Unraveling Avian Disease Patterns for Insights and Prevention

Purpose of this project:

Analyzing the Avian Disease Dataset serves several important purposes that contribute to the understanding, management, and prevention of avian diseases and their impact on both animal and human health.

About the dataset:

The dataset contains information related to avian disease outbreaks and sporadic cases reported from various countries and regions worldwide. The data is sourced from the WAHIS (World Animal Health Information System) platform of the World Organisation for Animal Health. The dataset includes 43089 rows × 18 columns.

Data dictionary:

Year:

The year in which the disease event was reported or occurred.

Semester:

The semester (half-year) during which the disease event was reported or occurred.

World region:

the global region in which the disease event took place.

Country:

The country where the disease event occurred.

Administrative Division:

The specific administrative division or geographical area within the country where the disease event occurred.

Disease:

The name or type of avian disease that was reported.

Serotype/Subtype/Genotype:

Specific characteristics or subtypes of the disease-causing agent (e.g., virus subtype).

Animal Category:

The category of animals affected by the disease (wild, domestic or both).

Species:

The species of birds affected by the disease.

Outbreak_id:

An identifier for the disease outbreak or case.

New outbreaks:

The number of new disease outbreaks reported.

Susceptible:

The number of susceptible animals at risk of contracting the disease.

Measuring units:

The units used to measure the disease-related quantities (e.g., animals, hives).

Cases:

The number of confirmed cases of the disease.

Killed and disposed of:

The number of animals that were killed and disposed of as a result of the disease.

Slaughtered:

The number of animals that were slaughtered to control the disease.

Deaths:

The number of deaths resulting from the disease.

Vaccinated:

The number of animals vaccinated as a preventive measure against the disease.

Part 1: Processing and analysis of avian diseases dataset in python

1. Importaion of the dataset about the avian diseases worldwide:

In [1]:
```
#Importation the libraries:
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
%matplotlib inline
# Importation of the dataset
avian=pd.read_csv(r"D:\Avian diseases project\avian_diseases.csv",  encoding='latin-1')
avian
```

Out[1]:

	Year	Semester	World region	Country	Administrative Division	Disease	Serotype/Subtype/Genotype	Animal Category	Species	Outbreak_id	New outbreaks	Suscep
0	2005	Jul-Dec 2005	Africa	Angola	Huila	Newcastle disease virus (Inf. with)	NaN	Both animal categories	NaN	NaN	1	
1	2005	Jul-Dec 2005	Africa	Angola	Huila	Newcastle disease virus (Inf. with)	NaN	Domestic	Birds	NaN	NaN	
2	2005	Jul-Dec 2005	Africa	Benin	Adjara	Newcastle disease virus (Inf. with)	NaN	Both animal categories	NaN	NaN	1	
3	2005	Jul-Dec 2005	Africa	Benin	Adjara	Newcastle disease virus (Inf. with)	NaN	Domestic	Birds	NaN	NaN	1
4	2005	Jul-Dec 2005	Africa	Benin	Adjohoun	Newcastle disease virus (Inf. with)	NaN	Both animal categories	NaN	NaN	1	
...	...	...	...	...	...	...	...	...	...	...	...	
43084	2023	Jul-Dec 2023	Europe	Poland	Dobrzyniewo Du?e	Newcastle disease virus (Inf. with)	NaN	Domestic	Birds	122327.0	1	
43085	2023	Jul-Dec 2023	Europe	Poland	Turo?? Ko? cielna	Newcastle disease virus (Inf. with)	NaN	Domestic	Birds	121679.0	1	4
43086	2023	Jul-Dec 2023	Europe	Poland	Turo?? Ko? cielna	Newcastle disease virus (Inf. with)	NaN	Domestic	Birds	122325.0	1	2
43087	2023	Jul-Dec 2023	Europe	Poland	Turo?? Ko? cielna	Newcastle disease virus (Inf. with)	NaN	Domestic	Birds	122326.0	1	
43088	2023	Jul-Dec 2023	Europe	United Kingdom	Aberdeenshire	High pathogenicity avian	H5N1	Domestic	Birds	121841.0	1	3

	Year	Semester	World region	Country	Administrative Division	Disease	Serotype/Subtype/Genotype	Animal Category	Species	Outbreak_id	New outbreaks	Suscep
						influenza viruses (po...						

43089 rows x 18 columns

2. Data processing

In [2]:
```
avian.head(10)
```

Out[2]:

	Year	Semester	World region	Country	Administrative Division	Disease	Serotype/Subtype/Genotype	Animal Category	Species	Outbreak_id	New outbreaks	Susceptible	M
0	2005	Jul-Dec 2005	Africa	Angola	Huila	Newcastle disease virus (Inf. with)	NaN	Both animal categories	NaN	NaN	1	NaN	
1	2005	Jul-Dec 2005	Africa	Angola	Huila	Newcastle disease virus (Inf. with)	NaN	Domestic	Birds	NaN	NaN	210	
2	2005	Jul-Dec 2005	Africa	Benin	Adjara	Newcastle disease virus (Inf. with)	NaN	Both animal categories	NaN	NaN	1	NaN	
3	2005	Jul-Dec 2005	Africa	Benin	Adjara	Newcastle disease virus (Inf. with)	NaN	Domestic	Birds	NaN	NaN	10,000	
4	2005	Jul-Dec 2005	Africa	Benin	Adjohoun	Newcastle disease virus (Inf. with)	NaN	Both animal categories	NaN	NaN	1	NaN	
5	2005	Jul-Dec 2005	Africa	Benin	Adjohoun	Newcastle disease virus (Inf. with)	NaN	Domestic	Birds	NaN	NaN	143,223	
6	2005	Jul-Dec 2005	Africa	Benin	Akpro-Misserete	Newcastle disease virus (Inf. with)	NaN	Both animal categories	NaN	NaN	1	NaN	
7	2005	Jul-Dec 2005	Africa	Benin	Akpro-Misserete	Newcastle disease virus (Inf. with)	NaN	Domestic	Birds	NaN	NaN	76,000	

	Year	Semester	World region	Country	Administrative Division	Disease	Serotype/Subtype/Genotype	Animal Category	Species	Outbreak_id	New outbreaks	Susceptible	M
8	2005	Jul-Dec 2005	Africa	Benin	Avrankou	Newcastle disease virus (Inf. with)	NaN	Both animal categories	NaN	NaN	1	NaN	
9	2005	Jul-Dec 2005	Africa	Benin	Avrankou	Newcastle disease virus (Inf. with)	NaN	Domestic	Birds	NaN	NaN	5,000	

In [3]:
```
avian.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 43089 entries, 0 to 43088
Data columns (total 18 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Year                       43089 non-null  int64  
 1   Semester                   43089 non-null  object 
 2   World region               43089 non-null  object 
 3   Country                    43089 non-null  object 
 4   Administrative Division    43089 non-null  object 
 5   Disease                    43089 non-null  object 
 6   Serotype/Subtype/Genotype  12851 non-null  object 
 7   Animal Category            43089 non-null  object 
 8   Species                    25023 non-null  object 
 9   Outbreak_id                4503 non-null   float64
 10  New outbreaks              22569 non-null  object 
 11  Susceptible                18057 non-null  object 
 12  Measuring units            24963 non-null  object 
 13  Cases                      21483 non-null  object 
 14  Killed and disposed of     18906 non-null  object 
 15  Slaughtered                17360 non-null  object 
 16  Deaths                     20456 non-null  object 
 17  Vaccinated                 13758 non-null  object 
dtypes: float64(1), int64(1), object(16)
memory usage: 5.9+ MB
```

From the output above: we conclude that there are columns with missing values and numeric values reported as objetcs that might disrupt further calculations

2.1 Data cleaning:

2.1.1. Detect the columns with missing values:

In [4]:
```
avian.isnull().any(axis=0)
```

Out[4]:
```
Year                          False
Semester                      False
World region                  False
Country                       False
Administrative Division       False
Disease                       False
Serotype/Subtype/Genotype      True
Animal Category               False
Species                        True
Outbreak_id                    True
New outbreaks                  True
Susceptible                    True
Measuring units                True
Cases                          True
Killed and disposed of         True
Slaughtered                    True
Deaths                         True
Vaccinated                     True
dtype: bool
```

We have the following columns with missing values: Serotype/Subtype/Genotype , Species, Outbreak_id , New outbreaks, Susceptible , Measuring units, Cases, Killed and disposed of , Slaughtered ,Deaths, Vaccinated

In [5]:
```
# let's check the values of each column that contains missing values:
variables= ['Serotype/Subtype/Genotype' , 'Species', 'Outbreak_id' ,'New outbreaks', 'Susceptible' , 'Measuring units', 'Cases',

for i in  variables:
    print(avian[i].unique())
```

```
[nan 'H5N2' 'H5N1' 'H5 (N untyped)' 'H7N7' 'other' 'H7N1' 'H5N3' 'H7N3'
 'H7N2' 'not typed' 'H7N9' 'H7N8' 'H5N8' 'H7N6' 'H7N3;not typed'
 'H5N7;not typed' 'H7N1;not typed' 'H5N1;H7N1' 'H7N4' 'H7N7;other'
 'H7N6;other' 'pending' 'H5N2;H7N3' 'H7' 'H5' 'H5N2;H7N1' 'H5;H7'
 'H5;H5N2;H7' 'H5;H5N2;H7N2' 'H5;H5N2' 'H7 (N untyped)' 'H5N2;H7N1;H7N7'
 'O' 'A' 'H5N1;H7N7' 'H5N2;H7N7' 'H5N2;H5N3' 'H7 (N untyped);H7N7'
 'H5;H5N3' 'H5N6' 'H5N2;H5N8' 'H5N1;H5N6' 'H5N1;H5N2;H5N6' 'H5N2;H5N6'
 'H5N1;H5N3;H5N6' 'H5 (N untyped);H5N1' 'H5 (N untyped);H5N1;H5N2'
 'H5 (N untyped);H5N1;H5N2;H5N8' 'H5 (N untyped);H5N2' 'H5N1;H5N2'
 'H5N1;H5N2;H5N8' 'H5N2;H5N3;H5N8' 'H5N3;H5N8' 'H5N2;H7N2'
 'H5N1;H5N2;H5N9' 'H5N9' 'H5N1;H5N8' 'H5N5;H5N8' 'H5N1;H5N9'
 'H5N1;H5N3;H5N8' 'H5N5' 'H5N1;H5N3' 'H7 (N untyped);H7N9'
 'H5N2;H5N6;H5N8' 'H5N6;H5N8' 'H5N1;H7N9' 'H5 (N untyped);H5N2;H5N8'
 'H5N1;H5N2;H5N5' 'H5N7' 'H5;H7N3' 'H5N2;H5N5' 'H5 (N untyped);H5N2;H5N5'
 'H5N4']
[nan 'Birds' 'Wildlife (species unspecified)' 'Other species' 'Sheep'
 'Cattle' 'Bees' 'American Mink' 'Equidae' 'Northwestern Crow'
 'Black-headed Gull' 'Crested Goshawk' 'Crested Myna' 'Grey Heron'
 'House Crow' 'Little Egret' 'Oriental Magpie-Robin' 'Peregrin falcon'
 'Scaly-breasted Munia' 'Large-billed Crow' 'Cats'
 'Rock Pigeon (Rock Dove)' 'Passeridae (unidentified)'
 'Eurasian Collared-Dove' 'European Turtle-Dove' 'Burrowing parrot'
 'Columbidae (unidentified)' 'Great black-backed Gull'
 'Phaethontidae (unidentified)' 'Turquoise Parrot'
 'Psittacidae (unidentified)' 'Double-crested cormorant' 'Barn Swallow'
 'Common Grackle' 'Eurasian Hobby' 'Common Kestrel' 'Lesser kestrel'
 'Common pheasant' 'Anatidae (unidentified)' 'Common Coot' 'Grey Parrot'
 'Herring Gull' 'Mallard' 'Northern Gannet' 'Red-breasted Goose'
 'Senegal Parrot' 'Willow Ptarmigan' 'Indian Peafowl'
 'Yellow-crowned Parrot' 'Ibex' 'Budgerigar' 'Cockatiel' 'Common Myna'
 'Red-vented Bulbul' 'Zebra Dove' 'Yellow-legged Gull'
 'Phalacrocoracidae (unidentified)' 'Rock Partridge' 'Common magpie'
 'Charadriidae (unidentified)' 'Ciconiidae (unidentified)'
 "Steller's Sea Eagle" 'Common Wood-Pigeon' 'Black Kite'
 'Egyptian Vulture' 'Eurasian Griffon' 'Mourning Collared-Dove'
 'Northern Long-tailed Tit' 'Blue Tit' 'Yellowhammer' 'Ring-necked Dove'
 'Laughing Dove' 'Barn Owl (Common Barn-Owl)' 'Gyrfalcon'
 'Northern Pintail' 'Mandarin Duck' 'Gruidae (unidentified)'
 'Hooded crane' 'White-naped crane' 'Tundra Swan' 'Emerald Dove'
 'Red Jungle Fowl' 'Eurasian Wigeon' 'American wigeon' 'Evening Grosbeak'
 'Pine Grosbeak' 'Clay-colored thrush' 'Plain chachalaca' 'Gadwall'
 'Green-winged Teal' 'Bald Eagle' 'Great Horned owl' 'Canada Goose'
 'Ring-necked Duck' 'Snow Goose' "Cooper's Hawk" 'Northern Shoveler'
 'Wood Duck' 'Red-tailed Hawk' 'Snowy Owl' 'Tiger' 'Common Pochard'
```

```
 'Greater Scaup' 'Whooper Swan' 'Black-necked Grebe' 'Bar-headed Goose'
 'Light-vented bulbul' 'Black-crowned Night-Heron' 'Spur-winged Lapwing'
 'Dalmatian pelican' 'White Stork' 'Chaffinch' 'Toco Toucan' 'Rook'
 'Common Tern' 'Great Crested Grebe' 'Mute Swan' 'Wild turkey'
 'Great Black-headed Gull' 'Pale thrush' 'Great Egret'
 'Eurasian buzzard (common buzzard)' 'Western bronze-naped pigeon'
 'Great Cormorant' 'Black-winged Stilt' 'Common Moorhen'
 'Common ringed plover' 'Common Teal' 'Eurasian Spoonbill'
 'Ferruginous Pochard' 'Glossy Ibis' 'Green Sandpiper' 'Kentish plover'
 'Laridae (unidentified)' 'Little ringed plover' 'Little stint'
 'Marbled teal' 'Pied avocet' 'Ruddy Shelduck' 'Ruff' 'Wood Sandpiper'
 'Corvidae (unidentified)' 'Ardeidae (unidentified)' 'Bean Goose'
 'Greater Flamingo' 'Little Grebe' 'Phoenicopteridae (unidentified)'
 'Tufted Duck' 'Greylag Goose' 'Western Cattle Egret' 'Armenian Gull'
 'Pygmy cormorant' 'Common Crane' 'Black Swan' 'Cackling Goose'
 'Northern Goshawk' 'Strigidae (unidentified)'
 'Greater White-fronted Goose' 'Eurasian Curlew'
 'Numididae (unidentified)' 'Common Raven' 'Mew Gull' 'White-tailed Eagle'
 'Common Eider' 'Eurasian Eagle-Owl' 'Rallidae (unidentified)'
 'Pelecanidae (unidentified)' 'Dromaiidae (unidentified)'
 'Phasianidae (unidentified)' 'Struthionidae (unidentified)'
 'Falconidae (unidentified)' 'Hooded Crow' 'Dogs' 'Common Shelduck'
 'Lesser Black-backed Gull' 'Accipitridae (unidentified)'
 'Eurasian Sparrowhawk' 'Common Goldeneye' 'Red-crested Pochard'
 'Muscovy Duck' 'Great tit' 'Common Starling' 'Northern Cardinal'
 'Fringillidae (unidentified)' 'Crimson Rosella' 'Carrion Crow'
 'Great grey owl' 'Diamond Dove' 'Emu' 'Long-eared Owl' "Bonelli's Eagle"
 'African Olive-pigeon' 'Blue Ground-Dove' 'Blue-grey Tanager'
 'Crested Quail-dove' 'Eurasian Hoopoe' 'Grey-hooded parakeet'
 'Lemon Dove' 'Pearly Parakeet' 'Pink-headed Imperial-pigeon'
 'Silver-beaked Tanager' 'Spotted Dove' 'Spotted wood owl' 'Sun parakeet'
 'White-crowned pigeon' 'Eurasian Jackdaw' 'Picidae (unidentified)'
 'Masked lapwing' "Sclater's Crowned Pigeon" 'Spotted dove'
 'Tambourine dove' 'Victoria Crowned-Pigeon' 'European Goldfinch'
 'Tawny owl' 'Turdidae (unidentified)' 'Sheep/goats (mixed herd)' 'Swine'
 'Crested Pigeon' 'Luzon Hornbill' 'Madagascar Turtle-dove'
 'White Imperial-pigeon' 'Eurasian Woodcock' 'Eurasian Blackbird'
 'Island Canary' 'Eurasian scops owl' 'Little Owl' 'House Sparrow'
 'Willow tit' 'Afep Pigeon' 'Coliidae (unidentified)'
 'Rheidae (unidentified)' 'European Serin']
[    nan   6901.   6900. ... 122325. 122326. 121841.]
['1' nan '2' '3' '25' '4' '5' '115' '17' '7' '6' '10' '28' '9' '44' '15'
 '14' '19' '18' '58' '417' '69' '36' '8' '12' '50' '124' '83' '130' '121'
 '91' '141' '23' '22' '40' '30' '29' '70' '31' '11' '56' '21' '54' '47'
```

```
 '388' '82' '102' '59' '37' '64' '45' '60' '48' '24' '670' '52' '79' '108'
 '88' '114' '143' '20' '35' '32' '33' '27' '105' '43' '26' '107' '93' '13'
 '680' '195' '46' '57' '144' '38' '42' '184' '274' '16' '68' '34' '94'
 '87' '116' '118' '81' '151' '39' '55' '136' '78' '51' '215' '168' '345'
 '75' '66' '348' '100' '74' '112' '133' '67' '639' '89' '131' '49' '253'
 '80' '65' '122' '1,194' '663' '41' '178' '921' '106' '320' '76' '222'
 '110' '171' '355' '409' '276' '1,282' '616' '188' '536' '77' '62' '199'
 '104' '61' '1,797' '872' '291' '620' '72' '251' '92' '63' '225' '166'
 '1,549' '744' '332' '200' '71' '450' '250' '120' '2,481' '1,121' '721'
 '865' '147' '53' '95' '1,948' '937' '904' '165' '1,049' '146' '752' '204'
 '387' '189' '816' '506' '164' '192' '153' '98' '3,168' '1,556' '1,440'
 '145' '128' '132' '127' '341' '696' '380' '84' '185' '117' '175' '2,050'
 '1,272' '1,470' '103' '126' '532' '247' '304' '606' '407' '172' '73'
 '212' '97' '86' '138' '1,155' '886' '512' '352' '150' '194' '2,436' '925'
 '1,365' '602' '474' '405' '214' '3,235' '167' '372' '96' '1,406' '123'
 '479' '378' '547' '321' '191' '125' '3,020' '726' '1,135' '99' '605'
 '640' '404' '227' '140' '681' '305' '350' '484' '364' '129' '530' '316'
 '113' '1,936' '485' '623' '302' '1,244' '948' '707' '435' '180' '2,193'
 '694' '179' '221' '434' '173' '85' '244' '568' '457' '159' '271' '488'
 '825' '452' '154' '284' '349' '156' '685' '436' '176' '111' '973' '323'
 '424' '466' '300' '286' '216' '475' '480' '182' '213' '208' '395' '303'
 '160' '249' '158' '425' '210' '205' '428' '169' '333' '313' '232' '163'
 '248' '239' '90' '478' '236' '148' '245' '224' '152' '233' '109' '223'
 '197' '0' '259' '1,449' '481' '287']
[nan '210' '10,000' ... '13,460' '6,389' '43,410']
[nan 'Animal' 'Hives']
[nan '93' '532' ... '3,563' '14,340' '1,787']
[nan '0' '27' ... '13,922' '3,050' '31,189']
[nan '0' '215' '5' '64' '80' '35' '13' '15' '24' '4' '1' '243' '800' '10'
 '71' '31' '375' '40' '11' '21,173' '4,929' '50' '8' '3' '51' '25' '100'
 '102' '320' '196' '683' '2' '17' '1,100' '241' '180' '59' '141,000' '19'
 '297' '316' '11,021' '157,617' '1,670' '140' '84' '920' '45' '6' '52'
 '250' '689' '1,549' '874' '494' '21' '92' '30,470' '23,466' '152' '72'
 '12,770' '12' '39' '41' '127' '27' '200' '57' '43,990' '756' '33' '90'
 '74' '78' '107' '4,961' '115' '144,000' '1,000' '1,400' '32' '105' '30'
 '42' '205' '17,878' '20' '29' '46' '29,540' '20,739' '330' '22' '132'
 '24,000' '743' '600' '328' '183' '34' '6,361' '62,730' '978' '249'
 '6,837' '2,090' '7' '54' '62' '14' '261' '354' '4,910' '24,579' '12,200'
 '400' '2,849' '18' '9' '201' '160,000' '350' '48' '85' '213' '60' '99'
 '47' '28' '59,044' '580' '69,080' '120' '552' '559' '696' '228' '12,587'
 '16' '55' '333' '262' '117' '304' '47,085' '60,000' '68' '6,177' '599'
 '70' '310' '150' '1,570' '165' '225' '198' '928' '50,000' '610' '3,830'
 '11,500' '450' '37' '44' '160' '216' '240' '219' '32,737' '137' '458'
 '1,069' '421' '23' '217,618' '3,095' '100,000' '3,000' '14,000' '453'
```

'73' '2,763' '397' '173' '374' '212' '752' '837' '635' '20,452' '155'
'15,083' '900' '416' '680' '850,816' '30,000' '36' '270' '4,924' '2,674'
'194' '688' '700' '300' '310,900' '223,160' '3,660' '210' '2,633' '4,403'
'2,747' '39,138' '728' '5,750' '186' '154' '176' '64,500' '21,500'
'51,250' '18,000' '141' '1,120' '2,915' '53' '151' '43' '46,000' '101'
'347' '420' '2,950' '86' '26' '591,780' '50,600' '4,000' '42,201'
'303,000' '255' '1,784' '284,015' '263' '11,000' '381' '12,930' '15,679'
'1,273' '12,500' '23,500' '685' '2,000' '3,500' '10,000' '349' '403'
'111,136' '8,100' '879' '1,995' '3,436' '96' '63' '454' '75' '461'
'1,420' '124' '40,874' '430' '52,901' '476' '1,350' '7,206' '6,919'
'23,334' '26,266' '199' '352' '390' '500' '2,479' '61' '148' '264' '142'
'5,300' '34,455' '615' '6,080' '82' '44,999' '4,240' '8,259' '169'
'2,991' '2,528' '790' '5,580' '720' '537' '778' '128' '340' '204' '193'
'203' '299' '125' '129' '38' '258' '730' '355' '79' '146' '113' '1,302'
'937,200' '20,700' '112,757' '9,996' '34,259' '12,700' '58' '97' '106'
'6,000' '1,200' '465' '67' '77' '145' '55,059' '36,633' '3,600' '904'
'13,000' '2,256,533' '4,021' '8,000' '3,955' '1,300' '145,277' '218' '93'
'83' '8,400' '1,720' '144' '7,800' '44,522' '35,820' '90,357' '693,400'
'4,650' '6,050' '9,900' '3,413' '6,576' '27,000' '191' '45,000' '82,844'
'73,718' '497,803' '86,075' '3,061' '26,656' '655' '8,950' '5,950'
'2,166' '564' '247' '1,444' '55,903' '603,250' '45,350' '86,400' '460'
'91' '2,684' '1,441' '267' '1,490' '7,665' '79,845' '71,530' '352,716'
'49,107' '27,952' '2,439,909' '762,226' '1,490,528' '834,380' '151,132'
'2,995,168' '139,287' '156,000' '496,230' '4,144,808' '2,040,754'
'2,049,538' '2,500' '49' '39,445' '84,000' '167' '139,950' '1,981'
'6,654' '1,013' '22,253' '39,884' '198,159' '591,328' '14,703' '218,967'
'875,428' '234,196' '1,102,769' '19,400' '2,435,870' '136,000'
'1,044,135' '439,070' '3,800' '12,000' '3,890,878' '63,110' '2,970'
'14,300' '1,500' '1,011' '185' '110' '360' '56' '3,586' '37,000'
'4,753,029' '117,540' '249,583' '594,882' '533,254' '4,282,820' '70,000'
'19,500' '48,683' '2,203,537' '524,437' '680,994' '3,910' '20,000'
'6,489' '521' '75,000' '3,400' '14,524' '138' '65' '50,289' '217'
'440,309' '29,708' '276,294' '18,390' '804,674' '510' '388,081' '35,000'
'834' '116,000' '382,050' '368' '66,000' '197,500' '155,000' '990'
'218,770' '103' '161' '110,314' '3,490' '247,713' '5,215,415' '1,208,132'
'136,670' '110,000' '3,526,341' '806,350' '277,264' '4,775,617' '309,184'
'208' '76' '152,000' '122' '6,448' '118,538' '18,650' '104,511' '28,590'
'6,064' '18,171' '17,310' '1,486,476' '414,682' '672,031' '331,200'
'334,400' '82,669' '174,163' '3,929,031' '410' '40,000' '1,505,991'
'19,940' '126,164' '5,400' '77,118' '257,312' '53,000' '1,063' '81'
'862,060' '342' '1,956,522' '133' '25,153' '220' '863,063' '1,020'
'2,755' '149,900' '31,540' '33,930' '148,700' '3,960' '220,380' '192,500'
'4,534' '3,880' '192,000' '95,000']

```
[nan '93' '108' ... '3,159' '13,976' '29,488']
[nan '0' '68' ... '97,700' '158,250' '131,860']
```

In [6]:
```
# let's check how many values in each column :
unique_values_counts= avian.nunique()
print( '\nNumber of values in each column including the nans:\n')
print(unique_values_counts)
```

```
Number of values in each column including the nans:

Year                          19
Semester                      38
World region                   5
Country                      163
Administrative Division     3679
Disease                       10
Serotype/Subtype/Genotype     73
Animal Category                3
Species                      231
Outbreak_id                 4494
New outbreaks                326
Susceptible                 8108
Measuring units                2
Cases                       5273
Killed and disposed of      4950
Slaughtered                  548
Deaths                      3486
Vaccinated                  1979
dtype: int64
```

In [7]:
```
# let's check how many values in each categorical column :
var=['World region','Country','Disease', 'Animal Category','Species', 'Measuring units']
unique_values_counts_cat= avian[var].nunique()
print( '\nNumber of values in categorical columns including the nans:\n')
print(unique_values_counts_cat)
```

```
Number of values in categorical columns including the nans:

World region         5
Country            163
Disease             10
Animal Category      3
Species            231
Measuring units      2
dtype: int64
```

2.1.2. Check if there is any misspelling values:

In [9]:

```
a=avian['World region'].unique()
b=avian['Country'].unique()
c=avian['Disease'].unique()
d=avian['Animal Category'].unique()
e=avian['Species'].unique()
f=avian['Measuring units'].unique()
print("\n\na.World region values:\n\n", a,  "\n\n\nb.Country names:\n\n",np.sort(b), "\n\n\nc.Diseases names:\n\n",np.sort(c),
      "\n\n\nd.Animal Category:\n\n",d, '\n\n\ne.Species:\n\n',e, '\n\nf.Measuring units:\n', f)
```

a.World region values:

```
['Africa' 'Americas' 'Asia' 'Europe' 'Oceania']
```

b.Country names:

```
['Afghanistan' 'Albania' 'Algeria' 'Angola' 'Argentina' 'Armenia'
'Australia' 'Austria' 'Azerbaijan' 'Bahrain' 'Bangladesh' 'Barbados'
'Belarus' 'Belgium' 'Belize' 'Benin' 'Bhutan' 'Bolivia'
'Bosnia and Herzegovina' 'Botswana' 'Brazil' 'Bulgaria' 'Burkina Faso'
'Burundi' 'Cabo verde' 'Cambodia' 'Cameroon' 'Canada' 'Cayman Islands'
'Central African (Rep.)' 'Chad' 'Chile' "China (People's Rep. of)"
'Chinese Taipei' 'Colombia' 'Congo (Dem. Rep. of the)'
'Congo (Rep. of the)' 'Costa Rica' "Cote D'Ivoire" 'Croatia' 'Cuba'
'Cyprus' 'Czech Republic' 'Denmark' 'Djibouti' 'Dominican (Rep.)'
'Ecuador' 'Egypt' 'El Salvador' 'Eritrea' 'Estonia' 'Eswatini' 'Ethiopia'
'Fiji' 'Finland' 'France' 'French Polynesia' 'Gabon' 'Gambia' 'Georgia'
'Germany' 'Ghana' 'Greece' 'Guatemala' 'Guinea' 'Guinea-Bissau' 'Haiti'
'Honduras' 'Hong Kong' 'Hungary' 'India' 'Indonesia' 'Iran' 'Iraq'
'Ireland' 'Israel' 'Italy' 'Japan' 'Jordan' 'Kazakhstan' 'Kenya'
"Korea (Dem People's Rep. of)" 'Korea (Rep. of)' 'Kuwait' 'Kyrgyzstan'
'Laos' 'Latvia' 'Lebanon' 'Lesotho' 'Libya' 'Liechtenstein' 'Lithuania'
'Luxembourg' 'Madagascar' 'Malawi' 'Malaysia' 'Mali' 'Martinique'
'Mauritania' 'Mexico' 'Moldova' 'Mongolia' 'Montenegro' 'Morocco'
'Mozambique' 'Myanmar' 'Namibia' 'Nepal' 'Netherlands' 'New Zealand'
'Nicaragua' 'Niger' 'Nigeria' 'North Macedonia' 'Norway' 'Oman'
'Pakistan' 'Palestine' 'Panama' 'Papua New Guinea' 'Paraguay' 'Peru'
'Philippines' 'Poland' 'Portugal' 'Qatar' 'Reunion' 'Romania' 'Russia'
'Rwanda' 'Sao Tome and Principe' 'Saudi Arabia' 'Senegal' 'Serbia'
'Serbia and Montenegro' 'Sierra Leone' 'Singapore' 'Slovakia' 'Slovenia'
'South Africa' 'South Sudan (Rep. of)' 'Spain' 'Sri Lanka' 'St. Helena'
'Sudan' 'Sweden' 'Switzerland' 'Tanzania' 'Thailand' 'Togo' 'Tunisia'
'Turkmenistan' 'T\x9frkiye' 'Uganda' 'Ukraine' 'United Kingdom'
'United States of America' 'Uruguay' 'Venezuela' 'Vietnam' 'Yemen'
'Zambia' 'Zimbabwe']
```

c.Diseases names:

```
['Avian chlamydiosis' 'Avian infectious bronchitis'
'Avian infectious laryngotracheitis' 'Avian tuberculosis (-2005)'
'High pathogenicity avian influenza viruses (poultry) (Inf. with)'
'Low pathogenic avian influenza (poultry) (2006-2021)'
```

```
 'Low pathogenicity avian influenza viruses transmissible to humans (Inf. with) (2022-)'
 'Mycoplasma gallisepticumÊ(Avian mycoplasmosis) (Inf. with)'
 'Mycoplasma synoviae (Avian mycoplasmosis) (Inf. with) (2006-)'
 'Newcastle disease virus (Inf. with)']
```

d.Animal Category:

```
 ['Both animal categories' 'Domestic' 'Wild']
```

e.Species:

```
 [nan 'Birds' 'Wildlife (species unspecified)' 'Other species' 'Sheep'
 'Cattle' 'Bees' 'American Mink' 'Equidae' 'Northwestern Crow'
 'Black-headed Gull' 'Crested Goshawk' 'Crested Myna' 'Grey Heron'
 'House Crow' 'Little Egret' 'Oriental Magpie-Robin' 'Peregrin falcon'
 'Scaly-breasted Munia' 'Large-billed Crow' 'Cats'
 'Rock Pigeon (Rock Dove)' 'Passeridae (unidentified)'
 'Eurasian Collared-Dove' 'European Turtle-Dove' 'Burrowing parrot'
 'Columbidae (unidentified)' 'Great black-backed Gull'
 'Phaethontidae (unidentified)' 'Turquoise Parrot'
 'Psittacidae (unidentified)' 'Double-crested cormorant' 'Barn Swallow'
 'Common Grackle' 'Eurasian Hobby' 'Common Kestrel' 'Lesser kestrel'
 'Common pheasant' 'Anatidae (unidentified)' 'Common Coot' 'Grey Parrot'
 'Herring Gull' 'Mallard' 'Northern Gannet' 'Red-breasted Goose'
 'Senegal Parrot' 'Willow Ptarmigan' 'Indian Peafowl'
 'Yellow-crowned Parrot' 'Ibex' 'Budgerigar' 'Cockatiel' 'Common Myna'
 'Red-vented Bulbul' 'Zebra Dove' 'Yellow-legged Gull'
 'Phalacrocoracidae (unidentified)' 'Rock Partridge' 'Common magpie'
 'Charadriidae (unidentified)' 'Ciconiidae (unidentified)'
 "Steller's Sea Eagle" 'Common Wood-Pigeon' 'Black Kite'
 'Egyptian Vulture' 'Eurasian Griffon' 'Mourning Collared-Dove'
 'Northern Long-tailed Tit' 'Blue Tit' 'Yellowhammer' 'Ring-necked Dove'
 'Laughing Dove' 'Barn Owl (Common Barn-Owl)' 'Gyrfalcon'
 'Northern Pintail' 'Mandarin Duck' 'Gruidae (unidentified)'
 'Hooded crane' 'White-naped crane' 'Tundra Swan' 'Emerald Dove'
 'Red Jungle Fowl' 'Eurasian Wigeon' 'American wigeon' 'Evening Grosbeak'
 'Pine Grosbeak' 'Clay-colored thrush' 'Plain chachalaca' 'Gadwall'
 'Green-winged Teal' 'Bald Eagle' 'Great Horned owl' 'Canada Goose'
 'Ring-necked Duck' 'Snow Goose' "Cooper's Hawk" 'Northern Shoveler'
 'Wood Duck' 'Red-tailed Hawk' 'Snowy Owl' 'Tiger' 'Common Pochard'
 'Greater Scaup' 'Whooper Swan' 'Black-necked Grebe' 'Bar-headed Goose'
 'Light-vented bulbul' 'Black-crowned Night-Heron' 'Spur-winged Lapwing'
```

```
 'Dalmatian pelican' 'White Stork' 'Chaffinch' 'Toco Toucan' 'Rook'
 'Common Tern' 'Great Crested Grebe' 'Mute Swan' 'Wild turkey'
 'Great Black-headed Gull' 'Pale thrush' 'Great Egret'
 'Eurasian buzzard (common buzzard)' 'Western bronze-naped pigeon'
 'Great Cormorant' 'Black-winged Stilt' 'Common Moorhen'
 'Common ringed plover' 'Common Teal' 'Eurasian Spoonbill'
 'Ferruginous Pochard' 'Glossy Ibis' 'Green Sandpiper' 'Kentish plover'
 'Laridae (unidentified)' 'Little ringed plover' 'Little stint'
 'Marbled teal' 'Pied avocet' 'Ruddy Shelduck' 'Ruff' 'Wood Sandpiper'
 'Corvidae (unidentified)' 'Ardeidae (unidentified)' 'Bean Goose'
 'Greater Flamingo' 'Little Grebe' 'Phoenicopteridae (unidentified)'
 'Tufted Duck' 'Greylag Goose' 'Western Cattle Egret' 'Armenian Gull'
 'Pygmy cormorant' 'Common Crane' 'Black Swan' 'Cackling Goose'
 'Northern Goshawk' 'Strigidae (unidentified)'
 'Greater White-fronted Goose' 'Eurasian Curlew'
 'Numididae (unidentified)' 'Common Raven' 'Mew Gull' 'White-tailed Eagle'
 'Common Eider' 'Eurasian Eagle-Owl' 'Rallidae (unidentified)'
 'Pelecanidae (unidentified)' 'Dromaiidae (unidentified)'
 'Phasianidae (unidentified)' 'Struthionidae (unidentified)'
 'Falconidae (unidentified)' 'Hooded Crow' 'Dogs' 'Common Shelduck'
 'Lesser Black-backed Gull' 'Accipitridae (unidentified)'
 'Eurasian Sparrowhawk' 'Common Goldeneye' 'Red-crested Pochard'
 'Muscovy Duck' 'Great tit' 'Common Starling' 'Northern Cardinal'
 'Fringillidae (unidentified)' 'Crimson Rosella' 'Carrion Crow'
 'Great grey owl' 'Diamond Dove' 'Emu' 'Long-eared Owl' "Bonelli's Eagle"
 'African Olive-pigeon' 'Blue Ground-Dove' 'Blue-grey Tanager'
 'Crested Quail-dove' 'Eurasian Hoopoe' 'Grey-hooded parakeet'
 'Lemon Dove' 'Pearly Parakeet' 'Pink-headed Imperial-pigeon'
 'Silver-beaked Tanager' 'Spotted Dove' 'Spotted wood owl' 'Sun parakeet'
 'White-crowned pigeon' 'Eurasian Jackdaw' 'Picidae (unidentified)'
 'Masked lapwing' "Sclater's Crowned Pigeon" 'Spotted dove'
 'Tambourine dove' 'Victoria Crowned-Pigeon' 'European Goldfinch'
 'Tawny owl' 'Turdidae (unidentified)' 'Sheep/goats (mixed herd)' 'Swine'
 'Crested Pigeon' 'Luzon Hornbill' 'Madagascar Turtle-dove'
 'White Imperial-pigeon' 'Eurasian Woodcock' 'Eurasian Blackbird'
 'Island Canary' 'Eurasian scops owl' 'Little Owl' 'House Sparrow'
 'Willow tit' 'Afep Pigeon' 'Coliidae (unidentified)'
 'Rheidae (unidentified)' 'European Serin']

f.Measuring units:
 [nan 'Animal' 'Hives']
```

In [10]:
```
# there are no duplicates in country : "Korea (Dem People's Rep. of)" 'Korea (Rep. of) represent North and South Korea.
# There are countries misspelled such as :T\x9frkiye.
# To ajust that, we will use interpolate:

avian['Country'].interpolate()
np.sort(avian['Country'].unique())
```

Out[10]:
```
array(['Afghanistan', 'Albania', 'Algeria', 'Angola', 'Argentina',
       'Armenia', 'Australia', 'Austria', 'Azerbaijan', 'Bahrain',
       'Bangladesh', 'Barbados', 'Belarus', 'Belgium', 'Belize', 'Benin',
       'Bhutan', 'Bolivia', 'Bosnia and Herzegovina', 'Botswana',
       'Brazil', 'Bulgaria', 'Burkina Faso', 'Burundi', 'Cabo verde',
       'Cambodia', 'Cameroon', 'Canada', 'Cayman Islands',
       'Central African (Rep.)', 'Chad', 'Chile',
       "China (People's Rep. of)", 'Chinese Taipei', 'Colombia',
       'Congo (Dem. Rep. of the)', 'Congo (Rep. of the)', 'Costa Rica',
       "Cote D'Ivoire", 'Croatia', 'Cuba', 'Cyprus', 'Czech Republic',
       'Denmark', 'Djibouti', 'Dominican (Rep.)', 'Ecuador', 'Egypt',
       'El Salvador', 'Eritrea', 'Estonia', 'Eswatini', 'Ethiopia',
       'Fiji', 'Finland', 'France', 'French Polynesia', 'Gabon', 'Gambia',
       'Georgia', 'Germany', 'Ghana', 'Greece', 'Guatemala', 'Guinea',
       'Guinea-Bissau', 'Haiti', 'Honduras', 'Hong Kong', 'Hungary',
       'India', 'Indonesia', 'Iran', 'Iraq', 'Ireland', 'Israel', 'Italy',
       'Japan', 'Jordan', 'Kazakhstan', 'Kenya',
       "Korea (Dem People's Rep. of)", 'Korea (Rep. of)', 'Kuwait',
       'Kyrgyzstan', 'Laos', 'Latvia', 'Lebanon', 'Lesotho', 'Libya',
       'Liechtenstein', 'Lithuania', 'Luxembourg', 'Madagascar', 'Malawi',
       'Malaysia', 'Mali', 'Martinique', 'Mauritania', 'Mexico',
       'Moldova', 'Mongolia', 'Montenegro', 'Morocco', 'Mozambique',
       'Myanmar', 'Namibia', 'Nepal', 'Netherlands', 'New Zealand',
       'Nicaragua', 'Niger', 'Nigeria', 'North Macedonia', 'Norway',
       'Oman', 'Pakistan', 'Palestine', 'Panama', 'Papua New Guinea',
       'Paraguay', 'Peru', 'Philippines', 'Poland', 'Portugal', 'Qatar',
       'Reunion', 'Romania', 'Russia', 'Rwanda', 'Sao Tome and Principe',
       'Saudi Arabia', 'Senegal', 'Serbia', 'Serbia and Montenegro',
       'Sierra Leone', 'Singapore', 'Slovakia', 'Slovenia',
       'South Africa', 'South Sudan (Rep. of)', 'Spain', 'Sri Lanka',
       'St. Helena', 'Sudan', 'Sweden', 'Switzerland', 'Tanzania',
       'Thailand', 'Togo', 'Tunisia', 'Turkmenistan', 'T\x9frkiye',
       'Uganda', 'Ukraine', 'United Kingdom', 'United States of America',
       'Uruguay', 'Venezuela', 'Vietnam', 'Yemen', 'Zambia', 'Zimbabwe'],
      dtype=object)
```

interpolate did not work, we know that The value 'T\x9frkiye' appears to be misspelled and might be intended to represent the country name 'Türkiye', which is the Turkish name for Turkey. So, we will replace it:

In [11]:
```
avian['Country'] = avian['Country'].replace('T\x9frkiye', 'Türkiye')
np.sort(avian['Country'].unique())
```

Out[11]:
```
array(['Afghanistan', 'Albania', 'Algeria', 'Angola', 'Argentina',
       'Armenia', 'Australia', 'Austria', 'Azerbaijan', 'Bahrain',
       'Bangladesh', 'Barbados', 'Belarus', 'Belgium', 'Belize', 'Benin',
       'Bhutan', 'Bolivia', 'Bosnia and Herzegovina', 'Botswana',
       'Brazil', 'Bulgaria', 'Burkina Faso', 'Burundi', 'Cabo verde',
       'Cambodia', 'Cameroon', 'Canada', 'Cayman Islands',
       'Central African (Rep.)', 'Chad', 'Chile',
       "China (People's Rep. of)", 'Chinese Taipei', 'Colombia',
       'Congo (Dem. Rep. of the)', 'Congo (Rep. of the)', 'Costa Rica',
       "Cote D'Ivoire", 'Croatia', 'Cuba', 'Cyprus', 'Czech Republic',
       'Denmark', 'Djibouti', 'Dominican (Rep.)', 'Ecuador', 'Egypt',
       'El Salvador', 'Eritrea', 'Estonia', 'Eswatini', 'Ethiopia',
       'Fiji', 'Finland', 'France', 'French Polynesia', 'Gabon', 'Gambia',
       'Georgia', 'Germany', 'Ghana', 'Greece', 'Guatemala', 'Guinea',
       'Guinea-Bissau', 'Haiti', 'Honduras', 'Hong Kong', 'Hungary',
       'India', 'Indonesia', 'Iran', 'Iraq', 'Ireland', 'Israel', 'Italy',
       'Japan', 'Jordan', 'Kazakhstan', 'Kenya',
       "Korea (Dem People's Rep. of)", 'Korea (Rep. of)', 'Kuwait',
       'Kyrgyzstan', 'Laos', 'Latvia', 'Lebanon', 'Lesotho', 'Libya',
       'Liechtenstein', 'Lithuania', 'Luxembourg', 'Madagascar', 'Malawi',
       'Malaysia', 'Mali', 'Martinique', 'Mauritania', 'Mexico',
       'Moldova', 'Mongolia', 'Montenegro', 'Morocco', 'Mozambique',
       'Myanmar', 'Namibia', 'Nepal', 'Netherlands', 'New Zealand',
       'Nicaragua', 'Niger', 'Nigeria', 'North Macedonia', 'Norway',
       'Oman', 'Pakistan', 'Palestine', 'Panama', 'Papua New Guinea',
       'Paraguay', 'Peru', 'Philippines', 'Poland', 'Portugal', 'Qatar',
       'Reunion', 'Romania', 'Russia', 'Rwanda', 'Sao Tome and Principe',
       'Saudi Arabia', 'Senegal', 'Serbia', 'Serbia and Montenegro',
       'Sierra Leone', 'Singapore', 'Slovakia', 'Slovenia',
       'South Africa', 'South Sudan (Rep. of)', 'Spain', 'Sri Lanka',
       'St. Helena', 'Sudan', 'Sweden', 'Switzerland', 'Tanzania',
       'Thailand', 'Togo', 'Tunisia', 'Turkmenistan', 'Türkiye', 'Uganda',
       'Ukraine', 'United Kingdom', 'United States of America', 'Uruguay',
       'Venezuela', 'Vietnam', 'Yemen', 'Zambia', 'Zimbabwe'],
      dtype=object)
```

The diseases are not well written. We will change it:

In [12]:
```
avian['Disease']=avian['Disease'].replace('Newcastle disease virus (Inf. with)', 'Newcastle disease virus')
avian['Disease']=avian['Disease'].replace('Mycoplasma gallisepticumÊ(Avian mycoplasmosis) (Inf. with)', 'Mycoplasma gallisepticu
avian['Disease']=avian['Disease'].replace('High pathogenicity avian influenza viruses (poultry) (Inf. with)', 'High pathogenicity
avian['Disease']=avian['Disease'].replace('Avian tuberculosis (-2005)', 'Avian tuberculosis')
avian['Disease']=avian['Disease'].replace('Mycoplasma synoviae (Avian mycoplasmosis) (Inf. with) (2006-)', 'Mycoplasma synoviae'
avian['Disease']=avian['Disease'].replace('Low pathogenic avian influenza (poultry) (2006-2021)', 'Low pathogenic avian influenza
avian['Disease']=avian['Disease'].replace('Low pathogenicity avian influenza viruses transmissible to humans (Inf. with) (2022-)
avian['Disease'].unique()
```

Out[12]:
```
array(['Newcastle disease virus', 'Mycoplasma gallisepticum',
       'Avian chlamydiosis', 'High pathogenicity avian influenza viruses',
       'Avian infectious bronchitis',
       'Avian infectious laryngotracheitis', 'Avian tuberculosis',
       'Mycoplasma synoviae', 'Low pathogenic avian influenza',
       'Low pathogenic avian influenza viruses transmissible to humans'],
      dtype=object)
```

2.2. Data transformation

Conversion of data types:

-Convert columns: New outbreaks , Susceptible,Cases,Killed and disposed of, Slaughtered ,Deaths, Vaccinated from object to integer.

-When sorting the Species, we encountered a problem in data type

In [13]:
```
  #To change New outbreaks , Susceptible,Cases,Killed and disposed of, Slaughtered ,Deaths, Vaccinated from object to integer.
    # Let's take a look at the values:
a=avian['Susceptible'].unique()
b=avian['Cases'].unique()
c=avian['Killed and disposed of'].unique()
d=avian['Slaughtered'].unique()
e=avian['Deaths'].unique()
f=avian['Vaccinated'].unique()
g=avian['New outbreaks'].unique()
print("New outbreaks:",g , '\nSusceptible:',a, '\nCases:',b, '\nKilled and disposed of:',c, '\nSlaughtered:',d,
      '\nDeaths:',e, '\nVaccinated:',f)
```

```
New outbreaks: ['1' nan '2' '3' '25' '4' '5' '115' '17' '7' '6' '10' '28' '9' '44' '15'
 '14' '19' '18' '58' '417' '69' '36' '8' '12' '50' '124' '83' '130' '121'
 '91' '141' '23' '22' '40' '30' '29' '70' '31' '11' '56' '21' '54' '47'
 '388' '82' '102' '59' '37' '64' '45' '60' '48' '24' '670' '52' '79' '108'
 '88' '114' '143' '20' '35' '32' '33' '27' '105' '43' '26' '107' '93' '13'
 '680' '195' '46' '57' '144' '38' '42' '184' '274' '16' '68' '34' '94'
 '87' '116' '118' '81' '151' '39' '55' '136' '78' '51' '215' '168' '345'
 '75' '66' '348' '100' '74' '112' '133' '67' '639' '89' '131' '49' '253'
 '80' '65' '122' '1,194' '663' '41' '178' '921' '106' '320' '76' '222'
 '110' '171' '355' '409' '276' '1,282' '616' '188' '536' '77' '62' '199'
 '104' '61' '1,797' '872' '291' '620' '72' '251' '92' '63' '225' '166'
 '1,549' '744' '332' '200' '71' '450' '250' '120' '2,481' '1,121' '721'
 '865' '147' '53' '95' '1,948' '937' '904' '165' '1,049' '146' '752' '204'
 '387' '189' '816' '506' '164' '192' '153' '98' '3,168' '1,556' '1,440'
 '145' '128' '132' '127' '341' '696' '380' '84' '185' '117' '175' '2,050'
 '1,272' '1,470' '103' '126' '532' '247' '304' '606' '407' '172' '73'
 '212' '97' '86' '138' '1,155' '886' '512' '352' '150' '194' '2,436' '925'
 '1,365' '602' '474' '405' '214' '3,235' '167' '372' '96' '1,406' '123'
 '479' '378' '547' '321' '191' '125' '3,020' '726' '1,135' '99' '605'
 '640' '404' '227' '140' '681' '305' '350' '484' '364' '129' '530' '316'
 '113' '1,936' '485' '623' '302' '1,244' '948' '707' '435' '180' '2,193'
 '694' '179' '221' '434' '173' '85' '244' '568' '457' '159' '271' '488'
 '825' '452' '154' '284' '349' '156' '685' '436' '176' '111' '973' '323'
 '424' '466' '300' '286' '216' '475' '480' '182' '213' '208' '395' '303'
 '160' '249' '158' '425' '210' '205' '428' '169' '333' '313' '232' '163'
 '248' '239' '90' '478' '236' '148' '245' '224' '152' '233' '109' '223'
 '197' '0' '259' '1,449' '481' '287']
Susceptible: [nan '210' '10,000' ... '13,460' '6,389' '43,410']
Cases: [nan '93' '532' ... '3,563' '14,340' '1,787']
Killed and disposed of: [nan '0' '27' ... '13,922' '3,050' '31,189']
Slaughtered: [nan '0' '215' '5' '64' '80' '35' '13' '15' '24' '4' '1' '243' '800' '10'
 '71' '31' '375' '40' '11' '21,173' '4,929' '50' '8' '3' '51' '25' '100'
 '102' '320' '196' '683' '2' '17' '1,100' '241' '180' '59' '141,000' '19'
 '297' '316' '11,021' '157,617' '1,670' '140' '84' '920' '45' '6' '52'
 '250' '689' '1,549' '874' '494' '21' '92' '30,470' '23,466' '152' '72'
 '12,770' '12' '39' '41' '127' '27' '200' '57' '43,990' '756' '33' '90'
 '74' '78' '107' '4,961' '115' '144,000' '1,000' '1,400' '32' '105' '30'
 '42' '205' '17,878' '20' '29' '46' '29,540' '20,739' '330' '22' '132'
 '24,000' '743' '600' '328' '183' '34' '6,361' '62,730' '978' '249'
 '6,837' '2,090' '7' '54' '62' '14' '261' '354' '4,910' '24,579' '12,200'
 '400' '2,849' '18' '9' '201' '160,000' '350' '48' '85' '213' '60' '99'
 '47' '28' '59,044' '580' '69,080' '120' '552' '559' '696' '228' '12,587'
 '16' '55' '333' '262' '117' '304' '47,085' '60,000' '68' '6,177' '599'
 '70' '310' '150' '1,570' '165' '225' '198' '928' '50,000' '610' '3,830'
```

'11,500' '450' '37' '44' '160' '216' '240' '219' '32,737' '137' '458'
'1,069' '421' '23' '217,618' '3,095' '100,000' '3,000' '14,000' '453'
'73' '2,763' '397' '173' '374' '212' '752' '837' '635' '20,452' '155'
'15,083' '900' '416' '680' '850,816' '30,000' '36' '270' '4,924' '2,674'
'194' '688' '700' '300' '310,900' '223,160' '3,660' '210' '2,633' '4,403'
'2,747' '39,138' '728' '5,750' '186' '154' '176' '64,500' '21,500'
'51,250' '18,000' '141' '1,120' '2,915' '53' '151' '43' '46,000' '101'
'347' '420' '2,950' '86' '26' '591,780' '50,600' '4,000' '42,201'
'303,000' '255' '1,784' '284,015' '263' '11,000' '381' '12,930' '15,679'
'1,273' '12,500' '23,500' '685' '2,000' '3,500' '10,000' '349' '403'
'111,136' '8,100' '879' '1,995' '3,436' '96' '63' '454' '75' '461'
'1,420' '124' '40,874' '430' '52,901' '476' '1,350' '7,206' '6,919'
'23,334' '26,266' '199' '352' '390' '500' '2,479' '61' '148' '264' '142'
'5,300' '34,455' '615' '6,080' '82' '44,999' '4,240' '8,259' '169'
'2,991' '2,528' '790' '5,580' '720' '537' '778' '128' '340' '204' '193'
'203' '299' '125' '129' '38' '258' '730' '355' '79' '146' '113' '1,302'
'937,200' '20,700' '112,757' '9,996' '34,259' '12,700' '58' '97' '106'
'6,000' '1,200' '465' '67' '77' '145' '55,059' '36,633' '3,600' '904'
'13,000' '2,256,533' '4,021' '8,000' '3,955' '1,300' '145,277' '218' '93'
'83' '8,400' '1,720' '144' '7,800' '44,522' '35,820' '90,357' '693,400'
'4,650' '6,050' '9,900' '3,413' '6,576' '27,000' '191' '45,000' '82,844'
'73,718' '497,803' '86,075' '3,061' '26,656' '655' '8,950' '5,950'
'2,166' '564' '247' '1,444' '55,903' '603,250' '45,350' '86,400' '460'
'91' '2,684' '1,441' '267' '1,490' '7,665' '79,845' '71,530' '352,716'
'49,107' '27,952' '2,439,909' '762,226' '1,490,528' '834,380' '151,132'
'2,995,168' '139,287' '156,000' '496,230' '4,144,808' '2,040,754'
'2,049,538' '2,500' '49' '39,445' '84,000' '167' '139,950' '1,981'
'6,654' '1,013' '22,253' '39,884' '198,159' '591,328' '14,703' '218,967'
'875,428' '234,196' '1,102,769' '19,400' '2,435,870' '136,000'
'1,044,135' '439,070' '3,800' '12,000' '3,890,878' '63,110' '2,970'
'14,300' '1,500' '1,011' '185' '110' '360' '56' '3,586' '37,000'
'4,753,029' '117,540' '249,583' '594,882' '533,254' '4,282,820' '70,000'
'19,500' '48,683' '2,203,537' '524,437' '680,994' '3,910' '20,000'
'6,489' '521' '75,000' '3,400' '14,524' '138' '65' '50,289' '217'
'440,309' '29,708' '276,294' '18,390' '804,674' '510' '388,081' '35,000'
'834' '116,000' '382,050' '368' '66,000' '197,500' '155,000' '990'
'218,770' '103' '161' '110,314' '3,490' '247,713' '5,215,415' '1,208,132'
'136,670' '110,000' '3,526,341' '806,350' '277,264' '4,775,617' '309,184'
'208' '76' '152,000' '122' '6,448' '118,538' '18,650' '104,511' '28,590'
'6,064' '18,171' '17,310' '1,486,476' '414,682' '672,031' '331,200'
'334,400' '82,669' '174,163' '3,929,031' '410' '40,000' '1,505,991'
'19,940' '126,164' '5,400' '77,118' '257,312' '53,000' '1,063' '81'
'862,060' '342' '1,956,522' '133' '25,153' '220' '863,063' '1,020'
'2,755' '149,900' '31,540' '33,930' '148,700' '3,960' '220,380' '192,500'

```
 '4,534' '3,880' '192,000' '95,000']
Deaths: [nan '93' '108' ... '3,159' '13,976' '29,488']
Vaccinated: [nan '0' '68' ... '97,700' '158,250' '131,860']
```

In [14]:
```
# The numbers were written in french with commas. They are integers because it represents animals.
#So we will have to delete the commas, replace null values with 0 and transforme values to integers:

avian['Cases'] = avian['Cases'].str.replace(',', '')
avian['Cases'] = pd.to_numeric(avian['Cases'], errors='coerce').fillna(0).astype(int)

avian['Susceptible'] = avian['Susceptible'].str.replace(',', '')
avian['Susceptible'] = pd.to_numeric(avian['Susceptible'], errors='coerce').fillna(0).astype(int)


avian['Killed and disposed of'] = avian['Killed and disposed of'].str.replace(',', '')
avian['Killed and disposed of'] = pd.to_numeric(avian['Killed and disposed of'], errors='coerce').fillna(0).astype(int)
avian['Slaughtered'] = avian['Slaughtered'].str.replace(',', '')
avian['Slaughtered'] = pd.to_numeric(avian['Slaughtered'], errors='coerce').fillna(0).astype(int)

avian['Deaths']= avian['Deaths'].str.replace(',', '')
avian['Deaths'] = pd.to_numeric(avian['Deaths'], errors='coerce').fillna(0).astype(int)

avian['Vaccinated']= avian['Vaccinated'].str.replace(',', '')
avian['Vaccinated'] = pd.to_numeric(avian['Vaccinated'], errors='coerce').fillna(0).astype(int)

avian['New outbreaks']=avian['New outbreaks'].str.replace(',', '')
avian['New outbreaks'] = pd.to_numeric(avian['New outbreaks'], errors='coerce').fillna(0).astype(int)
```

In [15]:
```
avian.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 43089 entries, 0 to 43088
Data columns (total 18 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   Year                        43089 non-null  int64  
 1   Semester                    43089 non-null  object 
 2   World region                43089 non-null  object 
 3   Country                     43089 non-null  object 
 4   Administrative Division     43089 non-null  object 
 5   Disease                     43089 non-null  object 
 6   Serotype/Subtype/Genotype   12851 non-null  object 
 7   Animal Category             43089 non-null  object 
 8   Species                     25023 non-null  object 
 9   Outbreak_id                 4503 non-null   float64
 10  New outbreaks               43089 non-null  int32  
 11  Susceptible                 43089 non-null  int32  
 12  Measuring units             24963 non-null  object 
 13  Cases                       43089 non-null  int32  
 14  Killed and disposed of      43089 non-null  int32  
 15  Slaughtered                 43089 non-null  int32  
 16  Deaths                      43089 non-null  int32  
 17  Vaccinated                  43089 non-null  int32  
dtypes: float64(1), int32(7), int64(1), object(9)
memory usage: 4.8+ MB
```

Data exploration:

In [16]:
```
avian.describe()
```

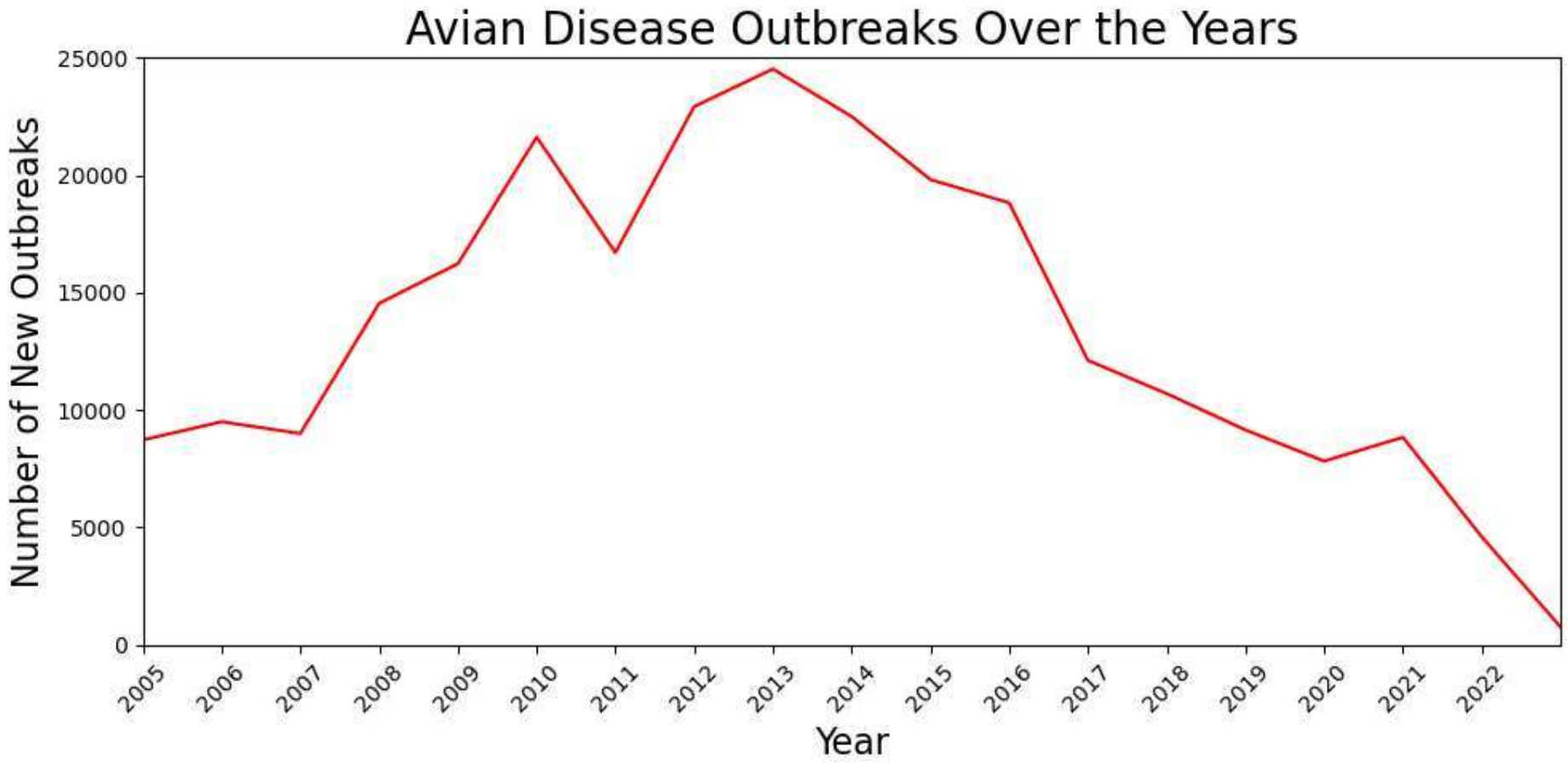
Out[16]:

	Year	Outbreak_id	New outbreaks	Susceptible	Cases	Killed and disposed of	Slaughtered	Deaths	Vaccinated
count	43089.000000	4503.000000	43089.000000	4.308900e+04	4.308900e+04	4.308900e+04	4.308900e+04	4.308900e+04	4.308900e+04
mean	2014.494674	102913.849656	6.009330	1.973358e+05	2.240895e+04	1.183465e+04	2.126127e+03	7.937363e+03	3.307593e+04
std	5.129286	12683.821199	58.000728	5.634785e+06	8.188160e+05	1.772139e+05	7.121194e+04	1.895099e+05	1.337743e+06
min	2005.000000	16.000000	0.000000	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	2010.000000	98690.500000	0.000000	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
50%	2015.000000	102590.000000	1.000000	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
75%	2018.000000	110288.500000	1.000000	2.000000e+03	1.740000e+02	0.000000e+00	0.000000e+00	3.700000e+01	0.000000e+00
max	2023.000000	122327.000000	3235.000000	5.291180e+08	6.250133e+07	1.995850e+07	5.215415e+06	1.124272e+07	1.800000e+08

1.What is the trend of avian disease outbreaks over the years?

In [17]:

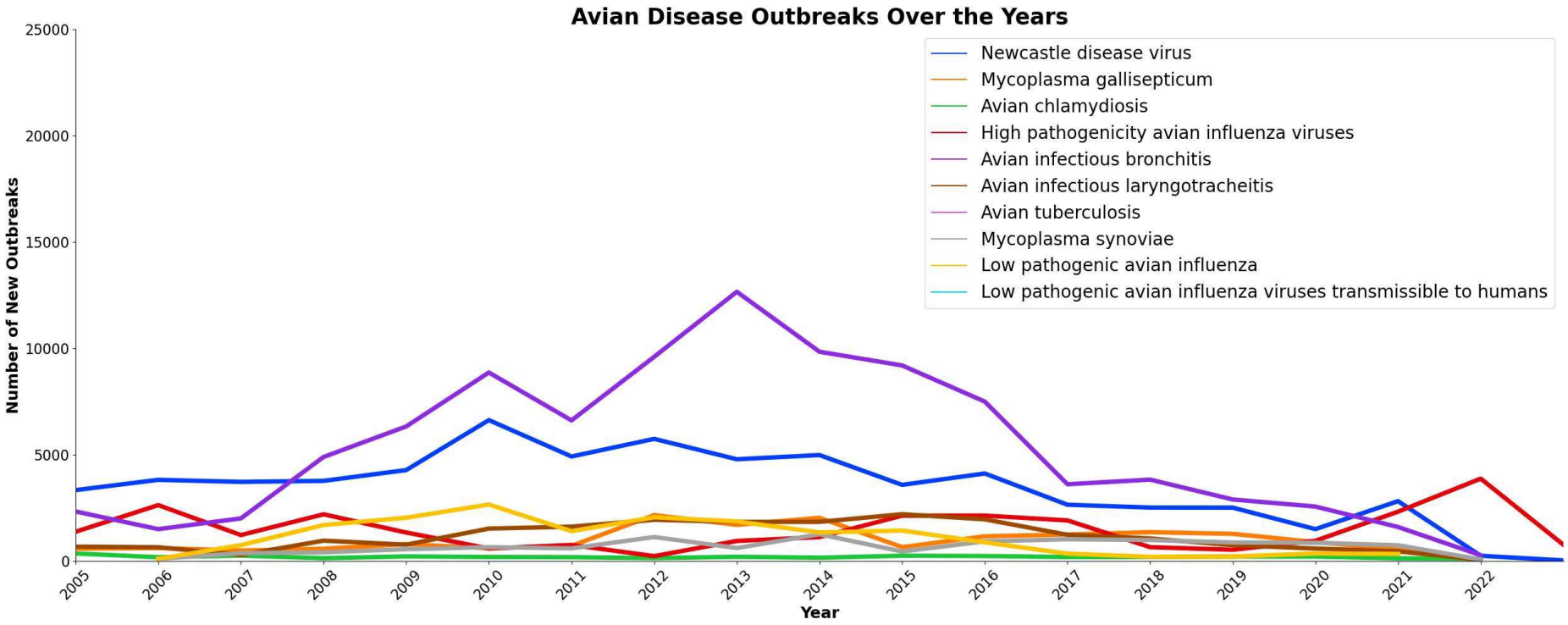
```
plt.figure(figsize = (10, 5))
sns.lineplot(
    data = avian ,
    x ='Year',
    y ="New outbreaks",
    errorbar=None,
    estimator = sum,
    color='red')

plt.ylim(0,25000)
plt.xlim(2005, 2023)
x_ticks = range(2005, 2023)
plt.xticks(x_ticks, x_ticks)
plt.xticks(rotation=45)
plt.xlabel('Year', fontsize=16)
plt.ylabel('Number of New Outbreaks',fontsize=16)
plt.title('Avian Disease Outbreaks Over the Years',fontsize=20)
plt.tight_layout()
plt.show()
```



Let's see the total number of new outbreaks per disease over the years:

In [18]:
```
plt.figure(figsize = (25, 10))
sns.lineplot(
    data = avian ,
    x ='Year',
    y ="New outbreaks",
    errorbar=None,
    estimator = sum,
     hue = "Disease",
    palette='bright',
 linewidth=5)
plt.ylim(0,25000)
```

```
plt.xlim(2005, 2023)
plt.yticks(fontsize=16)
x_ticks = range(2005, 2023)
plt.xticks(x_ticks, x_ticks, fontsize=16)
plt.xticks(rotation=45)
plt.xlabel('Year',fontsize=18,fontweight = "bold")
plt.ylabel('Number of New Outbreaks',fontsize=18,fontweight = "bold")
plt.title('Avian Disease Outbreaks Over the Years',fontsize=25,fontweight = "bold")
plt.legend(fontsize=20)
sns.despine()
plt.tight_layout()
plt.show()
```

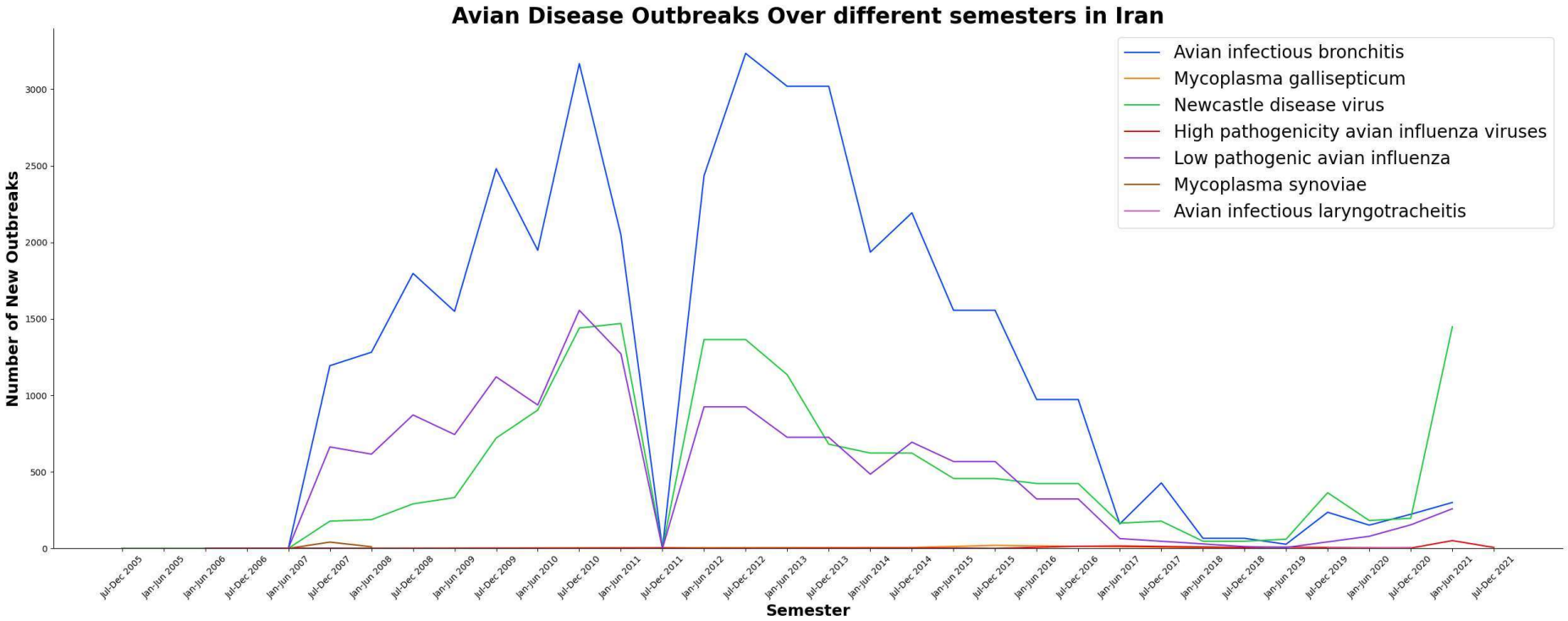


Let's analyze the trend of avian disease outbreaks over different semesters in Iran

In [19]:
```
#First, we have to make a query:
irn=avian.query("Country=='Iran'")
```

Create a line plot to visualize the data.

In [20]:
```
plt.figure(figsize = (25, 10))
sns.lineplot(
    data = irn ,
    x ='Semester',
    y ="New outbreaks",
    errorbar=None,
    estimator = sum,
     hue = "Disease",
    palette='bright')
 #linewidth=5)
plt.xticks(rotation=45)
plt.ylim(0)
plt.xlabel('Semester',fontsize=18,fontweight = "bold")
plt.ylabel('Number of New Outbreaks',fontsize=18,fontweight = "bold")
plt.title('Avian Disease Outbreaks Over different semesters in Iran',fontsize=25,fontweight = "bold")
plt.legend(fontsize=20)
sns.despine()
plt.tight_layout()
plt.show()
```



2.Which species were most affected by avian diseases?

A bar chart will be created to display the top 10 affected species:

In [21]:
```
#First of all, we have to explore how many species do we have and how many missing values:
avian['Species'].value_counts()
```

Out[21]:
```
Birds                              23724
Wildlife (species unspecified)       254
Rock Pigeon (Rock Dove)               71
Anatidae (unidentified)               56
Psittacidae (unidentified)            37
                                     ...
Marbled teal                           1
Little stint                           1
Little ringed plover                   1
Kentish plover                         1
European Serin                         1
Name: Species, Length: 231, dtype: int64
```

In [22]:
```
##we have 231 species, it would be diffiult to represent graphically.Let' see the total number of values in species:
avian['Species'].value_counts().sum()
```

Out[22]:
```
25023
```

In [23]:
```
# Let's see how many missing values are there:
avian['Species'].isnull().sum()
```

Out[23]:
```
18066
```

In [24]:
```
#Let's see the percentage of missing values
percentage_missing_values= round((avian['Species'].isnull().sum()/avian['Species'].value_counts().sum())*100,2)
percentage_missing_values
```

Out[24]:
```
72.2
```

In [25]:
```
#it's a great percentage. we cannot remove them.
```

In [26]:
```
#To represent the top 10 affected species, we have to create a sub-dataframe that represents the top 10 affected species:
top_affected_species = avian.groupby('Species')['Cases'].sum().nlargest(13)
top_affected_species
```

Out[26]:

```
Species
Birds                                965045104
Other species                           468574
Wildlife (species unspecified)           37493
Great Black-headed Gull                  14166
Columbidae (unidentified)                 3641
Anatidae (unidentified)                   1401
Phasianidae (unidentified)                 806
Common pheasant                            785
Ruddy Shelduck                             658
Psittacidae (unidentified)                 596
Yellow-legged Gull                         501
Tufted Duck                                277
Mute Swan                                  272
Name: Cases, dtype: int32
```

In [27]:

```
top_affected_dff = top_affected_species.reset_index()
top_affected_dff.columns = ['Species', 'Cases']
top_affected_dff.head(13)
```

Out[27]:

	Species	Cases
0	Birds	965045104
1	Other species	468574
2	Wildlife (species unspecified)	37493
3	Great Black-headed Gull	14166
4	Columbidae (unidentified)	3641
5	Anatidae (unidentified)	1401
6	Phasianidae (unidentified)	806
7	Common pheasant	785
8	Ruddy Shelduck	658
9	Psittacidae (unidentified)	596
10	Yellow-legged Gull	501
11	Tufted Duck	277
12	Mute Swan	272

There is a significant difference between the category of birds which does not represent a specific specie, other species and wildlife and all the rest of categories, which made it hard to represent visually. Therefore, we will remove birds, other species and wildlife:

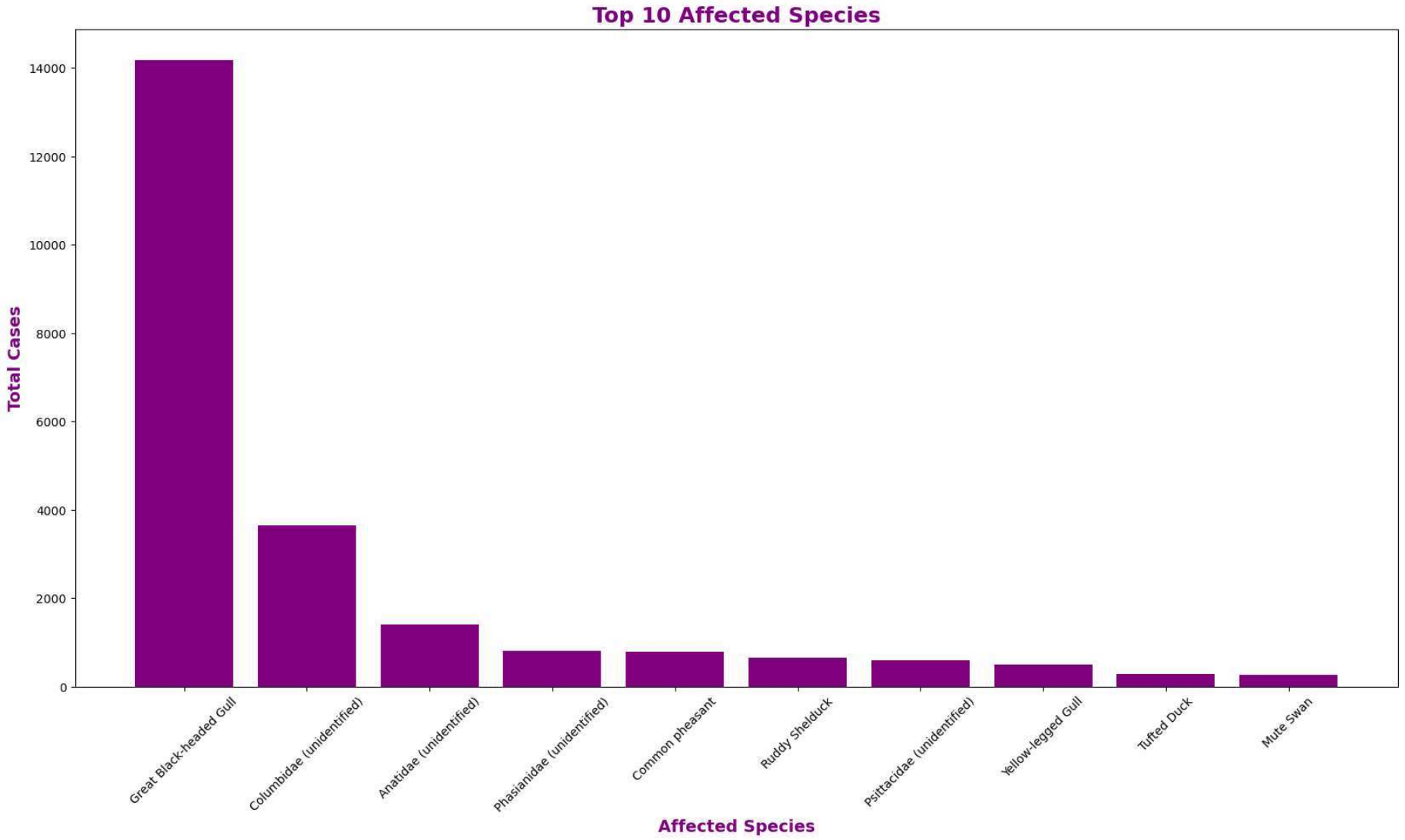
In [28]:
```
#We had to change the top 10 to 13 largest in the code above:
top_affected_ds = top_affected_dff.query("(Species != 'Birds') and (Species != 'Other species') and (Species != 'Wildlife (specie
top_affected_ds
```

Out[28]:

	Species	Cases
3	Great Black-headed Gull	14166
4	Columbidae (unidentified)	3641
5	Anatidae (unidentified)	1401
6	Phasianidae (unidentified)	806
7	Common pheasant	785
8	Ruddy Shelduck	658
9	Psittacidae (unidentified)	596
10	Yellow-legged Gull	501
11	Tufted Duck	277
12	Mute Swan	272

In [29]:

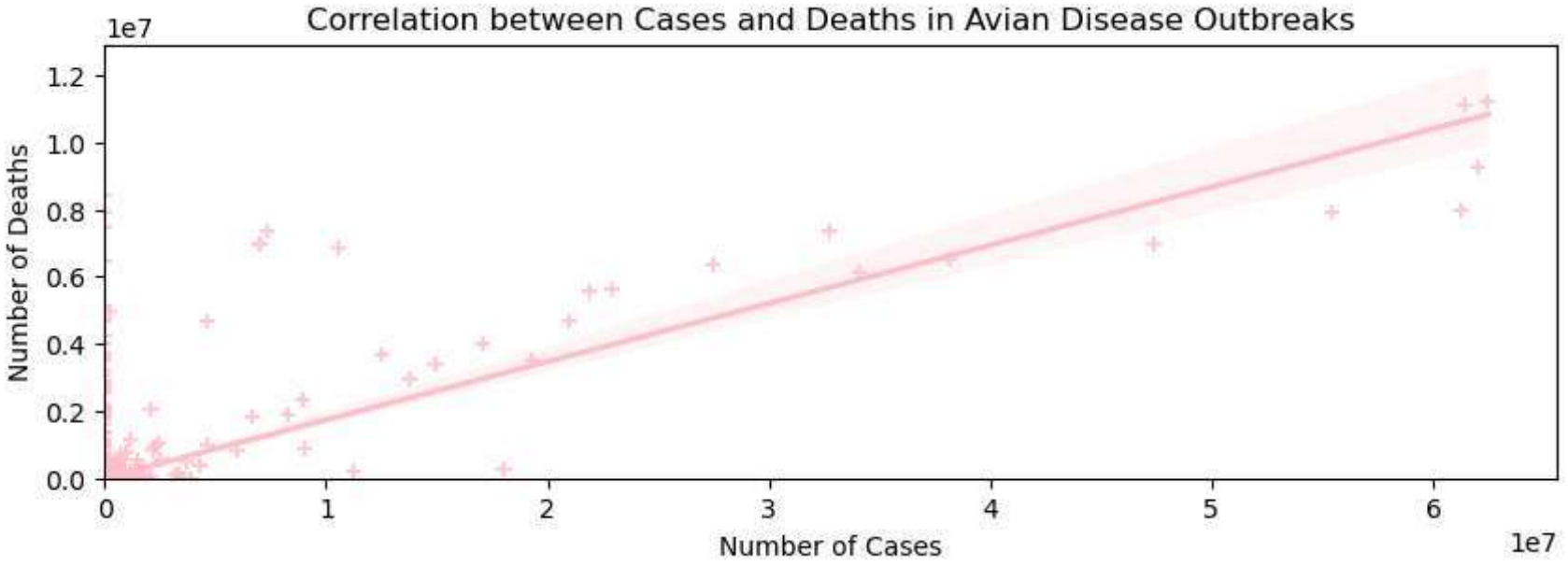
```
plt.figure(figsize=(20, 10))
plt.bar(data=top_affected_ds,
             x="Species",
             height="Cases", color='purple'
            )

plt.xticks(rotation=45)
plt.xlabel('Affected Species', fontsize=14 ,color='purple', fontweight='bold')
plt.ylabel('Total Cases',fontsize=14 ,color='purple', fontweight='bold')
plt.title('Top 10 Affected Species', fontsize=18 ,color='purple', fontweight='bold')
plt.show()
```



3.Is there any correlation between the number of cases and the number of deaths in avian diseases outbreaks?

Let's create a scatter plot to visualize the relationship.

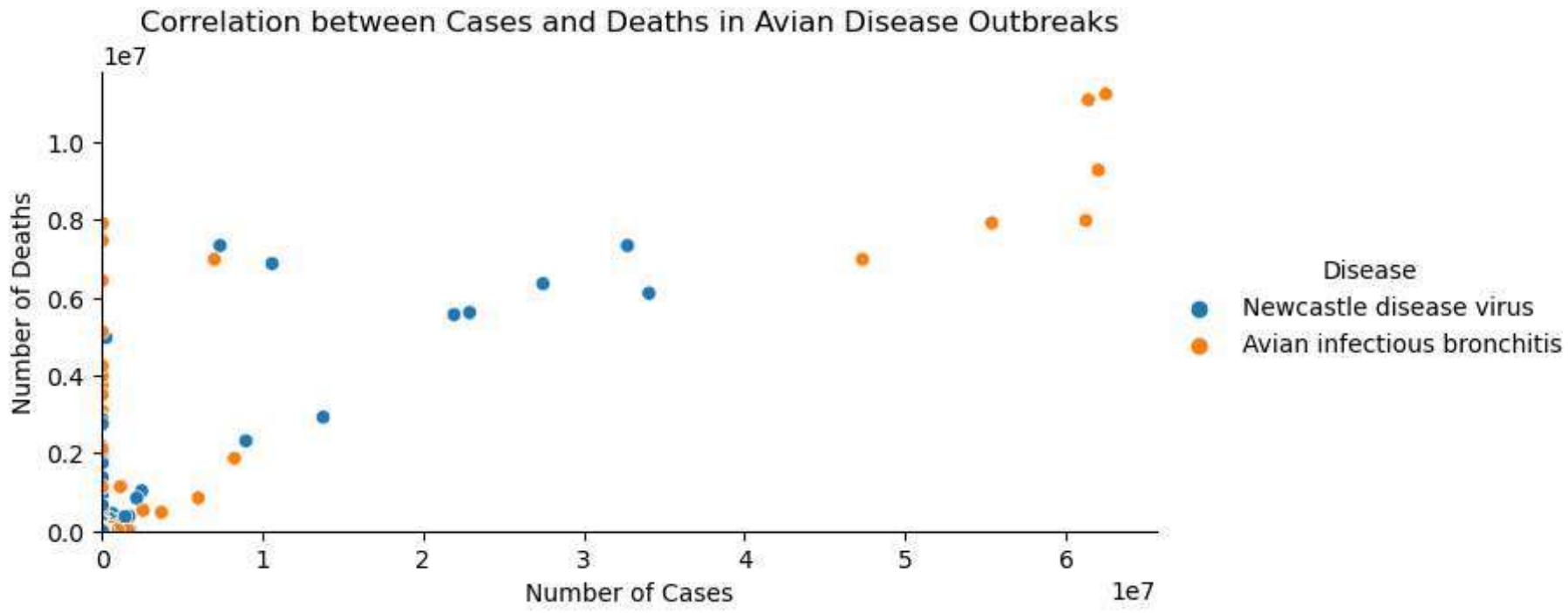
In [30]:
```
plt.figure(figsize=(10, 3))
sns.regplot(data = avian ,
             x ='Cases',
             y="Deaths",
             color = "pink",
           marker = '+')
plt.xlim(0)
plt.ylim(0)
plt.xlabel('Number of Cases')
plt.ylabel('Number of Deaths')
plt.title('Correlation between Cases and Deaths in Avian Disease Outbreaks')
plt.show()
```



Let's pick two diseases:Avian infectious bronchitis and Newcastle disease virus to observe the correlation between cases and deaths:

In [31]:
```
#First: create a subdataframe that contains the two diseases only:
diseases=['Avian infectious bronchitis','Newcastle disease virus']
df_avian=avian.query('Disease==@ diseases')
```

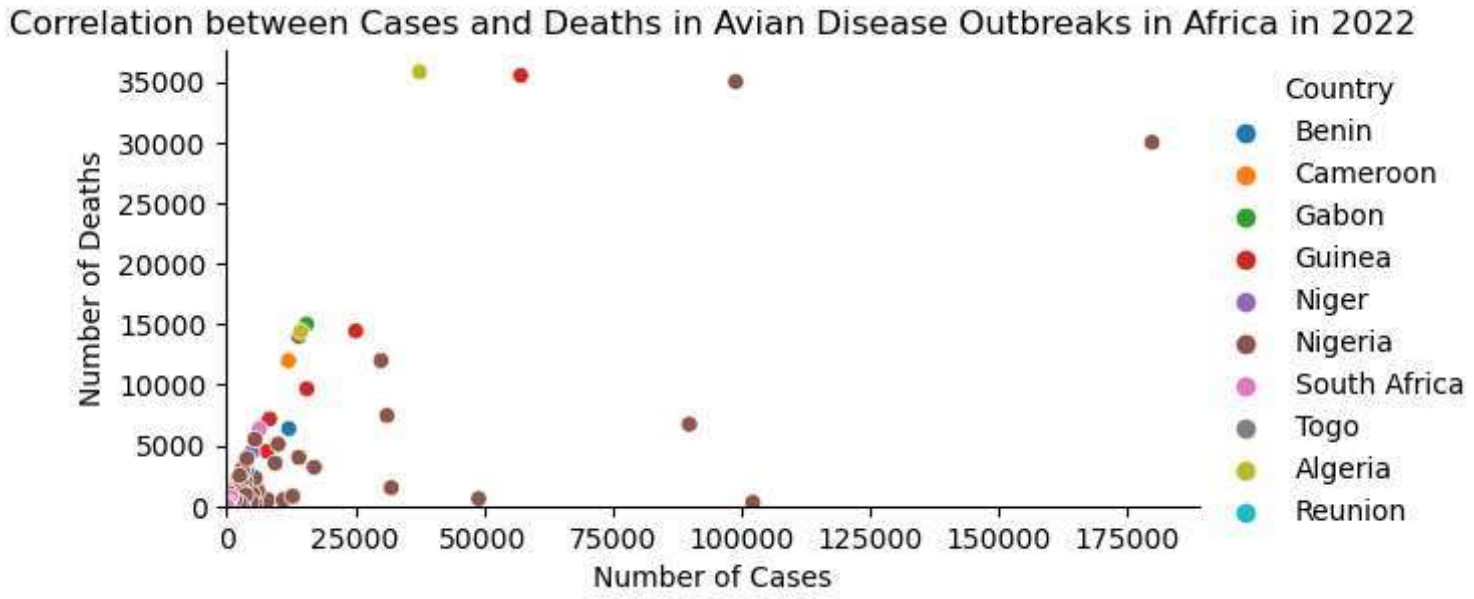
In [32]:
```
sns.relplot(data=df_avian,
            x="Cases",
            y="Deaths",
            hue="Disease",
            height = 3.5,
            aspect=2
           )
plt.xlim(0)
plt.ylim(0)
plt.xlabel('Number of Cases')
plt.ylabel('Number of Deaths')
plt.title('Correlation between Cases and Deaths in Avian Disease Outbreaks')
plt.show()
```



Let's choose african continent,High pathogenicity avian influenza viruses for the year 2022 and see the correlation between deaths and cases:

In [34]:
```
#create a dataframe from the query:
afr_influenza=avian.query("Year == 2022 and `World region` == 'Africa' and Disease == 'High pathogenicity avian influenza viruses
```

In [35]:
```
sns.relplot(data=afr_influenza,
            x="Cases",
            y="Deaths",
            hue='Country',
            height = 3,
            aspect=2
           )
plt.xlim(0)
plt.ylim(0)
plt.xlabel('Number of Cases')
plt.ylabel('Number of Deaths')
plt.title('Correlation between Cases and Deaths in Avian Disease Outbreaks in Africa in 2022')
plt.show()
```



4. How does the vaccination coverage affect the number of avian disease cases?

Calculate the correlation coefficient between 'Vaccinated' and 'Cases'.

In [36]:
```
correlation_coefficient = avian['Vaccinated'].corr(avian['Cases'])
print('correlation coefficient:', correlation_coefficient )
```

```
correlation coefficient: 6.894279314781178e-05
```

the correlation coefficient is close to 0 which indicates that there's little to no linear relationship between the two variables.

5. What is the overall fatality rate of avian disease outbreaks?

Calculate the percentage of deaths compared to total cases. But, we have to group cases and deaths by disease first

In [37]:
```
a=avian.groupby('Disease')['Cases','Deaths'].sum()
b=a.reset_index()
b
```

Out[37]:

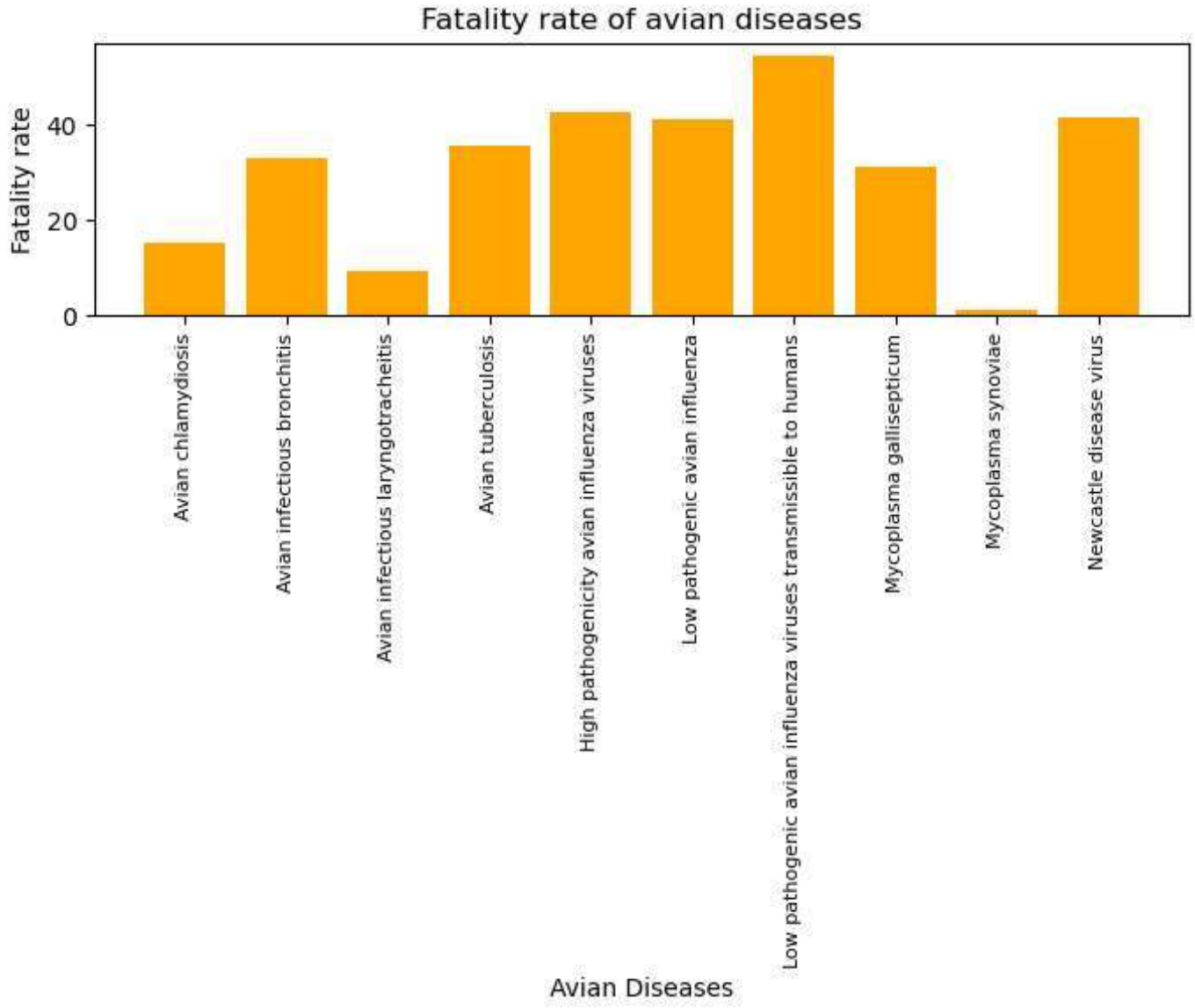
	Disease	Cases	Deaths
0	Avian chlamydiosis	48783	7385
1	Avian infectious bronchitis	413117367	135693538
2	Avian infectious laryngotracheitis	16676468	1544361
3	Avian tuberculosis	160	57
4	High pathogenicity avian influenza viruses	71867477	30526400
5	Low pathogenic avian influenza	162279324	66322804
6	Low pathogenic avian influenza viruses transmi...	10100	5480
7	Mycoplasma gallisepticum	42628187	13226579
8	Mycoplasma synoviae	31006196	396946
9	Newcastle disease virus	227945380	94289474

In [38]:
```
b['Fatality_rate']=round((b['Deaths']/b['Cases'])*100,2)
b
```

Out[38]:

	Disease	Cases	Deaths	Fatality_rate
0	Avian chlamydiosis	48783	7385	15.14
1	Avian infectious bronchitis	413117367	135693538	32.85
2	Avian infectious laryngotracheitis	16676468	1544361	9.26
3	Avian tuberculosis	160	57	35.62
4	High pathogenicity avian influenza viruses	71867477	30526400	42.48
5	Low pathogenic avian influenza	162279324	66322804	40.87
6	Low pathogenic avian influenza viruses transmi...	10100	5480	54.26
7	Mycoplasma gallisepticum	42628187	13226579	31.03
8	Mycoplasma synoviae	31006196	396946	1.28
9	Newcastle disease virus	227945380	94289474	41.36

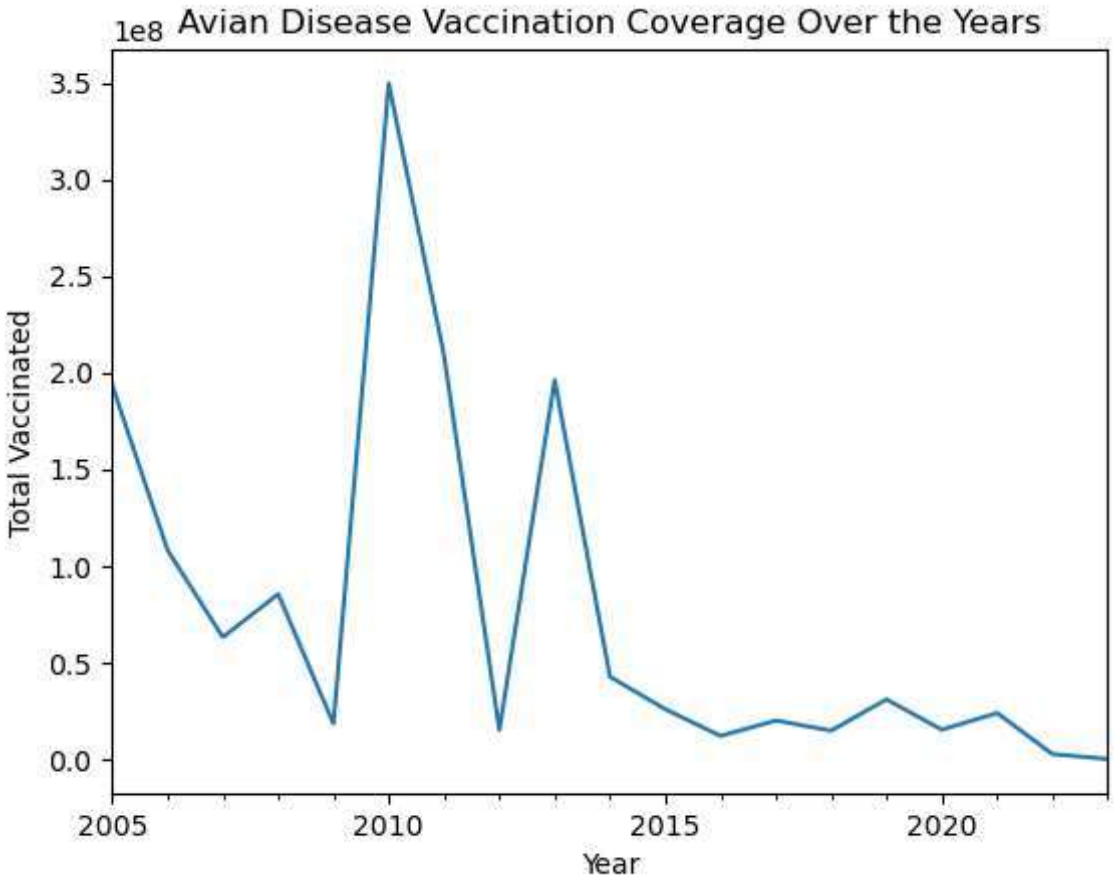
In [39]:
```
# We can visualize the fatality rate of each disease:
plt.figure(figsize=(8, 2))
plt.bar(data=b, x='Disease', height= 'Fatality_rate', color='orange')
plt.ylim(0)
plt.xlabel('Avian Diseases')
plt.ylabel('Fatality rate')
plt.title('Fatality rate of avian diseases')
plt.xticks(rotation=90, fontsize=8)
plt.show()
```



6. How has vaccination coverage evolved over the years?

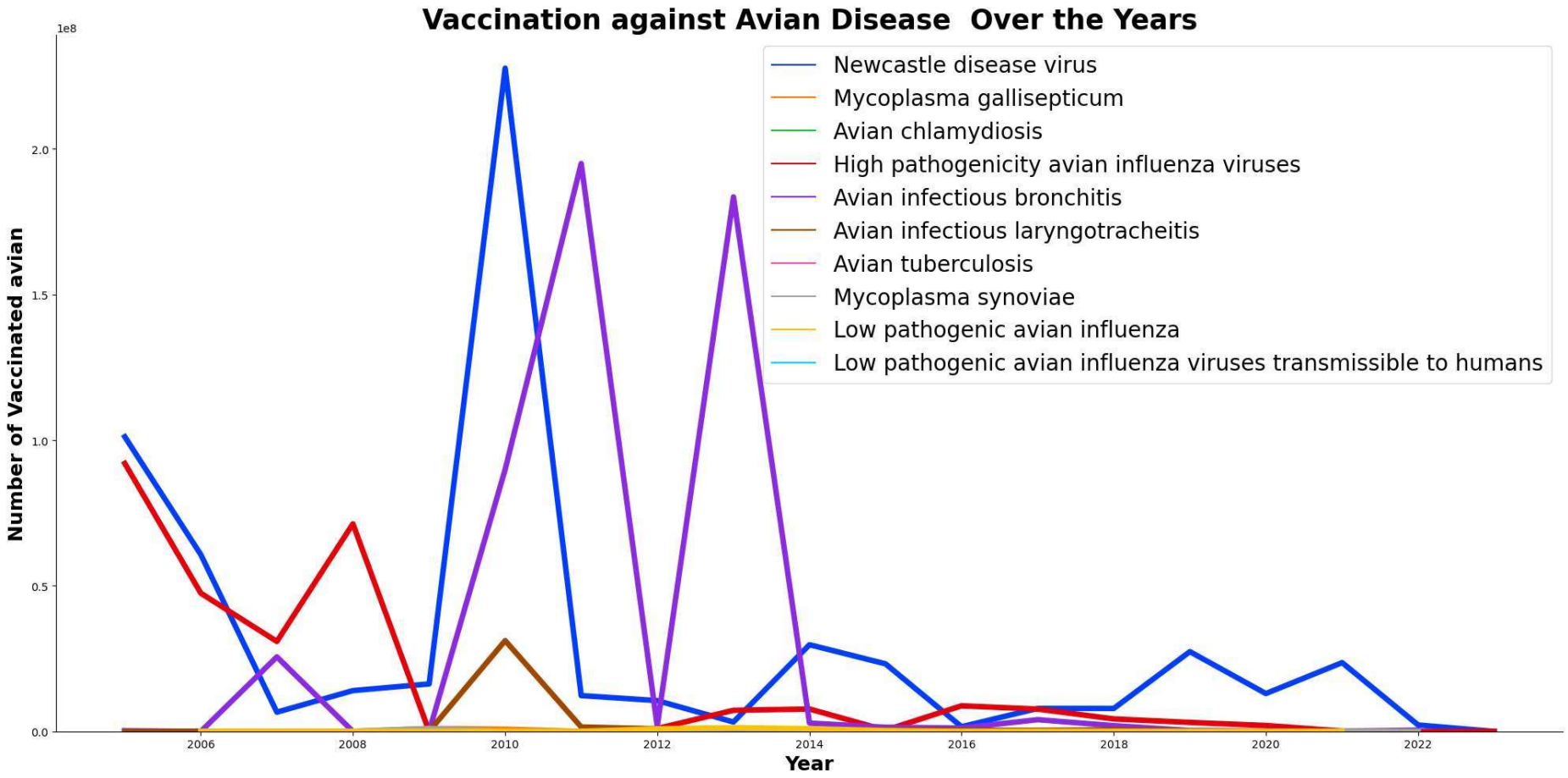
We can"t call the number of vaccinated , a vaccination coverage because we would need to calculate it as follow: number of vaccinated/ Number of avian species * 100 , wich is challenging. However, we will visualize the trends of number of vaccinated avian over the years:

In [40]:
```
# Creation of a line plot to visualize the trend.

avian['Year'] = pd.to_datetime(avian['Year'], format='%Y')
avian.groupby('Year')['Vaccinated'].sum().plot(kind='line')
plt.xlabel('Year')
plt.ylabel('Total Vaccinated')
plt.title('Avian Disease Vaccination Coverage Over the Years')
plt.show()
```



In [41]:
```
# Let's see the vaccination per disease:

plt.figure(figsize = (20, 10))
sns.lineplot(
    data = avian ,
```

```
    x ='Year',
    y ="Vaccinated",
    errorbar=None,
    estimator = sum,
     hue = "Disease",
    palette='bright',
 linewidth=5)
plt.ylim(0)
plt.xlabel('Year',fontsize=18,fontweight = "bold")
plt.ylabel('Number of Vaccinated avian',fontsize=18,fontweight = "bold")
plt.title('Vaccination against Avian Disease  Over the Years',fontsize=25,fontweight = "bold")
plt.legend(fontsize=20)
sns.despine()
plt.tight_layout()
plt.show()
```



7. What is the overall morbidity and mortality rate of avian diseases ?

Calculate the percentages of deaths and cases over all suscpetible avians: Mortality rate= Number of deaths/Number of susceptible animals
Morbidity rate= Number of cases/Number of susceptible animals

In [42]:
```
avv2=avian
avv2['Morbidity_rate']=round((avv2['Cases']/avv2['Susceptible'])*100,2)
avv2['Mortality_rate']=round((avv2['Deaths']/avv2['Susceptible'])*100,2)
avv2.head(100)
```

Out[42]:

	Year	Semester	World region	Country	Administrative Division	Disease	Serotype/Subtype/Genotype	Animal Category	Species	Outbreak_id	New outbreaks	Susceptibl
0	2005-01-01	Jul-Dec 2005	Africa	Angola	Huila	Newcastle disease virus	NaN	Both animal categories	NaN	NaN	1	
1	2005-01-01	Jul-Dec 2005	Africa	Angola	Huila	Newcastle disease virus	NaN	Domestic	Birds	NaN	0	21
2	2005-01-01	Jul-Dec 2005	Africa	Benin	Adjara	Newcastle disease virus	NaN	Both animal categories	NaN	NaN	1	
3	2005-01-01	Jul-Dec 2005	Africa	Benin	Adjara	Newcastle disease virus	NaN	Domestic	Birds	NaN	0	1000
4	2005-01-01	Jul-Dec 2005	Africa	Benin	Adjohoun	Newcastle disease virus	NaN	Both animal categories	NaN	NaN	1	
...	...	...	...	...	...	...	...	...	...	...	...	
95	2005-01-01	Jul-Dec 2005	Africa	Nigeria	Nigeria	Newcastle disease virus	NaN	Both animal categories	NaN	NaN	5	
96	2005-01-01	Jul-Dec 2005	Africa	Nigeria	Nigeria	Newcastle disease virus	NaN	Domestic	Birds	NaN	0	2210
97	2005-01-01	Jul-Dec 2005	Africa	South Africa	South Africa	Avian chlamydiosis	NaN	Both animal categories	NaN	NaN	1	
98	2005-01-01	Jul-Dec 2005	Africa	South Africa	South Africa	Avian chlamydiosis	NaN	Domestic	Birds	NaN	0	
99	2005-01-01	Jul-Dec 2005	Africa	South Africa	South Africa	Mycoplasma gallisepticum	NaN	Both animal categories	NaN	NaN	1	

100 rows × 20 columns

In [43]:
```
# Calculate the mean value of the mortality rate and the morbidity rate of each disease:
perc=avv2.groupby('Disease')['Mortality_rate','Morbidity_rate'].agg('mean')
perc
```

Out[43]:

Disease	Mortality_rate	Morbidity_rate
Avian chlamydiosis	NaN	NaN
Avian infectious bronchitis	NaN	NaN
Avian infectious laryngotracheitis	NaN	NaN
Avian tuberculosis	NaN	NaN
High pathogenicity avian influenza viruses	NaN	NaN
Low pathogenic avian influenza	NaN	NaN
Low pathogenic avian influenza viruses transmissible to humans	6.795	11.255
Mycoplasma gallisepticum	NaN	NaN
Mycoplasma synoviae	NaN	NaN
Newcastle disease virus	NaN	NaN

We could not calculate the average of the mortality rate and the morbidity rate because there were rows considered "inf" or "NaN" due to th lack of information regarding either susceptible or deaths and cases which were considered zero(0) eralier. Therefore, we have to delete all rows with this value "inf" in order to calculate the rate of mortality and morbidity.

In [44]:
```
#First, let's find out how many rows with 'inf' we have:
inf_count_overall = avv2['Mortality_rate'].isin([np.inf]) | avv2['Morbidity_rate'].isin([np.inf])
# Get the total count of rows with 'inf' values
total_inf_count = inf_count_overall.sum()
total_inf_count
```

Out[44]:
```
4718
```

In [45]:
```
#delete rows with 'inf':
avv2 = avv2.drop(avv2[avv2.isin([np.inf]).any(axis=1)].index)
```

```
avv2
```

Out[45]:

	Year	Semester	World region	Country	Administrative Division	Disease	Serotype/Subtype/Genotype	Animal Category	Species	Outbreak_id	New outbreaks	Susce
0	2005-01-01	Jul-Dec 2005	Africa	Angola	Huila	Newcastle disease virus	NaN	Both animal categories	NaN	NaN	1	
1	2005-01-01	Jul-Dec 2005	Africa	Angola	Huila	Newcastle disease virus	NaN	Domestic	Birds	NaN	0	
2	2005-01-01	Jul-Dec 2005	Africa	Benin	Adjara	Newcastle disease virus	NaN	Both animal categories	NaN	NaN	1	
3	2005-01-01	Jul-Dec 2005	Africa	Benin	Adjara	Newcastle disease virus	NaN	Domestic	Birds	NaN	0	
4	2005-01-01	Jul-Dec 2005	Africa	Benin	Adjohoun	Newcastle disease virus	NaN	Both animal categories	NaN	NaN	1	
...	...	...	...	...	...	...	...	...	...	...	...	
43084	2023-01-01	Jul-Dec 2023	Europe	Poland	Dobrzyniewo Du?e	Newcastle disease virus	NaN	Domestic	Birds	122327.0	1	
43085	2023-01-01	Jul-Dec 2023	Europe	Poland	Turo?? Ko? cielna	Newcastle disease virus	NaN	Domestic	Birds	121679.0	1	
43086	2023-01-01	Jul-Dec 2023	Europe	Poland	Turo?? Ko? cielna	Newcastle disease virus	NaN	Domestic	Birds	122325.0	1	
43087	2023-01-01	Jul-Dec 2023	Europe	Poland	Turo?? Ko? cielna	Newcastle disease virus	NaN	Domestic	Birds	122326.0	1	
43088	2023-01-01	Jul-Dec 2023	Europe	United Kingdom	Aberdeenshire	High pathogenicity avian influenza viruses	H5N1	Domestic	Birds	121841.0	1	

38371 rows × 20 columns

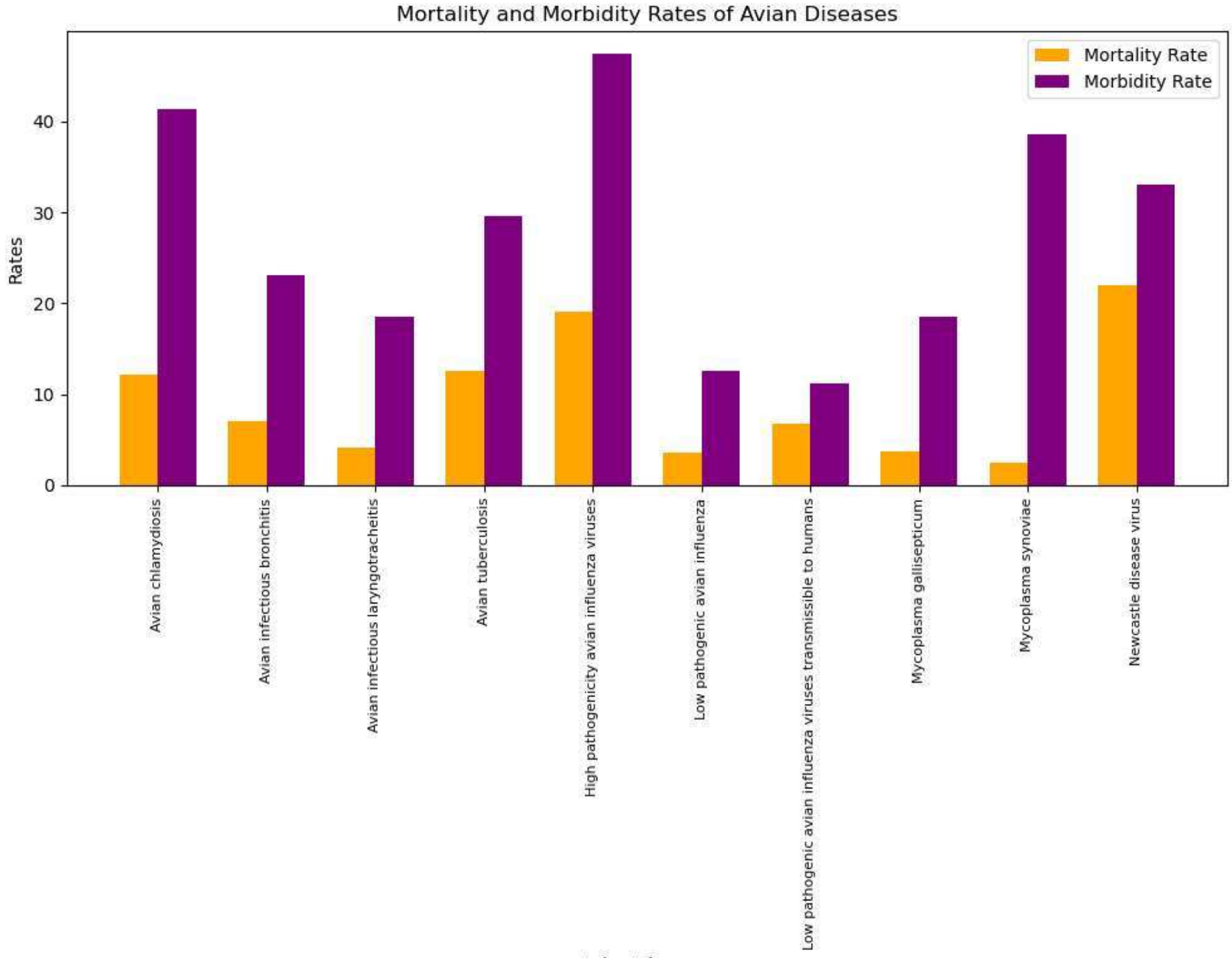
In [46]:
```
# Calculate the mean value of the mortality rate and the morbidity rate of each disease:
perct=avv2.groupby('Disease')['Mortality_rate','Morbidity_rate'].mean()
rates=perct.round(2).reset_index()
rates
```

Out[46]:

	Disease	Mortality_rate	Morbidity_rate
0	Avian chlamydiosis	12.19	41.34
1	Avian infectious bronchitis	6.99	23.07
2	Avian infectious laryngotracheitis	4.09	18.59
3	Avian tuberculosis	12.56	29.66
4	High pathogenicity avian influenza viruses	19.11	47.52
5	Low pathogenic avian influenza	3.54	12.63
6	Low pathogenic avian influenza viruses transmi...	6.80	11.26
7	Mycoplasma gallisepticum	3.70	18.50
8	Mycoplasma synoviae	2.52	38.53
9	Newcastle disease virus	22.05	33.09

In [47]:
```
# We can visualize the mortality and morbidity rate of each disease:


# Calculate the number of bars
num_bars = len(rates)

# Set the width of the bars
bar_width = 0.35

# Create an array of indexes for the bars
ind = np.arange(num_bars)

# Create the figure and axes
plt.figure(figsize=(10, 8))
```

```
# Create grouped bars for 'Mortality_rate' and 'Morbidity_rate'
plt.bar(ind, rates['Mortality_rate'], width=bar_width, label='Mortality Rate', color='orange')
plt.bar(ind + bar_width, rates['Morbidity_rate'], width=bar_width, label='Morbidity Rate', color='purple')
plt.xlabel('Avian Diseases')
plt.ylabel('Rates')
plt.title('Mortality and Morbidity Rates of Avian Diseases')
plt.xticks(ind + bar_width / 2, rates['Disease'], rotation=90, fontsize=8)
plt.legend()

plt.tight_layout()
plt.show()
```


Mortality and Morbidity Rates of Avian Diseases
Mortality Rate
Morbidity Rate
Rates
0
10
20
30
40
Avian chlamydiosis
Avian infectious bronchitis
Avian infectious laryngotracheitis
Avian tuberculosis
High pathogenicity avian influenza viruses
Low pathogenic avian influenza
Low pathogenic avian influenza viruses transmissible to humans
Mycoplasma gallisepticum
Mycoplasma synoviae
Newcastle disease virus

8.Which animal category is most vulnerable to avian diseases?

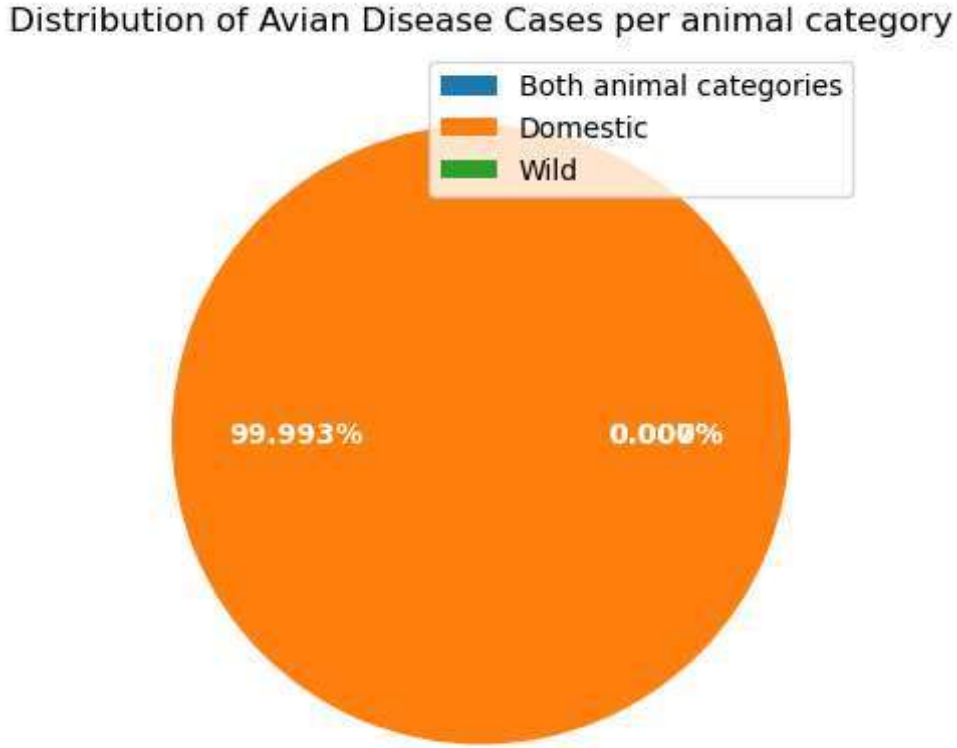
Create a pie chart to visualize the distribution of cases among animal categories.

In [48]:
```
animal_category_cases = avian.groupby('Animal Category')['Cases'].sum()
anil_cat = animal_category_cases.reset_index()
anil_cat
```

Out[48]:

	Animal Category	Cases
0	Both animal categories	0
1	Domestic	965515745
2	Wild	63697

In [49]:
```
plt.figure(figsize = (5, 5))
labels = anil_cat['Animal Category'].values
plt.pie(anil_cat['Cases'],
        autopct = '%.3f%%',
        textprops = {
                     'fontweight' : 'bold',
                     'color' : 'w'})
plt.legend(labels)
plt.title('Distribution of Avian Disease Cases per animal category')
plt.show()
```


Distribution of Avian Disease Cases per animal category
Both animal categories
Domestic
Wild
99.993%
0.000%