Data Cleaning

In [5]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
import os
for dirname, _, filenames in os.walk('animal_diet.csv'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

In [6]:
```
df = pd.read_csv('animal_diet.csv')
```

In [8]:
```
df.shape
```

Out[8]:
```
(63, 12)
```

In [13]:
```
df.head(63).style.set_properties(**{'background-color': 'yellow',
                                    'color': 'black'})
```

Out[13]:

	Animal	Height (cm)	Weight (kg)	Color	Lifespan (years)	Diet	Habitat	Conservation Status
0	Elephant	270-310	2700-6000	Grey	60-70	Herbivore	Savannah, Forest	Vulnerable
1	Lion	80-110	120-250	Tan	Oct-14	Carnivore	Grasslands, Savannas	Vulnerable
2	Wild Dog	75-80	18-36	Multicolored	10-Dec	Carnivore	Savannahs	Endangered
3	Bison	152-186	318-1000	Brown	15-20	Herbivore	Grasslands, Plains	Near Threatened
4	Anteater	52-91	22-41	Brown, White	15-20	Insectivore	Grasslands, Forests	Least Concern
5	Horse	140-160	380-1000	Various	25-30	Herbivore	Middle East, North Africa	Not Applicable
6	Fox	25-30	2.5-9	White	03-Jun	Omnivore	Tundra	Least Concern
7	Bengal Fox	35-40	2.5-4	Yellowish-gray	06-Aug	Omnivore	Grasslands	Least Concern
8	Bengal Tiger	90-110	220-260	Orange, Black	Oct-15	Carnivore	Grasslands, Mangroves	Endangered
9	Black Rhinoceros	132-180	800-1400	Gray, Black	35-50	Herbivore	Grasslands, Forests	Critically Endangered
10	Bornean Orangutan	96-150	30-82	Reddish-brown	35-45	Omnivore	Rainforests	Critically Endangered
11	Brown Bear	70-120	70-780	Brown	20-30	Omnivore	Forests, Mountains	Least Concern
12	Burmese Python	Up to 460	Up to 90	Brown, Black	20-25	Carnivore	Grasslands, Forests	Least Concern
13	Cheetah	70-90	20-72	Tan with Black Spots	10-Dec	Carnivore	Grasslands, Savannas	Vulnerable
14	Chimpanzee	Up to 160	26-70	Black	40-50	Omnivore	Rainforests, Grasslands	Endangered
15	Dalmatian Dog	50-61	23-32	White, Black	Oct-13	Omnivore	Worldwide	Not Applicable
16	Dhole	50-55	13-20	Red, Brown	Oct-15	Carnivore	Forests, Grasslands	Endangered
17	Dingo	48-58	13-20	Tan, Brown	Oct-15	Carnivore	Grasslands, Forests	Least Concern
18	Eastern Gorilla	140-188	73-204	Black	40-50	Herbivore	Rainforests	Endangered
19	Echidna	30-45	2.5-6.5	Brown, Tan	15-20	Insectivore	Forests,	Least Concern

	Animal	Height (cm)	Weight (kg)	Color	Lifespan (years)	Diet	Habitat	Conservation Status
							Grasslands	
20	Emperor Penguin	100	22-45	Black, White	15-20	Carnivore	Antarctic	Near Threatened
21	Emperor Tamarin	23-26	0.3-0.4	Black, White	Unknown	Omnivore	Rainforests	Least Concern
22	European Hedgehog	14-30	0.6-1.2	Brown, Spiny	02-May	Omnivore	Gardens, Grasslands	Least Concern
23	Fennec Fox	20-40	0.8-1.6	Cream	Unknown	Omnivore	Sahara Desert	Least Concern
24	Flying Fox	Up to 40	Up to 1.1	Brown, Gray	15-23	Herbivore	Forests	Least Concern
25	Fossa	70-90	Up to 5	Brown	15-20	Carnivore	Rainforests, Madagascar	Vulnerable
26	Gal�pagos Tortoise	Up to 183	Up to 417	Brown, Grey	100-150	Herbivore	Gal�pagos Islands	Vulnerable
27	King Cobra	Up to 550	Up to 9	Olive-green	20-30	Carnivore	Grasslands, Forests	Vulnerable
28	Lemur	30-65	0.6-4.5	Varies	15-25	Herbivore	Forests	Critically Endangered
29	Lion-tailed Macaque	45-60	2.5-10	Black	20-30	Omnivore	Western Ghats, India	Endangered
30	Malayan Krait	Up to 150	Up to 2.2	Black, Yellow	Oct-15	Carnivore	Forests, Grasslands	Least Concern
31	Mandrill	Up to 75	Up to 55	Brown, Green	20-40	Omnivore	Rainforests	Vulnerable
32	Maned Wolf	67-107	20-25	Reddish-Brown	06-Aug	Omnivore	Grasslands, Forests	Near Threatened
33	Markhor	Up to 115	Up to 110	Brown, Gray	Oct-13	Herbivore	Mountains	Near Threatened
34	Meerkat	25-35	0.6-1	Brown, Tan	Dec-14	Omnivore	Arid regions	Least Concern
35	Mountain Gorilla	120-200	70-200	Black, Brown	35-40	Herbivore	Rainforests, Mountains	Endangered
36	Naked Mole Rat	Up to 9	Up to 80	Pink, Wrinkled	Up to 32	Herbivore	Underground Tunnels	Least Concern
37	Slow Loris	20-38	0.3-1.6	Yellowish-Brown	20-24	Omnivore	Forests	Vulnerable
38	Snow Leopard	50-70	22-75	Grayish-yellow, Black	10	Carnivore	Mountains, Steppes	Vulnerable
39	Spectacled Bear	120-200	35-77	Black, Tan	20-25	Omnivore	Forests	Vulnerable

	Animal	Height (cm)	Weight (kg)	Color	Lifespan (years)	Diet	Habitat	Conservation Status
40	Spider Monkey	35-64	06-Sep	Brown, Black	25-36	Omnivore	Rainforests	Vulnerable
41	Spotted Hyena	70-92	44-64	Yellowish-brown	90-110	Carnivore	Grasslands, Savannahs	Least Concern
42	Squirrel Monkey	25-35	0.5-1	Yellowish-Orange, Black	15-20	Omnivore	Rainforests	Least Concern
43	Star-Nosed Mole	Up to 16	Up to 0.06	Black, Pink	16-22	Insectivore	Wetlands, Forests	Least Concern
44	Sumatran Orangutan	66-100	23-56	Reddish-Brown	30-50	Omnivore	Rainforests	Critically Endangered
45	Sumatran Rhino	100-145	750-950	Reddish-Brown	430-640	Herbivore	Rainforests	Critically Endangered
46	Sumatran Rhinoceros	112-145	500-950	Brown	425-475	Herbivore	Rainforests	Critically Endangered
47	Sumatran Tiger	70-87	47-121	Orange, Black Stripes	140-230	Carnivore	Rainforests	Critically Endangered
48	Sun Bear	120-150	27-80	Black	25-30	Omnivore	Tropical Forests	Vulnerable
49	Tapir	70-120	150-400	Brown, Black	390-395	Herbivore	Rainforests, Grasslands	Vulnerable
50	Tarsier	Up to 16	Up to 0.2	Gray, Brown	05-Jul	Insectivore	Rainforests	Vulnerable
51	Tasmanian Devil	30-76	14	Black, White	21-28	Carnivore	Forests, Grasslands	Endangered
52	Tasmanian Tiger	100-130	20-30	Yellow-brown	121-128	Carnivore	Australia	Extinct
53	Three-Toed Sloth	46-58	Up to 8	Brown, Gray	30-40	Herbivore	Rainforests	Least Concern
54	Giraffe	400-600	800-1600	Brown, Yellow	20-25	Herbivore	Savannas, Grasslands	Vulnerable
55	Cow	150	300	Brown	20	Herbivore	Grasslands	Least Concern
56	Bull	170	400	Black	18	Herbivore	Meadows	Near Threatened
57	Buffalo	160	350	White	22	Herbivore	Pastures	Vulnerable
58	Yak	155	320	Spotted	15	Herbivore	Savannas	Endangered
59	Banteng	180	450	Red	25	Herbivore	Woodlands	Critically Endangered

	Animal	Height (cm)	Weight (kg)	Color	Lifespan (years)	Diet	Habitat	Conservation Status
60	Wildebeest	145	280	Tan	17	Herbivore	Plains	Extinct
61	Zebu	140	260	Gray	22	Herbivore	Farmlands	Vulnerable
62	Highland Cattle	152	320	Ginger	18	Herbivore	Highlands	Least Concern

In [14]:
```
pd.DataFrame(df.columns,columns=['column name'])
```

Out[14]:

	column name
0	Animal
1	Height (cm)
2	Weight (kg)
3	Color
4	Lifespan (years)
5	Diet
6	Habitat
7	Conservation Status
8	Family
9	Gestation Period (days)
10	Feeding Behavior
11	Nutritional Content

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 63 entries, 0 to 62
Data columns (total 12 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   Animal                   63 non-null     object
 1   Height (cm)              63 non-null     object
 2   Weight (kg)              63 non-null     object
 3   Color                    63 non-null     object
 4   Lifespan (years)         63 non-null     object
 5   Diet                     63 non-null     object
 6   Habitat                  63 non-null     object
 7   Conservation Status      63 non-null     object
 8   Family                   63 non-null     object
 9   Gestation Period (days)  63 non-null     object
 10  Feeding Behavior         63 non-null     object
 11  Nutritional Content      63 non-null     object
dtypes: object(12)
memory usage: 6.0+ KB
```

Data Cleaning

In [16]:
```
df.columns
```

Out[16]:
```
Index(['Animal', 'Height (cm)', 'Weight (kg)', 'Color', 'Lifespan (years)',
       'Diet', 'Habitat', 'Conservation Status', 'Family',
       'Gestation Period (days)', 'Feeding Behavior', 'Nutritional Content'],
      dtype='object')
```

In [17]:
```
df
```

Out[17]:

	Animal	Height (cm)	Weight (kg)	Color	Lifespan (years)	Diet	Habitat	Conservation Status	
0	Elephant	270-310	2700-6000	Grey	60-70	Herbivore	Savannah, Forest	Vulnerable	
1	Lion	80-110	120-250	Tan	Oct-14	Carnivore	Grasslands, Savannas	Vulnerable	
2	Wild Dog	75-80	18-36	Multicolored	10-Dec	Carnivore	Savannahs	Endangered	
3	Bison	152-186	318-1000	Brown	15-20	Herbivore	Grasslands, Plains	Near Threatened	
4	Anteater	52-91	22-41	Brown, White	15-20	Insectivore	Grasslands, Forests	Least Concern	Myrr
...	...	...	...	...	...	...	...	...	
58	Yak	155	320	Spotted	15	Herbivore	Savannas	Endangered	
59	Banteng	180	450	Red	25	Herbivore	Woodlands	Critically Endangered	
60	Wildebeest	145	280	Tan	17	Herbivore	Plains	Extinct	
61	Zebu	140	260	Gray	22	Herbivore	Farmlands	Vulnerable	
62	Highland Cattle	152	320	Ginger	18	Herbivore	Highlands	Least Concern	

63 rows × 12 columns

In [23]:
```
columns_with_upto=['Height (cm)','Weight (kg)','Lifespan (years)','Gestation Perioc
```

In [22]:
```
df
```

Out[22]:

	Animal	Height (cm)	Weight (kg)	Color	Lifespan (years)	Diet	Habitat	Conservation Status	
0	Elephant	270-310	2700-6000	Grey	60-70	Herbivore	Savannah, Forest	Vulnerable	
1	Lion	80-110	120-250	Tan	Oct-14	Carnivore	Grasslands, Savannas	Vulnerable	
2	Wild Dog	75-80	18-36	Multicolored	10-Dec	Carnivore	Savannahs	Endangered	
3	Bison	152-186	318-1000	Brown	15-20	Herbivore	Grasslands, Plains	Near Threatened	
4	Anteater	52-91	22-41	Brown, White	15-20	Insectivore	Grasslands, Forests	Least Concern	Myrr
...	...	...	...	...	...	...	...	...	
58	Yak	155	320	Spotted	15	Herbivore	Savannas	Endangered	
59	Banteng	180	450	Red	25	Herbivore	Woodlands	Critically Endangered	
60	Wildebeest	145	280	Tan	17	Herbivore	Plains	Extinct	
61	Zebu	140	260	Gray	22	Herbivore	Farmlands	Vulnerable	
62	Highland Cattle	152	320	Ginger	18	Herbivore	Highlands	Least Concern	

63 rows × 12 columns

In [25]:
```
df.columns
```

Out[25]:
```
Index(['Animal', 'Height (cm)', 'Weight (kg)', 'Color', 'Lifespan (years)',
       'Diet', 'Habitat', 'Conservation Status', 'Family',
       'Gestation Period (days)', 'Feeding Behavior', 'Nutritional Content'],
      dtype='object')
```

In [27]:
```
for x in ['Animal', 'Height (cm)', 'Weight (kg)', 'Color', 'Lifespan (years)',
       'Diet', 'Habitat', 'Conservation Status', 'Family',
       'Gestation Period (days)', 'Feeding Behavior', 'Nutritional Content']:
    ab=df[x][df[x].str.contains('-')]
    df[x][df[x].str.contains('-')]=ab.str.split('-').str.get(-1)
```

In [28]:
```
df
```

Out[28]:

	Animal	Height (cm)	Weight (kg)	Color	Lifespan (years)	Diet	Habitat	Conservation Status	
0	Elephant	310	6000	Grey	70	Herbivore	Savannah, Forest	Vulnerable	
1	Lion	110	250	Tan	14	Carnivore	Grasslands, Savannas	Vulnerable	
2	Wild Dog	80	36	Multicolored	Dec	Carnivore	Savannahs	Endangered	
3	Bison	186	1000	Brown	20	Herbivore	Grasslands, Plains	Near Threatened	
4	Anteater	91	41	Brown, White	20	Insectivore	Grasslands, Forests	Least Concern	Myrr
...	...	...	...	...	...	...	...	...	
58	Yak	155	320	Spotted	15	Herbivore	Savannas	Endangered	
59	Banteng	180	450	Red	25	Herbivore	Woodlands	Critically Endangered	
60	Wildebeest	145	280	Tan	17	Herbivore	Plains	Extinct	
61	Zebu	140	260	Gray	22	Herbivore	Farmlands	Vulnerable	
62	Highland Cattle	152	320	Ginger	18	Herbivore	Highlands	Least Concern	

63 rows × 12 columns

In [29]:
```
df['Nutritional Content'][df['Nutritional Content'].str.endswith('(usually)')]=3
```

In [30]:
```
df
```

Out[30]:

	Animal	Height (cm)	Weight (kg)	Color	Lifespan (years)	Diet	Habitat	Conservation Status	
0	Elephant	310	6000	Grey	70	Herbivore	Savannah, Forest	Vulnerable	
1	Lion	110	250	Tan	14	Carnivore	Grasslands, Savannas	Vulnerable	
2	Wild Dog	80	36	Multicolored	Dec	Carnivore	Savannahs	Endangered	
3	Bison	186	1000	Brown	20	Herbivore	Grasslands, Plains	Near Threatened	
4	Anteater	91	41	Brown, White	20	Insectivore	Grasslands, Forests	Least Concern	Myrr
...	...	...	...	...	...	...	...	...	
58	Yak	155	320	Spotted	15	Herbivore	Savannas	Endangered	
59	Banteng	180	450	Red	25	Herbivore	Woodlands	Critically Endangered	
60	Wildebeest	145	280	Tan	17	Herbivore	Plains	Extinct	
61	Zebu	140	260	Gray	22	Herbivore	Farmlands	Vulnerable	
62	Highland Cattle	152	320	Ginger	18	Herbivore	Highlands	Least Concern	

63 rows × 12 columns

missing values and nulls

In [31]:
```
df.isna().sum()
```

Out[31]:
```
Animal                   0
Height (cm)              0
Weight (kg)              0
Color                    0
Lifespan (years)         0
Diet                     0
Habitat                  0
Conservation Status      0
Family                   0
Gestation Period (days)  0
Feeding Behavior         0
Nutritional Content      0
dtype: int64
```

In [32]:
```
df.replace('Not Applicable',np.nan,inplace=True)
```

In [33]:
```
df.dropna(inplace=True)
```

In [34]:
```
df[df['Height (cm)'].isna()]
```

Out[34]:

Animal	Height (cm)	Weight (kg)	Color	Lifespan (years)	Diet	Habitat	Conservation Status	Family	Gestation Period (days)	Feed Behav

In [35]:
```
df.dropna(subset=['Height (cm)'],inplace=True)
```

In [36]:
```
df.isna().sum()
```

Out[36]:
```
Animal                    0
Height (cm)               0
Weight (kg)               0
Color                     0
Lifespan (years)          0
Diet                      0
Habitat                   0
Conservation Status       0
Family                    0
Gestation Period (days)   0
Feeding Behavior          0
Nutritional Content       0
dtype: int64
```

In [37]:
```
df.dropna(inplace=True)
```

fixing special character values for each columns

In [38]:
```
df.isna().sum()
```

Out[38]:
```
Animal                    0
Height (cm)               0
Weight (kg)               0
Color                     0
Lifespan (years)          0
Diet                      0
Habitat                   0
Conservation Status       0
Family                    0
Gestation Period (days)   0
Feeding Behavior          0
Nutritional Content       0
dtype: int64
```

In [39]:
```
df.loc[df['Height (cm)'] == '']
```

Out[39]:

Animal	Height (cm)	Weight (kg)	Color	Lifespan (years)	Diet	Habitat	Conservation Status	Family	Gestation Period (days)	Feed Behav

In [40]:
```
df.drop(df[df['Height (cm)'] == ''].index,inplace= True)
```

In [41]:
```
df.replace(',', '', regex=True,inplace=True)
```

In [42]:
```
df.replace('Varies',np.nan,inplace=True)
```

In [43]:
```
df.dropna(inplace=True)
```

In [44]:
```
df['Lifespan (years)']=df['Lifespan (years)'].str.replace(r'\+', '', regex=True)
```

In [46]:
```
df.replace(r'\(in burrow\)','',regex=True,inplace=True)
```

In [47]:
```
df['Gestation Period (days)'][df['Gestation Period (days)'].str.contains(r'.*days.*
```

In [48]:
```
df['Gestation Period (days)'][df['Gestation Period (days)'].str.contains(r'.*weeks.
```

Out[48]:
```
Series([], Name: Gestation Period (days), dtype: object)
```

In [49]:
```
df['Gestation Period (days)'][df['Gestation Period (days)'].str.contains(r'.*weeks.
```

In [50]:
```
df['Gestation Period (days)'][df['Gestation Period (days)'].str.contains(r'.*months
```

Out[50]:
```
Series([], Name: Gestation Period (days), dtype: object)
```

In [51]:
```
df['Gestation Period (days)'] = df['Gestation Period (days)'].apply(
    lambda x: f'{int(x.split()[0]) * 30}' if 'months' in x else x
)
```

In [53]:
```
df['Nutritional Content']=df['Nutritional Content'].str.replace(r')',"")
```

In [55]:
```
df['Nutritional Content']=df['Nutritional Content'].str.replace(r'Hundreds',"100")
```

In [57]:
```
non_convertible_values = []

for value in df['Nutritional Content']:
    try:
        float(value)
    except ValueError:
        non_convertible_values.append(value)

df['Nutritional Content'][df['Nutritional Content'].isin(non_convertible_values)] =
```

In [58]:
```
df.columns
```

Out[58]:
```
Index(['Animal', 'Height (cm)', 'Weight (kg)', 'Color', 'Lifespan (years)',
       'Diet', 'Habitat', 'Conservation Status', 'Family',
       'Gestation Period (days)', 'Feeding Behavior', 'Nutritional Content'],
      dtype='object')
```

Exploratory Data Analysis(EDA)

In [59]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 60 entries, 0 to 62
Data columns (total 12 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Animal                  60 non-null     object
 1   Height (cm)             60 non-null     object
 2   Weight (kg)             60 non-null     object
 3   Color                   60 non-null     object
 4   Lifespan (years)        60 non-null     object
 5   Diet                    60 non-null     object
 6   Habitat                 60 non-null     object
 7   Conservation Status     60 non-null     object
 8   Family                  60 non-null     object
 9   Gestation Period (days) 60 non-null     object
 10  Feeding Behavior        60 non-null     object
 11  Nutritional Content     60 non-null     object
dtypes: object(12)
memory usage: 6.1+ KB
```

In [60]:
```
pd.DataFrame(df.describe()).style\
    .set_properties(**{'background-color': 'lightgreen',
                       'color': 'black'})
```

Out[60]:

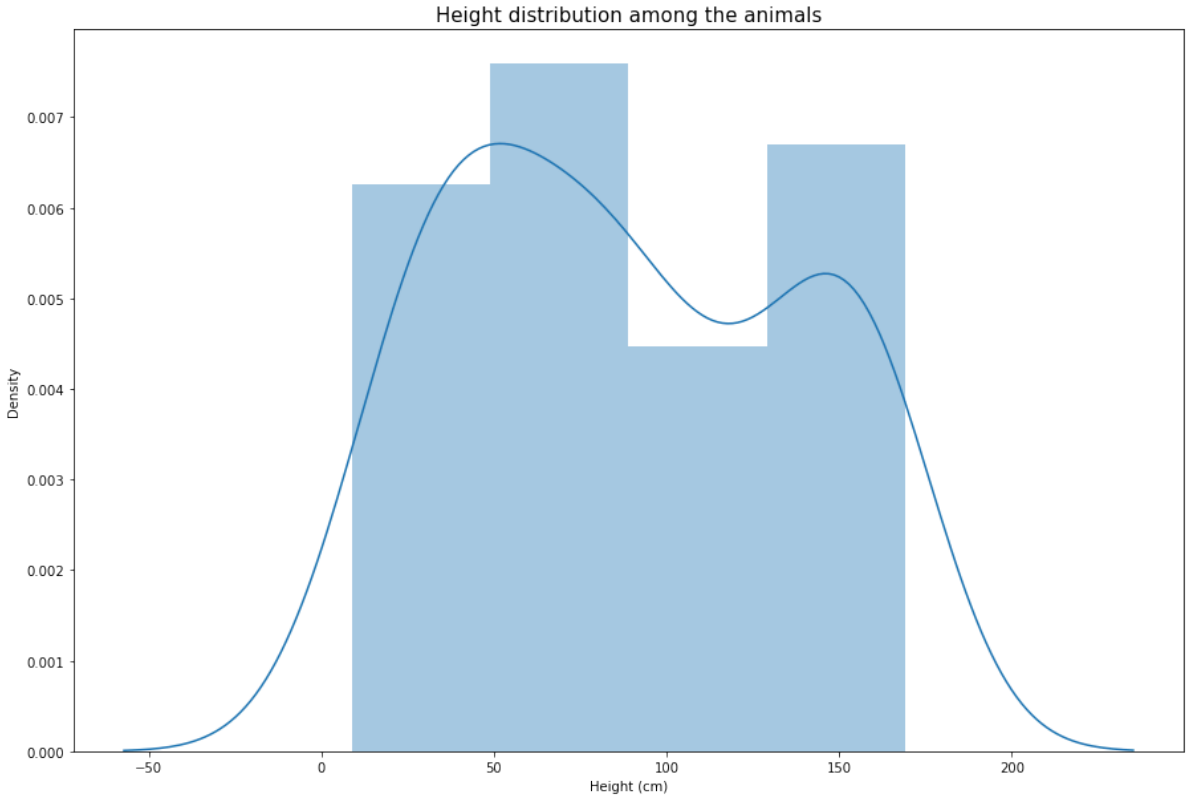
	Animal	Height (cm)	Weight (kg)	Color	Lifespan (years)	Diet	Habitat	Conservation Status	Family
count	60	60	60	60	60	60	60	60	60
unique	60	42	51	36	32	4	36	6	31
top	Sumatran Rhinoceros	150	320	Brown	20	Herbivore	Rainforests	Least Concern	Bovidae
freq	1	4	2	9	7	22	12	20	10

In [62]:
```
print("The average height for the animals are {:.1f}cm, and the maximum weight for
```

```
The average height for the animals are 516851336448550639958240602321380499692741914907355050430542676803892285553113804909647636900599193263334164176285575988511235370892585155313224220672.0cm, and the maximum weight for an animal is Sepkg.
```

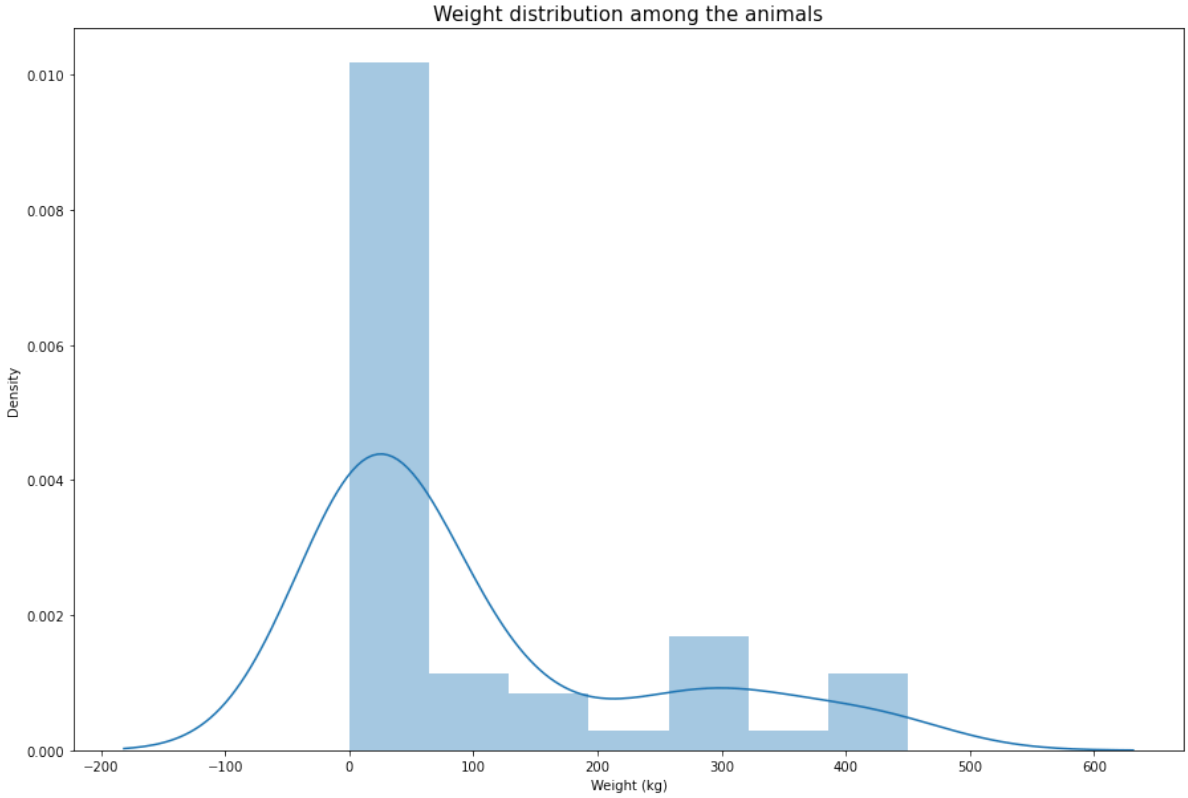
In [65]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv("animal_diet.csv")

def convert_height(x):
    if '-' in x:
        return sum(map(int, x.split('-'))) / 2
    elif x.startswith('Up to'):
        return int(x.split()[-1])
    else:
        return int(x)

df['Height (cm)'] = df['Height (cm)'].apply(convert_height)

plt.figure(figsize=(15,10))
data = df.copy()
data = data[data['Height (cm)'] < data['Height (cm)'].quantile(0.90)]
sns.distplot(data['Height (cm)'].sort_values())
```

```
plt.title("Height distribution among the animals", fontsize=15)
plt.xlabel("Height (cm)")
plt.ylabel("Density")
plt.show()
```



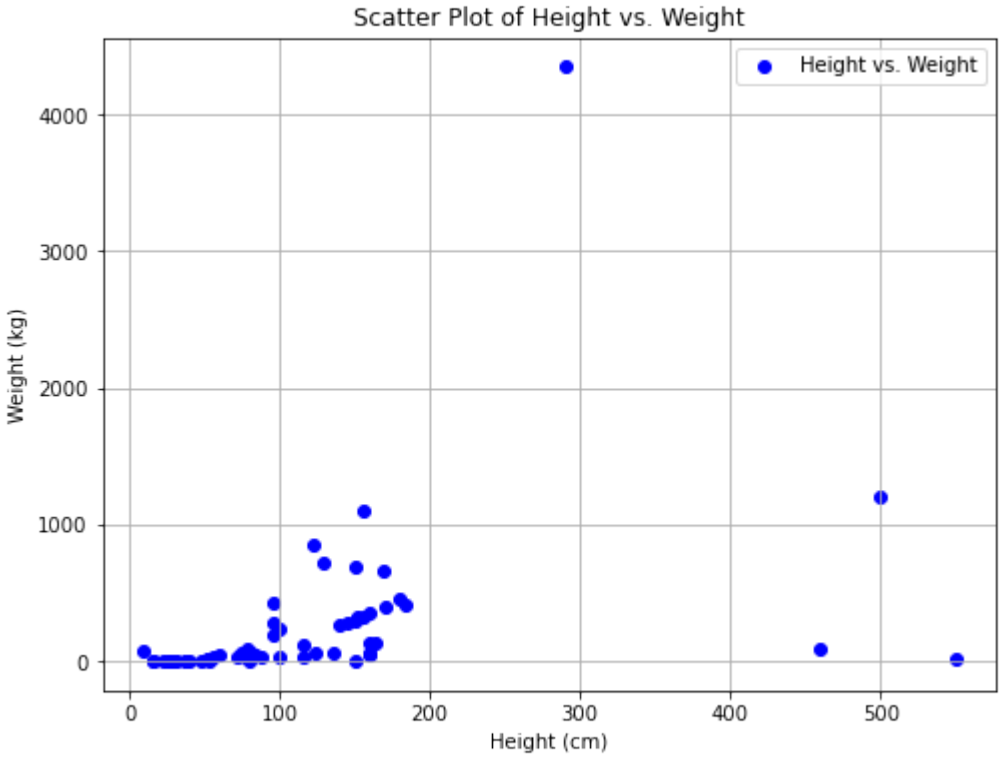
In [68]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv("animal_diet.csv")

def convert_weight(x):
    try:
        if '-' in x:
            return sum(map(float, x.split('-'))) / 2
        elif x.startswith('Up to'):
            return float(x.split()[-1])
        else:
            return float(x)
    except ValueError:
        return None

df['Weight (kg)'] = df['Weight (kg)'].apply(convert_weight)

plt.figure(figsize=(15,10))
data = df.dropna(subset=['Weight (kg)'])
data = data[data['Weight (kg)'] < data['Weight (kg)'].quantile(0.90)]
sns.distplot(data['Weight (kg)'].sort_values())
plt.title("Weight distribution among the animals", fontsize=15)
plt.xlabel("Weight (kg)")
plt.ylabel("Density")
plt.show()
```

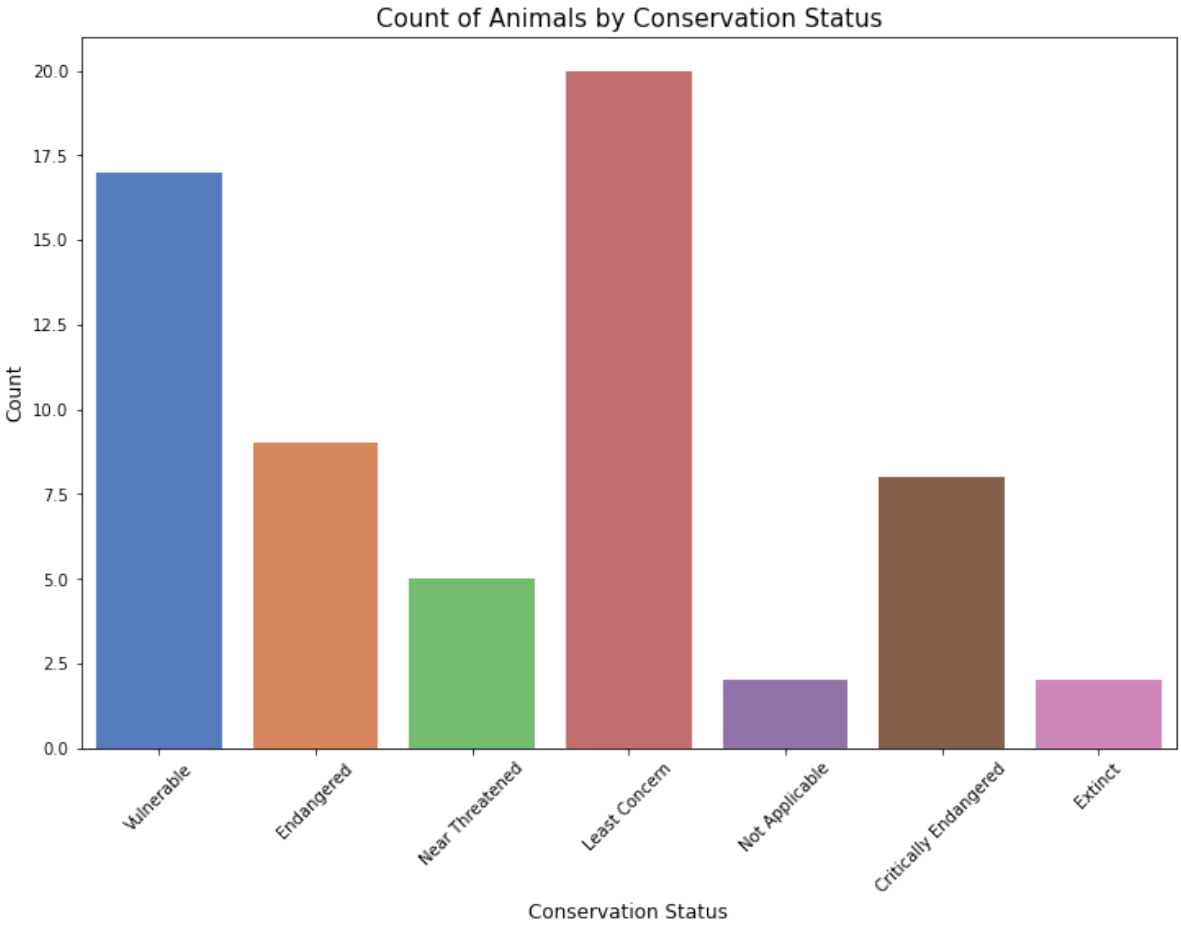


In [69]:
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("animal_diet.csv")

def convert_height(x):
    try:
        if '-' in x:
            return sum(map(float, x.split('-'))) / 2
        elif x.startswith('Up to'):
            return float(x.split()[-1])
        else:
            return float(x)
    except ValueError:
        return None

df['Height (cm)'] = df['Height (cm)'].apply(convert_height)

def convert_weight(x):
    try:
        if '-' in x:
            return sum(map(float, x.split('-'))) / 2
        elif x.startswith('Up to'):
            return float(x.split()[-1])
        else:
            return float(x)
    except ValueError:
        return None

df['Weight (kg)'] = df['Weight (kg)'].apply(convert_weight)

data = df.dropna(subset=['Height (cm)', 'Weight (kg)'])

plt.figure(figsize=(8, 6))
plt.scatter(data['Height (cm)'], data['Weight (kg)'], c='blue', marker='o', label='
plt.xlabel('Height (cm)')
plt.ylabel('Weight (kg)')
plt.title('Scatter Plot of Height vs. Weight')
```

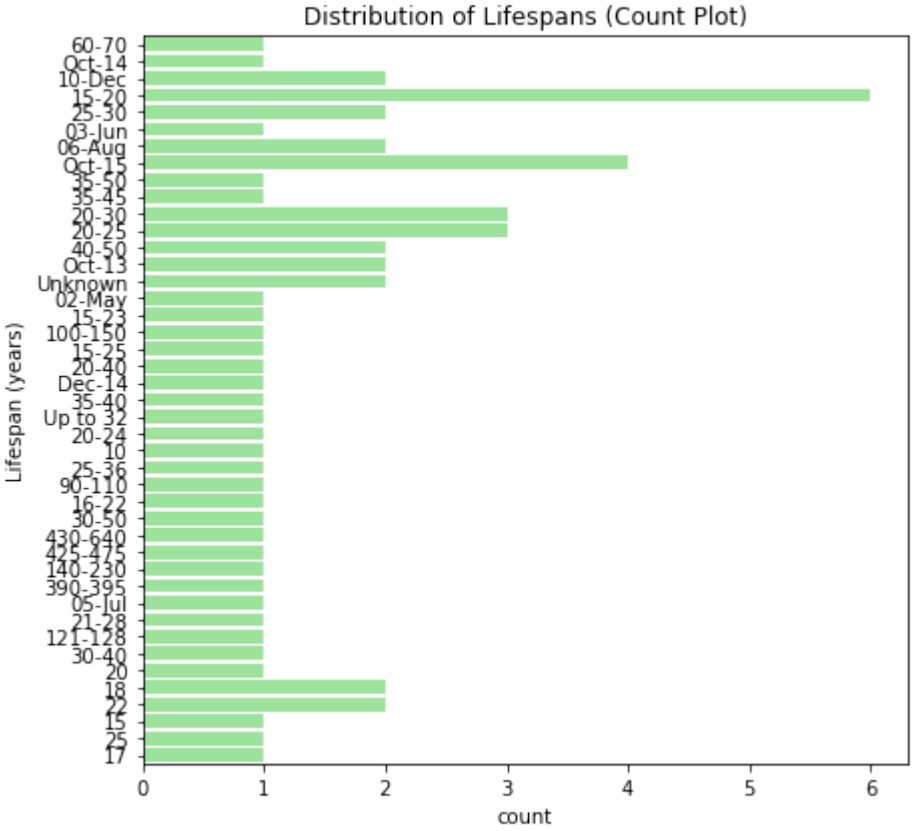
```
plt.legend()
plt.grid(True)
plt.show()
```



In [71]:
```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(12, 8))
sns.countplot(x="Conservation Status", data=df, palette="muted")
plt.title("Count of Animals by Conservation Status", fontsize=15)
plt.xlabel("Conservation Status", fontsize=12)
plt.ylabel("Count", fontsize=12)
plt.xticks(rotation=45)
plt.show()
```

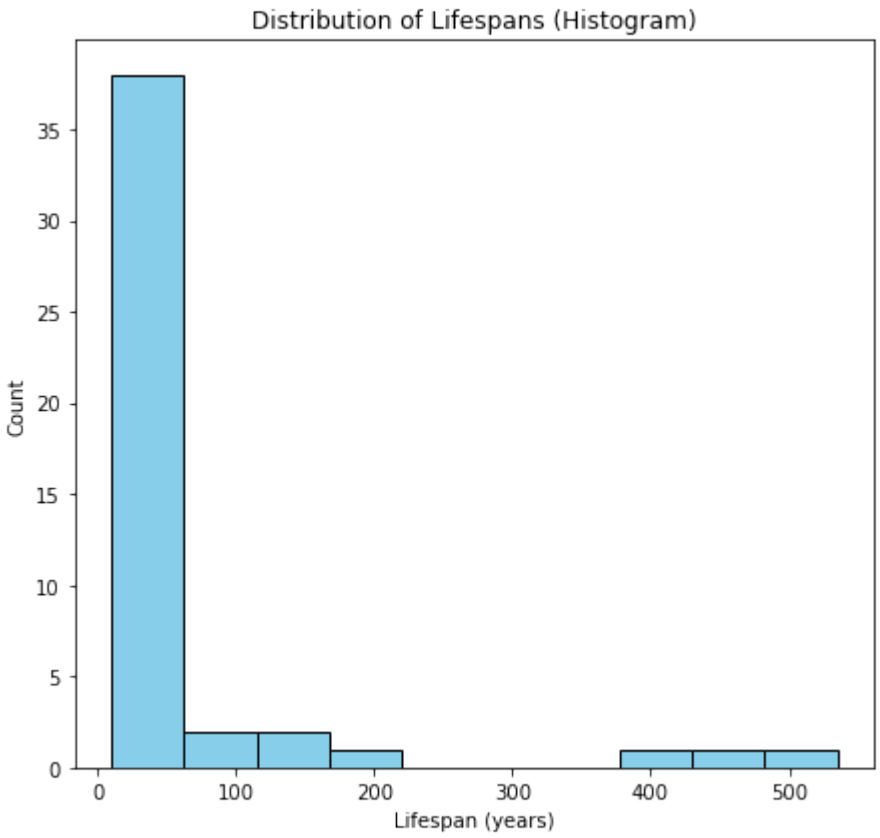


In [73]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv("animal_diet.csv")

plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 2)
sns.countplot(data=df, y='Lifespan (years)', color='lightgreen')
plt.ylabel('Lifespan (years)')
plt.title('Distribution of Lifespans (Count Plot)')

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

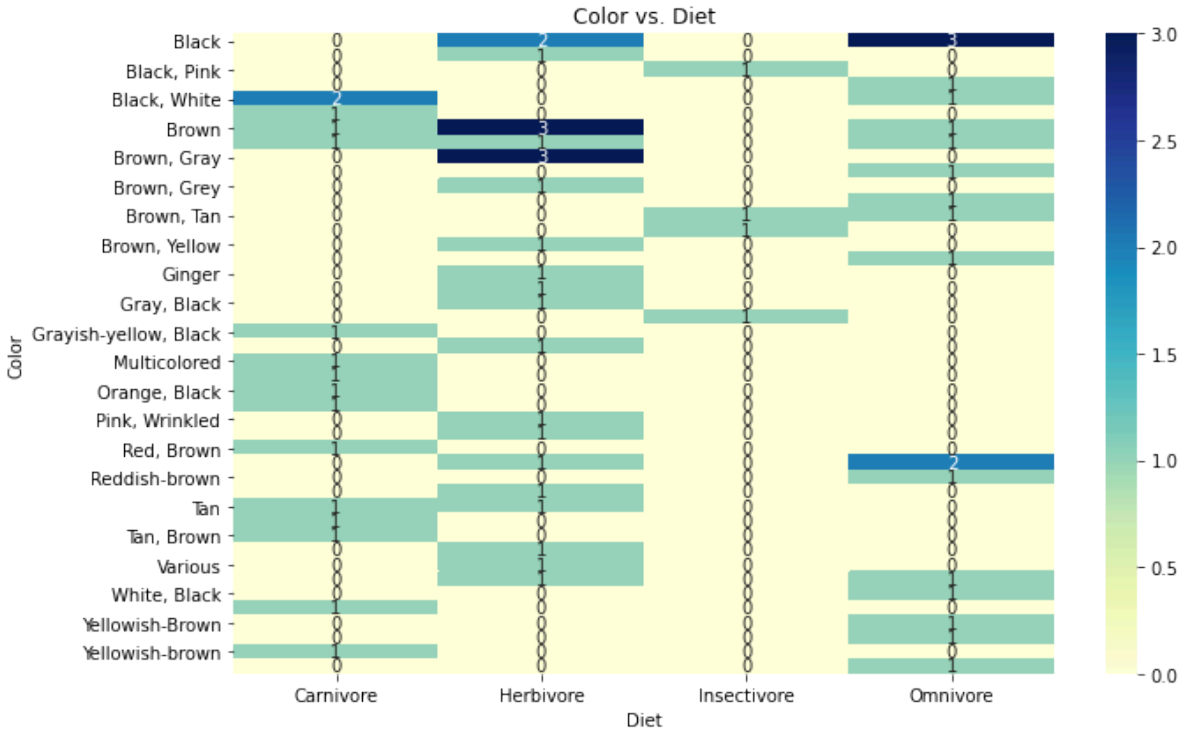


In [79]:

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_csv("animal_diet.csv")

# Convert string ranges to numeric values
def convert_lifespan(x):
    if '-' in x:
        try:
            start, end = map(int, x.split('-'))
            return (start + end) / 2
        except ValueError:
            return None
    elif x.startswith('Up to'):
        try:
            return int(x.split()[-1])
        except ValueError:
            return None
    elif x.isdigit():
        return int(x)
    else:
        return None

df['Lifespan (years)'] = df['Lifespan (years)'].apply(convert_lifespan)

lifespan_values = df['Lifespan (years)'].dropna()

plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 2)
plt.hist(lifespan_values, bins=10, color='skyblue', edgecolor='black')
plt.xlabel('Lifespan (years)')
plt.ylabel('Count')
plt.title('Distribution of Lifespans (Histogram)')

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

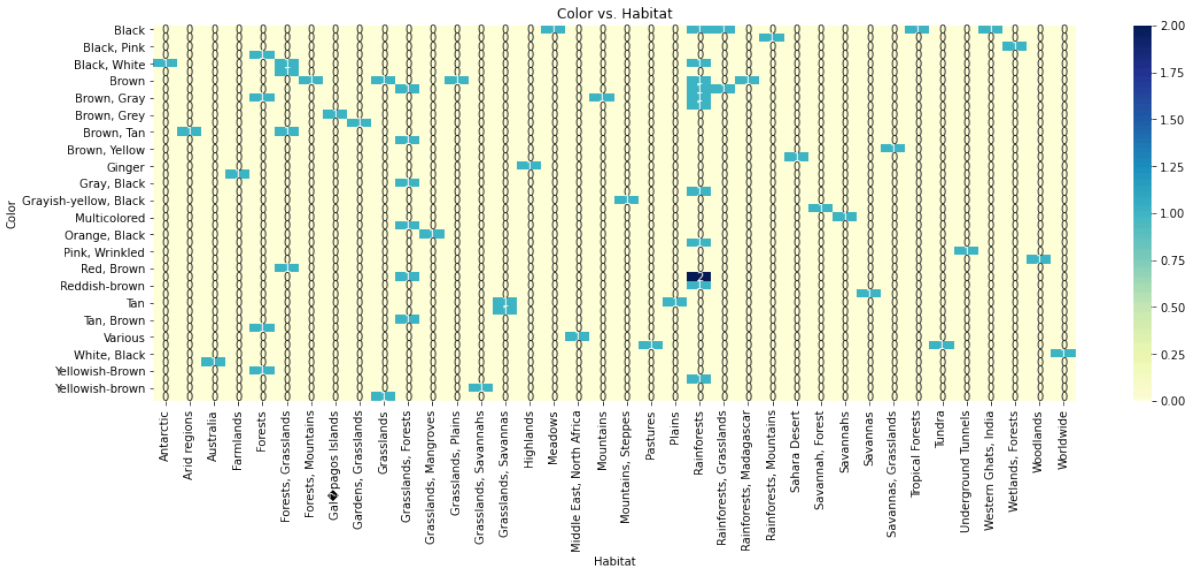


In [80]:
```
import seaborn as sns
import matplotlib.pyplot as plt

color_diet = pd.crosstab(df['Color'], df['Diet'])

color_habitat = pd.crosstab(df['Color'], df['Habitat'])

plt.figure(figsize=(18, 6))

plt.subplot(1, 2, 1)
sns.heatmap(color_diet, annot=True, fmt='d', cmap='YlGnBu')
plt.title('Color vs. Diet')

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

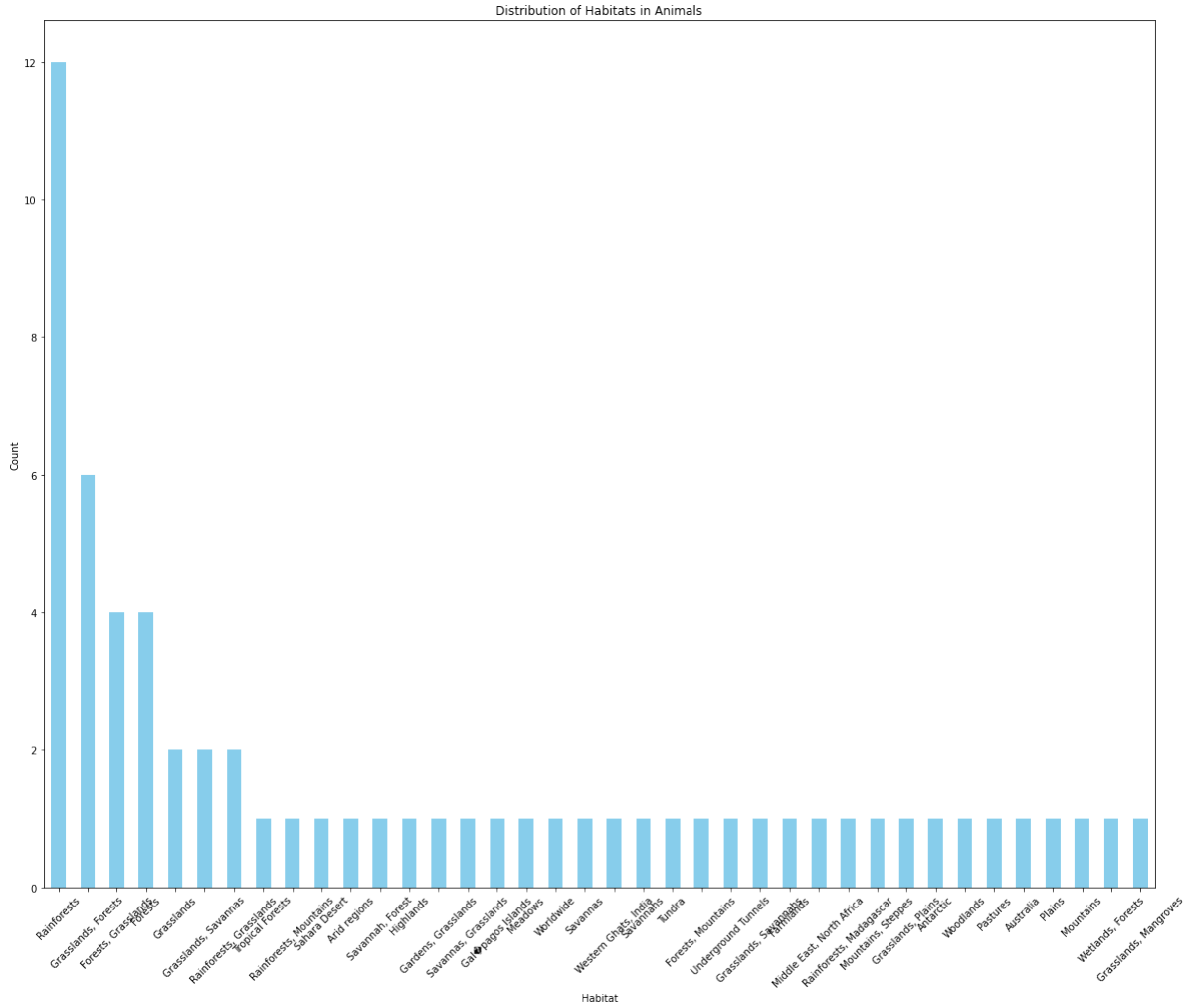


In [82]:
```
plt.figure(figsize=(18, 6))
sns.heatmap(color_habitat, annot=True, fmt='d', cmap='YlGnBu')
plt.title('Color vs. Habitat')
```

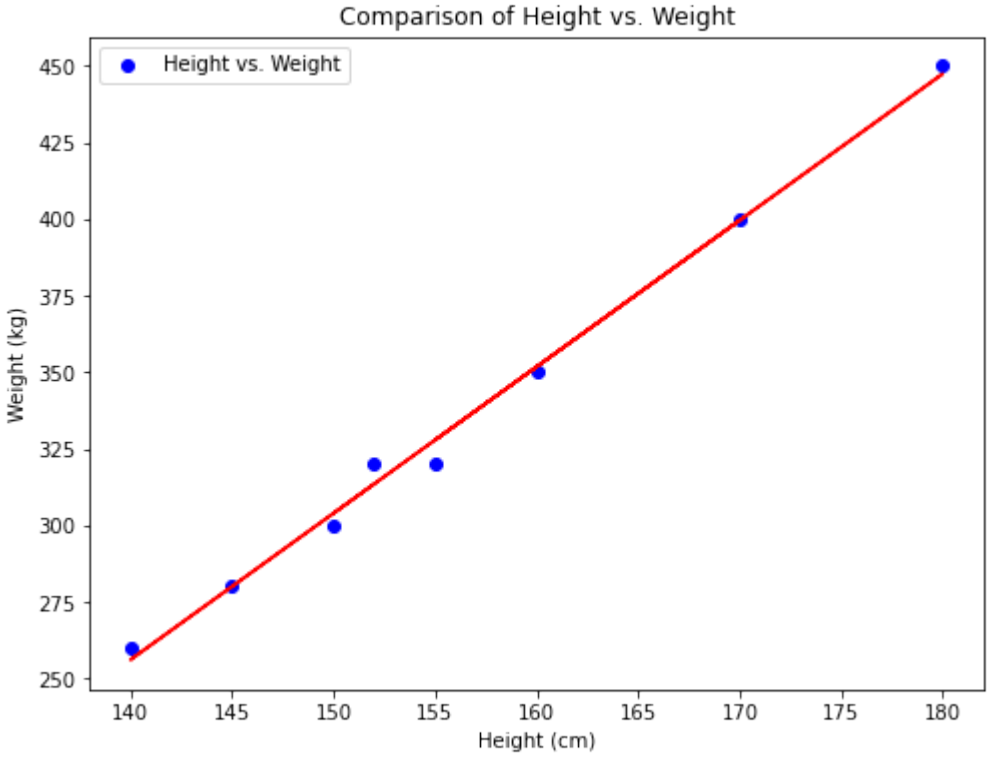
Out[82]:
```
Text(0.5, 1.0, 'Color vs. Habitat')
```



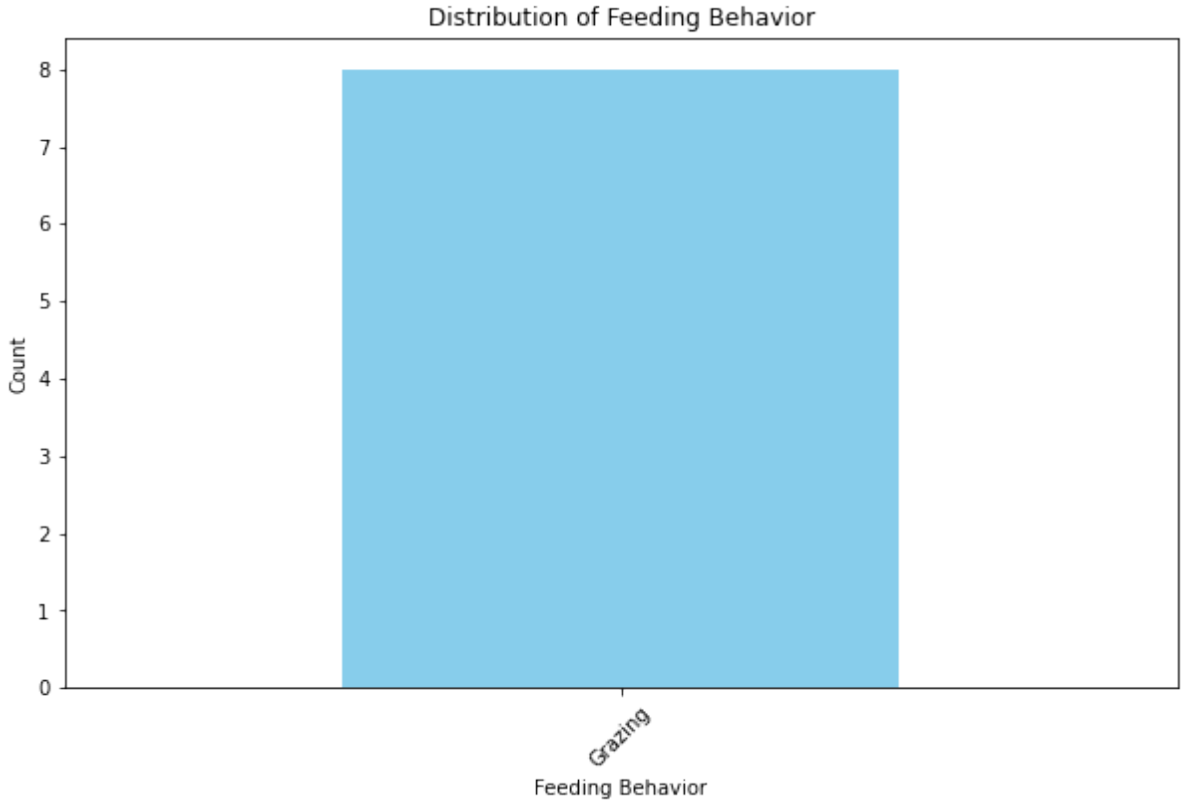
In [83]:
```
plt.figure(figsize=(20, 16))
df['Habitat'].value_counts().plot(kind='bar', color='skyblue')
plt.xlabel('Habitat')
plt.ylabel('Count')
plt.title('Distribution of Habitats in Animals')
plt.xticks(rotation=45)
plt.show()
```



In [85]:
```
df['Height (cm)'] = pd.to_numeric(df['Height (cm)'], errors='coerce')
df['Weight (kg)'] = pd.to_numeric(df['Weight (kg)'], errors='coerce')

df = df.dropna(subset=['Height (cm)', 'Weight (kg)'])

plt.figure(figsize=(8, 6))
plt.scatter(df['Height (cm)'], df['Weight (kg)'], c='blue', marker='o', label='Heig
plt.xlabel('Height (cm)')
plt.ylabel('Weight (kg)')
plt.title('Comparison of Height vs. Weight')
plt.legend()

fit = np.polyfit(df['Height (cm)'], df['Weight (kg)'], 1)
plt.plot(df['Height (cm)'], fit[0] * df['Height (cm)'] + fit[1], color='red')

# Show the plot
plt.show()
```

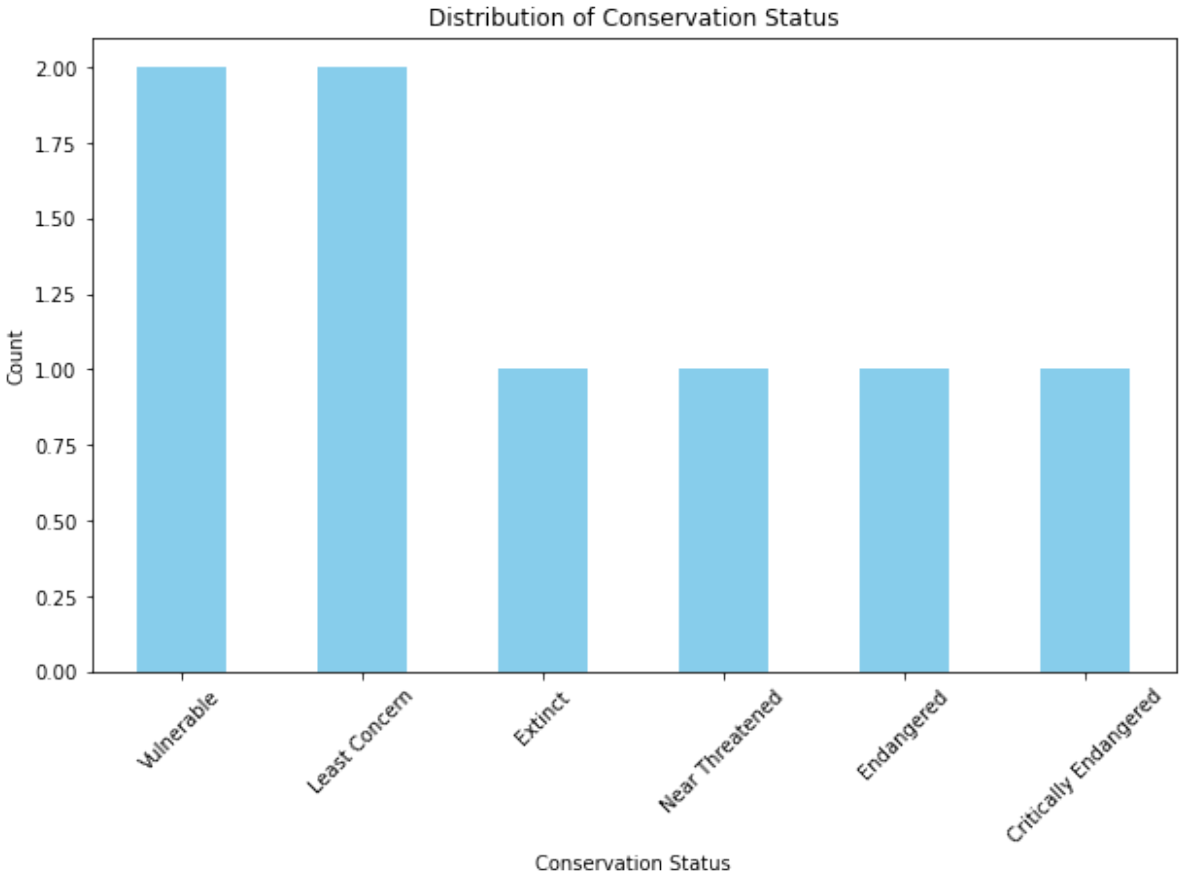


In [86]:
```
feeding_behavior_counts = df['Feeding Behavior'].value_counts()

plt.figure(figsize=(10, 6))

feeding_behavior_counts.plot(kind='bar', color='skyblue')
plt.xlabel('Feeding Behavior')
plt.ylabel('Count')
plt.title('Distribution of Feeding Behavior')
plt.xticks(rotation=45)
plt.show()
```

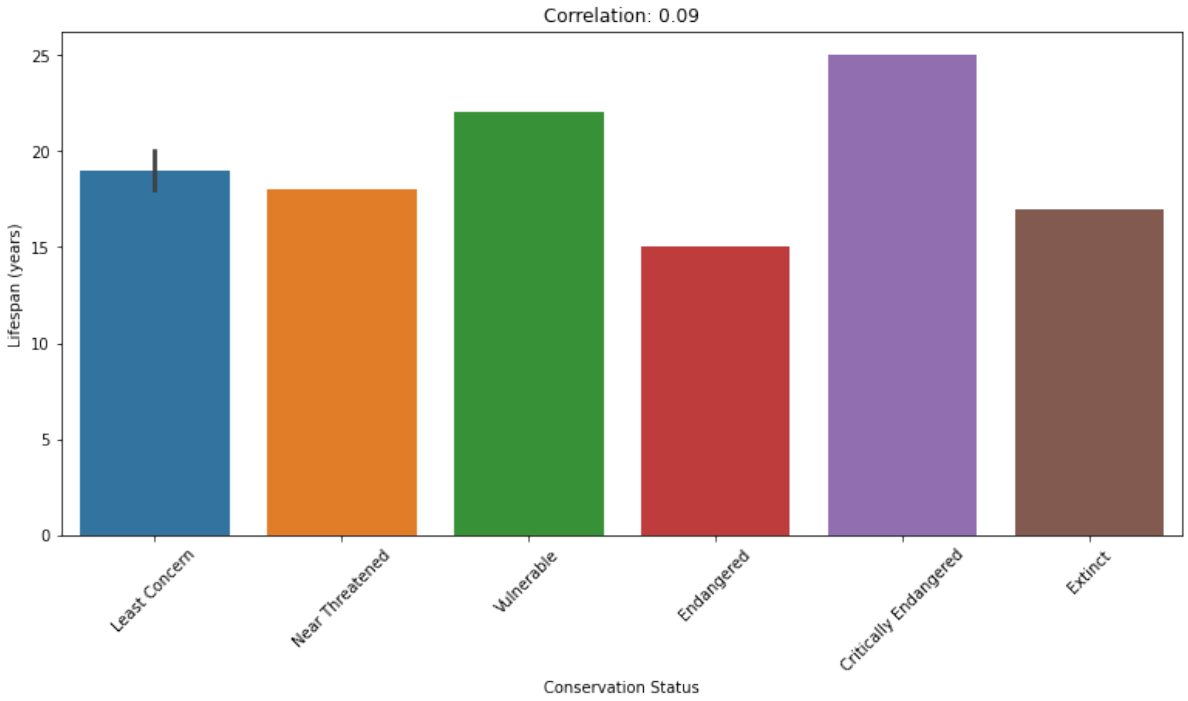


In [87]:
```
conservation_status_counts = df['Conservation Status'].value_counts()
```

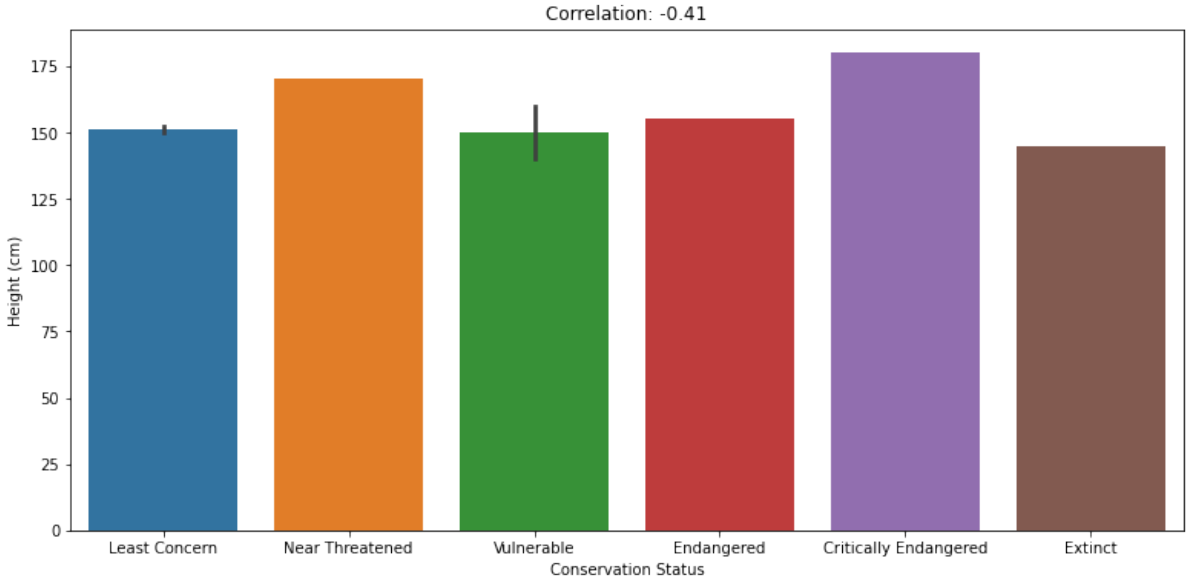
```
plt.figure(figsize=(10, 6))

conservation_status_counts.plot(kind='bar', color='skyblue')
plt.xlabel('Conservation Status')
plt.ylabel('Count')
plt.title('Distribution of Conservation Status')
plt.xticks(rotation=45)
plt.show()
```



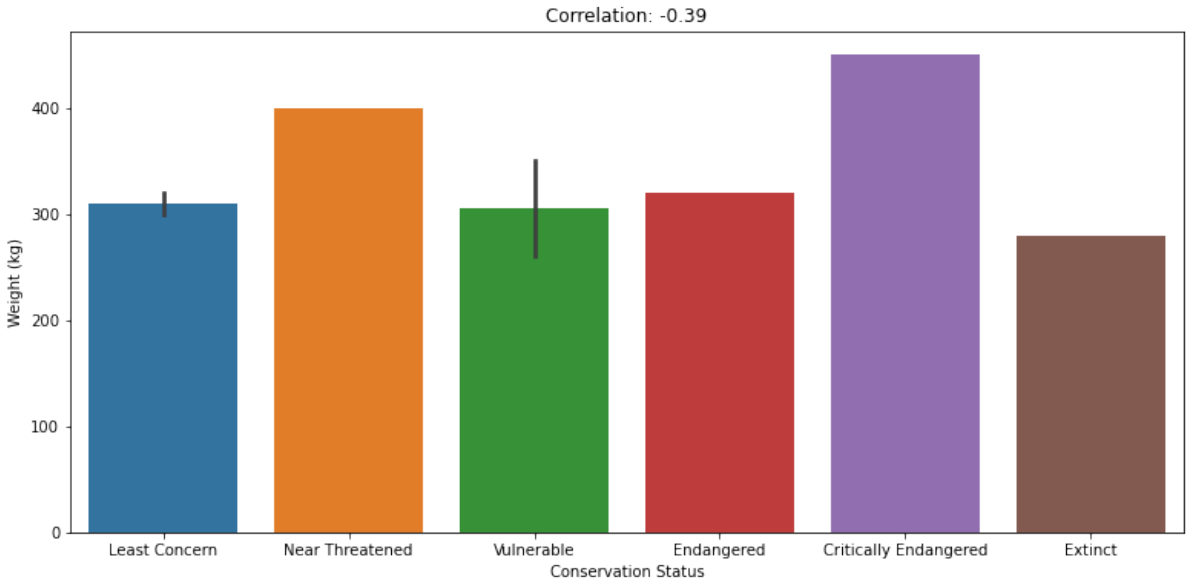
In [91]:
```
correlation = df['Conservation Status'].astype('category').cat.codes.corr(df['Lifes

plt.figure(figsize=(13, 6))
sns.barplot(x='Conservation Status', y='Lifespan (years)', data=df)
plt.title(f'Correlation: {correlation:.2f}')
plt.xticks(rotation=45)
plt.show()
```



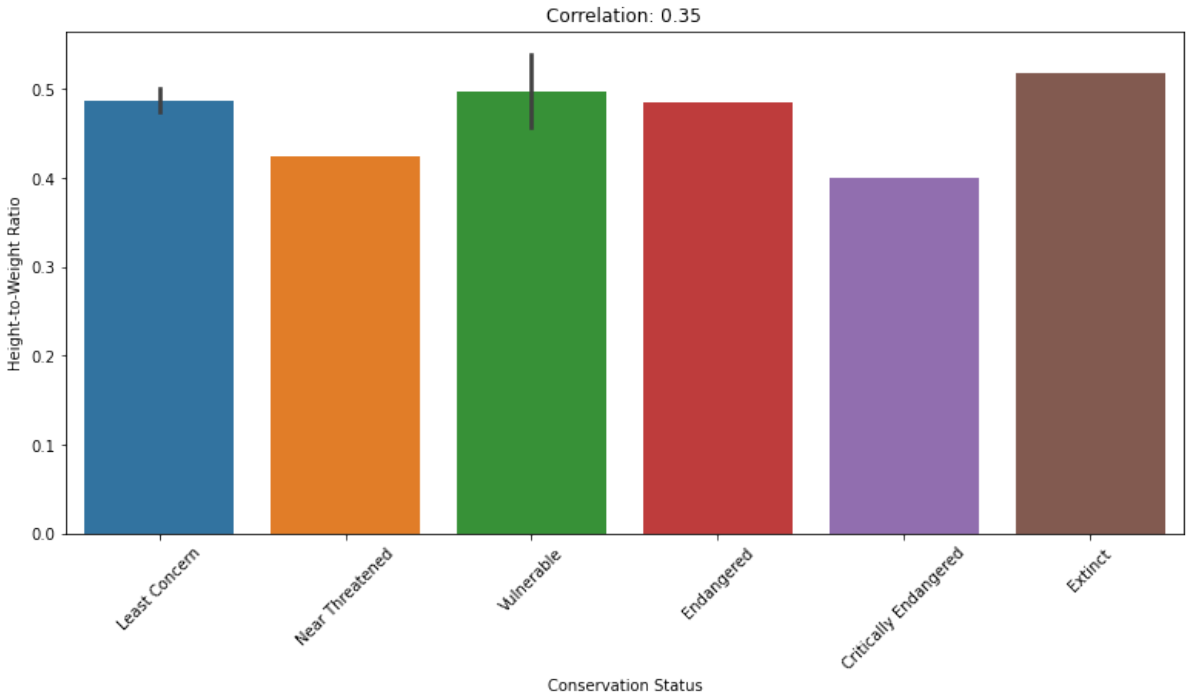
In [92]:
```
correlation = df['Conservation Status'].astype('category').cat.codes.corr(df['Heigh

plt.figure(figsize=(13, 6))
sns.barplot(x='Conservation Status', y='Height (cm)', data=df)
plt.title(f'Correlation: {correlation:.2f}')
plt.show()
```



In [93]:
```
correlation = df['Conservation Status'].astype('category').cat.codes.corr(df['Weigh

plt.figure(figsize=(13, 6))
sns.barplot(x='Conservation Status', y='Weight (kg)', data=df)
plt.title(f'Correlation: {correlation:.2f}')
plt.show()
```



In [95]:
```
df['Height-to-Weight Ratio'] = df['Height (cm)'] / df['Weight (kg)']

correlation = df['Conservation Status'].astype('category').cat.codes.corr(df['Heigh

plt.figure(figsize=(13, 6))
sns.barplot(x='Conservation Status', y='Height-to-Weight Ratio', data=df)
plt.title(f'Correlation: {correlation:.2f}')
plt.xticks(rotation=45)
plt.show()
```

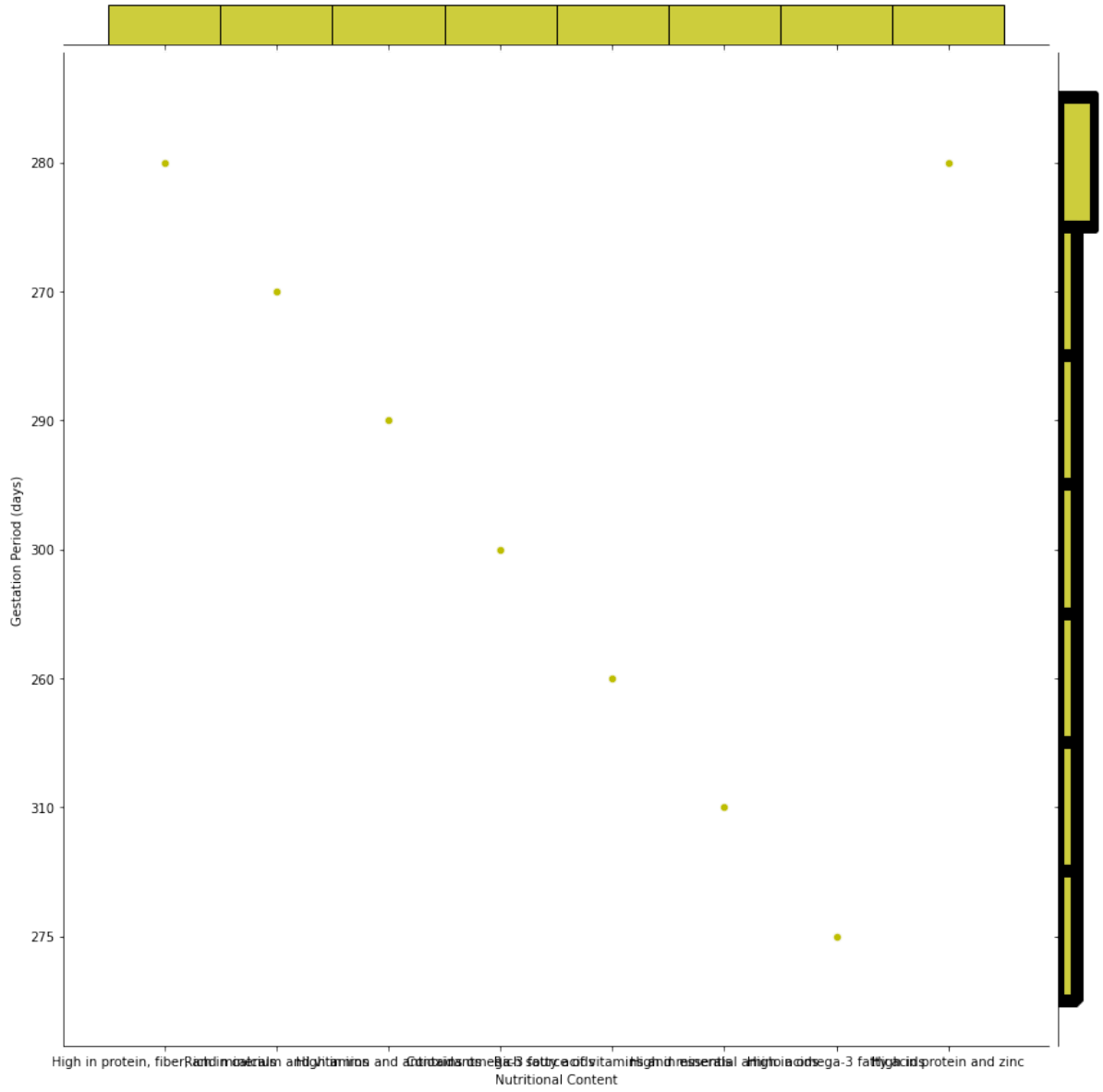


In [96]:
```
pd.DataFrame(df.groupby('Family')[['Habitat', 'Diet']].agg(pd.Series.unique))
```

Out[96]:

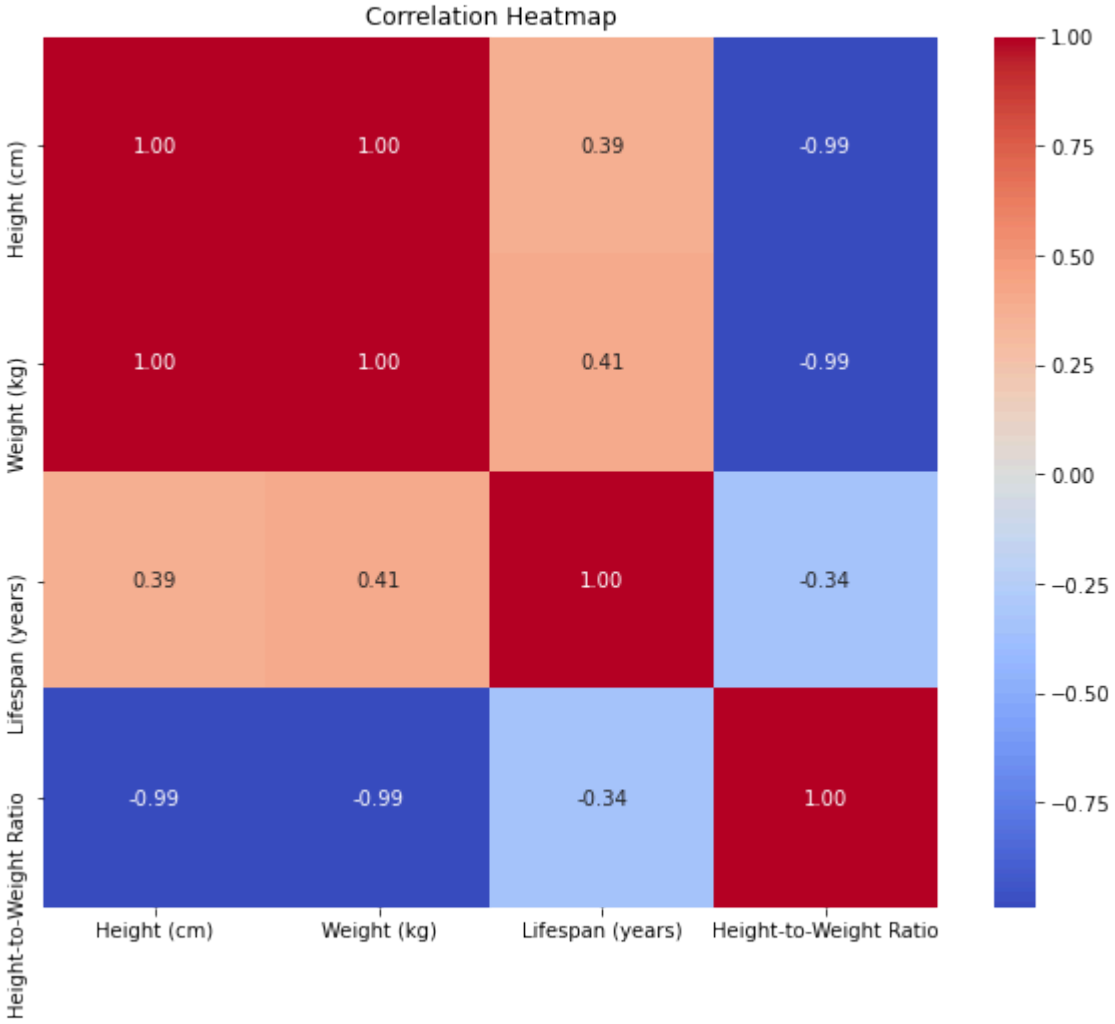
Family	Diet
Bovidae	Herbivore

In [98]:
```
sns.jointplot(y="Gestation Period (days)", x="Nutritional Content", data=df, height
plt.xticks(rotation=45)
plt.show()
```



In [102…

```
numeric_cols = [x for x in df.columns if df[x].dtype in ['float64', 'int64']]
correlation_matrix = df[numeric_cols].corr()

# Create a correlation heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap')
plt.show()
```

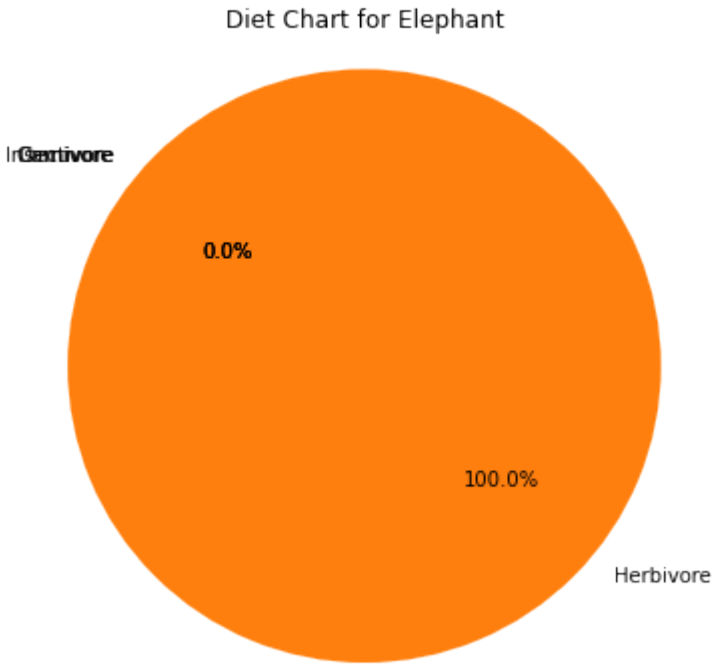


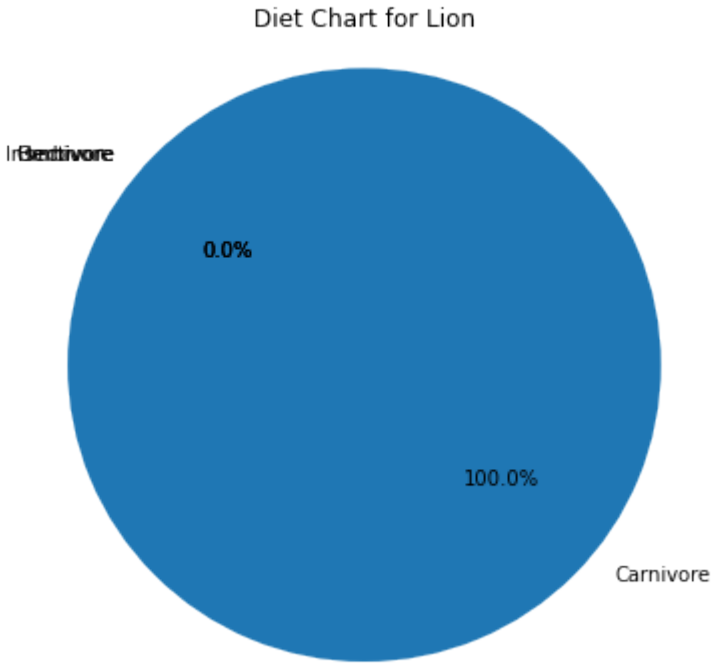
In [ ]:

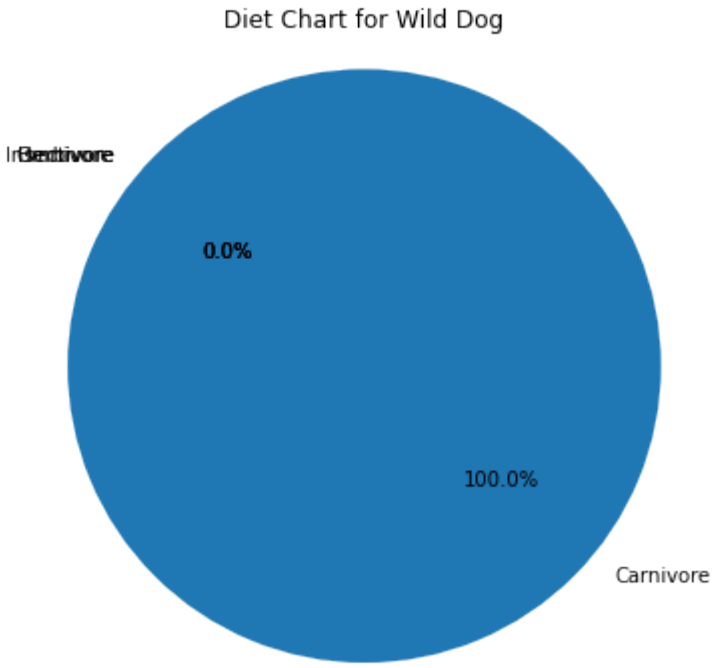
In [3]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
df = pd.read_csv("animal_diet.csv")

# Display the dataset
print("Original DataFrame:")
print(df)

# Generate diet chart for each animal
for index, row in df.iterrows():
    plt.figure(figsize=(8, 6))
    plt.title(f"Diet Chart for {row['Animal']}")
    plt.pie([row['Feeding Behavior'].count('Carnivore'),
             row['Feeding Behavior'].count('Herbivore'),
             row['Feeding Behavior'].count('Omnivore'),
             row['Feeding Behavior'].count('Insectivore')],
            labels=['Carnivore', 'Herbivore', 'Omnivore', 'Insectivore'],
            autopct='%1.1f%%', startangle=140)
    plt.axis('equal')
    plt.show()

# Basic exploratory data analysis
print("Summary Statistics:")
print(df.describe())
```

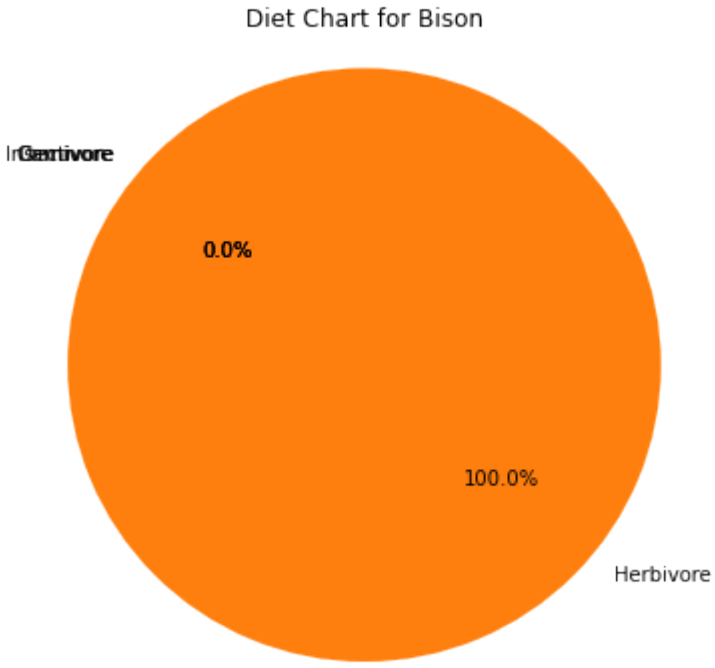
```
# Handling missing values
df.dropna(inplace=True)  # Drop rows with missing values, you can also fill missing

# Print DataFrame after handling missing values
print("DataFrame after handling missing values:")
print(df)

# Correlation heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()
```

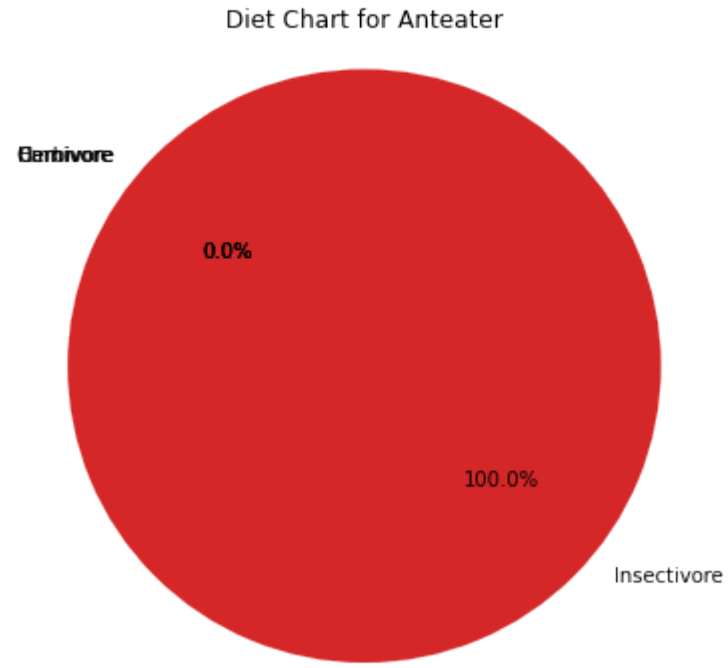
```
Original DataFrame:
             Animal Height (cm) Weight (kg)         Color Lifespan (years)  \
0          Elephant     270-310   2700-6000          Grey            60-70
1              Lion      80-110     120-250           Tan           Oct-14
2          Wild Dog       75-80       18-36  Multicolored           10-Dec
3             Bison     152-186    318-1000         Brown            15-20
4          Anteater       52-91       22-41  Brown, White            15-20
..              ...         ...         ...           ...              ...
58              Yak         155         320       Spotted               15
59           Banteng         180         450           Red               25
60       Wildebeest         145         280           Tan               17
61             Zebu         140         260          Gray               22
62  Highland Cattle         152         320        Ginger               18

           Diet               Habitat    Conservation Status           Family  \
0     Herbivore      Savannah, Forest             Vulnerable     Elephantidae
1     Carnivore  Grasslands, Savannas             Vulnerable         Felidae
2     Carnivore             Savannahs             Endangered         Canidae
3     Herbivore    Grasslands, Plains        Near Threatened         Bovidae
4   Insectivore   Grasslands, Forests          Least Concern  Myrmecophagidae
..          ...                   ...                    ...             ...
58    Herbivore              Savannas             Endangered         Bovidae
59    Herbivore             Woodlands  Critically Endangered         Bovidae
60    Herbivore                Plains                Extinct         Bovidae
61    Herbivore             Farmlands             Vulnerable         Bovidae
62    Herbivore             Highlands          Least Concern         Bovidae

   Gestation Period (days) Feeding Behavior  \
0                  640-660        Herbivore
1                   98-105        Carnivore
2                       70        Carnivore
3                  270-290        Herbivore
4                  190-210      Insectivore
..                     ...              ...
58                     300          Grazing
59                     260          Grazing
60                     310          Grazing
61                     275          Grazing
62                     280          Grazing

                    Nutritional Content
0              High in fiber, low in fat
1                        High in protein
2                        High in protein
3        High in protein, iron, and zinc
4                        High in protein
..                                  ...
58          Contains omega-3 fatty acids
59  Rich source of vitamins and minerals
60         High in essential amino acids
61           High in omega-3 fatty acids
62              High in protein and zinc

[63 rows x 12 columns]
```

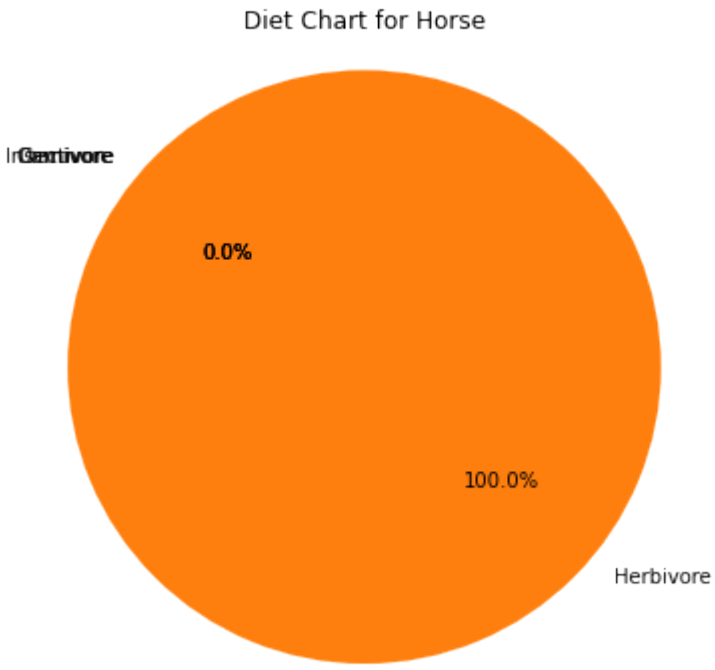

Diet Chart for Elephant
0.0%
100.0%
Herbivore

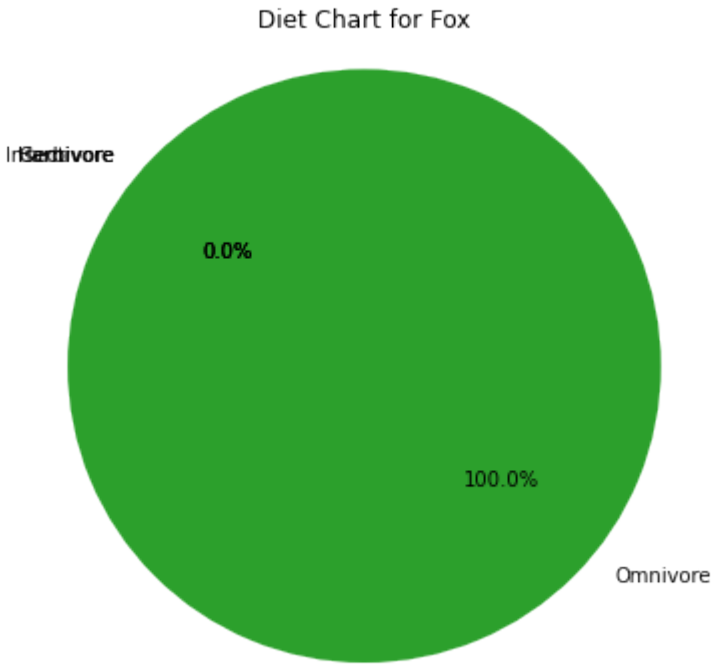

Diet Chart for Lion
0.0%
100.0%
Carnivore

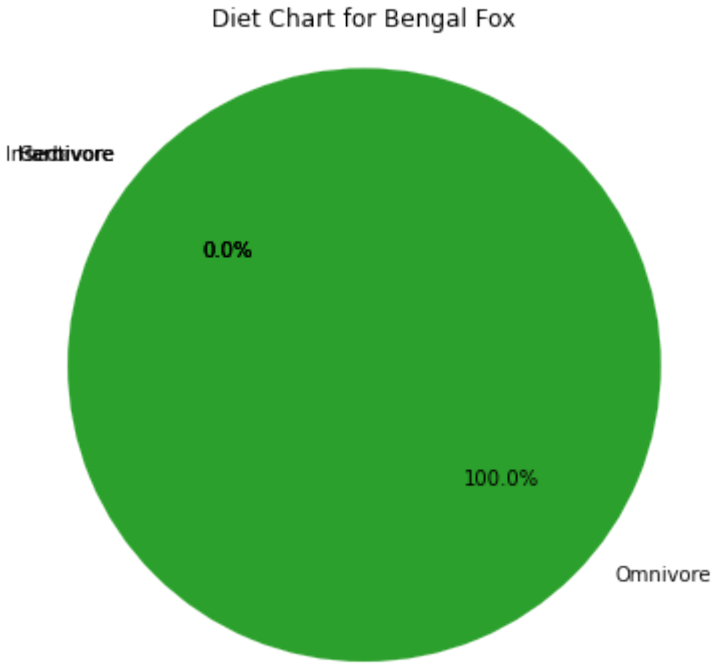

Diet Chart for Wild Dog
0.0%
100.0%
Carnivore

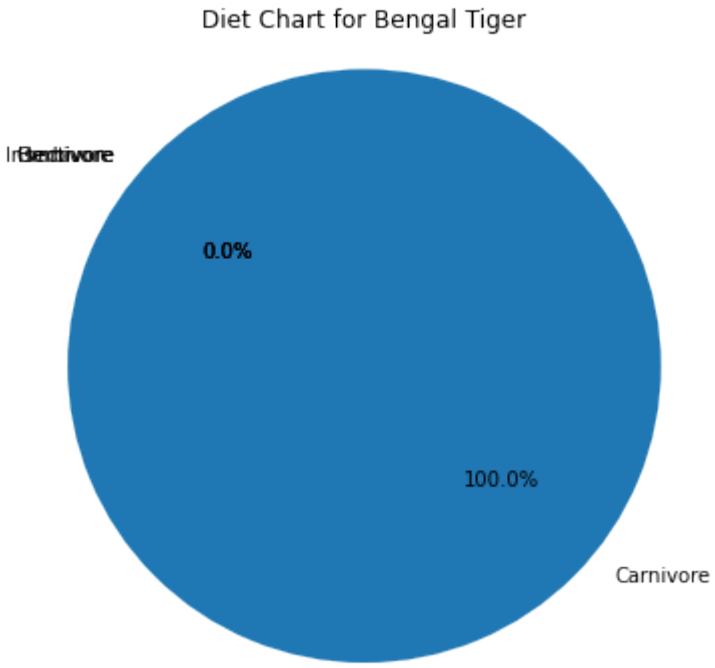

Diet Chart for Bison
0.0%
100.0%
Herbivore

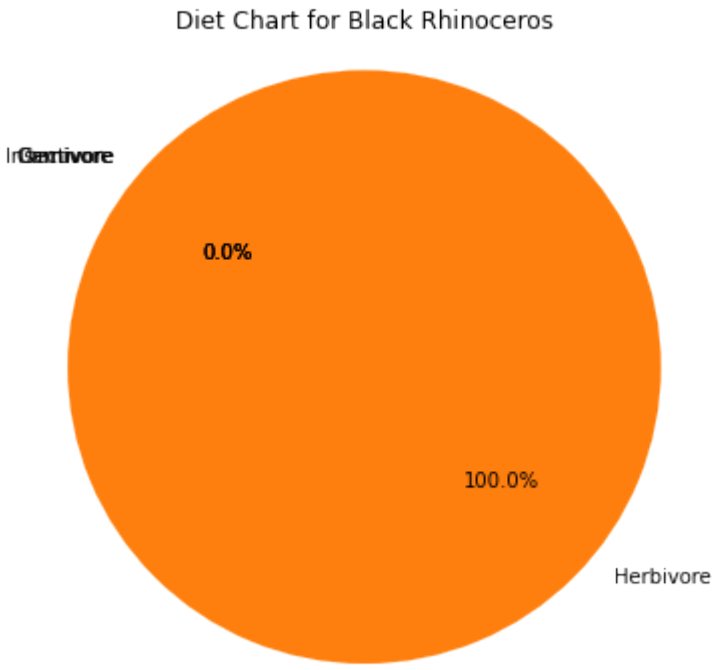

Diet Chart for Anteater
0.0%
100.0%
Insectivore

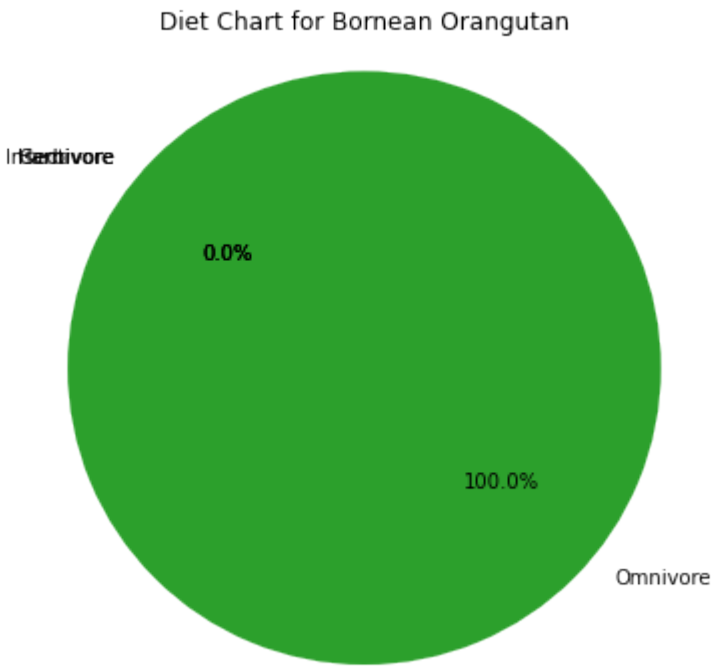

Diet Chart for Horse
0.0%
100.0%
Herbivore

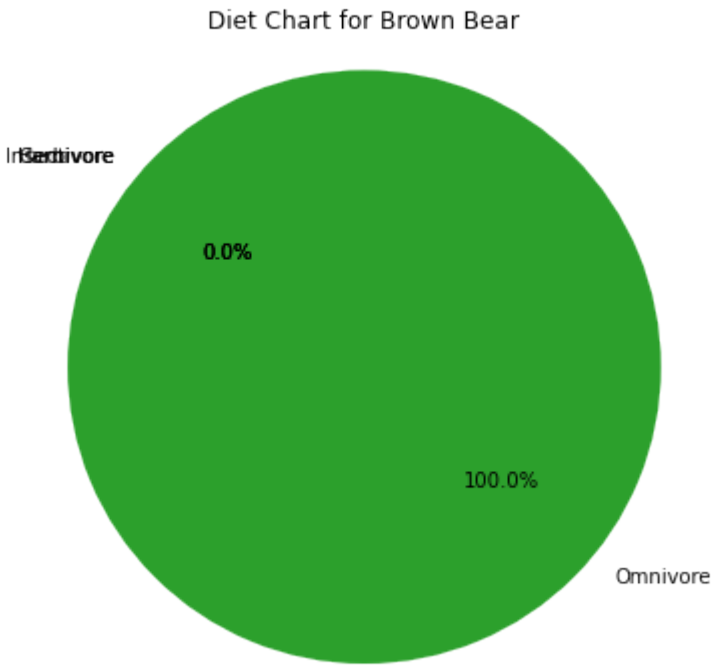

Diet Chart for Fox
Insectivore
0.0%
100.0%
Omnivore

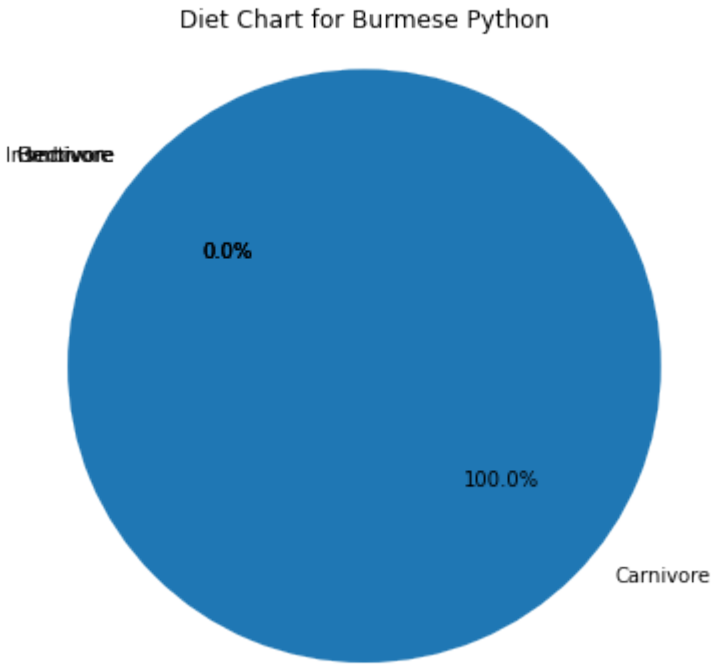

Diet Chart for Bengal Fox
Insectivore
0.0%
100.0%
Omnivore

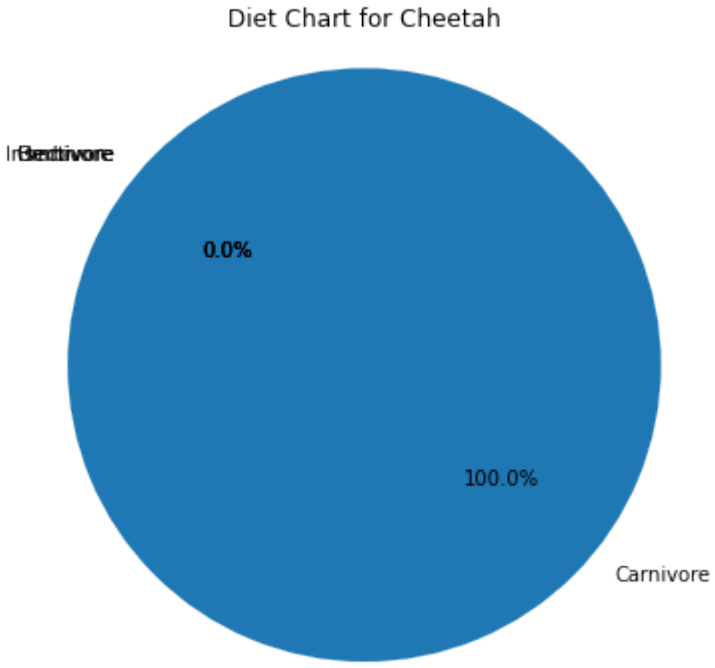

Diet Chart for Bengal Tiger
0.0%
100.0%
Carnivore

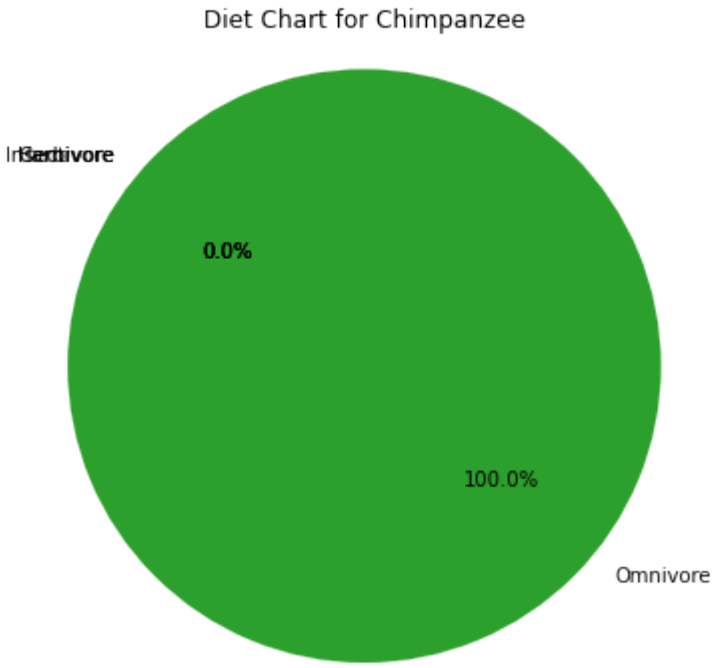

Diet Chart for Black Rhinoceros
0.0%
100.0%
Herbivore

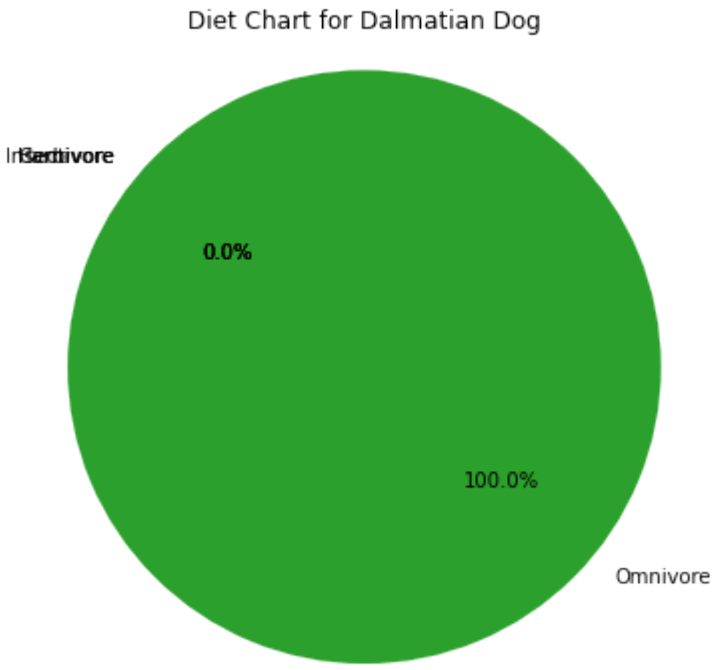

Diet Chart for Bornean Orangutan
Insectivore
0.0%
100.0%
Omnivore

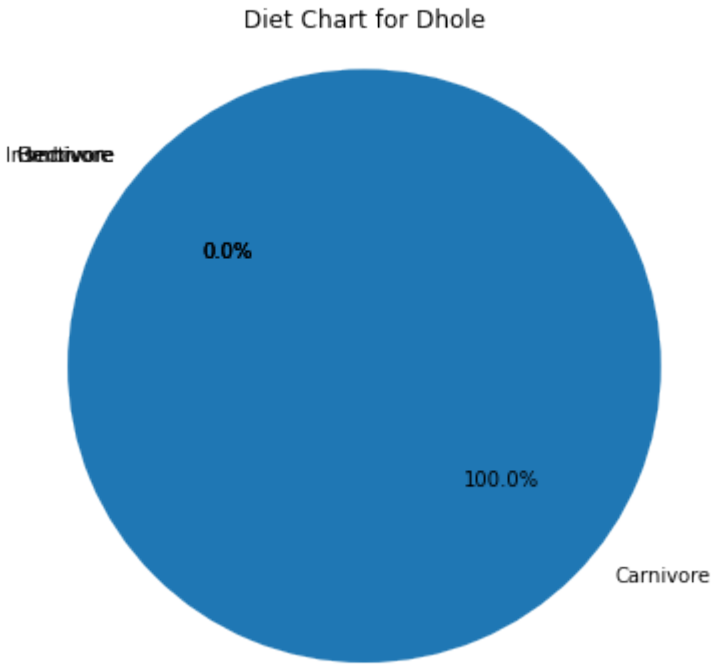

Diet Chart for Brown Bear
Insectivore
0.0%
100.0%
Omnivore

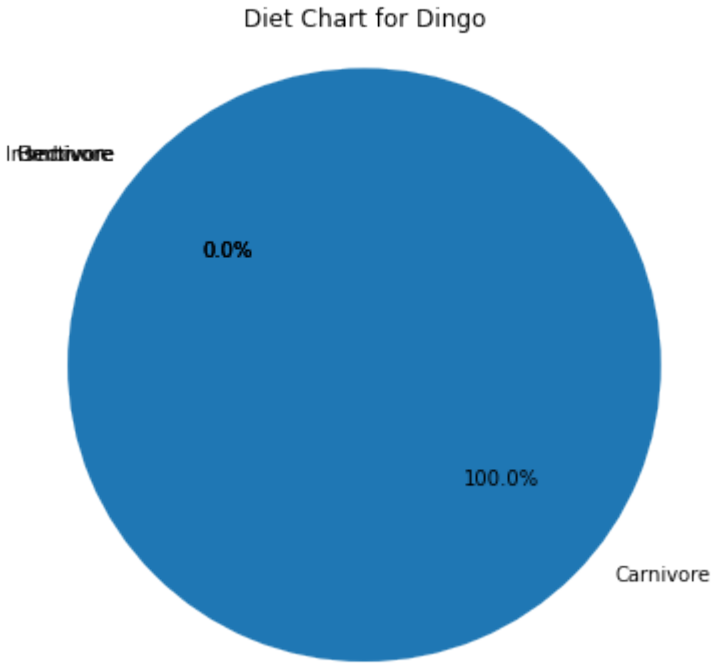

Diet Chart for Burmese Python
0.0%
100.0%
Carnivore

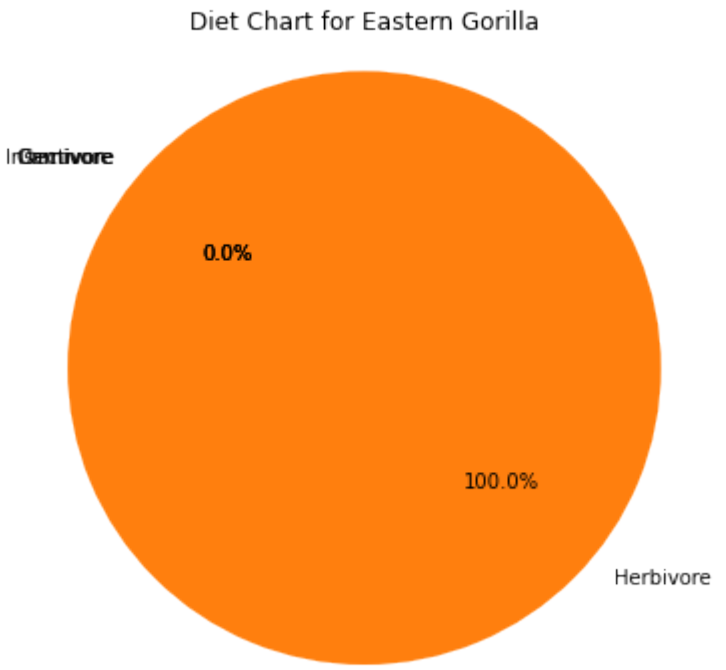

Diet Chart for Cheetah
0.0%
100.0%
Carnivore

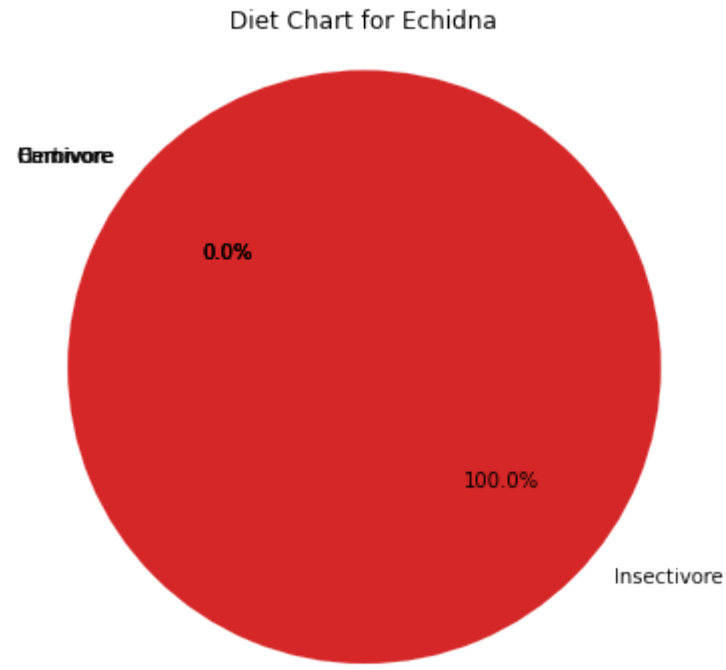

Diet Chart for Chimpanzee
Insectivore
0.0%
100.0%
Omnivore

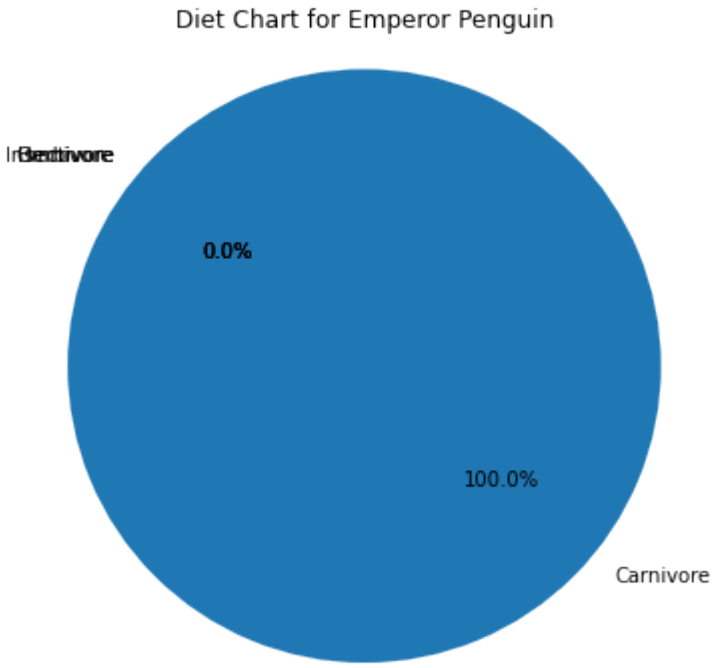

Diet Chart for Dalmatian Dog
Insectivore
0.0%
100.0%
Omnivore

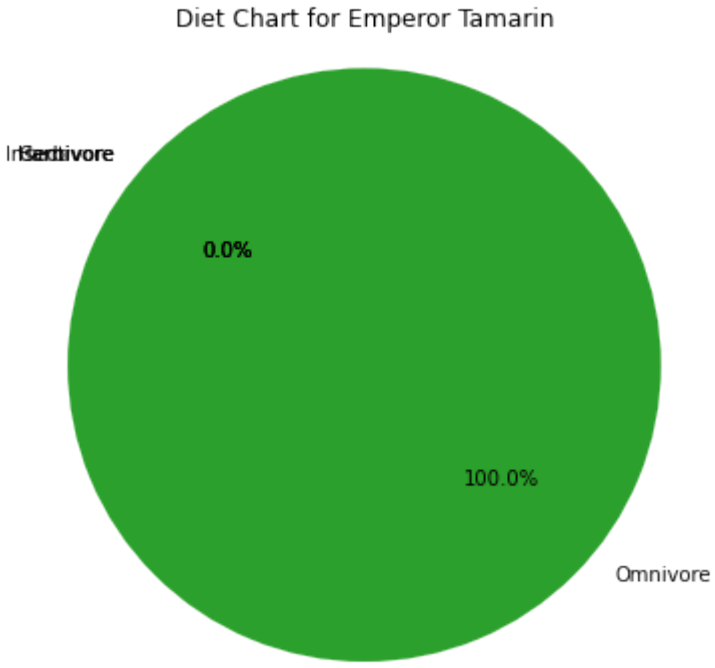

Diet Chart for Dhole
0.0%
100.0%
Carnivore

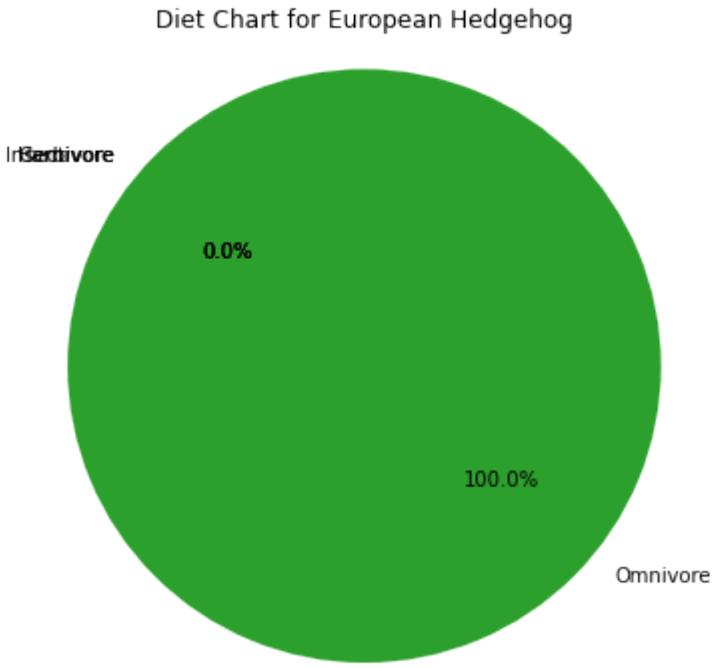

Diet Chart for Dingo
0.0%
100.0%
Carnivore

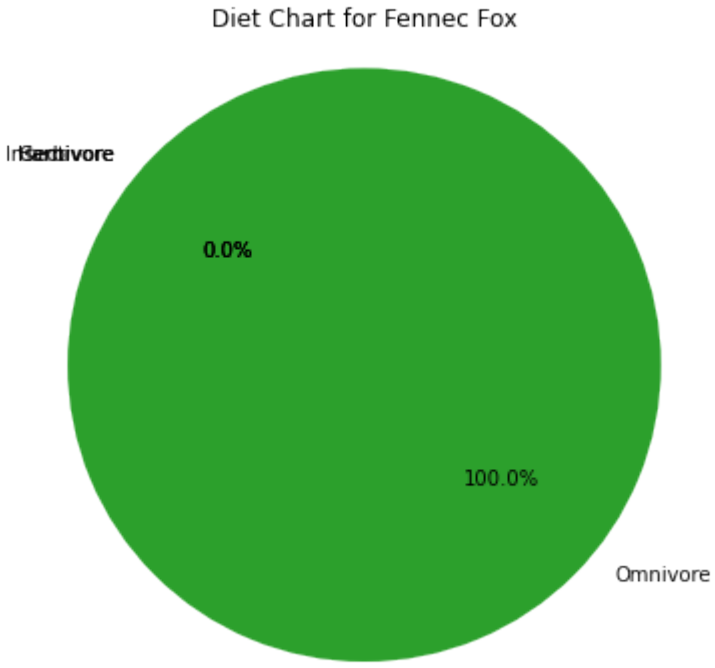

Diet Chart for Eastern Gorilla
0.0%
100.0%
Herbivore

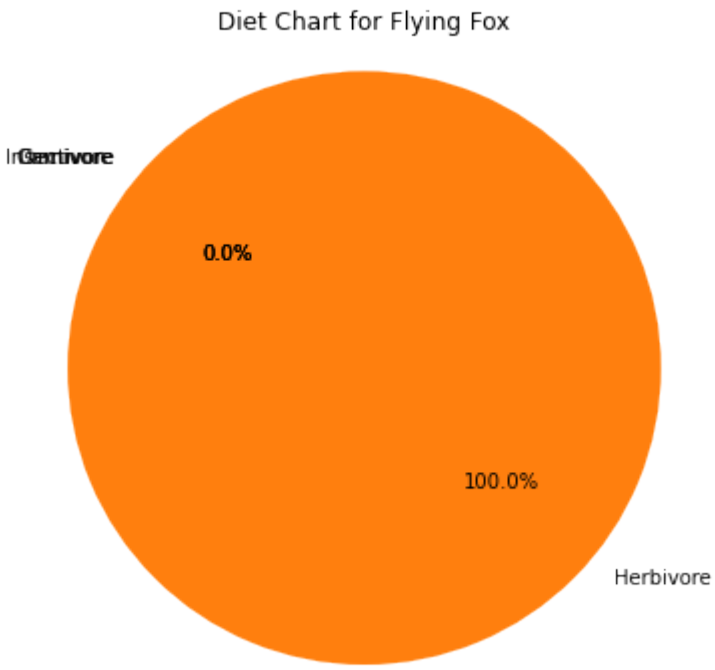

Diet Chart for Echidna
0.0%
100.0%
Insectivore

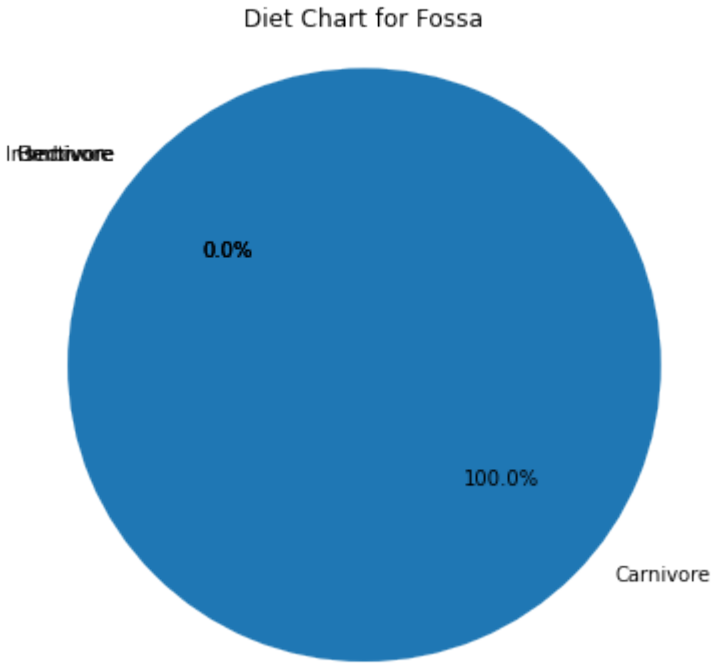

Diet Chart for Emperor Penguin
0.0%
100.0%
Carnivore

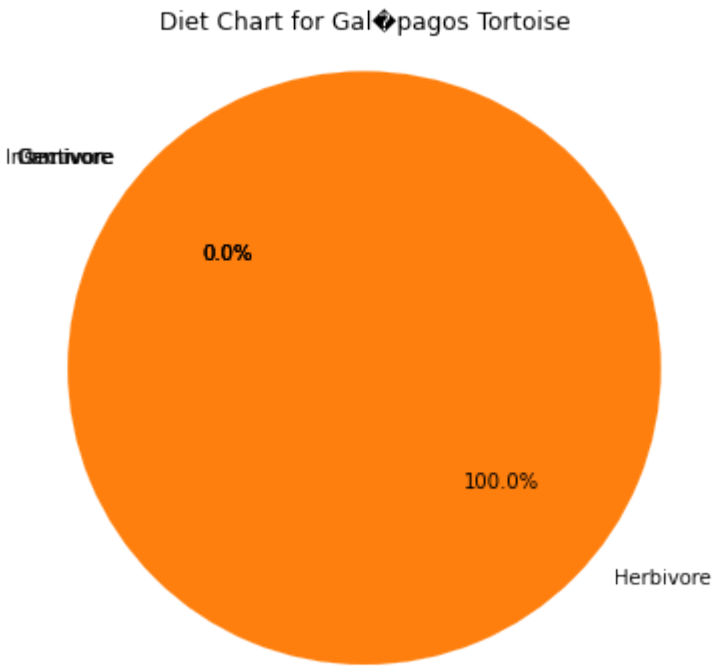

Diet Chart for Emperor Tamarin
0.0%
100.0%
Omnivore

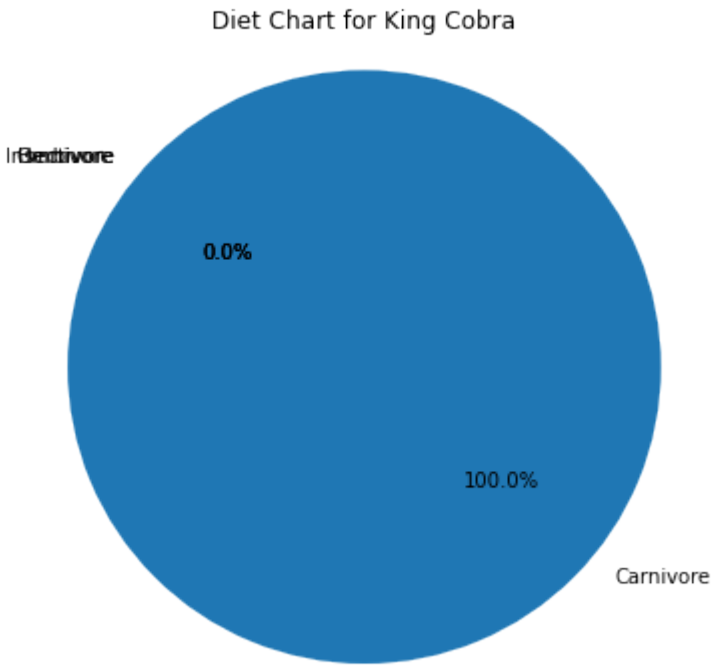

Diet Chart for European Hedgehog
Insectivore
0.0%
100.0%
Omnivore

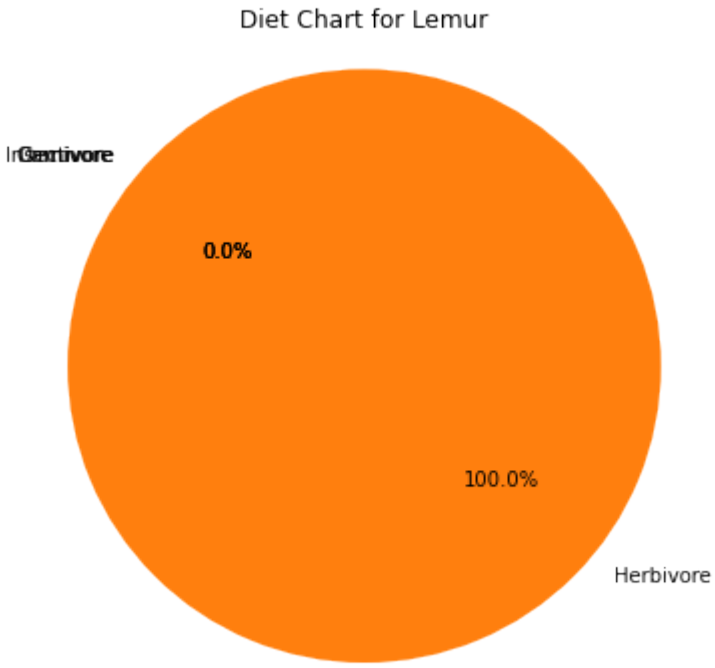

Diet Chart for Fennec Fox
Insectivore
0.0%
100.0%
Omnivore

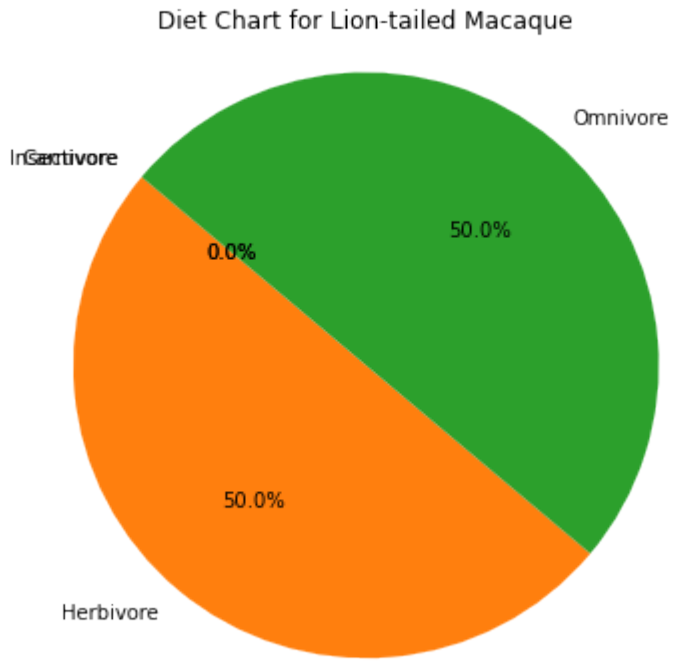

Diet Chart for Flying Fox
0.0%
100.0%
Herbivore

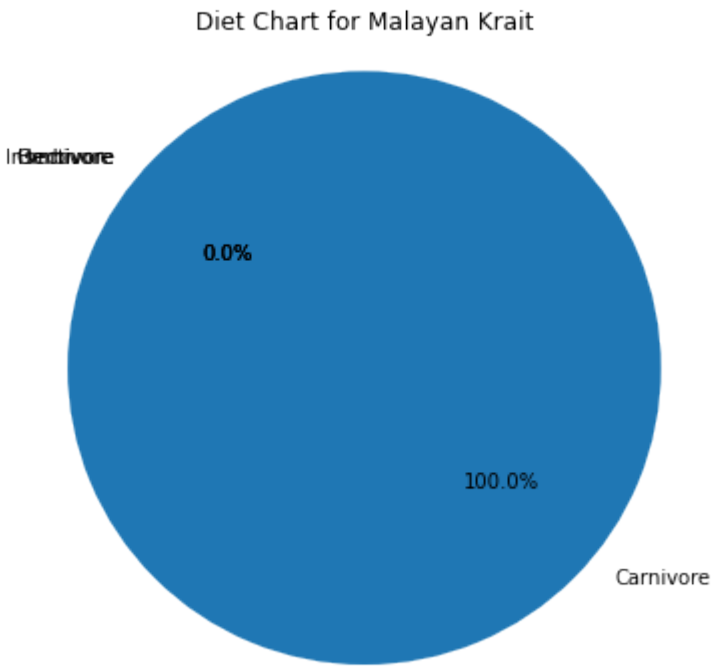

Diet Chart for Fossa
0.0%
100.0%
Carnivore

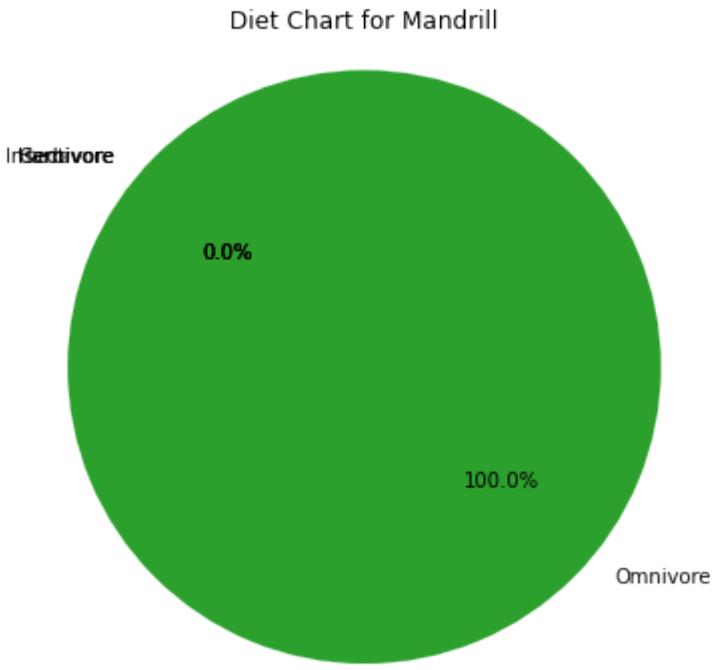

Diet Chart for Gal�pagos Tortoise
0.0%
100.0%
Herbivore

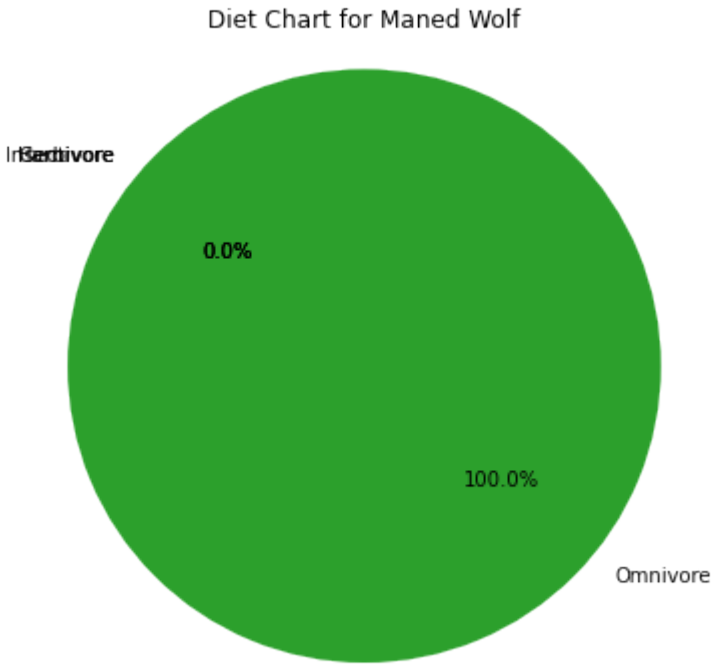

Diet Chart for King Cobra
0.0%
100.0%
Carnivore

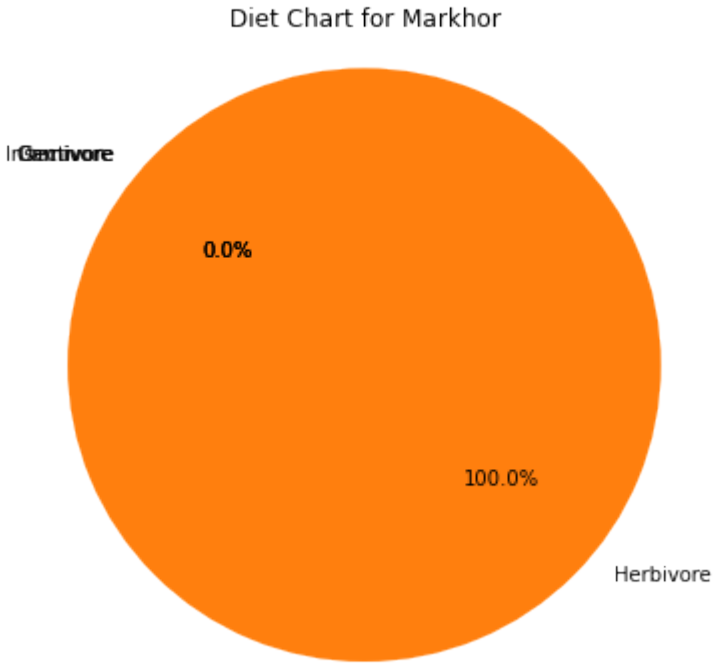

Diet Chart for Lemur
0.0%
100.0%
Herbivore

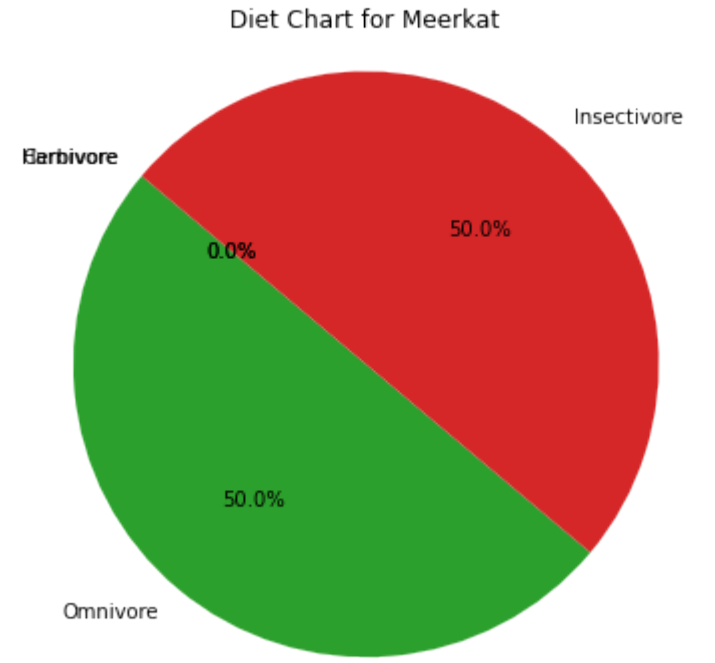

Diet Chart for Lion-tailed Macaque
Omnivore
50.0%
0.0%
50.0%
Herbivore

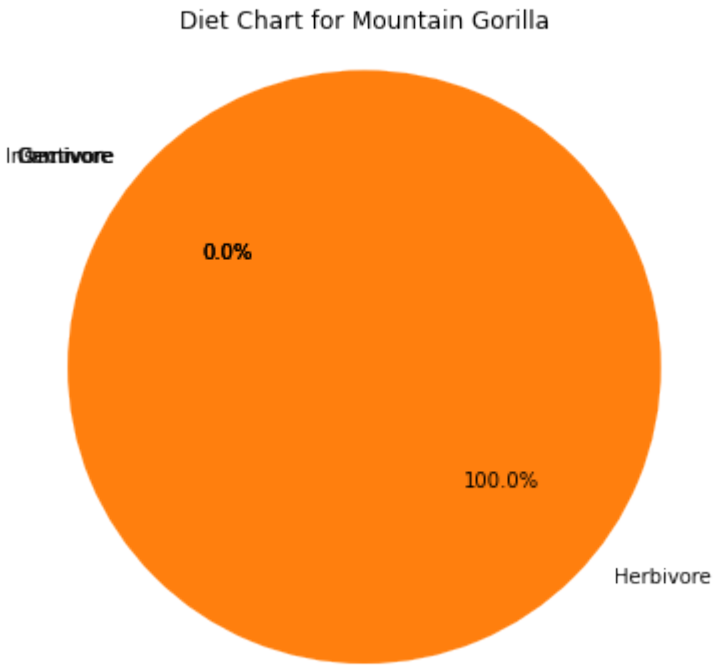

Diet Chart for Malayan Krait
0.0%
100.0%
Carnivore

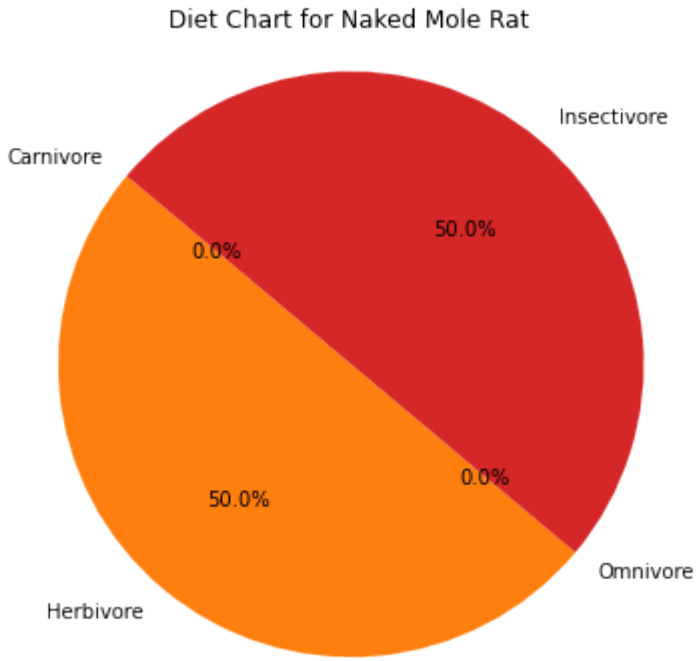

Diet Chart for Mandrill
0.0%
100.0%
Omnivore

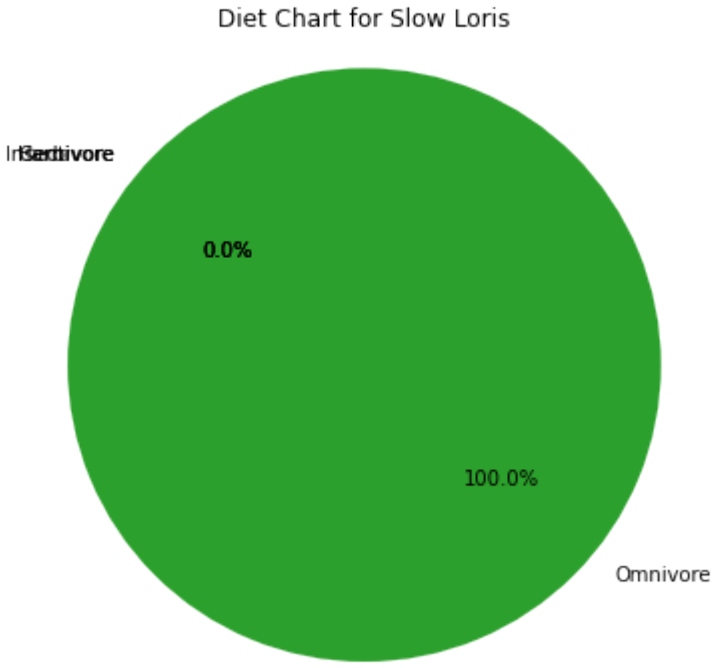

Diet Chart for Maned Wolf
0.0%
100.0%
Omnivore

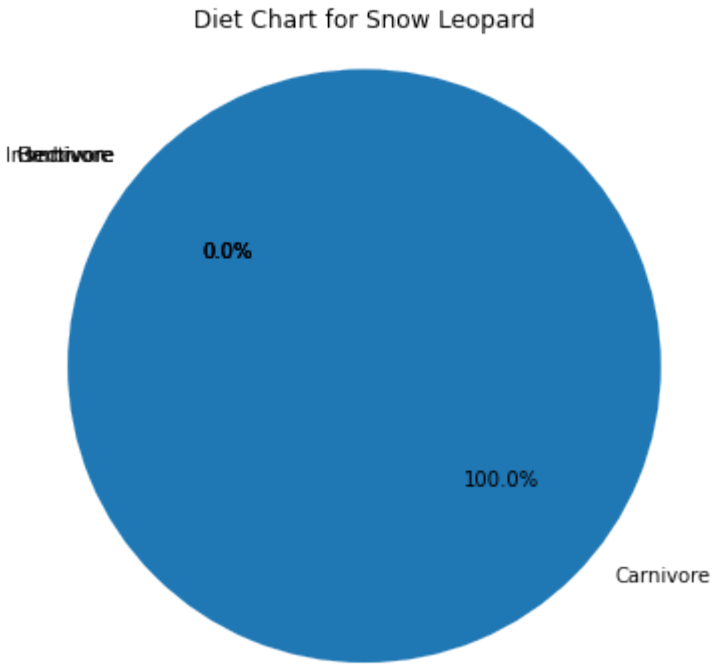

Diet Chart for Markhor
0.0%
100.0%
Herbivore

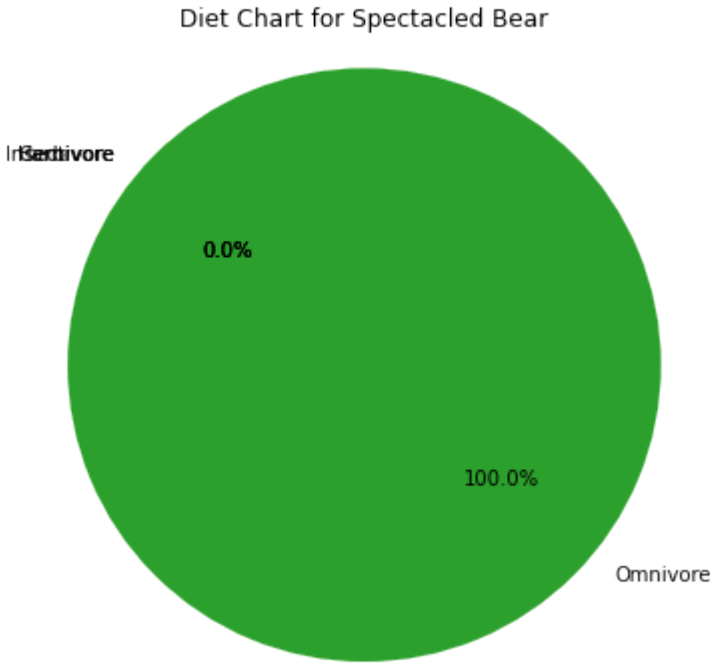

Diet Chart for Meerkat
Insectivore
50.0%
0.0%
50.0%
Omnivore

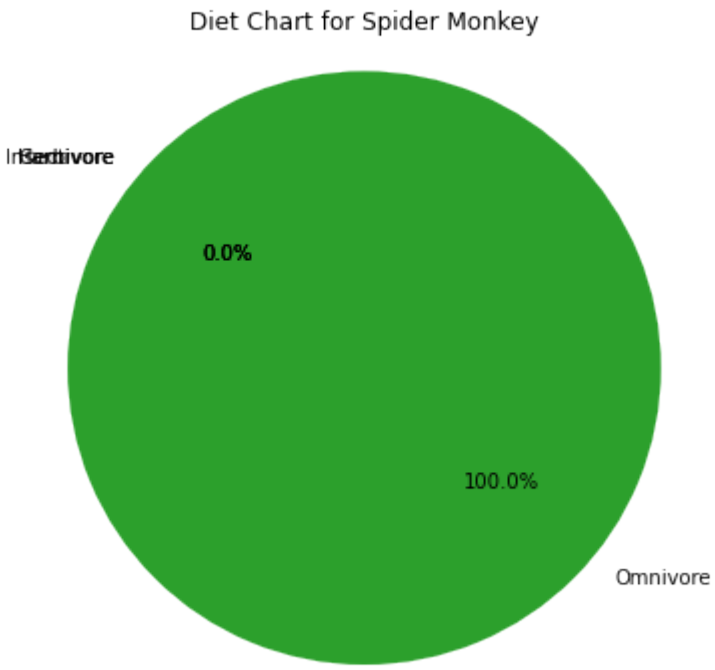

Diet Chart for Mountain Gorilla
0.0%
100.0%
Herbivore

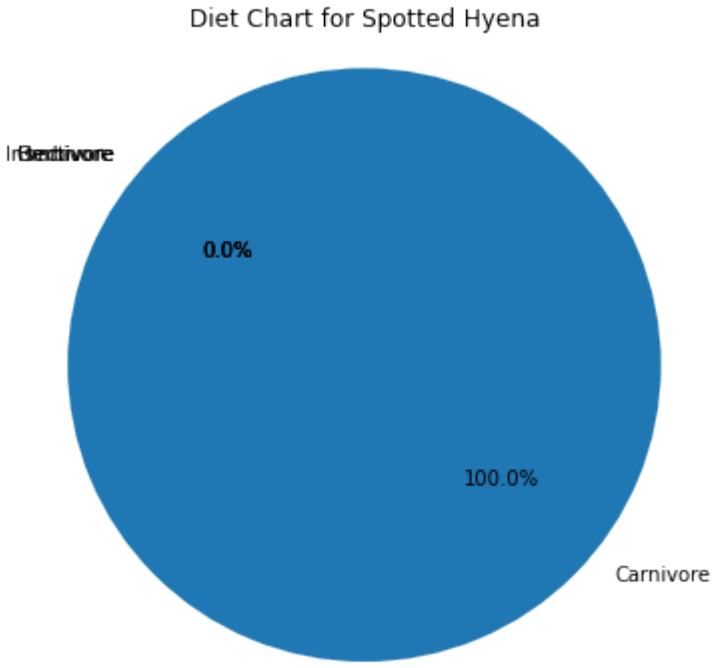

Diet Chart for Naked Mole Rat
Insectivore
Carnivore
50.0%
0.0%
0.0%
50.0%
Omnivore
Herbivore

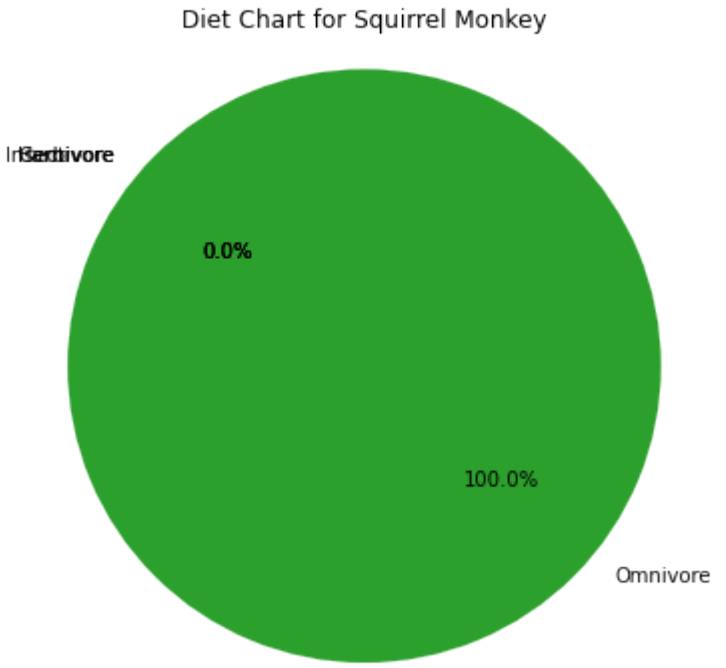

Diet Chart for Slow Loris
0.0%
100.0%
Omnivore

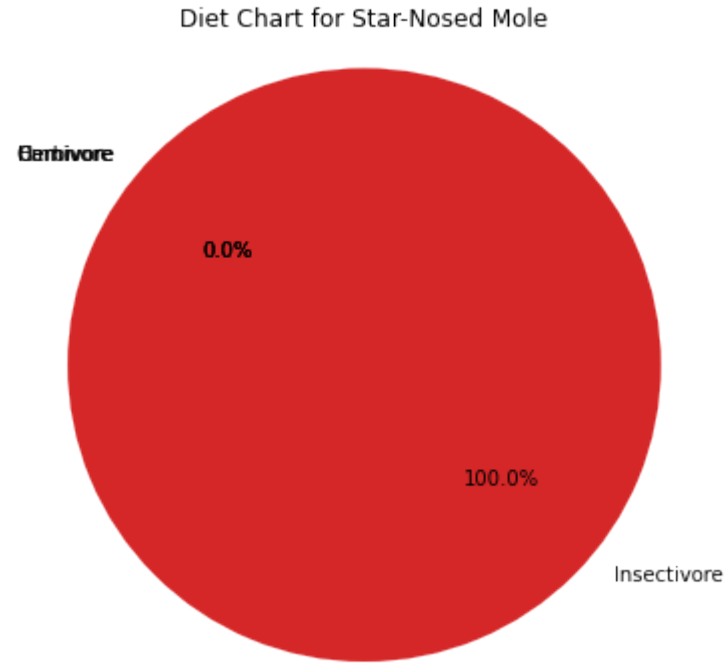

Diet Chart for Snow Leopard
0.0%
100.0%
Carnivore

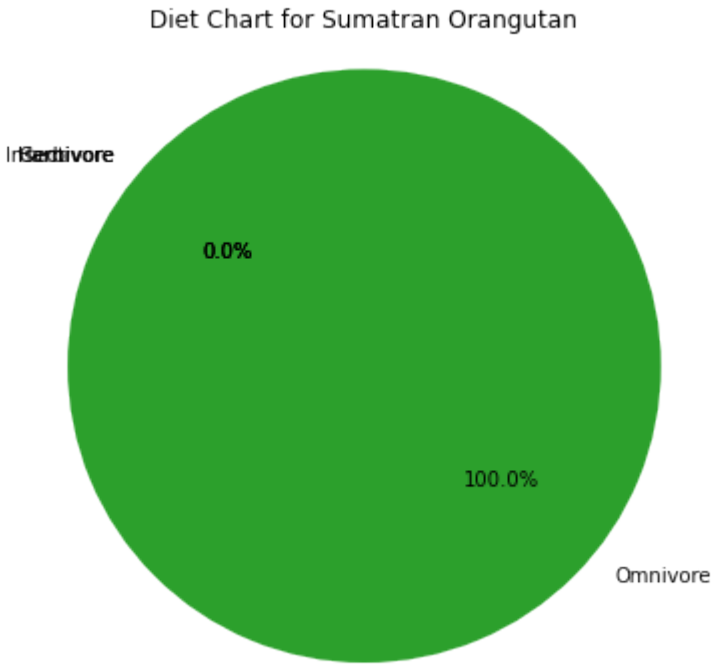

Diet Chart for Spectacled Bear
0.0%
100.0%
Omnivore

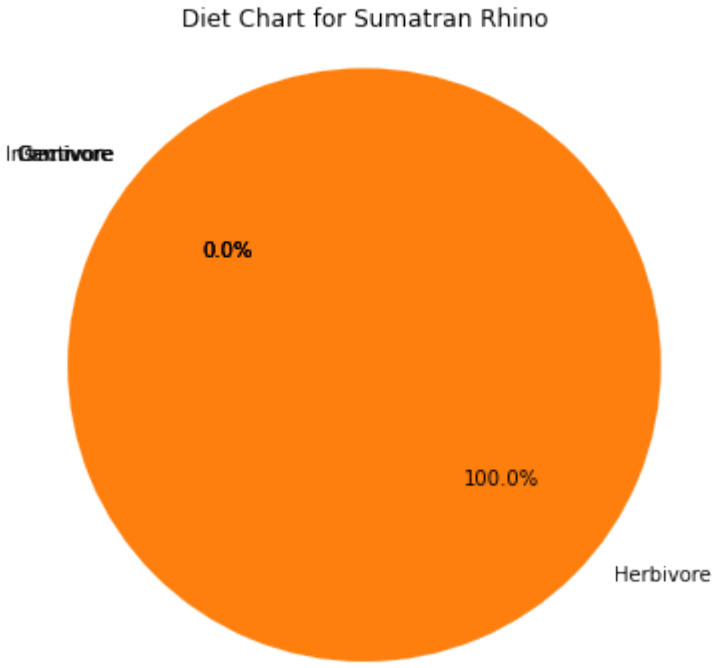

Diet Chart for Spider Monkey
0.0%
100.0%
Omnivore

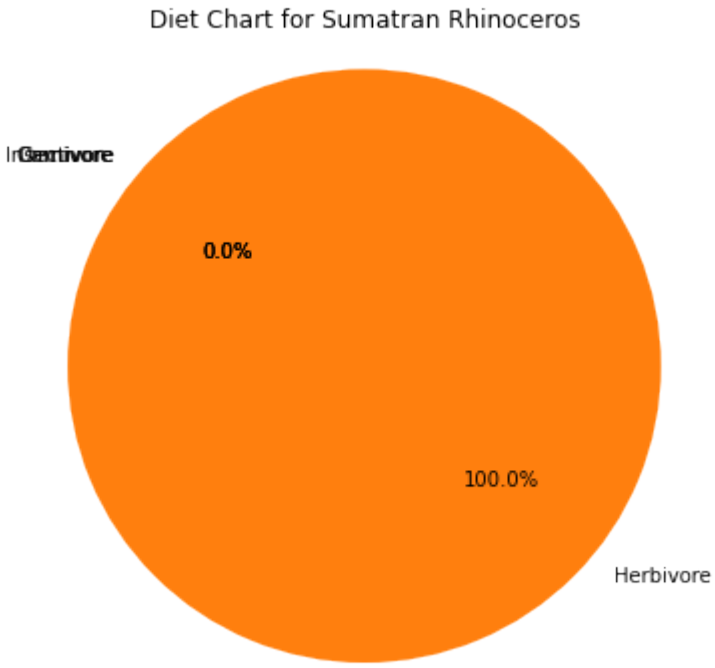

Diet Chart for Spotted Hyena
0.0%
100.0%
Carnivore

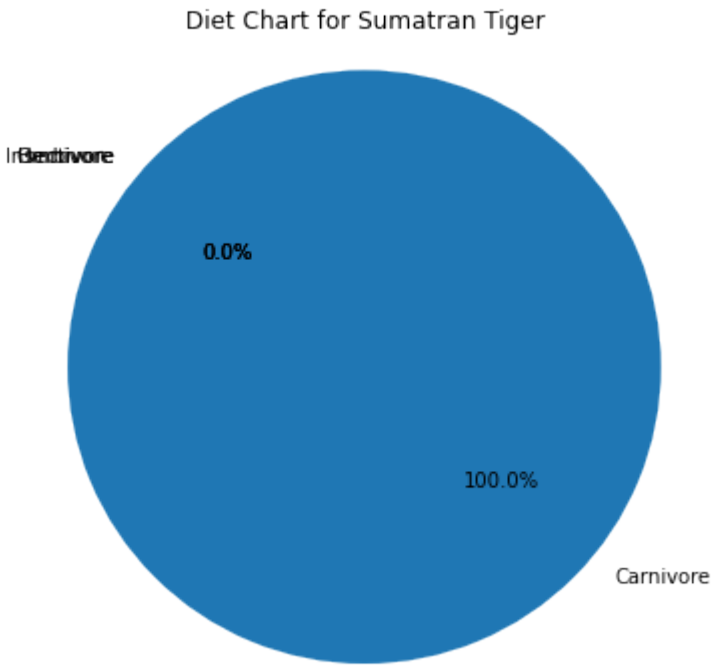

Diet Chart for Squirrel Monkey
0.0%
100.0%
Omnivore

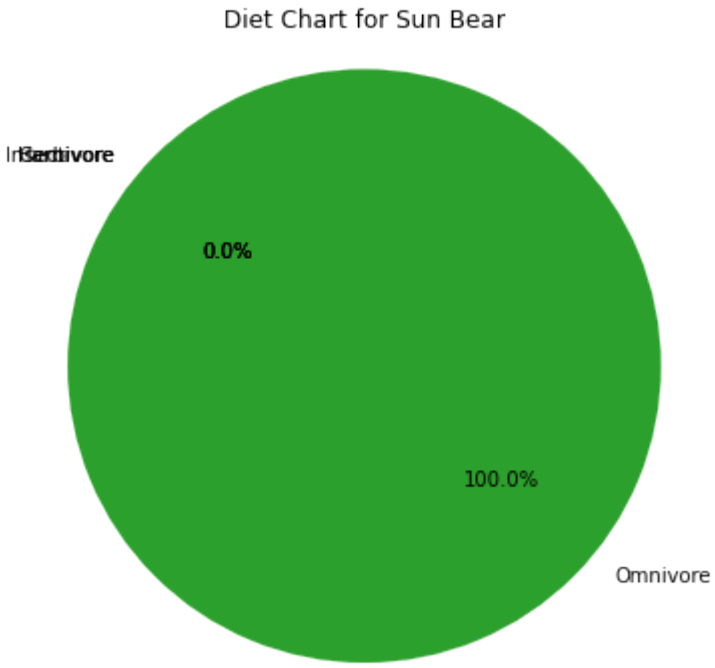

Diet Chart for Star-Nosed Mole
0.0%
100.0%
Insectivore

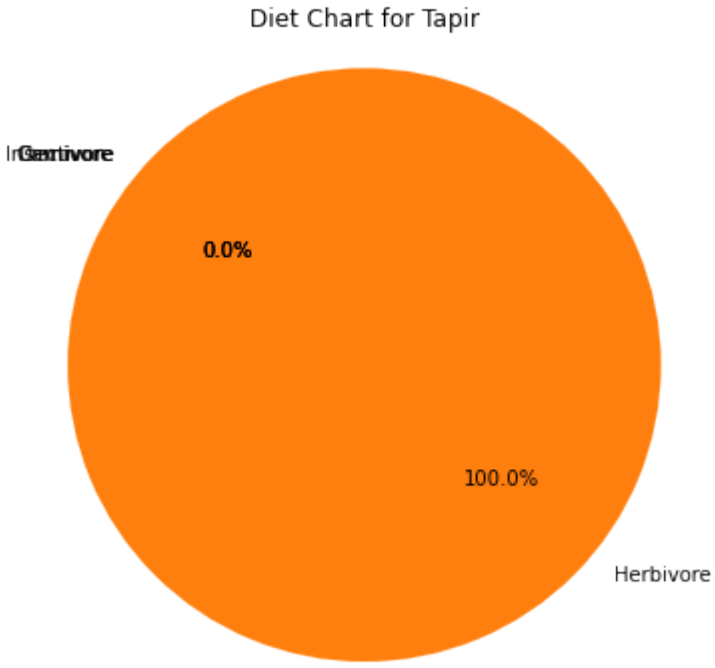

Diet Chart for Sumatran Orangutan
0.0%
100.0%
Omnivore

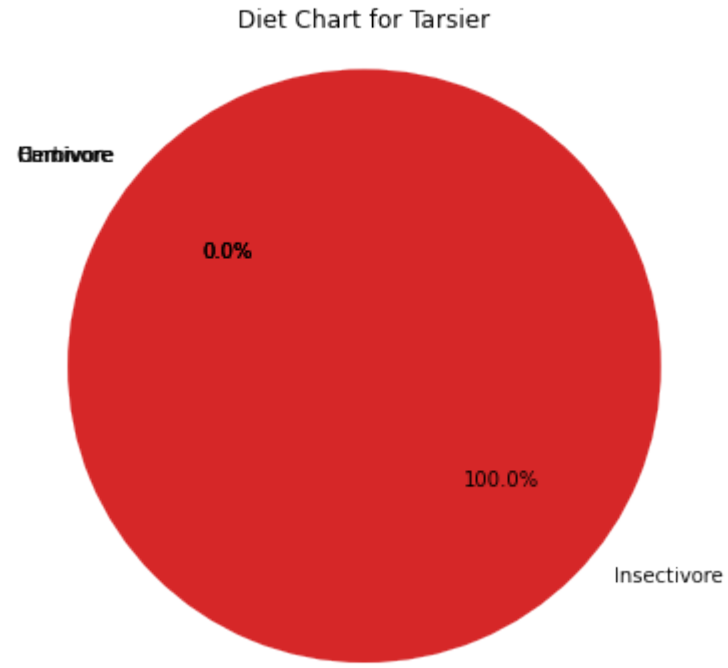

Diet Chart for Sumatran Rhino
0.0%
100.0%
Herbivore

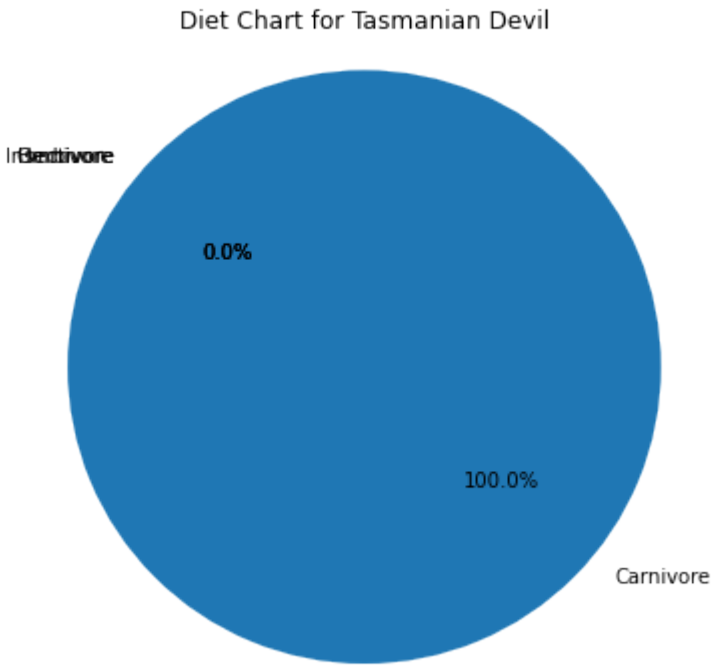

Diet Chart for Sumatran Rhinoceros
0.0%
100.0%
Herbivore

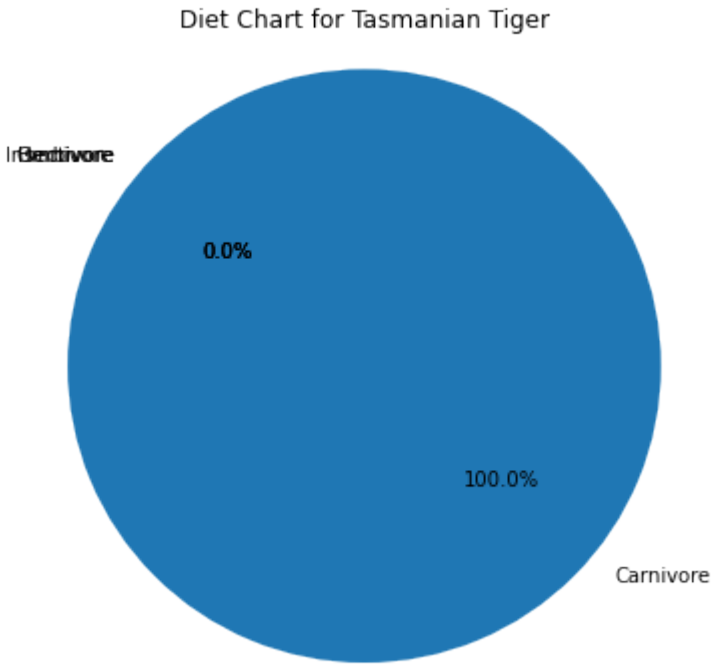

Diet Chart for Sumatran Tiger
0.0%
100.0%
Carnivore

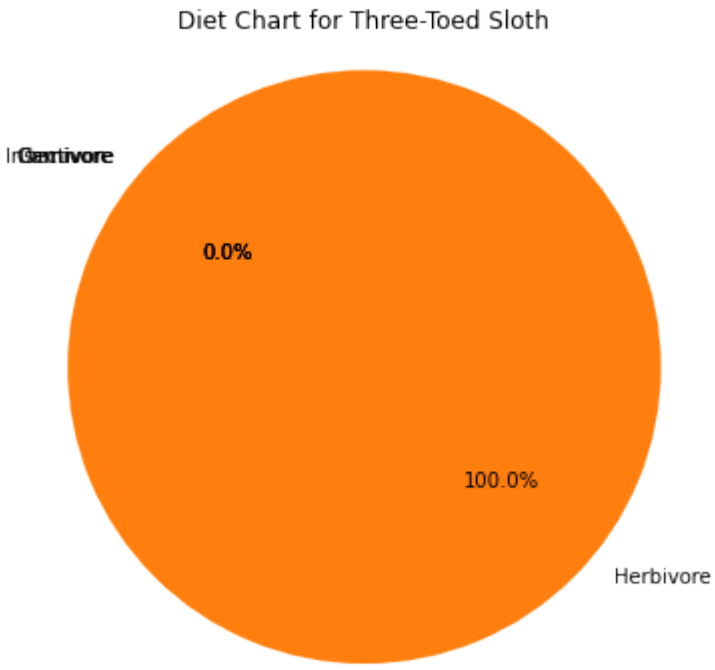

Diet Chart for Sun Bear
0.0%
100.0%
Omnivore

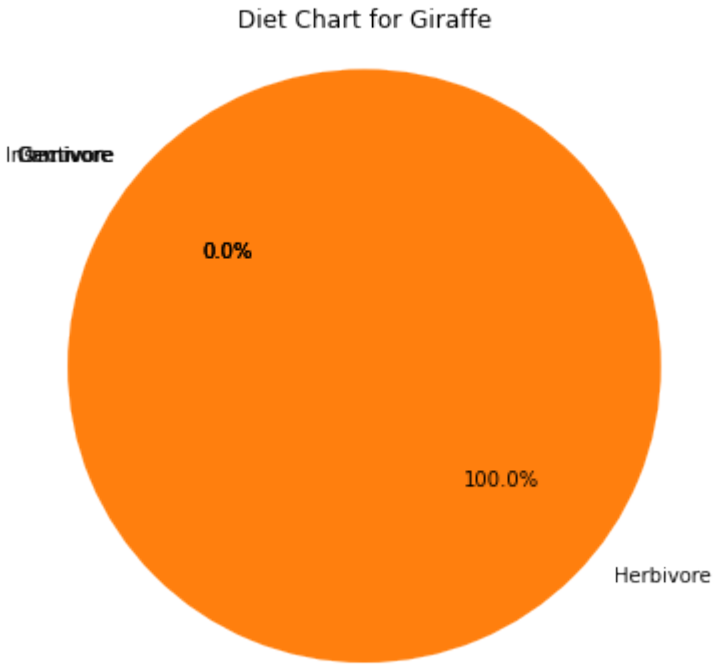

Diet Chart for Tapir
0.0%
100.0%
Herbivore


Diet Chart for Tarsier
0.0%
100.0%
Insectivore

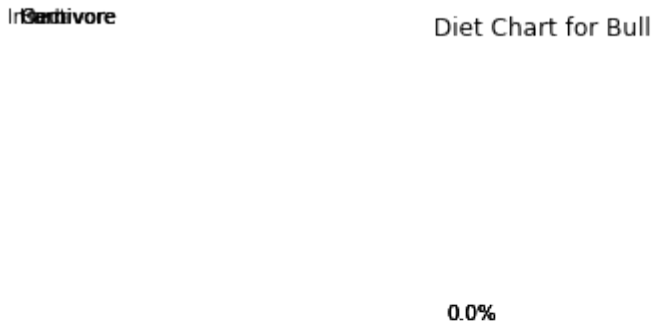

Diet Chart for Tasmanian Devil
0.0%
100.0%
Carnivore

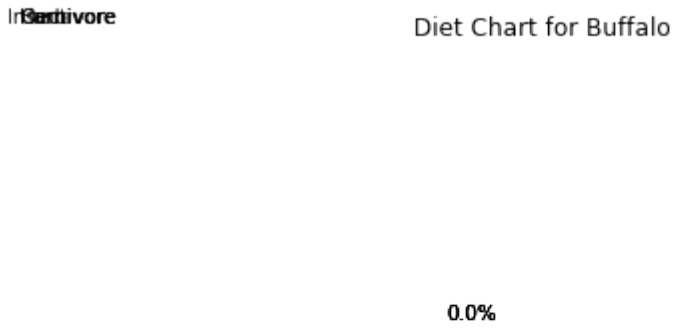

Diet Chart for Tasmanian Tiger
0.0%
100.0%
Carnivore


Diet Chart for Three-Toed Sloth
0.0%
100.0%
Herbivore

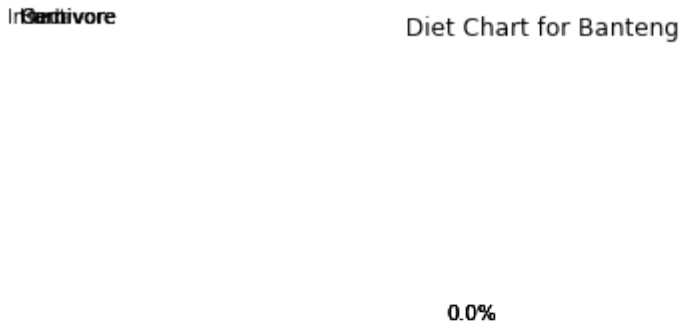


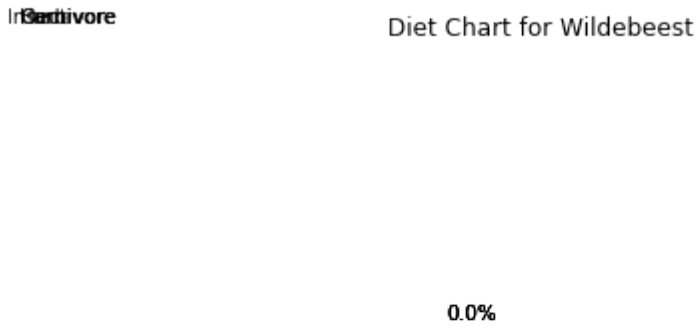

```
<ipython-input-3-43d9476236fb>:16: MatplotlibDeprecationWarning: normalize=None do
es not normalize if the sum is less than 1 but this behavior is deprecated since
3.3 until two minor releases later. After the deprecation period the default value
will be normalize=True. To prevent normalization pass normalize=False
  plt.pie([row['Feeding Behavior'].count('Carnivore'),
```

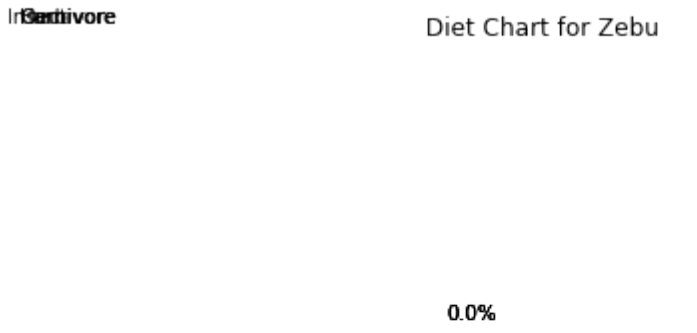

Herbivore
Diet Chart for Bull
0.0%

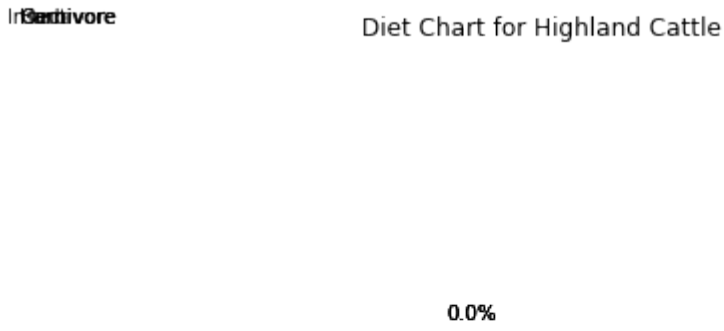

Herbivore
Diet Chart for Buffalo
0.0%


Herbivore
Diet Chart for Yak
0.0%


Herbivore
Diet Chart for Banteng
0.0%


Herbivore
Diet Chart for Wildebeest
0.0%


Herbivore
Diet Chart for Zebu
0.0%


Herbivore
Diet Chart for Highland Cattle
0.0%

```
Summary Statistics:
                     Animal Height (cm) Weight (kg)  Color Lifespan (years)  \
count                    63          63          63     63               63
unique                   63          58          61     44               43
top     Sumatran Rhinoceros       70-90       13-20  Brown            15-20
freq                      1           2           2      5                6

             Diet      Habitat Conservation Status   Family  \
count          63           63                  63       63
unique          4           38                   7       33
top     Herbivore  Rainforests       Least Concern  Bovidae
freq           24           12                  20       10

       Gestation Period (days) Feeding Behavior Nutritional Content
count                      63               63                  63
unique                     58                9                  17
top                       280         Omnivore              Varied
freq                        2               17                  21
DataFrame after handling missing values:
             Animal Height (cm) Weight (kg)         Color Lifespan (years)  \
0          Elephant     270-310   2700-6000          Grey            60-70
1              Lion      80-110     120-250           Tan           Oct-14
2          Wild Dog       75-80       18-36  Multicolored           10-Dec
3             Bison     152-186    318-1000         Brown            15-20
4          Anteater       52-91       22-41  Brown, White            15-20
..              ...         ...         ...           ...              ...
58              Yak         155         320       Spotted               15
59           Banteng         180         450           Red               25
60       Wildebeest         145         280           Tan               17
61             Zebu         140         260          Gray               22
62  Highland Cattle         152         320        Ginger               18

           Diet               Habitat    Conservation Status           Family  \
0     Herbivore      Savannah, Forest             Vulnerable     Elephantidae
1     Carnivore  Grasslands, Savannas             Vulnerable         Felidae
2     Carnivore             Savannahs             Endangered         Canidae
3     Herbivore    Grasslands, Plains        Near Threatened         Bovidae
4   Insectivore   Grasslands, Forests          Least Concern  Myrmecophagidae
..          ...                   ...                    ...             ...
58    Herbivore              Savannas             Endangered         Bovidae
59    Herbivore             Woodlands  Critically Endangered         Bovidae
60    Herbivore                Plains                Extinct         Bovidae
61    Herbivore             Farmlands             Vulnerable         Bovidae
62    Herbivore             Highlands          Least Concern         Bovidae

   Gestation Period (days) Feeding Behavior  \
0                  640-660       Herbivore
1                   98-105       Carnivore
2                       70       Carnivore
3                  270-290       Herbivore
4                  190-210      Insectivore
..                     ...             ...
58                     300          Grazing
59                     260          Grazing
60                     310          Grazing
61                     275          Grazing
62                     280          Grazing

                    Nutritional Content
0             High in fiber, low in fat
1                       High in protein
2                       High in protein
3       High in protein, iron, and zinc
4                       High in protein
```

```
..                                    ...
58          Contains omega-3 fatty acids
59  Rich source of vitamins and minerals
60         High in essential amino acids
61           High in omega-3 fatty acids
62              High in protein and zinc

[63 rows x 12 columns]
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-3-43d9476236fb> in <module>
     36 # Correlation heatmap
     37 plt.figure(figsize=(10, 8))
---> 38 sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f")
     39 plt.title("Correlation Heatmap")
     40 plt.show()

~\anaconda3\lib\site-packages\seaborn\_decorators.py in inner_f(*args, **kwargs)
     44             )
     45         kwargs.update({k: arg for k, arg in zip(sig.parameters, args)})
---> 46         return f(**kwargs)
     47     return inner_f
     48

~\anaconda3\lib\site-packages\seaborn\matrix.py in heatmap(data, vmin, vmax, cmap,
center, robust, annot, fmt, annot_kws, linewidths, linecolor, cbar, cbar_kws, cbar
_ax, square, xticklabels, yticklabels, mask, ax, **kwargs)
    533     """
    534     # Initialize the plotter object
--> 535     plotter = _HeatMapper(data, vmin, vmax, cmap, center, robust, annot, f
mt,
    536                           annot_kws, cbar, cbar_kws, xticklabels,
    537                           yticklabels, mask)

~\anaconda3\lib\site-packages\seaborn\matrix.py in __init__(self, data, vmin, vma
x, cmap, center, robust, annot, fmt, annot_kws, cbar, cbar_kws, xticklabels, ytick
labels, mask)
    153
    154         # Determine good default values for the colormapping
--> 155         self._determine_cmap_params(plot_data, vmin, vmax,
    156                                     cmap, center, robust)
    157

~\anaconda3\lib\site-packages\seaborn\matrix.py in _determine_cmap_params(self, pl
ot_data, vmin, vmax, cmap, center, robust)
    192                 vmin = np.nanpercentile(calc_data, 2)
    193             else:
--> 194                 vmin = np.nanmin(calc_data)
    195         if vmax is None:
    196             if robust:

<__array_function__ internals> in nanmin(*args, **kwargs)

~\anaconda3\lib\site-packages\numpy\lib\nanfunctions.py in nanmin(a, axis, out, ke
epdims)
    317         # Fast, but not safe for subclasses of ndarray, or object arrays,
    318         # which do not implement isnan (gh-9009), or fmin correctly (gh-89
75)
--> 319         res = np.fmin.reduce(a, axis=axis, out=out, **kwargs)
    320         if np.isnan(res).any():
    321             warnings.warn("All-NaN slice encountered", RuntimeWarning,

ValueError: zero-size array to reduction operation fmin which has no identity
```

```
<Figure size 720x576 with 0 Axes>
```

In [109…

```
import pandas as pd
import numpy as np
from abc import ABC, abstractmethod

class Animal(ABC):
    def __init__(
        self,
        weight,
        age,
        is_neutered,
        is_active,
        health_condition=None
    ):
        self.weight = weight
        self.age = age
        self.is_neutered = is_neutered
        self.is_active = is_active
        self.health_condition = health_condition
        super().__init__()

    @abstractmethod
    def estimate_MER_factor(self):
        raise NotImplementedError()

    @abstractmethod
    def estimate_MER_factor_weight_loss(self):
        raise NotImplementedError()

class DietPlanner():
    def __init__(
        self,
        animal,
        dry_food_dict,
        wet_food_dict,
        proportion_dry,
        proportion_wet,
        MER_factor=None
    ):
        self.animal = animal

        self.dry_food_dict = dry_food_dict
        self.wet_food_dict = wet_food_dict

        if proportion_dry + proportion_wet != 1.:
            raise ValueError('The sum of the proportion of dry and wet food should

        self.proportion_dry = proportion_dry
        self.proportion_wet = proportion_wet

        self.MER_factor = MER_factor

        return

    def plan_diet(self):
        if self.MER_factor is None:
            _MER_factor = self.animal.estimate_MER_factor()
        else:
            _MER_factor = self.MER_factor

        print(f"Estimated MER Factor: {_MER_factor}")
        # Add your diet planning logic here
        print("Diet planning logic goes here")
```

```
def load_animal_dataset(animal_diet):
    df = pd.read_csv(animal_diet)
    animals = []
    for index, row in df.iterrows():
        animal_data = {
            'weight': row['Weight (kg)'],
            'age': row['Lifespan (years)'],
            'is_neutered': False,  # You might want to adjust this based on actual
            'is_active': True,     # You might want to adjust this based on actual
            'health_condition': None
        }
        print(f"Loading data for {row['Animal']}")
        animals.append(GenericAnimal(animal_data))
    return animals

animals = load_animal_dataset('animal_diet.csv')
```

```
Loading data for Elephant
Loading data for Lion
Loading data for Wild Dog
Loading data for Bison
Loading data for Anteater
Loading data for Horse
Loading data for Fox
Loading data for Bengal Fox
Loading data for Bengal Tiger
Loading data for Black Rhinoceros
Loading data for Bornean Orangutan
Loading data for Brown Bear
Loading data for Burmese Python
Loading data for Cheetah
Loading data for Chimpanzee
Loading data for Dalmatian Dog
Loading data for Dhole
Loading data for Dingo
Loading data for Eastern Gorilla
Loading data for Echidna
Loading data for Emperor Penguin
Loading data for Emperor Tamarin
Loading data for European Hedgehog
Loading data for Fennec Fox
Loading data for Flying Fox
Loading data for Fossa
Loading data for Gal�pagos Tortoise
Loading data for King Cobra
Loading data for Lemur
Loading data for Lion-tailed Macaque
Loading data for Malayan Krait
Loading data for Mandrill
Loading data for Maned Wolf
Loading data for Markhor
Loading data for Meerkat
Loading data for Mountain Gorilla
Loading data for Naked Mole Rat
Loading data for Slow Loris
Loading data for Snow Leopard
Loading data for Spectacled Bear
Loading data for Spider Monkey
Loading data for Spotted Hyena
Loading data for Squirrel Monkey
Loading data for Star-Nosed Mole
Loading data for Sumatran Orangutan
Loading data for Sumatran Rhino
Loading data for Sumatran Rhinoceros
Loading data for Sumatran Tiger
Loading data for Sun Bear
Loading data for Tapir
Loading data for Tarsier
Loading data for Tasmanian Devil
Loading data for Tasmanian Tiger
Loading data for Three-Toed Sloth
Loading data for Giraffe
Loading data for Cow
Loading data for Bull
Loading data for Buffalo
Loading data for Yak
Loading data for Banteng
Loading data for Wildebeest
Loading data for Zebu
Loading data for Highland Cattle
```

In [ ]: