bird-strikes-data-analysis

July 21, 2024

1 Bird Strikes Data Analysis

[1]:

```
import pandas as pd #import pandas liberaries
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:

```
data=pd.read_csv(r'C:\Users\amitk\Downloads\Bird Strikes data.csv') # Read the␣
 ↪data
```

[3]:

```
data.head()
```

[3]:

```
   Record ID Aircraft: Type                Airport: Name Aircraft: Make/Model  \
0     202152       Airplane                 LAGUARDIA NY            B-737-400
1     208159       Airplane  DALLAS/FORT WORTH INTL ARPT                MD-80
2     207601       Airplane            LAKEFRONT AIRPORT                C-500
3     215953       Airplane          SEATTLE-TACOMA INTL            B-737-400
4     219878       Airplane                 NORFOLK INTL         CL-RJ100/200

   Wildlife: Number Struck Actual Effect: Impact to flight  Year  \
0                             859         Engine Shut Down  2000
1                             424                     None  2001
2                             261                     None  2001
3                             806    Precautionary Landing  2002
4                             942                     None  2003

  Effect: Indicated Damage Aircraft: Number of engines?  \
0            Caused damage                            2
1            Caused damage                            2
2                No damage                            2
3                No damage                            2
4                No damage                            2

  Aircraft: Airline/Operator  … Remains of wildlife collected?  \
0                US AIRWAYS*  …                          False
1          AMERICAN AIRLINES  …                          False
2                   BUSINESS  …                          False
```

```
3            ALASKA AIRLINES  …                           True
4            COMAIR AIRLINES  …                          False

  Remains of wildlife sent to Smithsonian Wildlife: Size  Conditions: Sky  \
0                                   False         Medium         No Cloud
1                                   False          Small       Some Cloud
2                                   False          Small         No Cloud
3                                   False          Small       Some Cloud
4                                   False          Small         No Cloud

       Wildlife: Species Pilot warned of birds or wildlife? Cost: Total $  \
0  Unknown bird - medium                                  N         30736
1            Rock pigeon                                  Y             0
2      European starling                                  N             0
3      European starling                                  Y             0
4      European starling                                  N             0

  Feet above ground Number of people injured  Is Aircraft Large?
0              1500                        0                 Yes
1                 0                        0                  No
2                50                        0                  No
3                50                        0                 Yes
4                50                        0                  No

[5 rows x 23 columns]
```

```
[4]: data.columns #shown columns
```

```
[4]: Index(['Record ID', 'Aircraft: Type', 'Airport: Name', 'Aircraft: Make/Model',
            'Wildlife: Number Struck Actual', 'Effect: Impact to flight', 'Year',
            'Effect: Indicated Damage', 'Aircraft: Number of engines?',
            'Aircraft: Airline/Operator', 'Origin State', 'When: Phase of flight',
            'Conditions: Precipitation', 'Remains of wildlife collected?',
            'Remains of wildlife sent to Smithsonian', 'Wildlife: Size',
            'Conditions: Sky', 'Wildlife: Species',
            'Pilot warned of birds or wildlife?', 'Cost: Total $',
            'Feet above ground', 'Number of people injured', 'Is Aircraft Large?'],
           dtype='object')
```

```
[5]: data.shape #no. of rows and columns
```

```
[5]: (24747, 23)
```

```
[6]: data.isnull().sum()
```

```
[6]: Record ID                                  0
     Aircraft: Type                             0
```

```
Airport: Name                              0
Aircraft: Make/Model                       0
Wildlife: Number Struck Actual             0
Effect: Impact to flight                   0
Year                                       0
Effect: Indicated Damage                   0
Aircraft: Number of engines?               0
Aircraft: Airline/Operator                 0
Origin State                               0
When: Phase of flight                      0
Conditions: Precipitation                  0
Remains of wildlife collected?             0
Remains of wildlife sent to Smithsonian    0
Wildlife: Size                             0
Conditions: Sky                            0
Wildlife: Species                          0
Pilot warned of birds or wildlife?         0
Cost: Total $                              0
Feet above ground                          0
Number of people injured                   0
Is Aircraft Large?                         0
dtype: int64
```

This data file is cleaned before read in Jupyter Notebook, so no need of cleaning process.

```
[7]: data.nunique() #unique values per columns
```

```
[7]: Record ID                                  24747
     Aircraft: Type                                 1
     Airport: Name                                976
     Aircraft: Make/Model                         279
     Wildlife: Number Struck Actual               105
     Effect: Impact to flight                       5
     Year                                          12
     Effect: Indicated Damage                       2
     Aircraft: Number of engines?                   5
     Aircraft: Airline/Operator                   288
     Origin State                                  60
     When: Phase of flight                          7
     Conditions: Precipitation                      8
     Remains of wildlife collected?                 2
     Remains of wildlife sent to Smithsonian        2
     Wildlife: Size                                 3
     Conditions: Sky                                3
     Wildlife: Species                            340
     Pilot warned of birds or wildlife?             2
     Cost: Total $                                760
```

```
Feet above ground                          252
Number of people injured                     4
Is Aircraft Large?                           2
dtype: int64
```

[8]:

```
data[['Wildlife: Number Struck Actual','Number of people injured']].describe() ␣
 ↪#Statical analysis
```

[8]:

```
       Wildlife: Number Struck Actual  Number of people injured
count                    24747.000000              24747.000000
mean                         2.689255                  0.000849
std                         12.506021                  0.047986
min                          1.000000                  0.000000
25%                          1.000000                  0.000000
50%                          1.000000                  0.000000
75%                          1.000000                  0.000000
max                        942.000000                  6.000000
```

[9]:

```
data[['Feet above ground','Cost: Total $']].describe()  #Statical analysis
```

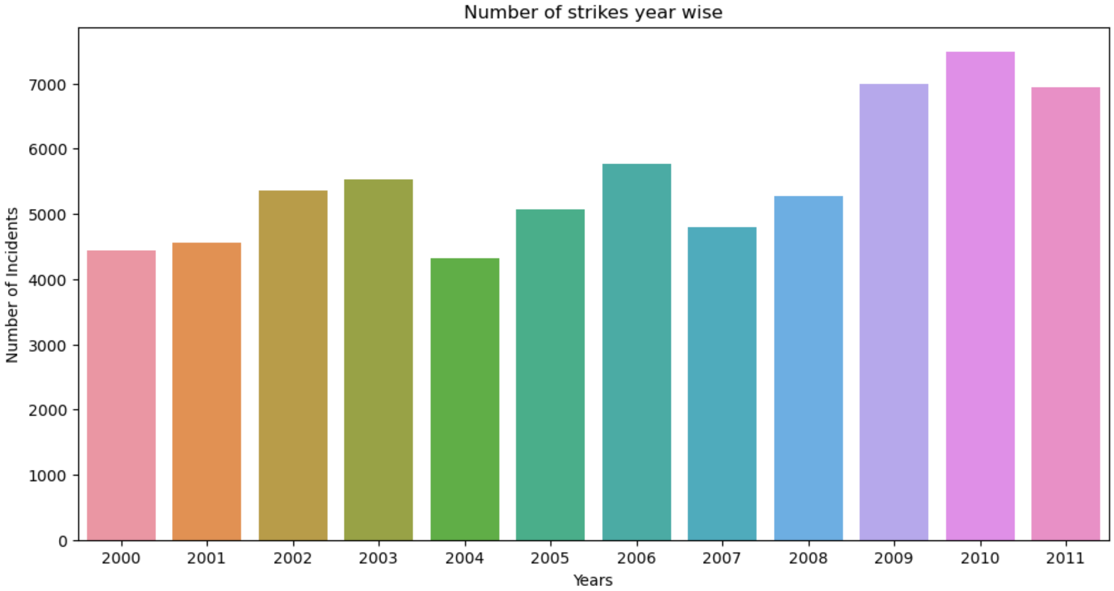
[9]:

```
       Feet above ground  Cost: Total $
count       24747.000000   2.474700e+04
mean          801.538449   5.485157e+03
std          1736.743268   1.231439e+05
min             0.000000   0.000000e+00
25%             0.000000   0.000000e+00
50%            50.000000   0.000000e+00
75%           700.000000   0.000000e+00
max         18000.000000   1.239775e+07
```

2 EDA

[10]:

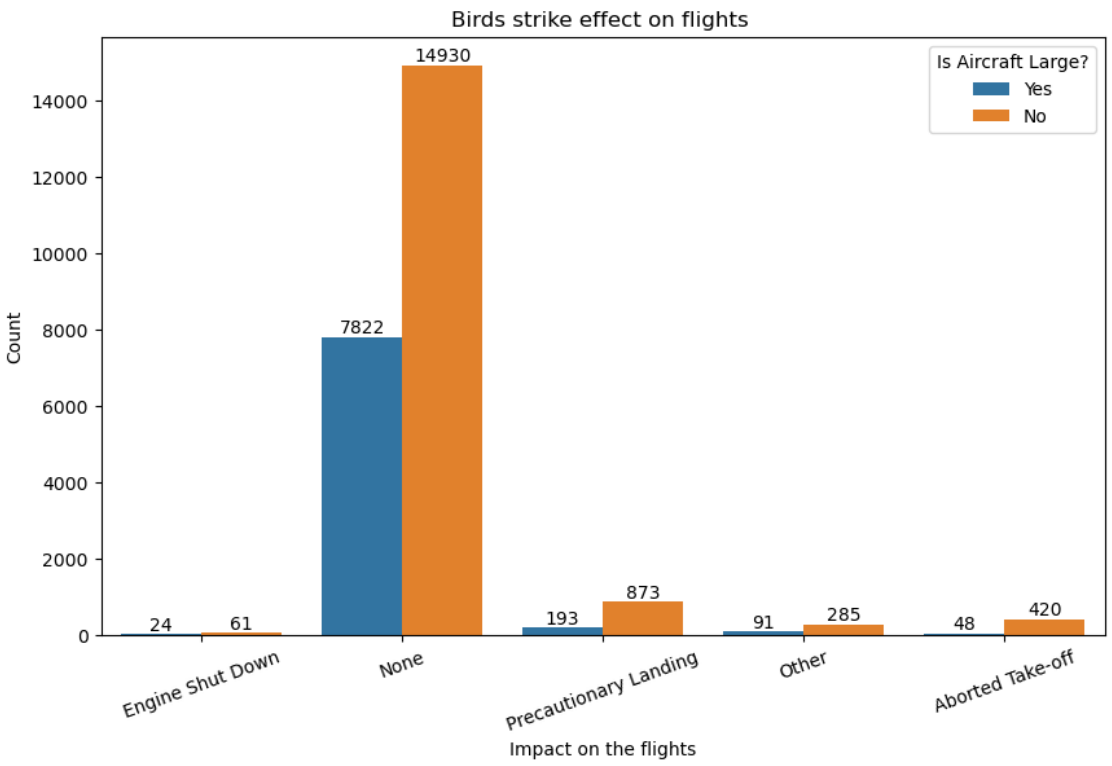
```
#Number of strikes year wise
plt.figure(figsize=(12,6))
strikes=data.groupby(['Year'], as_index=False)['Wildlife: Number Struck␣
 ↪Actual'].sum()
sns.barplot(x='Year', y='Wildlife: Number Struck Actual', data=strikes)
plt.ylabel('Number of Incidents')
plt.xlabel("Years")
plt.title("Number of strikes year wise")
plt.show()
```



Most of the strike events happened in 2010, then 2009 & 2011 respectively.

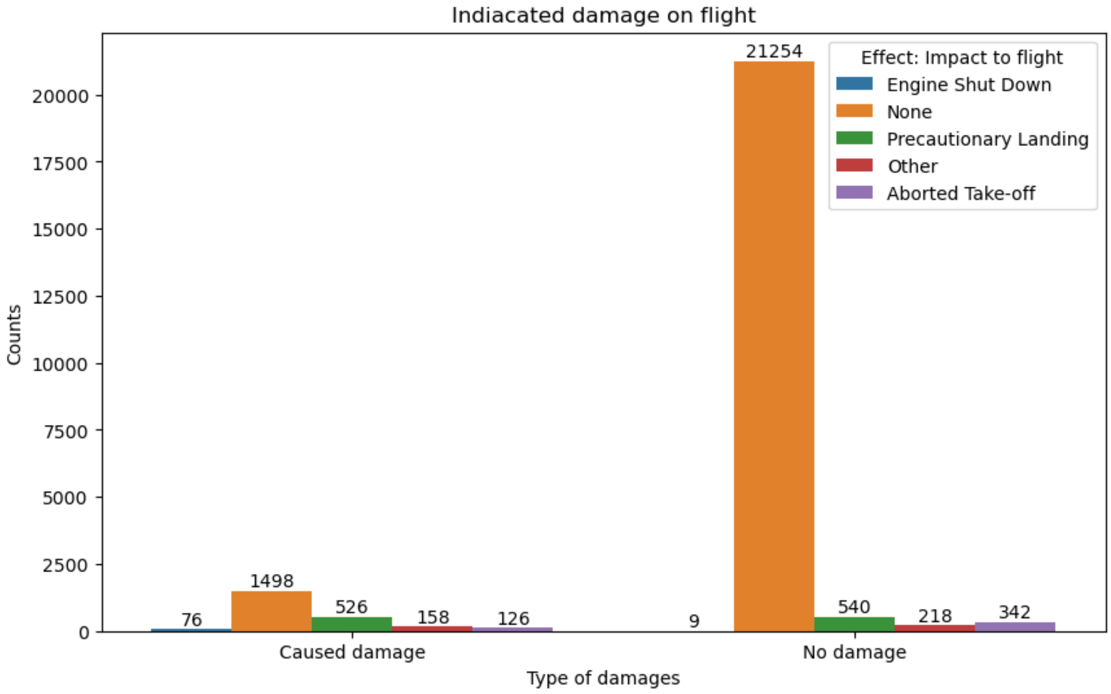
[11]:

```
plt.figure(figsize=(10,6))
effect=sns.countplot(x='Effect: Impact to flight', hue='Is Aircraft Large?',␣
 ↪data=data)
for bars in effect.containers:
    effect.bar_label(bars)
plt.title('Birds strike effect on flights ')
plt.xlabel('Impact on the flights')
plt.ylabel('Count')
plt.xticks(rotation=20)
plt.show()
```

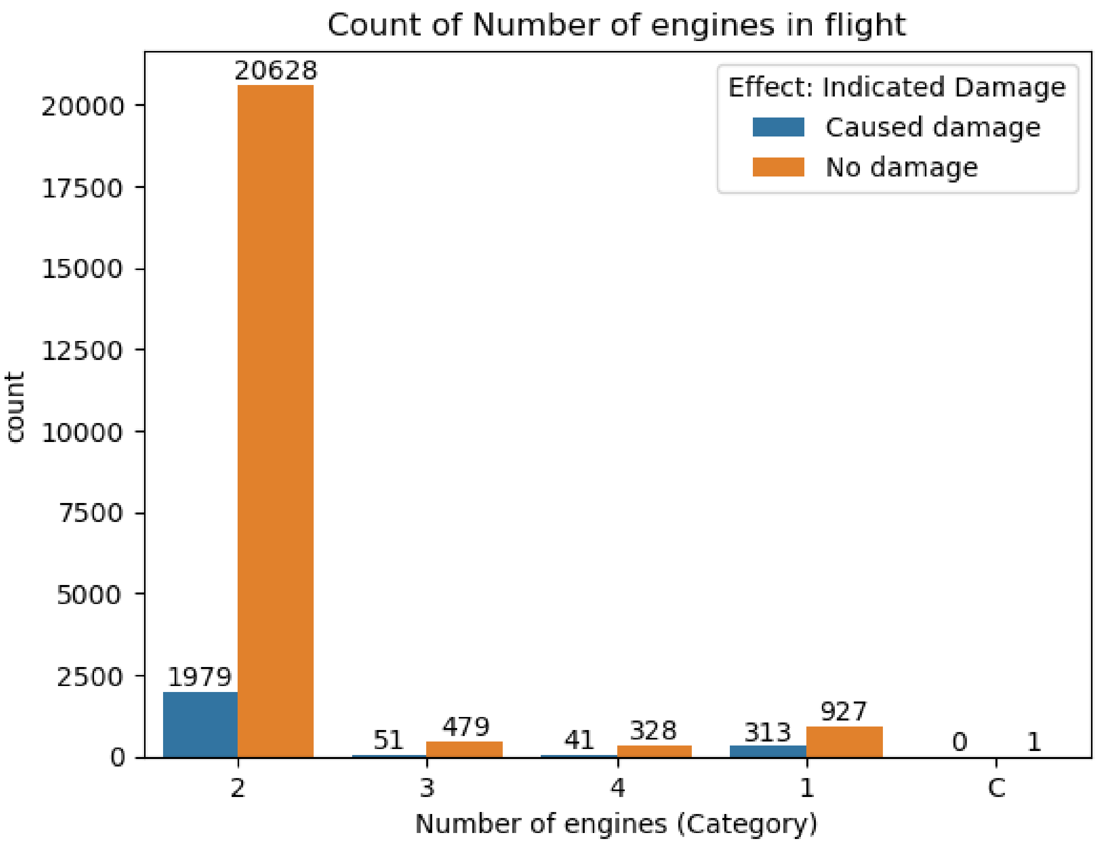


[12]:

```
# How flights get damaged after birds strikes
plt.figure(figsize=(10,6))
damage=sns.countplot(x='Effect: Indicated Damage', hue='Effect: Impact to␣
 ↪flight', data=data)
for bars in damage.containers:
    damage.bar_label(bars)
plt.xlabel('Type of damages')
plt.ylabel('Counts')
plt.title('Indiacated damage on flight')
plt.show()
```



[13]:
```python
# Count of Number of engines in flight with damaged/No damaged condition
plt.figure.figsize=(10,6)
engine=sns.countplot(x='Aircraft: Number of engines?', hue='Effect: Indicated␣
  ↪Damage', data=data)
for bars in engine.containers:
    engine.bar_label(bars)
plt.title('Count of Number of engines in flight')
plt.xlabel('Number of engines (Category)')
plt.show()
```



Two engines in an airplane are the most damaged among the others.

[14]:

```
import plotly.express as px
```

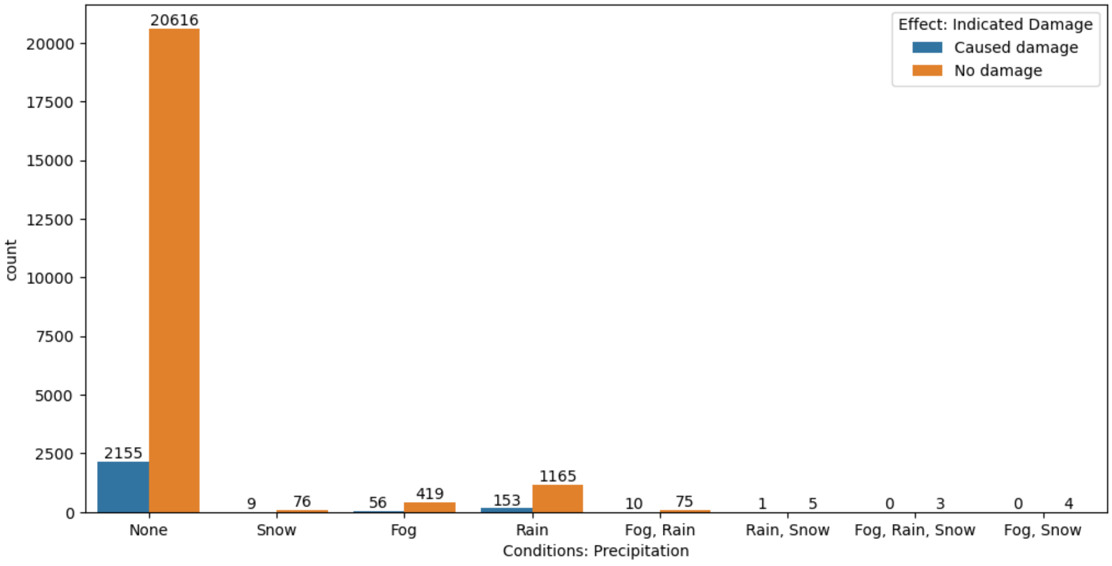
[15]:

```
# Phase of flights during strikes
phase=data['When: Phase of flight'].value_counts()
fig=px.pie(phase, values=phase.values, names=phase.index, title='Phase of␣
 ↪flights during the strikes')
fig.update_traces(textinfo="percent+label+value")
fig.show()
```

Most of the strikes take place at an approaching time.

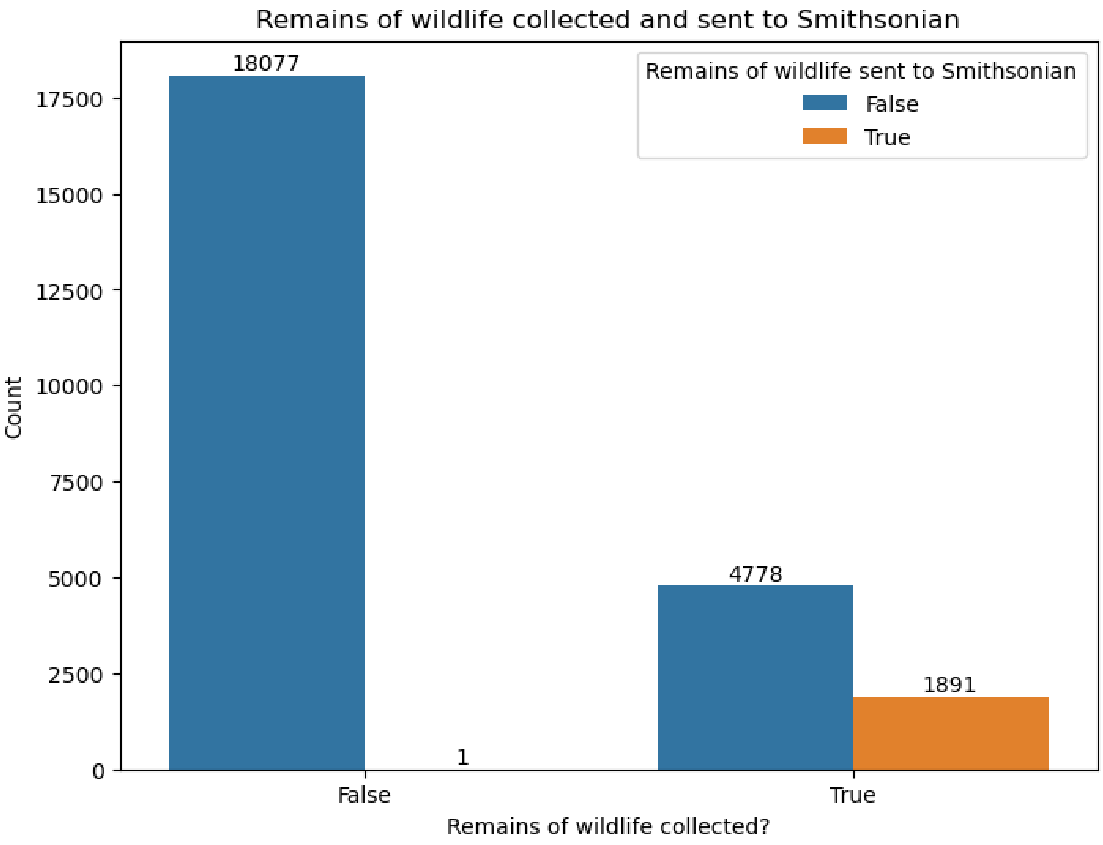
[16]:

```
#Conditions: Precipitation
plt.figure(figsize=(12,6))
rain=sns.countplot(data=data, x='Conditions: Precipitation', hue='Effect:␣
 ↪Indicated Damage')
for bars in rain.containers:
    rain.bar_label(bars)
plt.show()
```



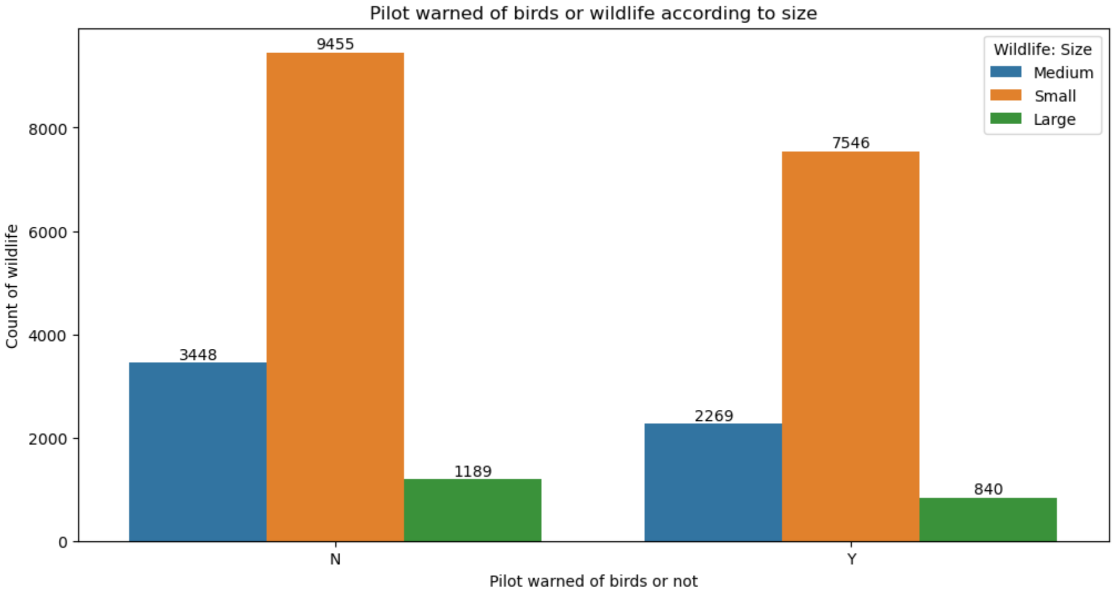
The weather conditions are clear most of the time, then comes rain, which becomes the cause of damage.

[17]:
```
# Remains of wildlife collected or not
plt.figure(figsize=(8,6))
collect=sns.countplot(x='Remains of wildlife collected?', hue='Remains of␣
 ↪wildlife sent to Smithsonian', data=data)
for bars in collect.containers:
    collect.bar_label(bars)
plt.ylabel('Count')
plt.title('Remains of wildlife collected and sent to Smithsonian')
plt.show()
```



[18]:

```
#Pilot warned of birds or wildlife according to size
plt.figure(figsize=(12,6))
warn=sns.countplot(x='Pilot warned of birds or wildlife?', data=data,␣
 ↪hue='Wildlife: Size')
for bars in warn.containers:
    warn.bar_label(bars)
plt.title('Pilot warned of birds or wildlife according to size')
plt.ylabel('Count of wildlife')
plt.xlabel('Pilot warned of birds or not')
plt.show()
```



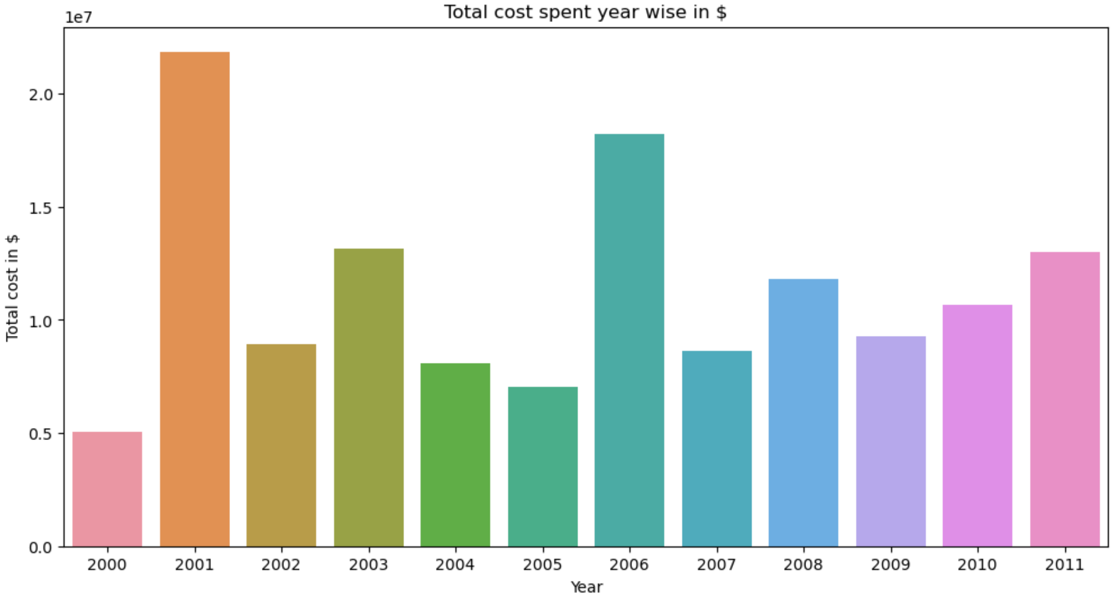
Most wildlife strikes with airplanes according to size: small, medium, and large.

```
[19]: # Sky condition during strike
      sky=data['Conditions: Sky'].value_counts()
      fig=px.pie(sky, names=sky.index, values=sky.values, title='Sky condition during␣
        ↪strike')
      fig.update_traces(textinfo='value+percent+label')
      fig.show()
```

Most of the time, the sky was clear; there was no rain or cloud.

```
[20]: #Total cost
      plt.figure(figsize=(12,6))
      cost=data.groupby(['Year'], as_index=False)['Cost: Total $'].sum()
      sns.barplot(x='Year',y='Cost: Total $', data=cost)
      plt.title('Total cost spent year wise in $')
      plt.ylabel('Total cost in $')
```

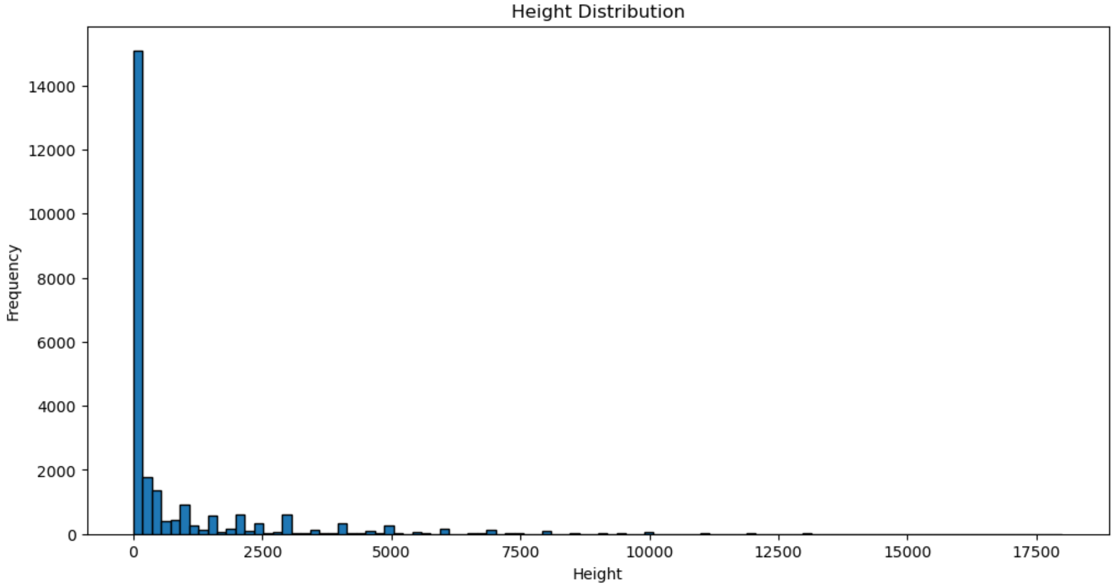
```
[20]: Text(0, 0.5, 'Total cost in $')
```



Most of the cost spent on airplanes by birds strikes was in 2001, then in 2006 and 2003.

[21]:

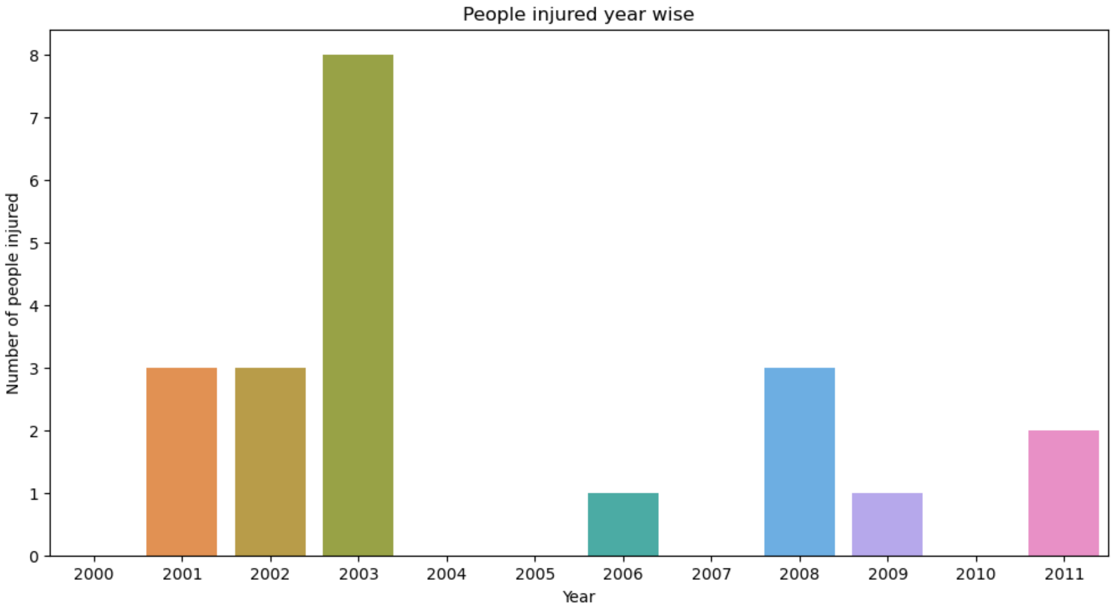
```
# Strike height in feet
plt.figure(figsize=(12, 6))
plt.hist(data['Feet above ground'], bins=100, edgecolor='black')
plt.xlabel('Height')
plt.ylabel('Frequency')
plt.title('Height Distribution')
plt.show()
```



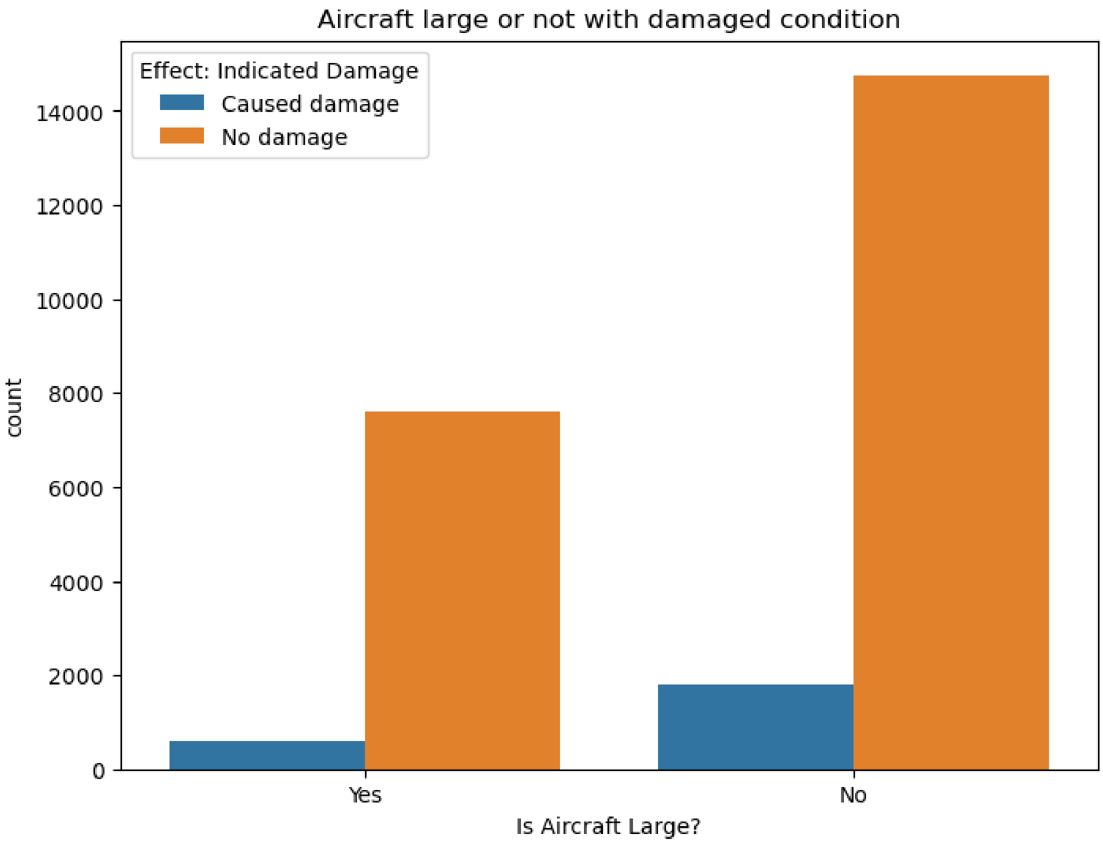
Most of strike incidents happened below 10000 feet height. In which most of it are below 200 feets.

```
[22]: # Peolple injured by strike year wise
      plt.figure(figsize=(12,6))
      injured=data.groupby(['Year'], as_index=False)['Number of people injured'].sum()
      sns.barplot(data=injured, x='Year', y='Number of people injured')
      plt.title('People injured year wise')
```

```
[22]: Text(0.5, 1.0, 'People injured year wise')
```



```
[23]: # Aircraft is large or not
      plt.figure(figsize=(8,6))
      sns.countplot(x='Is Aircraft Large?', hue='Effect: Indicated Damage', data=data)
      plt.title('Aircraft large or not with damaged condition')
      plt.show()
```



```
[24]: import plotly.express as px
```

```
[25]: airport=data.groupby(['Airport: Name'], as_index=False)['Wildlife: Number␣
       ↪Struck Actual'].sum().sort_values(by='Wildlife: Number Struck Actual',␣
       ↪ascending=False).head(10)
      airport
```

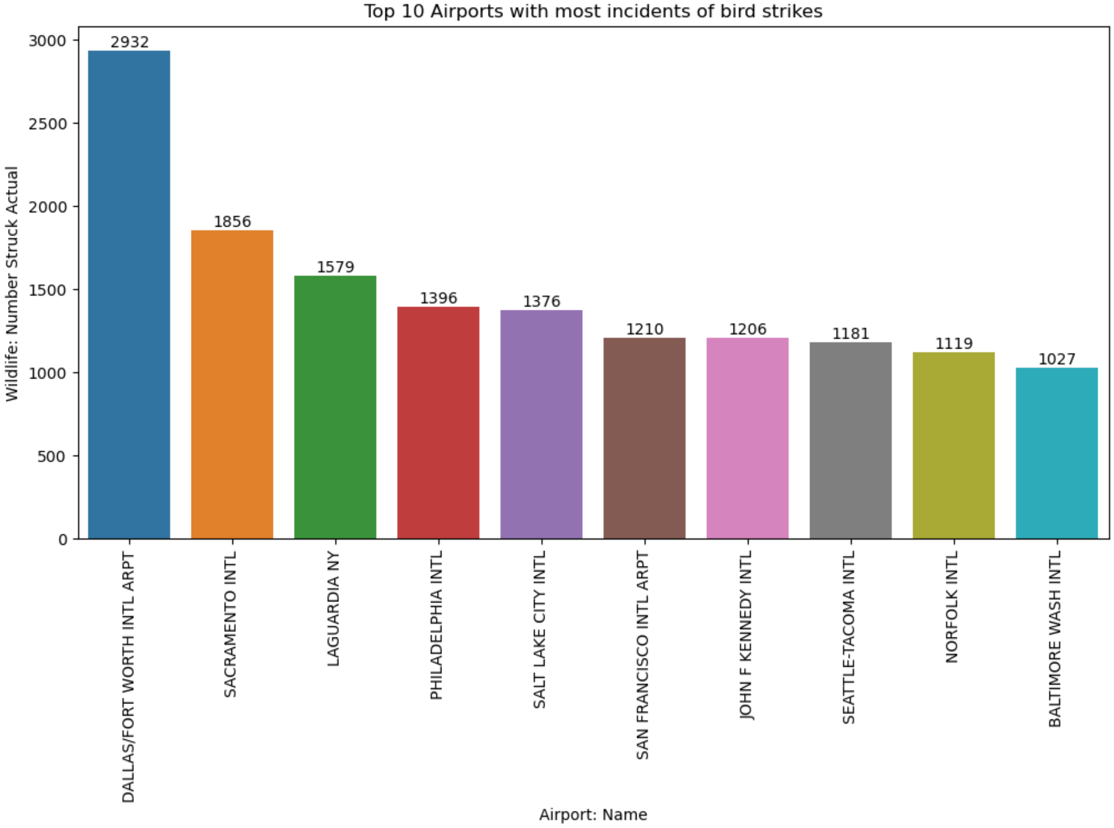
```
[25]:                      Airport: Name  Wildlife: Number Struck Actual
      188  DALLAS/FORT WORTH INTL ARPT                            2932
      776              SACRAMENTO INTL                            1856
      467                 LAGUARDIA NY                            1579
      698            PHILADELPHIA INTL                            1396
      783          SALT LAKE CITY INTL                            1376
      787      SAN FRANCISCO INTL ARPT                            1210
      424          JOHN F KENNEDY INTL                            1206
      805          SEATTLE-TACOMA INTL                            1181
      633                 NORFOLK INTL                            1119
      48           BALTIMORE WASH INTL                            1027
```

[26]:

```python
plt.figure(figsize=(12,6))
top=sns.barplot(data=airport, x='Airport: Name', y='Wildlife: Number Struck␣
 ↪Actual')
for bars in top.containers:
    top.bar_label(bars)
plt.title('Top 10 Airports with most incidents of bird strikes')
plt.xticks(rotation=90)
```

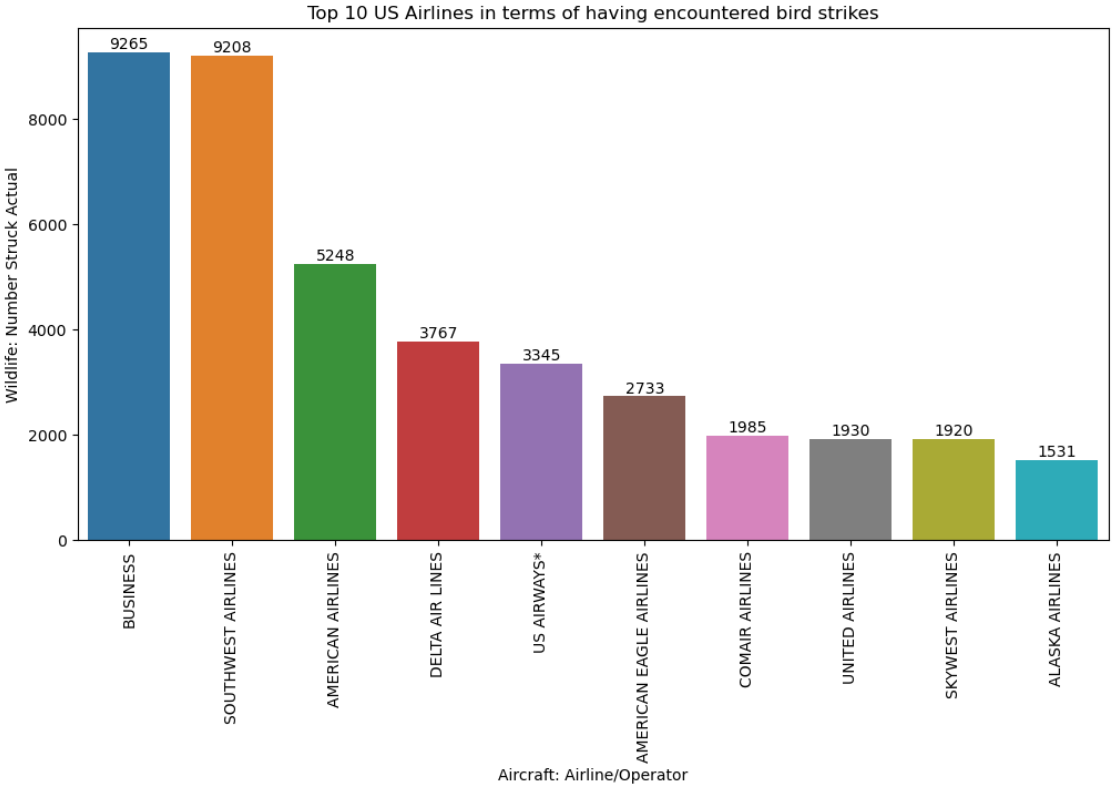
[26]:

```
(array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
 [Text(0, 0, 'DALLAS/FORT WORTH INTL ARPT'),
  Text(1, 0, 'SACRAMENTO INTL'),
  Text(2, 0, 'LAGUARDIA NY'),
  Text(3, 0, 'PHILADELPHIA INTL'),
  Text(4, 0, 'SALT LAKE CITY INTL'),
  Text(5, 0, 'SAN FRANCISCO INTL ARPT'),
  Text(6, 0, 'JOHN F KENNEDY INTL'),
  Text(7, 0, 'SEATTLE-TACOMA INTL'),
  Text(8, 0, 'NORFOLK INTL'),
  Text(9, 0, 'BALTIMORE WASH INTL')])
```



[27]:

```
plt.figure(figsize=(12,6))
operator=data.groupby(['Aircraft: Airline/Operator'], as_index=False)['Wildlife: Number Struck Actual'].sum().sort_values(by='Wildlife: Number Struck Actual', ascending=False).head(10)
graph=sns.barplot(data=operator, x='Aircraft: Airline/Operator', y='Wildlife: Number Struck Actual')
for bars in graph.containers:
    graph.bar_label(bars)
plt.xticks(rotation=90)
plt.title('Top 10 US Airlines in terms of having encountered bird strikes')
plt.show()
```



[ ]: