura                   1
Venus                     1
Verbeek                   1
Veltman                   1
Veith                     1
Vehafric                  1
VALENTIN                  1
Veazie                    1
Vavra, Martin A.          1
Vaughn                    1
Vermeys                   1
Vaughan                   1
Vari-eze                  1
Vari                      1
Vanwinkle                 1
Vans-rupert               1
Vanhoose/scorpio          1
Vere Eze                  1
Vestal                    1
Vernon                    1
Virgil D. Hawks           1
Vuncannon                 1
Vought                    1
Vortec                    1
Von Claparede Clemen      1
Von Berg                  1
Volmer                    1
Vollmers                  1
Vogt                      1
Vogel                     1
Virginia Aviation         1
Virgil Larson             1
Villine's                 1
Vernon D. Pitts           1
Villeneuve                1
Villarubia                1
Vigneau                   1
Victor Marsh              1
Victor E. Nigro           1
Vickers Slingsby          1
Vicari                    1
Via Inc                   1
Vangilder                 1
Name: count, dtype: int64
```

In [214]:

```
df.loc[df['Make'].str.lower().str.startswith('valent', na=False), 'Make'] = 'Valentin'
df.loc[df['Make'].str.lower().str.startswith('vans', na=False), 'Make'] = 'Vans'
df.loc[df['Make'].str.lower().str.startswith("van's", na=False), 'Make'] = 'Vans'
df.loc[df['Make'].str.lower().str.startswith('varga', na=False), 'Make'] = 'Varga'
df.loc[df['Make'].str.lower().str.startswith('vari', na=False), 'Make'] = 'Varieze'
```

In [215]:

```
df[df['Make'].str.lower().str.startswith('vaugh', na=False)].value_counts('Make').loc[la
mbda x : x>0].head(60)
```

Out[215]:

```
Make
VAUGHN              2
VAUGHAN GERALD R    1
Vaughan             1
Vaughn              1
Name: count, dtype: int64
```

In [216]:

```
df.loc[df['Make'].str.lower().str.startswith('vaugh', na=False), 'Make'] = 'Vaughn'
```

In [217]:

```
df[df['Make'].str.lower().str.startswith('w', na=False)].value_counts('Make').loc[lambda
x : x<2].head(60)
```

Out[217]:

```
Make
Wilde Mickey                     1
Wilderoder                       1
Wernega                          1
Wilford J. Tolman                1
Wente                            1
Wild Goose                       1
Wells/rand                       1
Wiles                            1
Welles                           1
Wiley                            1
Willbird                         1
Wilhelm                          1
Wilkinson Daniel V               1
Wilbur D. Batman                 1
Willey                           1
Willi/bensen                     1
William A. Waas                  1
William Allen                    1
Welborn                          1
William Bovard                   1
William C. Keiling               1
Wilburn                          1
Wesley                           1
Werner                           1
Western International            1
Weston                           1
Westland                         1
William E. Furr                  1
Whistle                          1
Westholm                         1
White Lightning Aircraft Corp.   1
Whitehead                        1
Whithington                      1
Whitlock                         1
Whitman                          1
Whittaker                        1
Wiedemann                        1
Whitty                           1
Westerhout/Gagnier               1
Weste                            1
Westcott                         1
West-heckman                     1
Whitty, Philip J.                1
West                             1
Wichawk                          1
Wessel                           1
Weyerts                          1
William D. Weeks                 1
W. E. Simmons Max Air            1
William F. Lair                  1
Wondrasek                        1
Woodstock                        1
Woodruff Lancair                 1
Woodrow Stolp-adams              1
Woodrow S.a.                     1
Woodley                          1
Wood/thorpe                      1
Wood-oldfield                    1
Wolter                           1
Wise                             1
Name: count, dtype: int64
```

In [218]:

```
df.loc[df['Make'].str.lower().str.startswith('waco', na=False), 'Make'] = 'Waco'
df.loc[df['Make'].str.lower().str.startswith('weatherl', na=False), 'Make'] = 'Weatherly'
df.loc[df['Make'].str.lower().str.startswith('weber', na=False), 'Make'] = 'Weber'
df.loc[df['Make'].str.lower().str.startswith('westland', na=False), 'Make'] = 'Westland Helicopters'
df.loc[df['Make'].str.lower().str.startswith('wheele', na=False), 'Make'] = 'Wheeler'
df.loc[df['Make'].str.lower().str.startswith('white', na=False), 'Make'] = 'White'
df.loc[df['Make'].str.lower().str.startswith('whittman', na=False), 'Make'] = 'Whittman'
df.loc[df['Make'].str.lower().str.startswith('williams hel', na=False), 'Make'] = 'Williams Helicopter'
df.loc[df['Make'].str.lower().str.startswith('wsk', na=False), 'Make'] = 'WSK'
```

In [219]:

```
df[df['Make'].str.lower().str.startswith('x', na=False)].value_counts('Make').loc[lambda x : x>0].head(60)
```

Out[219]:

```
Make
Xtremeair GMBH    4
X-AIR LLC         1
Name: count, dtype: int64
```

In [220]:

```
df[df['Make'].str.lower().str.startswith('y', na=False)].value_counts('Make').loc[lambda x : x>0].head(60)
```

Out[220]:

```
Make
Yakovlev                 45
Young                    12
YAMOKOSKI WILLIAM         1
Yasecko                   1
Younkin J R               1
Young-owens               1
Young-losey               1
Youkey                    1
Yothment                  1
York                      1
Yonchak                   1
Yoder                     1
Yoakley                   1
Yeager Inc.               1
Yates                     1
Yarnell                   1
YATES MIKE E              1
Yaple, Robert G.          1
Yamokoski                 1
Yadon                     1
YUNEEC                    1
YOUNGS WILLIAM D          1
YOUNG ROBERT HERMAN       1
YOUNG JOHN E              1
YOUNG DEE C               1
YOST EDWIN S              1
YORK RONALD J             1
YORK                      1
YATES W/LAZAR M           1
Yowell                    1
Name: count, dtype: int64
```

In [221]:

```
df.loc[df['Make'].str.lower().str.startswith('yamok', na=False), 'Make'] = 'Yamokoski'
```

In [222]:

In [222]:

```
df[df['Make'].str.lower().str.startswith('z', na=False)].value_counts('Make').loc[lambda x : x>0].head(60)
```

Out[222]:

```
Make
Zenith                      46
Zenair                      35
Zlin                         8
Zivko                        7
Zimmerman                    5
Zengel                       2
Zodiac                       1
Zorn                         1
Zito                         1
ZACH VANS                    1
Zilz                         1
Ziermann                     1
Zielke                       1
Zidek                        1
Zhuhai Helicopter Co.        1
Zukowski                     1
Z-HI-MAX                     1
Zemp                         1
Zeigler                      1
Zeidman                      1
Zeidler                      1
Zdybel                       1
Zbacnick                     1
Zawada                       1
Zaverton Elmer               1
Zappia Gabriel J             1
Zank                         1
Zacharius                    1
ZWICKER MURRAY R             1
ZUBER THOMAS P               1
ZUBAIR S KHAN                1
ZS DELTA-BIELSKO WROCLAW     1
ZEILER                       1
Zwart                        1
Name: count, dtype: int64
```

In [223]:

```
df.loc[df['Make'].str.lower().str.startswith('zenai', na=False), 'Make'] = 'Zenair'
df.loc[df['Make'].str.lower().str.startswith('zeni', na=False), 'Make'] = 'Zenith'
df.loc[df['Make'].str.lower().str.startswith('zimmerm', na=False), 'Make'] = 'Zimmerman'
df.loc[df['Make'].str.lower().str.startswith('zivk', na=False), 'Make'] = 'Zivko Aeronautics'
df.loc[df['Make'].str.lower().str.startswith('zli', na=False), 'Make'] = 'Zlin'
```

In [224]:

```
df.loc[df['Make'].str.lower().str.startswith('aero vodo', na=False), 'Make'] = 'Aero Vodochody'
df.loc[df['Make'].str.lower().str.startswith('aeromot', na=False), 'Make'] = 'Aeromot'
df.loc[df['Make'].str.lower().str.startswith('aeropro', na=False), 'Make'] = 'Aeropro CZ'
df.loc[df['Make'].str.lower().str.startswith('aerostar', na=False), 'Make'] = 'Aerostar'
df.loc[df['Make'].str.lower().str.startswith('aerotek', na=False), 'Make'] = 'Aerotek'
df.loc[df['Make'].str.lower().str.startswith('air cre', na=False), 'Make'] = 'Air Creation'
df.loc[df['Make'].str.lower().str.startswith('aircraft mfg', na=False), 'Make'] = 'Aircraft Mfg and Dev'
df.loc[df['Make'].str.lower().str.startswith('alon', na=False), 'Make'] = 'Alon'
df.loc[df['Make'].str.lower().str.startswith('amateur b', na=False), 'Make'] = 'Amateur Built'
df.loc[df['Make'].str.lower().str.startswith('atr', na=False), 'Make'] = 'ATR'
df.loc[df['Make'].str.lower().str.startswith('autogyr', na=False), 'Make'] = 'AutoGyro'
df.loc[df['Make'].str.lower().str.startswith('avid', na=False), 'Make'] = 'Avid'
df.loc[df['Make'].str.lower().str.startswith('balloon w', na=False), 'Make'] = 'Balloon
```

```
Works'
df.loc[df['Make'].str.lower().str.startswith('brantl', na=False), 'Make'] = 'Brantly'
df.loc[df['Make'].str.lower().str.startswith('british ae', na=False), 'Make'] = 'British Aerospace'
df.loc[df['Make'].str.lower().str.startswith('britten', na=False), 'Make'] = 'Britten Norman'
df.loc[df['Make'].str.lower().str.startswith('buckeye', na=False), 'Make'] = 'Buckeye'
df.loc[df['Make'].str.lower().str.startswith('burkhart', na=False), 'Make'] = 'Burkhart Grob'
df.loc[df['Make'].str.lower().str.startswith('canadair', na=False), 'Make'] = 'Canadair'
df.loc[df['Make'].str.lower().str.startswith('cassutt', na=False), 'Make'] = 'Cassutt'
df.loc[df['Make'].str.lower().str.startswith('cgs', na=False), 'Make'] = 'CGS Aviation'
df.loc[df['Make'].str.lower().str.startswith('classic airc', na=False), 'Make'] = 'Classic Aircraft Corp'
df.loc[df['Make'].str.lower().str.startswith('continental', na=False), 'Make'] = 'Continental Copters'
df.loc[df['Make'].str.lower().str.startswith('convair', na=False), 'Make'] = 'Convair'
df.loc[df['Make'].str.lower().str.startswith('cosmos', na=False), 'Make'] = 'Cosmos'
df.loc[df['Make'].str.lower().str.startswith('costruzioni', na=False), 'Make'] = 'Costruzioni Aeronautiche Tecna'
df.loc[df['Make'].str.lower().str.startswith('curtis', na=False), 'Make'] = 'Curtiss-Wright'
df.loc[df['Make'].str.lower().str.startswith('czech a', na=False), 'Make'] = 'Czech Aircraft Works'
df.loc[df['Make'].str.lower().str.startswith('czech s', na=False), 'Make'] = 'Czech Sport Aircraft'
df.loc[df['Make'].str.lower().str.startswith('downer', na=False), 'Make'] = 'Downer Aircraft Industries'
df.loc[df['Make'].str.lower().str.startswith('pipistrel', na=False), 'Make'] = 'Pipistrel'
df.loc[df['Make'].str.lower().str.startswith('socata', na=False), 'Make'] = 'Socata'
df.loc[df['Make'].str.lower().str.startswith('sonex', na=False), 'Make'] = 'Sonex'
df.loc[df['Make'].str.lower().str.startswith('stearm', na=False), 'Make'] = 'Stearman Aircraft'
df.loc[df['Make'].str.lower().str.startswith('steen', na=False), 'Make'] = 'Steen'
df.loc[df['Make'].str.lower().str.startswith('stemme', na=False), 'Make'] = 'Stemme'
df.loc[df['Make'].str.lower().str.startswith('stinson', na=False), 'Make'] = 'Stinson'
df.loc[df['Make'].str.lower().str.startswith('sukh', na=False), 'Make'] = 'Sukhoi'
df.loc[df['Make'].str.lower().str.startswith('swearing', na=False), 'Make'] = 'Swearingen'
```

In [225]:

```
df['Make'].value_counts().loc[lambda x : x>50].head(60)
```

Out[225]:

```
Make
Cessna                             26903
Piper                              14818
Beech                               5431
Bell                                2792
Boeing                              2733
Robinson Helicopter                 1675
Grumman                             1632
Mooney                              1373
Bellanca                            1037
Hughes Helicopters                   941
Air Tractor                          911
Schweizer                            802
Aeronca                              633
Maule                                589
Mcdonnell Douglas                    579
Champion                             517
de Havilland                         512
Airbus                               507
Cirrus Design                        467
Rockwell                             437
Stinson                              433
Aero Commander                       425
Luscombe                             412
Aerospatiale                         395
North American
North American                     392
Taylorcraft                        389
Hiller                             361
Enstrom                            303
Eurocopter                         295
Ayres                              286
Douglas                            284
Aviat                              259
Embraer                            257
Ercoupe                            247
Sikorsky                           232
Gulfstream                         226
Fairchild                          221
Vans                               172
Swearingen                         171
Bombardier                         169
Pitts                              161
Lake                               152
Waco                               152
Learjet                            145
Schleicher                         144
Balloon Works                      144
Mitsubishi                         142
Let                                139
Burkhart Grob                      137
Smith                              134
Aerostar                           134
McDonnell Douglas Helicopters      126
Socata                             126
Ryan                               125
Lockheed                           122
American                           121
Raytheon                           120
Helio                              116
WSK                                113
Diamond                            108
Name: count, dtype: int64
```

In [226]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 87951 entries, 0 to 88888
Data columns (total 27 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Event_Id                87951 non-null  object 
 1   Investigation_Type      87951 non-null  object 
 2   Accident_Number         87951 non-null  object 
 3   Event_Date              87951 non-null  object 
 4   Location                87951 non-null  object 
 5   Country                 87729 non-null  object 
 6   Airport_Code            49484 non-null  object 
 7   Airport_Name            52031 non-null  object 
 8   Injury_Severity         86961 non-null  object 
 9   Aircraft_damage         84848 non-null  object 
 10  Aircraft_Category       87951 non-null  object 
 11  Registration_Number     86601 non-null  object 
 12  Make                    87951 non-null  object 
 13  Model                   87859 non-null  object 
 14  Amateur_Built           87851 non-null  object 
 15  Number_of_Engines       81924 non-null  float64
 16  Engine_Type             87951 non-null  object 
 17  FAR_Description         31915 non-null  object 
 18  Purpose_of_flight       87951 non-null  object 
 19  Total_Fatal_Injuries    87951 non-null  float64
 20  Total_Serious_Injuries  87951 non-null  float64
 21  Total_Minor_Injuries    87951 non-null  float64
 22  Total_Uninjured         87951 non-null  float64
 23  Weather_Condition       87951 non-null  object 
 24  Broad_phase_of_flight   87951 non-null  object 
 25  Report_Status           81587 non-null  object 
 25   Report_Status           81587 non-null   object
 26   Publication_Date        74352 non-null   object
dtypes: float64(5), object(22)
memory usage: 18.8+ MB
```

The airport columns and FAR Description are quite empty and not useful to the intended analysis, so they can be removed

In [227]:

```
# Remove Columns labelled Airport_Code, Airport_Name, FAR_Description
df = df.drop(['Airport_Code', 'Airport_Name', 'FAR_Description'], axis=1)

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 87951 entries, 0 to 88888
Data columns (total 24 columns):
 #    Column                  Non-Null Count   Dtype
---   ------                  --------------   -----
 0    Event_Id                87951 non-null   object
 1    Investigation_Type      87951 non-null   object
 2    Accident_Number         87951 non-null   object
 3    Event_Date              87951 non-null   object
 4    Location                87951 non-null   object
 5    Country                 87729 non-null   object
 6    Injury_Severity         86961 non-null   object
 7    Aircraft_damage         84848 non-null   object
 8    Aircraft_Category       87951 non-null   object
 9    Registration_Number     86601 non-null   object
 10   Make                    87951 non-null   object
 11   Model                   87859 non-null   object
 12   Amateur_Built           87851 non-null   object
 13   Number_of_Engines       81924 non-null   float64
 14   Engine_Type             87951 non-null   object
 15   Purpose_of_flight       87951 non-null   object
 16   Total_Fatal_Injuries    87951 non-null   float64
 17   Total_Serious_Injuries  87951 non-null   float64
 18   Total_Minor_Injuries    87951 non-null   float64
 19   Total_Uninjured         87951 non-null   float64
 20   Weather_Condition       87951 non-null   object
 21   Broad_phase_of_flight   87951 non-null   object
 22   Report_Status           81587 non-null   object
 23   Publication_Date        74352 non-null   object
dtypes: float64(5), object(19)
memory usage: 16.8+ MB
```

In [228]:

```
# Aircraft_damage value_counts
df['Aircraft_damage'].value_counts(dropna=False)
```

Out[228]:

```
Aircraft_damage
Substantial    63641
Destroyed      18402
NaN             3103
Minor           2686
Unknown          119
Name: count, dtype: int64
```

In [229]:

```
# Fill in NaN values in Aircraft_damage with Unknown
df['Aircraft_damage'] = df['Aircraft_damage'].fillna('Unknown')

df['Aircraft_damage'].value_counts(dropna=False)
```

Out[229]:

```
Aircraft_damage
```

```
Substantial    63641
Destroyed      18402
Unknown         3222
Minor           2686
Name: count, dtype: int64
```

In [230]:

```
df['Aircraft_Category'].value_counts(dropna=False)
```

Out[230]:

```
Aircraft_Category
Airplane             72284
Helicopter            8644
Unknown               5039
Glider                 824
Balloon                624
Gyrocraft              208
Weight-Shift           170
Powered Parachute       90
Ultralight              59
Blimp                    4
UNK                      2
Powered-Lift             1
Rocket                   1
ULTR                     1
Name: count, dtype: int64
```

In [231]:

```
# Fill UNK with Unknown in Category column
df['Aircraft_Category'] = df['Aircraft_Category'].replace('UNK', 'Unknown')
```

In [232]:

```
# Fill ULTR with Ultralight in Category column
df['Aircraft_Category'] = df['Aircraft_Category'].replace('ULTR', 'Ultralight')
```

In [233]:

```
df['Aircraft_Category'].value_counts(dropna=False)
```

Out[233]:

```
Aircraft_Category
Airplane             72284
Helicopter            8644
Unknown               5041
Glider                 824
Balloon                624
Gyrocraft              208
Weight-Shift           170
Powered Parachute       90
Ultralight              60
Blimp                    4
Powered-Lift             1
Rocket                   1
Name: count, dtype: int64
```

In [234]:

```
df['Injury_Severity'].value_counts(dropna=False)
```

Out[234]:

```
Injury_Severity
Non-Fatal     66822
Fatal(1)       6086
Fatal          5257
Fatal(2)       3632
Incident       2113
               ...
```

```
...
Fatal(33)         1
Fatal(123)        1
Fatal(72)         1
Fatal(54)         1
Fatal(189)        1
Name: count, Length: 110, dtype: int64
```

In [235]:

```
# Show NaN count in Injury_Severity column
df['Injury_Severity'].isna().sum()
```

Out[235]:

```
990
```

In [236]:

```
# Fill in NaN values in Injury_Severity with Unknown
df['Injury_Severity'] = df['Injury_Severity'].fillna('Unknown')
```

In [237]:

```
# replace values starting with 'Fata' with 'Fatal' since the number of fatalities is alre
ady recorded in another column
df.loc[df['Injury_Severity'].str.startswith('Fata'), 'Injury_Severity'] = 'Fatal'
```

In [238]:

```
df['Injury_Severity'].value_counts(dropna=False)
```

Out[238]:

```
Injury_Severity
Non-Fatal      66822
Fatal          17540
Incident        2113
Unknown          990
Minor            217
Serious          173
Unavailable       96
Name: count, dtype: int64
```

In [239]:

```
# Fill Unavailable with Unknown in Injury_Severity column
df['Injury_Severity'] = df['Injury_Severity'].replace('Unavailable', 'Unknown')
```

In [240]:

```
df['Injury_Severity'].value_counts(dropna=False)
```

Out[240]:

```
Injury_Severity
Non-Fatal    66822
Fatal        17540
Incident      2113
Unknown       1086
Minor          217
Serious        173
Name: count, dtype: int64
```

In [241]:

```
df['Amateur_Built'].value_counts(dropna=False)
```

Out[241]:

```
Amateur_Built
No     79431
Yes     8420
NaN      100
NaN        100
Name: count, dtype: int64
```

In [242]:

```
# Fill in NaN values in Amateur_Built with Unknown
df['Amateur_Built'] = df['Amateur_Built'].fillna('Unknown')
```

In [243]:

```
# Show NaN count in Report_Status column
df['Report_Status'].isna().sum()
```

Out[243]:

```
6364
```

In [244]:

```
# Fill in NaN values in Amateur_Built with Unknown
df['Report_Status'] = df['Report_Status'].fillna('Unknown')
```

In [245]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 87951 entries, 0 to 88888
Data columns (total 24 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Event_Id               87951 non-null  object 
 1   Investigation_Type     87951 non-null  object 
 2   Accident_Number        87951 non-null  object 
 3   Event_Date             87951 non-null  object 
 4   Location               87951 non-null  object 
 5   Country                87729 non-null  object 
 6   Injury_Severity        87951 non-null  object 
 7   Aircraft_damage        87951 non-null  object 
 8   Aircraft_Category      87951 non-null  object 
 9   Registration_Number    86601 non-null  object 
 10  Make                   87951 non-null  object 
 11  Model                  87859 non-null  object 
 12  Amateur_Built          87951 non-null  object 
 13  Number_of_Engines      81924 non-null  float64
 14  Engine_Type            87951 non-null  object 
 15  Purpose_of_flight      87951 non-null  object 
 16  Total_Fatal_Injuries   87951 non-null  float64
 17  Total_Serious_Injuries 87951 non-null  float64
 18  Total_Minor_Injuries   87951 non-null  float64
 19  Total_Uninjured        87951 non-null  float64
 20  Weather_Condition      87951 non-null  object 
 21  Broad_phase_of_flight  87951 non-null  object 
 22  Report_Status          87951 non-null  object 
 23  Publication_Date       74352 non-null  object 
dtypes: float64(5), object(19)
memory usage: 16.8+ MB
```

In [246]:

```
# Show NaN count in Report_Status column
df['Country'].isna().sum()
```

Out[246]:

```
222
```

In [247]:

```
# Fill in NaN values in Country with Unknown
df['Country'] = df['Country'].fillna('Unknown')
```

In [248]:

```
# Fill in NaN values in Registration_Number with Unknown
df['Registration_Number'] = df['Registration_Number'].fillna('Unknown')
```

In [249]:

```
# Fill in NaN values in Model with Unknown
df['Model'] = df['Model'].fillna('Unknown')
```

In [250]:

```
# Fill in NaN values in Number_of_Engines with Unknown
df['Number_of_Engines'] = df['Number_of_Engines'].fillna('Unknown')
```

In [251]:

```
# Fill in NaN values in Publication_Date with Unknown
df['Publication_Date'] = df['Publication_Date'].fillna('Unknown')
```

In [252]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 87951 entries, 0 to 88888
Data columns (total 24 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Event_Id                87951 non-null  object 
 1   Investigation_Type      87951 non-null  object 
 2   Accident_Number         87951 non-null  object 
 3   Event_Date              87951 non-null  object 
 4   Location                87951 non-null  object 
 5   Country                 87951 non-null  object 
 6   Injury_Severity         87951 non-null  object 
 7   Aircraft_damage         87951 non-null  object 
 8   Aircraft_Category       87951 non-null  object 
 9   Registration_Number     87951 non-null  object 
 10  Make                    87951 non-null  object 
 11  Model                   87951 non-null  object 
 12  Amateur_Built           87951 non-null  object 
 13  Number_of_Engines       87951 non-null  object 
 14  Engine_Type             87951 non-null  object 
 15  Purpose_of_flight       87951 non-null  object 
 16  Total_Fatal_Injuries    87951 non-null  float64
 17  Total_Serious_Injuries  87951 non-null  float64
 18  Total_Minor_Injuries    87951 non-null  float64
 19  Total_Uninjured         87951 non-null  float64
 20  Weather_Condition       87951 non-null  object 
 21  Broad_phase_of_flight   87951 non-null  object 
 22  Report_Status           87951 non-null  object 
 23  Publication_Date        87951 non-null  object 
dtypes: float64(4), object(20)
memory usage: 16.8+ MB
```

At this point we've filled in all the columns with valid values or "Unknown" if the values were not capable of being filled in.

In [253]:

```
# Export df as a separate file for Tableau visualizations
df.to_csv('working-df/cleaned_aviation_data_complete.csv', index=False)
```

Aircraft Damage Levels

I'd like to create some numbers, percentages, and charts to explore the Aircraft Damage levels related to Injury Levels

In [254]:

```
# For airplane incidents, how many were destroyed, had minor damage or substantial damage
incidents_airplane = df[df['Aircraft_Category'] == 'Airplane']
incidents_airplane['Aircraft_damage'].value_counts()
```

Out[254]:

```
Aircraft_damage
Substantial    52692
Destroyed      14596
Unknown         2652
Minor           2344
Name: count, dtype: int64
```

In [255]:

```
# Sums of different injury categories for airplanes
fatalities_airplane = incidents_airplane['Total_Fatal_Injuries'].sum()
serious_injury_airplane = incidents_airplane['Total_Serious_Injuries'].sum()
minor_injury_airplane = incidents_airplane['Total_Minor_Injuries'].sum()
no_injury_airplane = incidents_airplane['Total_Uninjured'].sum()
airplane_people_total = fatalities_airplane + serious_injury_airplane + minor_injury_airp
lane + no_injury_airplane

airplane_people_total
```

Out[255]:

```
479844.0
```

In [256]:

```
substantial_damage_airplane = incidents_airplane[incidents_airplane['Aircraft_damage'] ==
'Substantial'].shape[0]
minor_damage_airplane = incidents_airplane[incidents_airplane['Aircraft_damage'] == 'Mino
r'].shape[0]
destroyed_airplane = incidents_airplane[incidents_airplane['Aircraft_damage'] == 'Destroy
ed'].shape[0]
```

In [257]:

```
#fatalities in the damage subsets
fatalities_substantial_damage_airplane = incidents_airplane[incidents_airplane['Aircraft_
damage'] == 'Substantial']['Total_Fatal_Injuries'].sum()
fatalities_minor_damage_airplane = incidents_airplane[incidents_airplane['Aircraft_damage
'] == 'Minor']['Total_Fatal_Injuries'].sum()
fatalities_destroyed_airplane = incidents_airplane[incidents_airplane['Aircraft_damage']
== 'Destroyed']['Total_Fatal_Injuries'].sum()
```

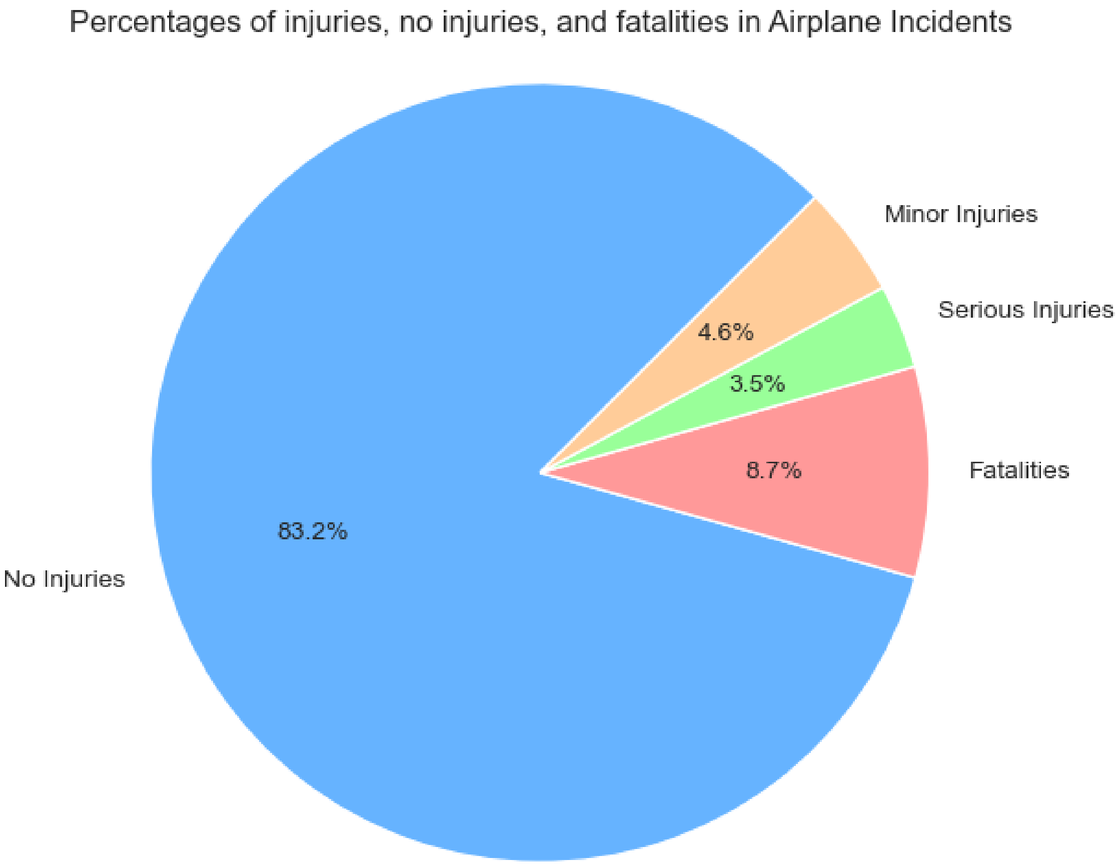
In [258]:

```
# what are the percentages of incidents_airplane injury levels
no_injury_airplane_percent = no_injury_airplane / airplane_people_total * 100
fatalities_airplane_percent = fatalities_airplane / airplane_people_total * 100
serious_injury_airplane_percent = serious_injury_airplane / airplane_people_total * 100
minor_injury_airplane_percent = minor_injury_airplane / airplane_people_total * 100
```

In [259]:

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Create pie chart with injury percentages
labels = ['No Injuries', 'Fatalities', 'Serious Injuries', 'Minor Injuries']
sizes = [no_injury_airplane_percent, fatalities_airplane_percent, serious_injury_airplane
_percent, minor_injury_airplane_percent]
colors = ['#66b3ff', '#ff9999', '#99ff99', '#ffcc99']
sns.set_style("whitegrid")
plt.figure(figsize=(6,6))
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', shadow=False, startangle
=45)
plt.axis('equal')
plt.title('Percentages of injuries, no injuries, and fatalities in Airplane Incidents')
plt.show()
```



Make the same graph for helicopters

In [260]:

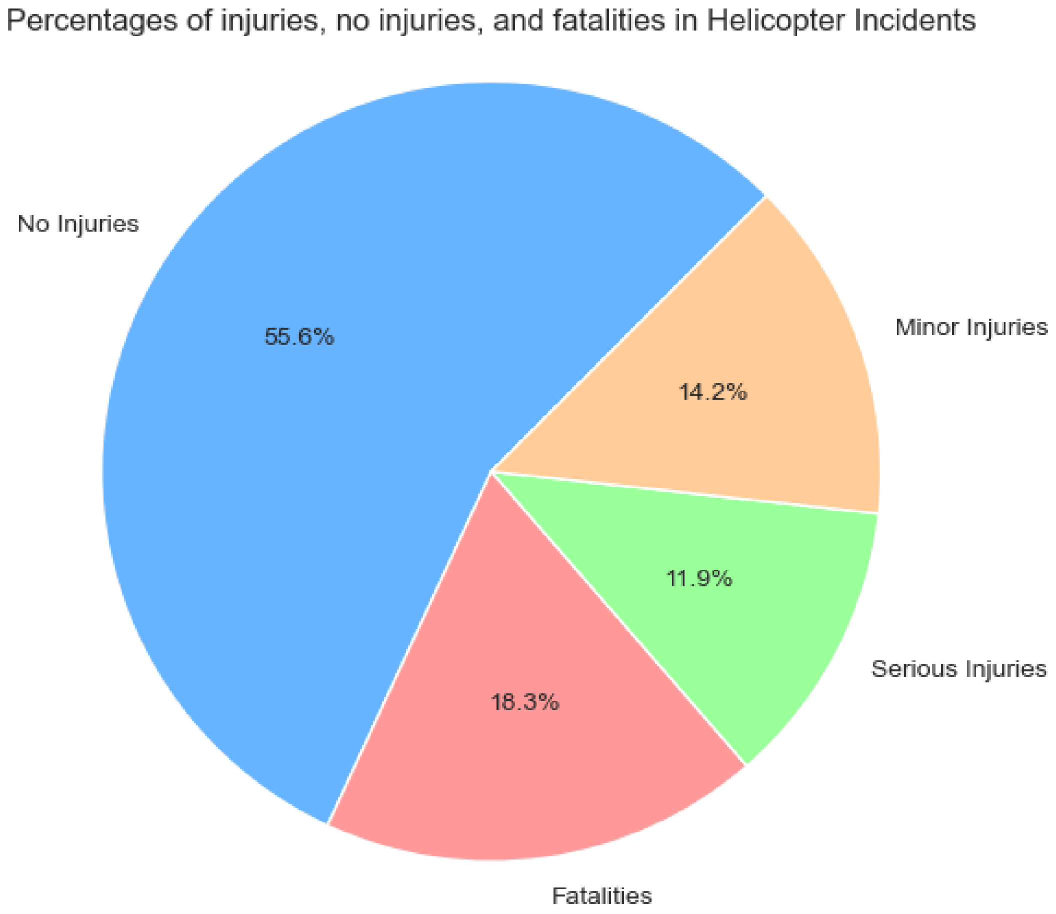
```
incidents_helicopter = df[df['Aircraft_Category'] == 'Helicopter']
# Sums of different injury categories for airplanes
fatalities_helicopter = incidents_helicopter['Total_Fatal_Injuries'].sum()
serious_injury_helicopter = incidents_helicopter['Total_Serious_Injuries'].sum()
minor_injury_helicopter = incidents_helicopter['Total_Minor_Injuries'].sum()
no_injury_helicopter = incidents_helicopter['Total_Uninjured'].sum()
helicopter_people_total = fatalities_helicopter + serious_injury_helicopter + minor_injur
y_helicopter + no_injury_helicopter

substantial_damage_helicopter = incidents_helicopter[incidents_helicopter['Aircraft_damag
e'] == 'Substantial'].shape[0]
minor_damage_helicopter = incidents_helicopter[incidents_helicopter['Aircraft_damage'] ==
'Minor'].shape[0]
destroyed_helicopter = incidents_helicopter[incidents_helicopter['Aircraft_damage'] == '
Destroyed'].shape[0]

#fatalities in the damage subsets
fatalities_substantial_damage_helicopter = incidents_helicopter[incidents_helicopter['Air
craft_damage'] == 'Substantial']['Total_Fatal_Injuries'].sum()
fatalities_minor_damage_helicopter = incidents_helicopter[incidents_helicopter['Aircraft_
damage'] == 'Minor']['Total_Fatal_Injuries'].sum()
fatalities_destroyed_helicopter = incidents_helicopter[incidents_helicopter['Aircraft_dam
age'] == 'Destroyed']['Total_Fatal_Injuries'].sum()

# percentages of incidents_helicopter in the injury column
no_injury_helicopter_percent = no_injury_helicopter / helicopter_people_total * 100
fatalities_helicopter_percent = fatalities_helicopter / helicopter_people_total * 100
serious_injury_helicopter_percent = serious_injury_helicopter / helicopter_people_total *
100
minor_injury_helicopter_percent = minor_injury_helicopter / helicopter_people_total * 100
```

```
# Create pie chart with injury percentages
labels = ['No Injuries', 'Fatalities', 'Serious Injuries', 'Minor Injuries']
sizes = [no_injury_helicopter_percent, fatalities_helicopter_percent, serious_injury_heli
copter_percent, minor_injury_helicopter_percent]
colors = ['#66b3ff', '#ff9999', '#99ff99', '#ffcc99']
sns.set_style("whitegrid")
plt.figure(figsize=(6,6))
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', shadow=False, startangle
=45)
plt.axis('equal')
plt.title('Percentages of injuries, no injuries, and fatalities in Helicopter Incidents')
plt.show()
```



Report Status

Even though the Report Status column is mostly empty, I feel that the data that does exist there may be interesting and would like to see if it can be sorted, cleaned, and used somehow.

In [261]:

```
df['Report_Status'].value_counts()
```

Out[261]:

```
Report_Status
Probable Cause
60867
Unknown
6364
Foreign
1974
<br /><br />
167
Factual
145

...
The pilot's incapacitation due to a ruptured berry aneurysm during takeoff.
```

```
1
The unauthorized operation of the helicopter by a non-certificated and unqualified indivi
dual who failed to maintain helicopter control.                                        1
A loss of engine power due to the pilot's failure to utilize carburetor heat while maneuv
ering.\r\n.                                                                             1
The pilot's failure to maintain adequate separation behind a corporate jet, which resulte
d in an encounter with wake turbulence and a subsequent loss of control.               1
The pilots loss of control due to a wind gust during landing.
1
Name: count, Length: 17075, dtype: int64
```

In [262]:

```
# create a subset of rows that is called informative_report that removes probable cause,
unknown, foreign, factual, and any other non-useful values

informative_report = df[df['Report_Status'] != 'Probable Cause']
informative_report = informative_report[informative_report['Report_Status'] != 'Unknown']
informative_report = informative_report[informative_report['Report_Status'] != 'Foreign']
informative_report = informative_report[informative_report['Report_Status'] != '<br /><br
/>']
informative_report = informative_report[informative_report['Report_Status'] != 'Factual']
informative_report = informative_report[informative_report['Report_Status'] != 'None.']
informative_report = informative_report[informative_report['Report_Status'] != '.']
informative_report = informative_report[informative_report['Report_Status'] != 'Prelimina
ry']
informative_report = informative_report[informative_report['Report_Status'] != 'Undetermi
ned.']

informative_report['Report_Status'].info()
```

```
<class 'pandas.core.series.Series'>
Index: 18380 entries, 63913 to 88767
Series name: Report_Status
Non-Null Count  Dtype
--------------  -----
18380 non-null  object
dtypes: object(1)
memory usage: 287.2+ KB
```

In [263]:

```
# In informative_report, replace "pilots" with "pilot's"
informative_report['Report_Status'] = informative_report['Report_Status'].str.replace('pi
lots', "pilot's")

informative_report['Report_Status'].value_counts()
```

Out[263]:

```
Report_Status
The pilot's failure to maintain directional control during the landing roll.
75
The pilot's failure to maintain directional control during landing.
60
A loss of engine power for undetermined reasons.
52
A total loss of engine power for undetermined reasons.
39
The loss of engine power for undetermined reasons.
29


..
The pilot's inadvertent encounter with severe weather, which resulted in the airplanes le
ft wing failing in positive overload. Contributing to the accident was the pilot's relian
ce on outdated weather information that he received on his in-cockpit Next-Generation Rad
ar (NEXRAD).                    1
The pilot's controlled flight into terrain while maneuvering at a low altitude in instrum
ent meteorological conditions. Contributing to the accident was the pilot's decision to p
erform a circling maneuver in weather below circling minimums instead of flying the misse
d approach instructions.     1
The airplanes encounter with unforecasted severe icing conditions that were characterized
by high ice accretion rates and the pilot's failure to use his command authority to depar
by high ice accretion rates and the pilot's failure to use his command authority to depar
t the icing conditions in an expeditious manner, which resulted in a loss of airplane con
trol.                          1
The pilot's improper decision to continue a visual flight into instrument meteorological
conditions, which resulted in a wire strike.
1
The pilot's loss of control due to a wind gust during landing.
1
Name: count, Length: 16939, dtype: int64
```

In [264]:

```
# create subset of rows named pilot_error that contain the word "pilot's" in the Report_S
tatus column
pilot_error = informative_report[informative_report['Report_Status'].str.contains("pilot'
s")]

pilot_error['Report_Status'].info()
```

```
<class 'pandas.core.series.Series'>
Index: 12414 entries, 63913 to 88767
Series name: Report_Status
Non-Null Count  Dtype
--------------  -----
12414 non-null  object
dtypes: object(1)
memory usage: 194.0+ KB
```

In [265]:

```
# What percentage of all the records are pilot_error
pilot_error.shape[0] / df.shape[0] * 100
```

Out[265]:

```
14.114677490875602
```

In [266]:

```
# non_pilot_report is informative_report without the pilot_error results
non_pilot_report = informative_report[~informative_report.index.isin(pilot_error.index)]

non_pilot_report['Report_Status'].info()
```

```
<class 'pandas.core.series.Series'>
Index: 5966 entries, 63917 to 88661
Series name: Report_Status
Non-Null Count  Dtype
--------------  -----
5966 non-null   object
dtypes: object(1)
memory usage: 93.2+ KB
```

In [267]:

```
# What percentage of all the records are non_pilot_error
non_pilot_report.shape[0] / df.shape[0] * 100
```

Out[267]:

```
6.783322531864333
```

In [268]:

```
# What percentage of the informative records are non_pilot_error and pilot_error
print(non_pilot_report.shape[0] / informative_report.shape[0] * 100)
print(pilot_error.shape[0] / informative_report.shape[0] * 100)
```

```
32.45919477693145
67.54080522306856
```

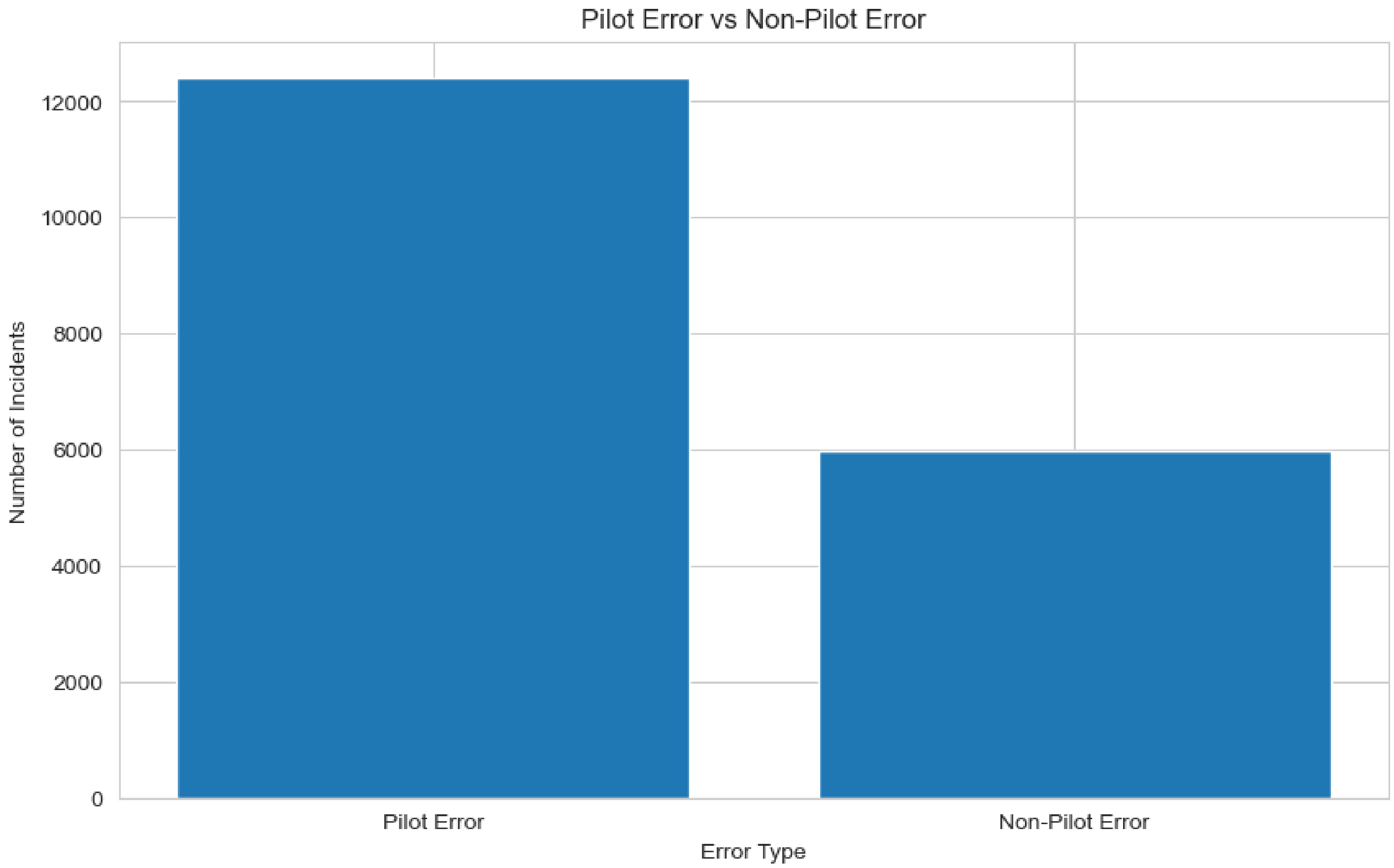
Report Status This previous section demonstrates that the vast majority of the report status column is not informative, having values such as "Probable Cause", "Foreign", and "Unknown". About 14% of the records

informative, having values such as "Probable Cause", "Foreign", and "Unknown". About 14% of the records (12,414) indicate pilot error as the main cause of the incident. Another 6.8% (5,966) contain a variety of causes for the incident, most of which point to mechanical or equipment issues.

So of these 18,380 informative values for Report Status, almost 68% are attributed to pilot error and about 32.5% attributed to various mechanical or equipment failures, many due to undetermined causes and some caused by human error in maintenance of equipment.

In [269]:

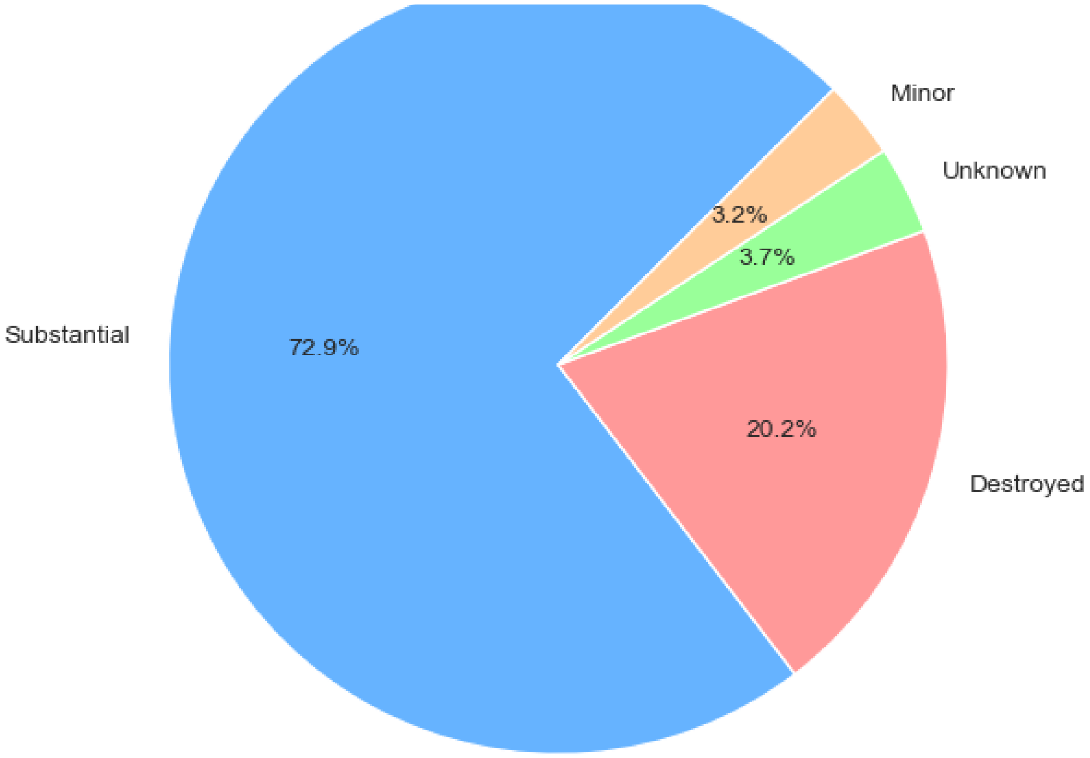
```
# create bar chart for pilot_error.shape and non_pilot_error.shape
plt.figure(figsize=(10, 6))
plt.bar(['Pilot Error', 'Non-Pilot Error'], [pilot_error.shape[0], non_pilot_report.shap
e[0]])
plt.title('Pilot Error vs Non-Pilot Error')
plt.xlabel('Error Type')
plt.ylabel('Number of Incidents')
plt.show()
```



Create two charts showing the damage percentage of planes and helicopters.

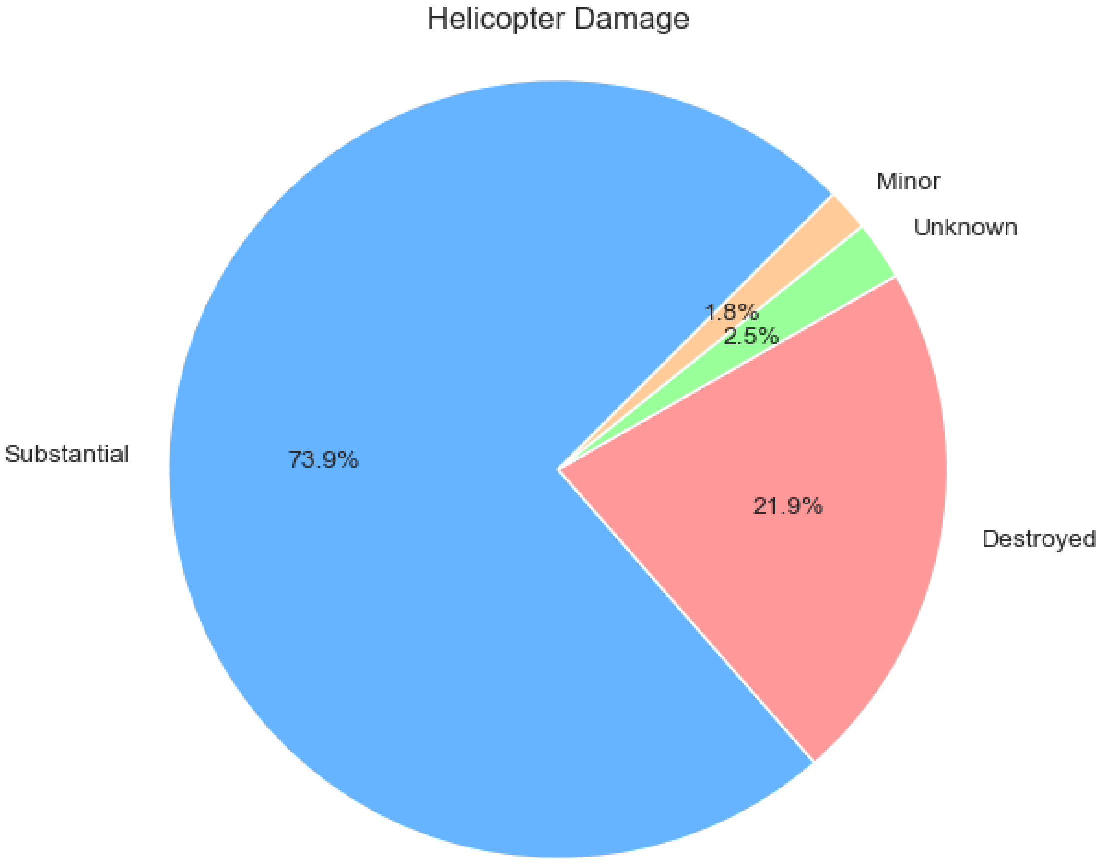
In [270]:

```
# airplane damage percentages
airplane_damage = incidents_airplane['Aircraft_damage'].value_counts()
labels = airplane_damage.index
sizes = airplane_damage
colors = ['#66b3ff', '#ff9999', '#99ff99', '#ffcc99']
explode = (0, 0, 0)
sns.set_style("whitegrid")
plt.figure(figsize=(6,6))
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', shadow=False, startangle
=45)
plt.axis('equal')
plt.title('Airplane Damage')
plt.show()
```



In [271]:

```
# do the same for heli
helicopter_damage = incidents_helicopter['Aircraft_damage'].value_counts()
labels = helicopter_damage.index
sizes = helicopter_damage
colors = ['#66b3ff', '#ff9999', '#99ff99', '#ffcc99']
explode = (0, 0, 0)
sns.set_style("whitegrid")
plt.figure(figsize=(6,6))
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', shadow=False, startangle
=45)
plt.axis('equal')
plt.title('Helicopter Damage')
plt.show()
```


Helicopter Damage
Minor
Unknown
Substantial
Destroyed