Data Analysis and Visualization with Python: A case Study

Arash

December 1, 2023

Contents

Introduction

The dataset under analysis, named ‘SleepInMammals,’ provides valuable information about sleep patterns and characteristics in various mammalian species. This dataset encompasses a diverse range of mammals, offering insights into their behaviors, physical attributes, and lifespans.

Dataset Overview

- **Source:** [Include source information]
- **Columns:**
  - **Species:** The species of the mammal.
  - **Body Weight (kg):** The body weight of the mammal in kilograms.
  - **Brain Weight (g):** The brain weight of the mammal in grams.
  - **Maximum Life Span (years):** The maximum lifespan of the mammal in years.
  - [Include other relevant columns]

Objectives

The primary objectives of this analysis are to:

1. Explore the relationships between body weight and brain weight.
2. Identify and handle outliers in the dataset.
3. Apply log transformations to certain variables for improved visualization.
4. Investigate the distribution of log-transformed variables.
5. Calculate and analyze the brain-to-body weight ratio.
6. Visualize the dataset through scatter plots, box plots, and bar plots to gain insights.

Through this analysis, we aim to uncover patterns, trends, and anomalies in the sleep patterns of mammals, contributing to a deeper understanding of the relationships between various physiological factors.

```python
from matplotlib import pyplot as plt
import numpy as np
import pandas as pd

animals = pd.read_csv('SleepInMammals.csv')

animals.head(5)
```

```
##              Species of animal  ...  Overall danger index
## 0  African giant pouched rat  ...                     3
## 1             Asian elephant  ...                     4
## 2                     Baboon  ...                     4
## 3              Big brown bat  ...                     1
## 4           Brazilian tapir  ...                     4
##
## [5 rows x 11 columns]
```

```
animals.info()
```

```
## <class 'pandas.core.frame.DataFrame'>
## RangeIndex: 40 entries, 0 to 39
## Data columns (total 11 columns):
##  #   Column                      Non-Null Count  Dtype
## ---  ------                      --------------  -----
##  0   Species of animal           40 non-null     object
##  1   Body Weight (kg)            40 non-null     float64
##  2   Brain Weight (g)            40 non-null     float64
##  3   Slow wave sleep (hrs/day)   40 non-null     float64
##  4   Paradoxical sleep (hrs/day)  40 non-null     float64
##  5   Total sleep (hrs/day)       40 non-null     float64
##  6   Maximum life span (years)   40 non-null     float64
##  7   Gestation time (days)       40 non-null     float64
##  8   Predation index             40 non-null     int64
##  9   Sleep exposure index        40 non-null     int64
##  10  Overall danger index        40 non-null     int64
## dtypes: float64(7), int64(3), object(1)
## memory usage: 3.6+ KB
```

```
print(animals.columns)
```

```
## Index(['Species of animal', 'Body Weight (kg)', 'Brain Weight (g)',
##        'Slow wave sleep (hrs/day)', 'Paradoxical sleep (hrs/day)',
##        'Total sleep (hrs/day)', 'Maximum life span (years)',
##        'Gestation time (days)', 'Predation index', 'Sleep exposure index',
##        'Overall danger index'],
##       dtype='object')
```

```
print(animals['Species of animal'])
```

```
## 0     African giant pouched rat
## 1                Asian elephant
## 2                        Baboon
## 3                 Big brown bat
## 4              Brazilian tapir
## 5                           Cat
## 6                    Chimpanzee
## 7                    Chinchilla
## 8                           Cow
## 9         Eastern American mole
## 10                      Echidna
## 11            European hedgehog
## 12                       Galago
## 13                         Goat
## 14               Golden hamster
## 15                    Gray seal
## 16              Ground squirrel
## 17                   Guinea pig
## 18                        Horse
## 19    Lesser short-tailed shrew
## 20             Little brown bat
## 21                        Human
## 22                        Mouse
## 23                   Musk shrew
## 24          N. American opossum
## 25        Nine-banded armadillo
## 26                   Owl monkey
## 27                 Patas monkey
## 28                   Phanlanger
## 29                          Pig
## 30                       Rabbit
## 31                          Rat
## 32                      Red fox
## 33                Rhesus monkey
## 34                        Sheep
## 35                       Tenrec
## 36                   Tree hyrax
## 37                   Tree shrew
## 38                       Vervet
## 39                Water opossum
## Name: Species of animal, dtype: object
```

Data Cleaning and Preparation

Let's clean and prepare the data for analysis.

```
# Define a dictionary mapping old column names to new column names
column_mapping = {
    'Species of animal': 'species',
    'Maximum life span (years)': 'age'
    #'old_name2': 'new_name2',
    # Add more mappings as needed
}

# Rename the columns using the rename method
animals.rename(columns=column_mapping, inplace=True)


print(animals['species'])
```

```
## 0     African giant pouched rat
## 1                Asian elephant
## 2                       Baboon
## 3                  Big brown bat
## 4              Brazilian tapir
## 5                          Cat
## 6                   Chimpanzee
## 7                   Chinchilla
## 8                          Cow
## 9          Eastern American mole
## 10                     Echidna
## 11            European hedgehog
## 12                       Galago
## 13                         Goat
## 14               Golden hamster
## 15                    Gray seal
## 16              Ground squirrel
## 17                   Guinea pig
## 18                        Horse
## 19    Lesser short-tailed shrew
## 20              Little brown bat
## 21                        Human
## 22                        Mouse
## 23                   Musk shrew
## 24          N. American opossum
## 25        Nine-banded armadillo
## 26                   Owl monkey
## 27                 Patas monkey
## 28                   Phanlanger
## 29                          Pig
## 30                       Rabbit
## 31                          Rat
## 32                      Red fox
## 33                Rhesus monkey
## 34                        Sheep
## 35                       Tenrec
## 36                   Tree hyrax
## 37                   Tree shrew
## 38                       Vervet
## 39               Water opossum
## Name: species, dtype: object
```

```
print(animals.describe())
```

```
##        Body Weight (kg)  ...  Overall danger index
## count         40.000000  ...             40.000000
## mean         105.745850  ...              2.700000
## std          411.627286  ...              1.417835
## min            0.005000  ...              1.000000
## 25%            0.260000  ...              1.000000
## 50%            2.250000  ...              2.500000
## 75%           14.827500  ...              4.000000
## max         2547.000000  ...              5.000000
##
## [8 rows x 10 columns]
```

```
# Select a specific column


# Select rows based on a condition
print(animals[animals['age'] > 25])
```

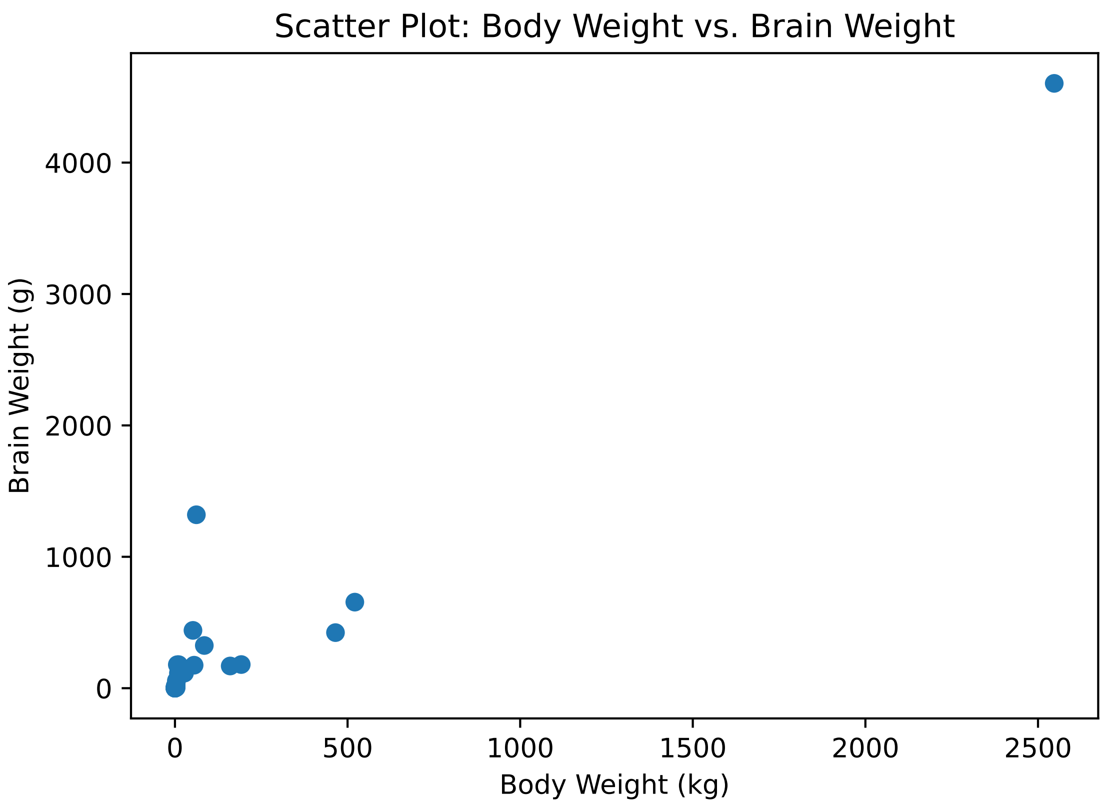
```
##            species  ...  Overall danger index
## 1    Asian elephant  ...                     4
## 2           Baboon  ...                     4
## 4   Brazilian tapir  ...                       4
## 5               Cat  ...                       1
## 6        Chimpanzee  ...                       1
## 8               Cow  ...                       5
## 10          Echidna  ...                       2
## 15        Gray seal  ...                       1
## 18            Horse  ...                       5
## 21            Human  ...                       1
## 29              Pig  ...                       4
## 33    Rhesus monkey  ...                       2
##
## [12 rows x 11 columns]
```

Data Visualization

Let's visualize the data to gain insights.

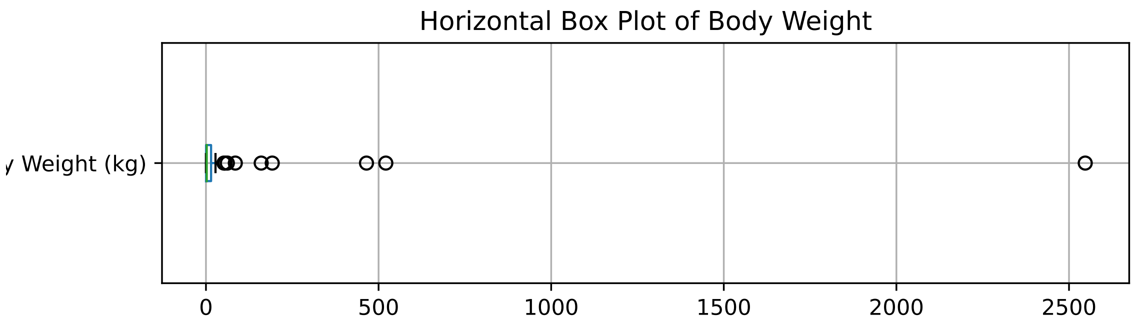
Scatter Plot: Body Weight vs. Brain Weight

```python
plt.scatter(animals['Body Weight (kg)'], animals['Brain Weight (g)'])
plt.xlabel('Body Weight (kg)')
plt.ylabel('Brain Weight (g)')
plt.title('Scatter Plot: Body Weight vs. Brain Weight')
plt.show()
```

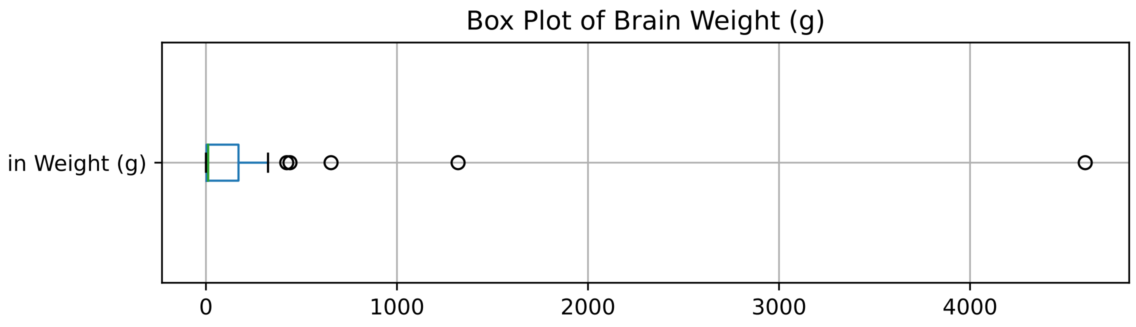


Box Plot: Body Weight and Brain Weight

```
# Box plot to identify outliers in 'Body Weight (kg)' column
plt.figure(figsize=(8, 2))  # Adjust the figsize based on your preferences
animals.boxplot(column='Body Weight (kg)', vert=False)  # Set vert to False for horizontal box plot
plt.title('Horizontal Box Plot of Body Weight')
plt.show()
```



```
animals.boxplot(column='Brain Weight (g)', vert=False)
plt.title('Box Plot of Brain Weight (g)')
plt.show()
```



Outlier Removal

Let’s remove outliers using z-score.

```
"""
# Removing outliers using z-score
z_scores_body_weight = (animals['Body Weight (kg)'] - animals['Body Weight (kg)'].mean()) / animals['Bo
z_scores_brain_weight = (animals['Brain Weight (g)'] - animals['Brain Weight (g)'].mean()) / animals['B

animals_no_outliers = animals[(abs(z_scores_body_weight) < 3) & (abs(z_scores_brain_weight) < 3)]

"""
```

```
## "\n# Removing outliers using z-score\nz_scores_body_weight = (animals['Body Weight (kg)'] - animals[
```

Log Transformation

Let’s apply a log transformation to certain columns.

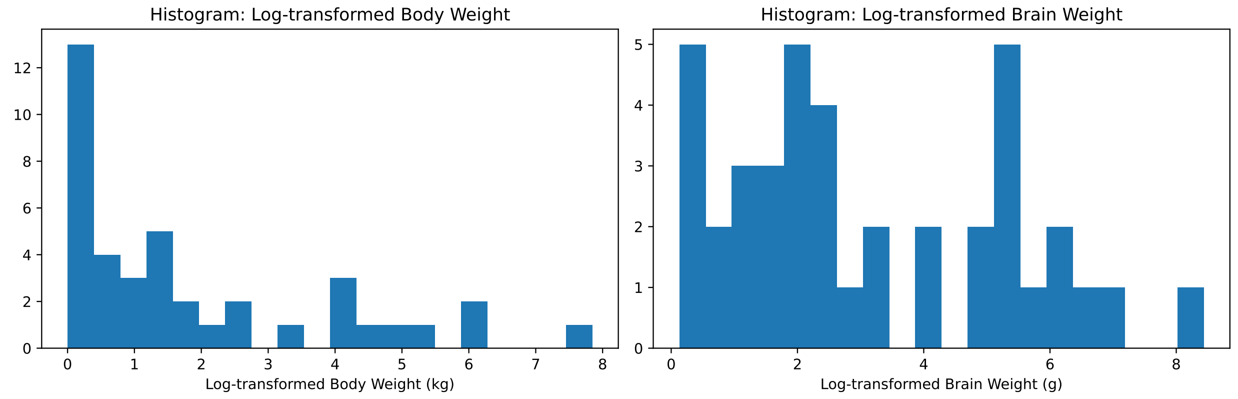
```
# ... (Python code for log transformation)
# Apply log transformation to 'Body Weight (kg)' and 'Brain Weight (g)'
animals['Body Weight (kg)_log'] = np.log1p(animals['Body Weight (kg)'])
animals['Brain Weight (g)_log'] = np.log1p(animals['Brain Weight (g)'])
```

Histograms of Log-Transformed Columns

```
# Visual inspection - histogram of log-transformed columns
fig, axes = plt.subplots(1, 2, figsize=(12, 4))
axes[0].hist(animals['Body Weight (kg)_log'], bins=20)
```

```
axes[0].set_title('Histogram: Log-transformed Body Weight')
axes[0].set_xlabel('Log-transformed Body Weight (kg)')

axes[1].hist(animals['Brain Weight (g)_log'], bins=20)
```

```
axes[1].set_title('Histogram: Log-transformed Brain Weight')
axes[1].set_xlabel('Log-transformed Brain Weight (g)')

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



Histograms and boxplots of Log-Transformed Columns

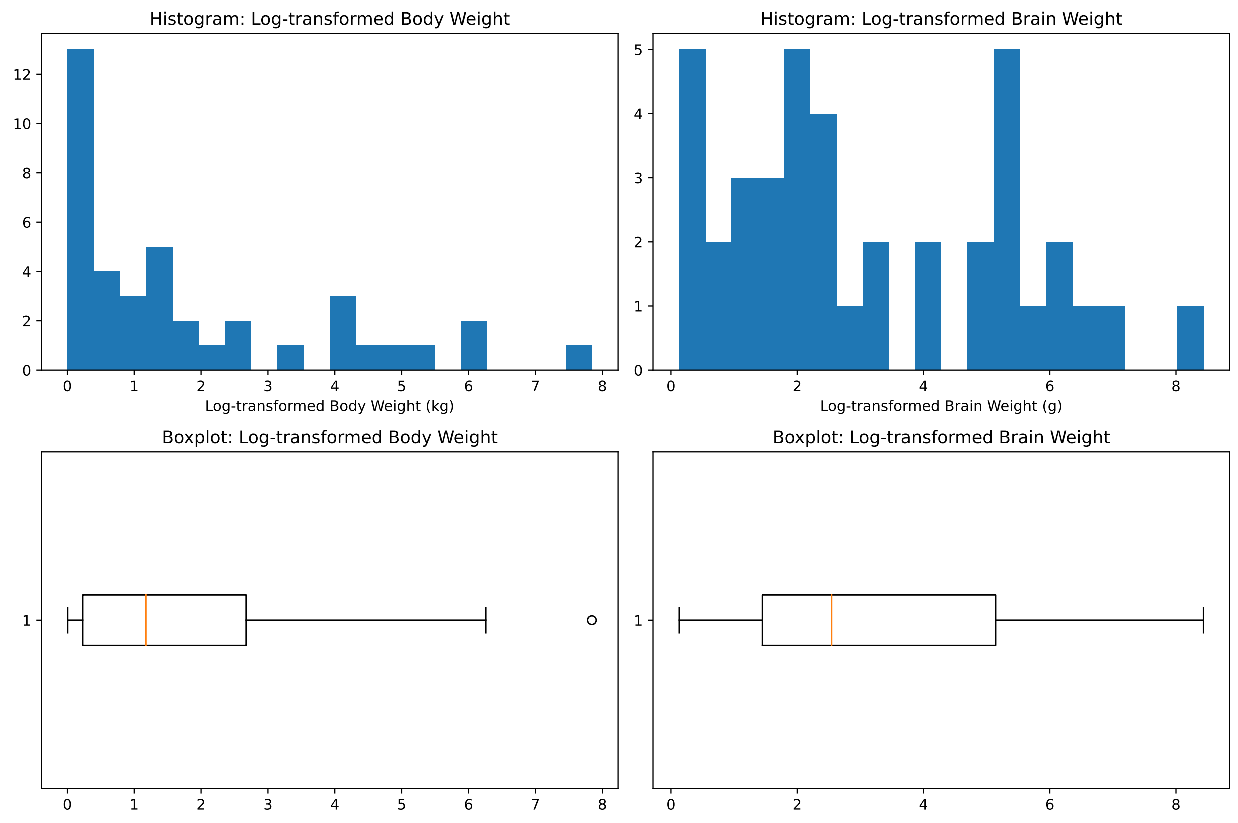
```
 # Visual inspection - histogram and horizontal boxplot of log-transformed
fig, axes = plt.subplots(2, 2, figsize=(12, 8))
axes[0, 0].hist(animals['Body Weight (kg)_log'], bins=20)
```

```
axes[0, 0].set_title('Histogram: Log-transformed Body Weight')
axes[0, 0].set_xlabel('Log-transformed Body Weight (kg)')

axes[0, 1].hist(animals['Brain Weight (g)_log'], bins=20)
```

```
axes[0, 1].set_title('Histogram: Log-transformed Brain Weight')
axes[0, 1].set_xlabel('Log-transformed Brain Weight (g)')

# Use horizontal boxplot
axes[1, 0].boxplot(animals['Body Weight (kg)_log'], vert=False)
```

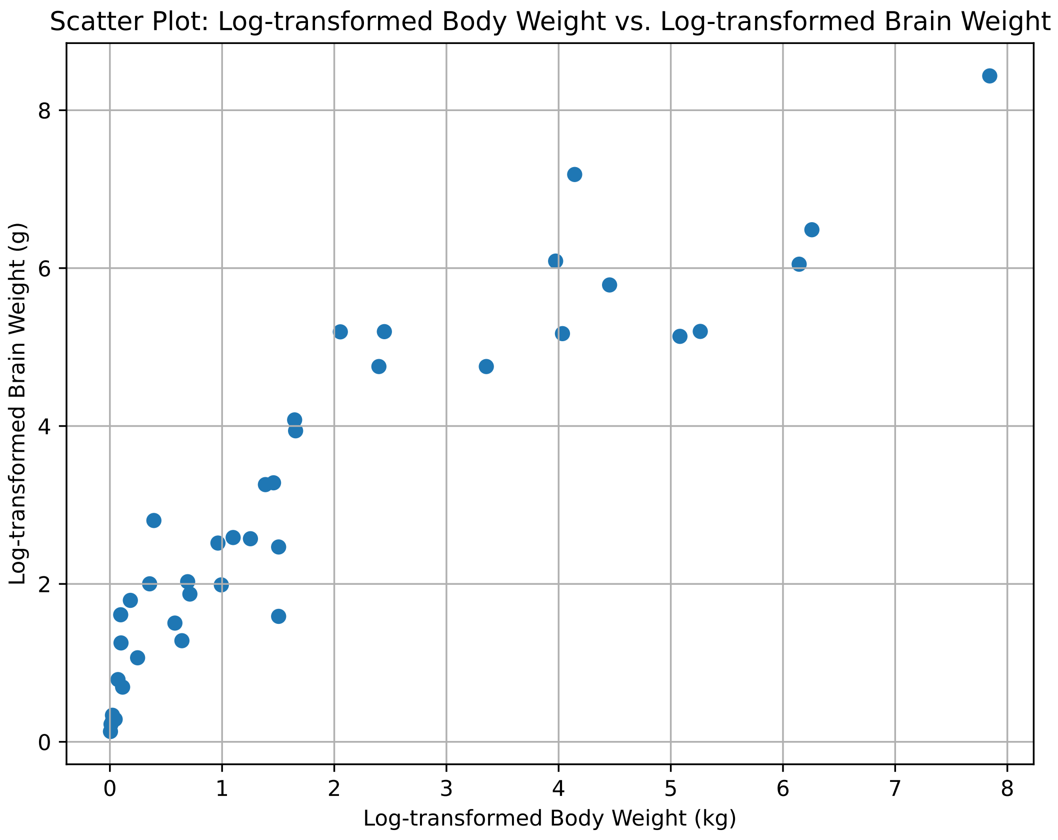
```
axes[1, 0].set_title('Boxplot: Log-transformed Body Weight')

axes[1, 1].boxplot(animals['Brain Weight (g)_log'], vert=False)
```

```
axes[1, 1].set_title('Boxplot: Log-transformed Brain Weight')

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



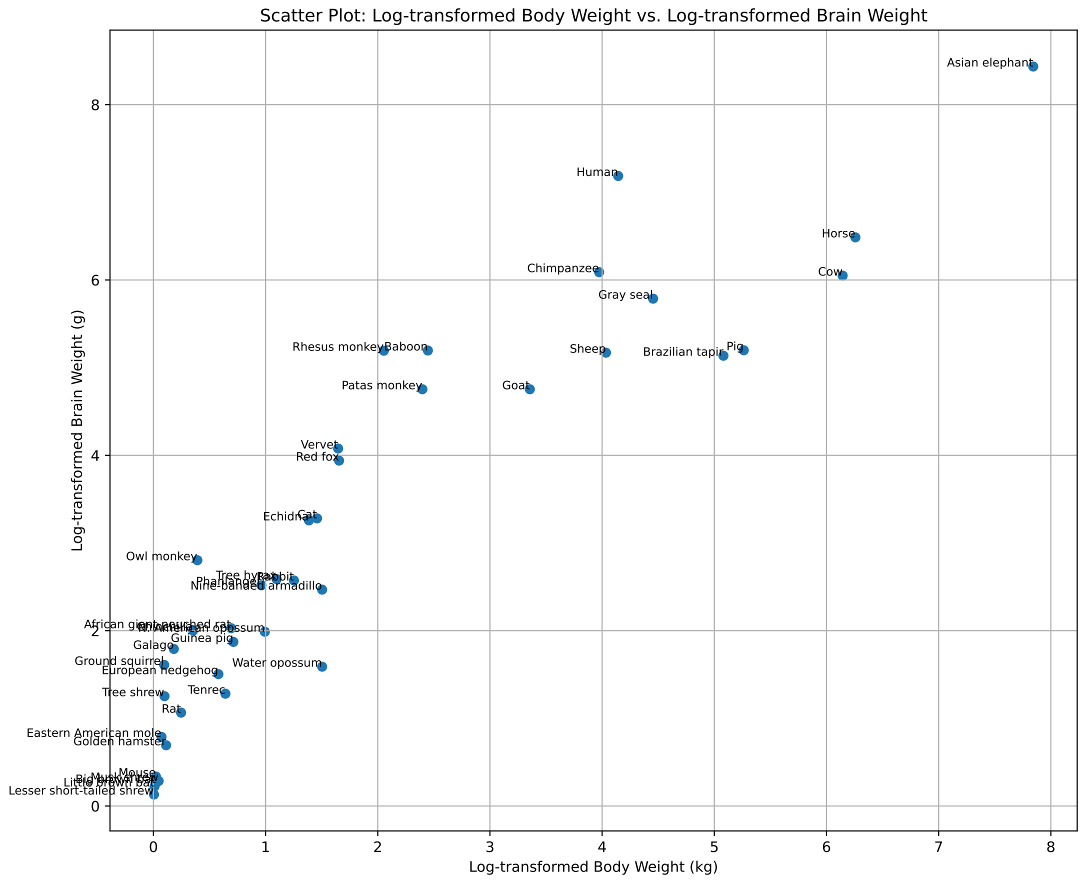
Scatter Plot

```python
# Scatter plot
plt.figure(figsize=(8, 6))
plt.scatter(animals['Body Weight (kg)_log'], animals['Brain Weight (g)_log'])
plt.title('Scatter Plot: Log-transformed Body Weight vs. Log-transformed Brain Weight')
plt.xlabel('Log-transformed Body Weight (kg)')
plt.ylabel('Log-transformed Brain Weight (g)')
plt.grid(True)
plt.show()
```



Scatter Plot with Text Labels

```python
## Scatter plot with text labels
plt.figure(figsize=(12, 10))
plt.scatter(animals['Body Weight (kg)_log'], animals['Brain Weight (g)_log'])

# Add text labels for each point
for i, animal in animals.iterrows():
    plt.text(animal['Body Weight (kg)_log'], animal['Brain Weight (g)_log'],
             animal['species'],
             fontsize=8,
             ha='right') #ha (horizontal alignment),

plt.title('Scatter Plot: Log-transformed Body Weight vs. Log-transformed Brain Weight')
plt.xlabel('Log-transformed Body Weight (kg)')
plt.ylabel('Log-transformed Brain Weight (g)')
plt.grid(True)
plt.show()
```



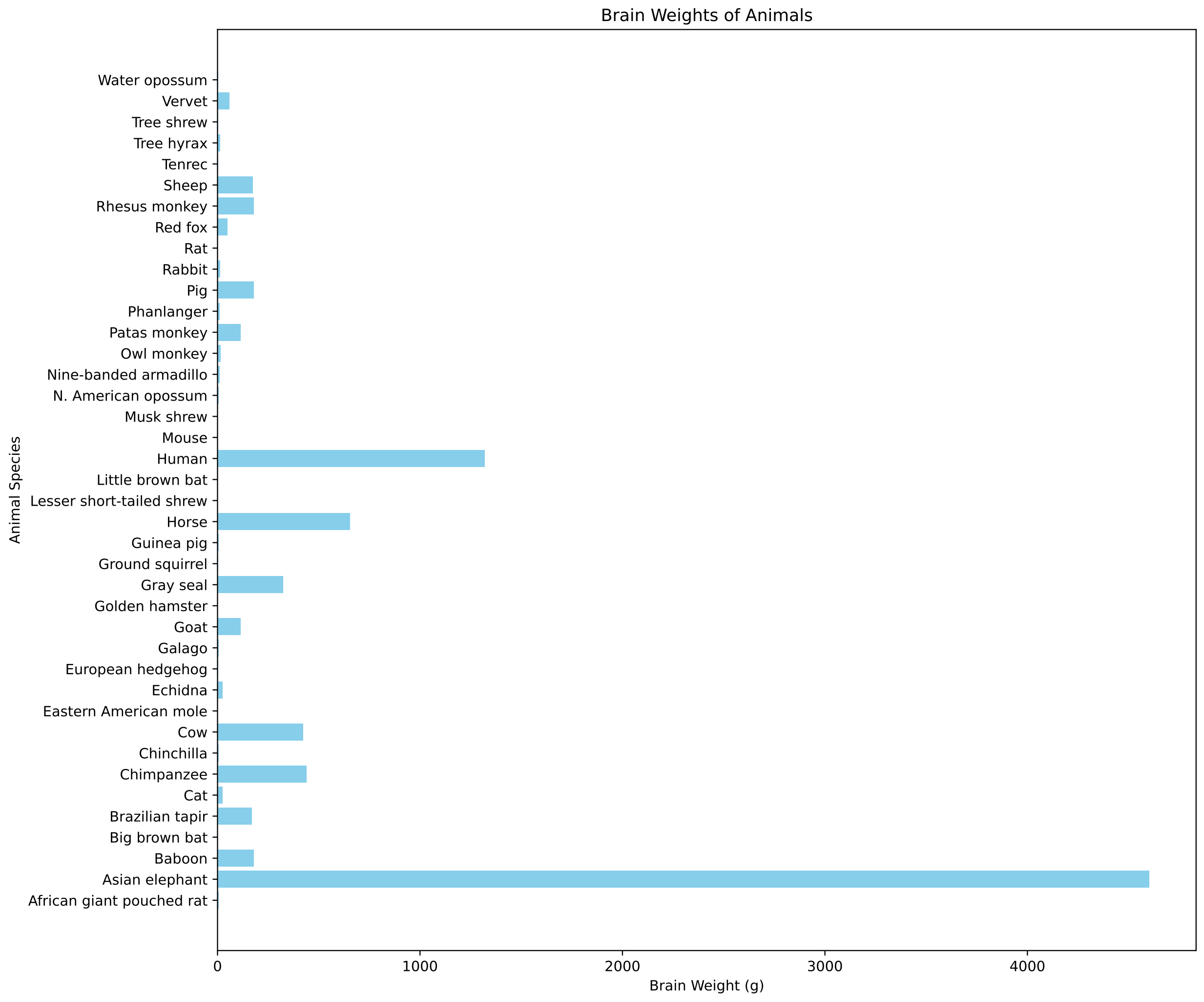
Save the Figure

```
plt.savefig('scatter_plot_with_labels.png')
```

Create a Horizontal Bar Plot

```
 # Create a horizontal bar plot for brain weights
fig, ax = plt.subplots(figsize=(12, 10))

# Bar plot
ax.barh(animals['species'], animals['Brain Weight (g)'], color='skyblue')
```

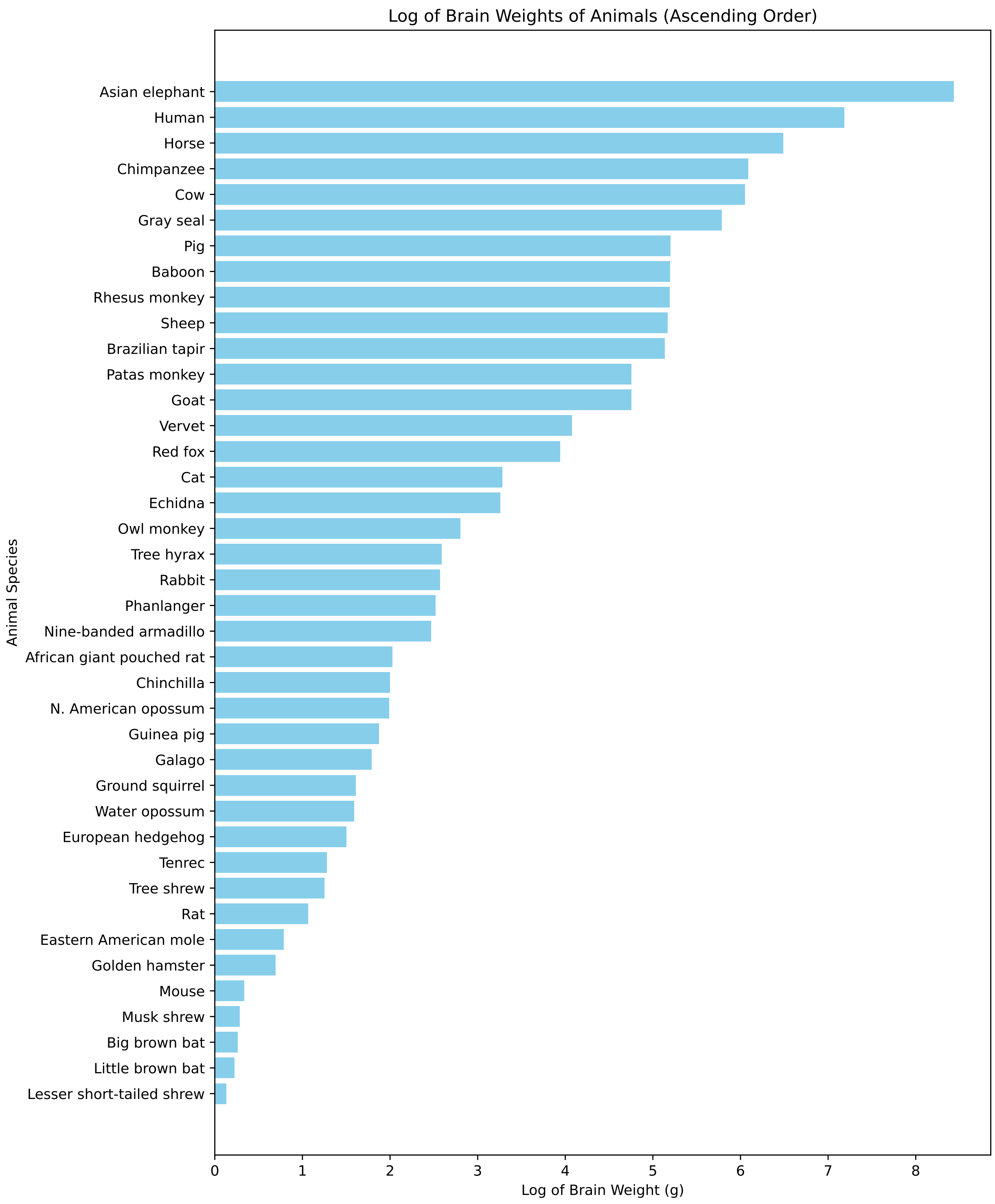
```


ax.set_title('Brain Weights of Animals')
ax.set_xlabel('Brain Weight (g)')
ax.set_ylabel('Animal Species')
plt.tight_layout()
plt.show()
```



Sort and Plot Brain Weights

```
# Sort DataFrame by Brain Weight in ascending order
animals_sorted = animals.sort_values(by='Brain Weight (g)')

# Create a horizontal bar plot for brain weights
fig, ax = plt.subplots(figsize=(10, 12))

# Bar plot
ax.barh(animals_sorted['species'], animals_sorted['Brain Weight (g)_log'],
        color='skyblue')
```

```
ax.set_title('Log of Brain Weights of Animals (Ascending Order)')
ax.set_xlabel('Log of Brain Weight (g)')
ax.set_ylabel('Animal Species')
plt.tight_layout()
plt.show()
```

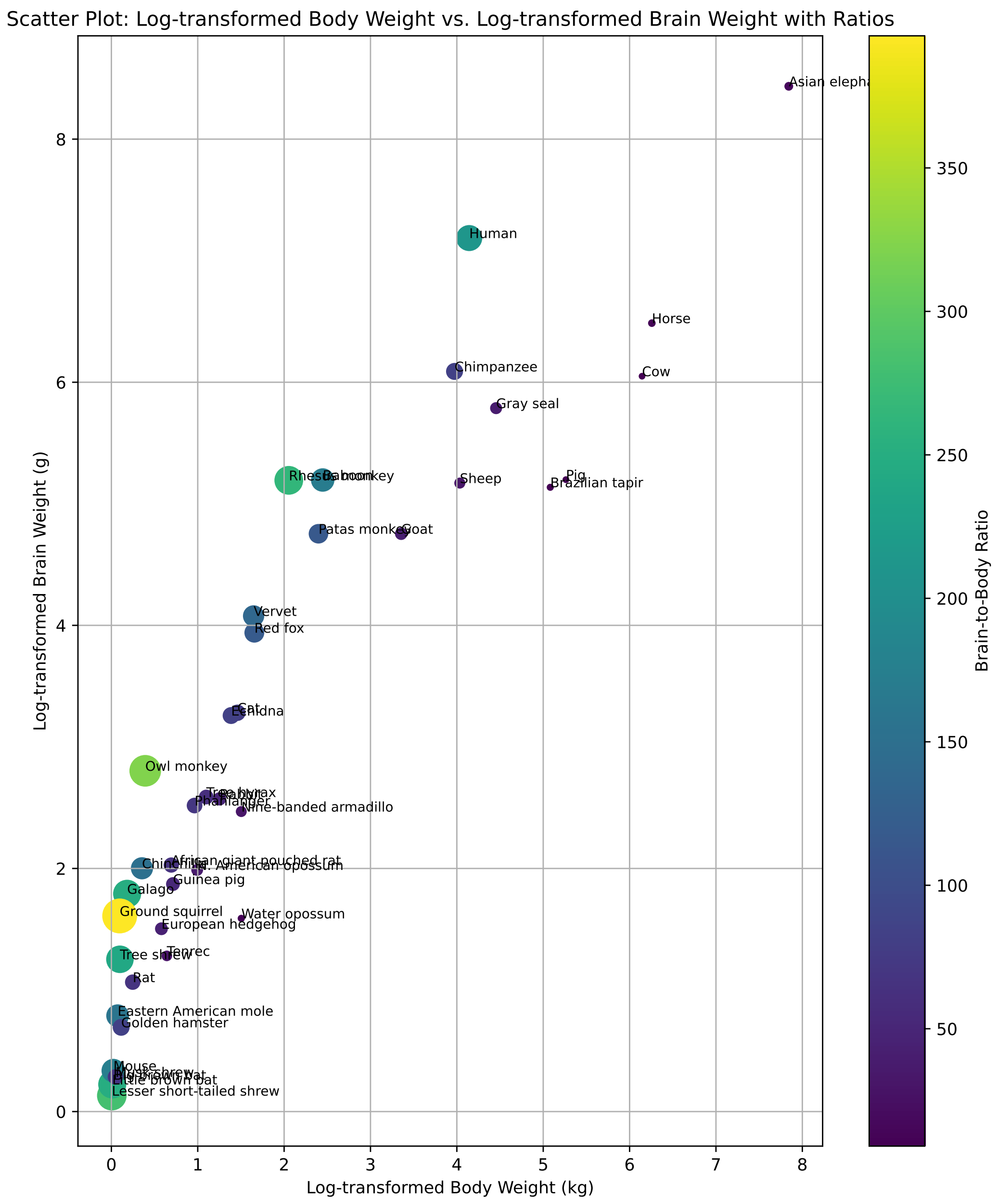

Log of Brain Weights of Animals (Ascending Order)
Asian elephant
Human
Horse
Chimpanzee
Cow
Gray seal
Pig
Baboon
Rhesus monkey
Sheep
Brazilian tapir
Patas monkey
Goat
Vervet
Red fox
Cat
Echidna
Owl monkey
Tree hyrax
Rabbit
Phanlanger
Nine-banded armadillo
African giant pouched rat
Chinchilla
N. American opossum
Guinea pig
Galago
Ground squirrel
Water opossum
European hedgehog
Tenrec
Tree shrew
Rat
Eastern American mole
Golden hamster
Mouse
Musk shrew
Big brown bat
Little brown bat
Lesser short-tailed shrew
Animal Species
0
1
2
3
4
5
6
7
8
Log of Brain Weight (g)

Calculate Brain-to-Body Weight Ratio

```
# Calculate brain-to-body weight ratio
animals['Brain-to-Body Ratio'] = animals['Brain Weight (g)'] / animals['Body Weight (kg)'] * 10
```

Scatter Plot with Ratios

```
# Scatter plot with text labels using annotate
plt.figure(figsize=(10, 12))
scatter = plt.scatter(animals['Body Weight (kg)_log'], animals['Brain Weight (g)_log'],
                      c=animals['Brain-to-Body Ratio'],
                      s = animals['Brain-to-Body Ratio'],
                      cmap='viridis')

# Add text labels for each point
for i, animal in animals.iterrows():
    plt.text(animal['Body Weight (kg)_log'], animal['Brain Weight (g)_log'],
             animal['species'],
             fontsize=8)

plt.title('Scatter Plot: Log-transformed Body Weight vs. Log-transformed Brain Weight with Ratios')
plt.xlabel('Log-transformed Body Weight (kg)')
plt.ylabel('Log-transformed Brain Weight (g)')

# Add colorbar for the ratios
cbar = plt.colorbar(scatter)
cbar.set_label('Brain-to-Body Ratio')

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

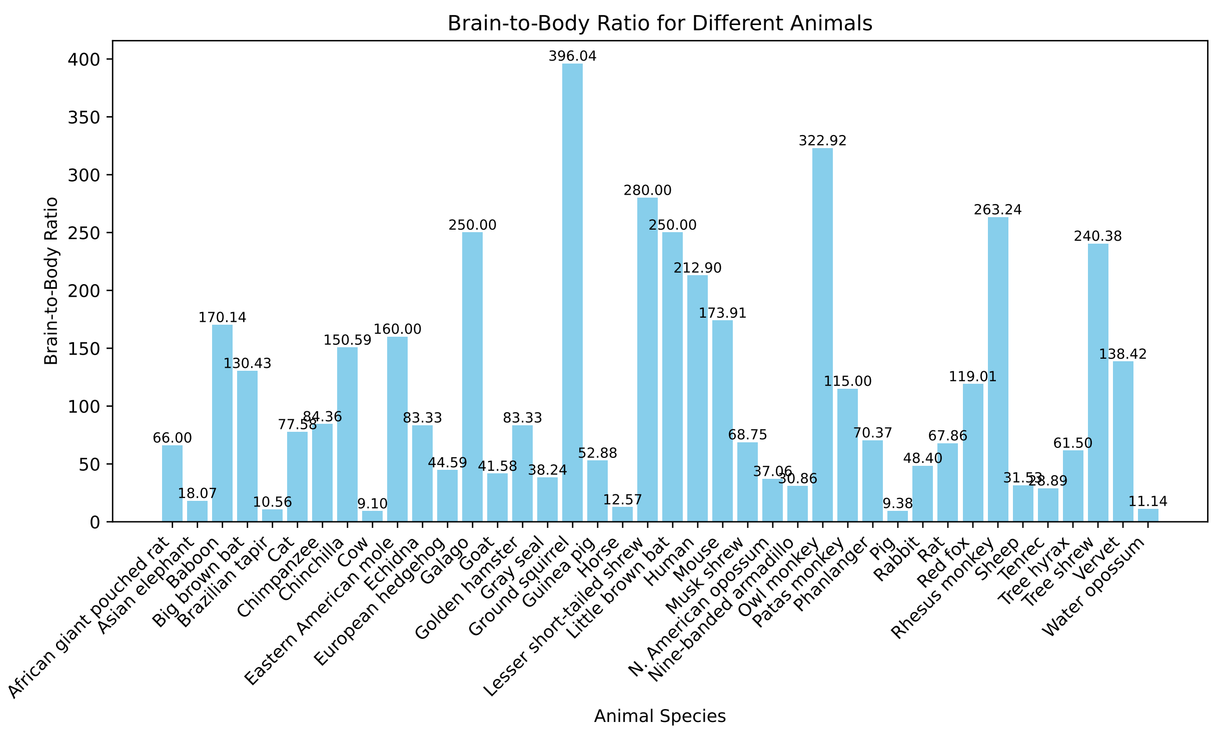


Scatter Plot: Log-transformed Body Weight vs. Log-transformed Brain Weight with Ratios
Log-transformed Body Weight (kg)
Log-transformed Brain Weight (g)
Brain-to-Body Ratio
Asian elepha
Human
Horse
Chimpanzee
Cow
Gray seal
Rhesus monkey
Baboon
Sheep
Pig
Brazilian tapir
Patas monkey
Goat
Vervet
Red fox
Cat
Echidna
Owl monkey
Tree hyrax
Rabbit
Phalanger
Nine-banded armadillo
African giant pouched rat
Chinchilla
N. American opossum
Guinea pig
Galago
Ground squirrel
Water opossum
European hedgehog
Tenrec
Tree shrew
Rat
Eastern American mole
Golden hamster
Mouse
Musk shrew
Big brown bat
Little brown bat
Lesser short-tailed shrew

Bar Plot for Brain-to-Body Ratio

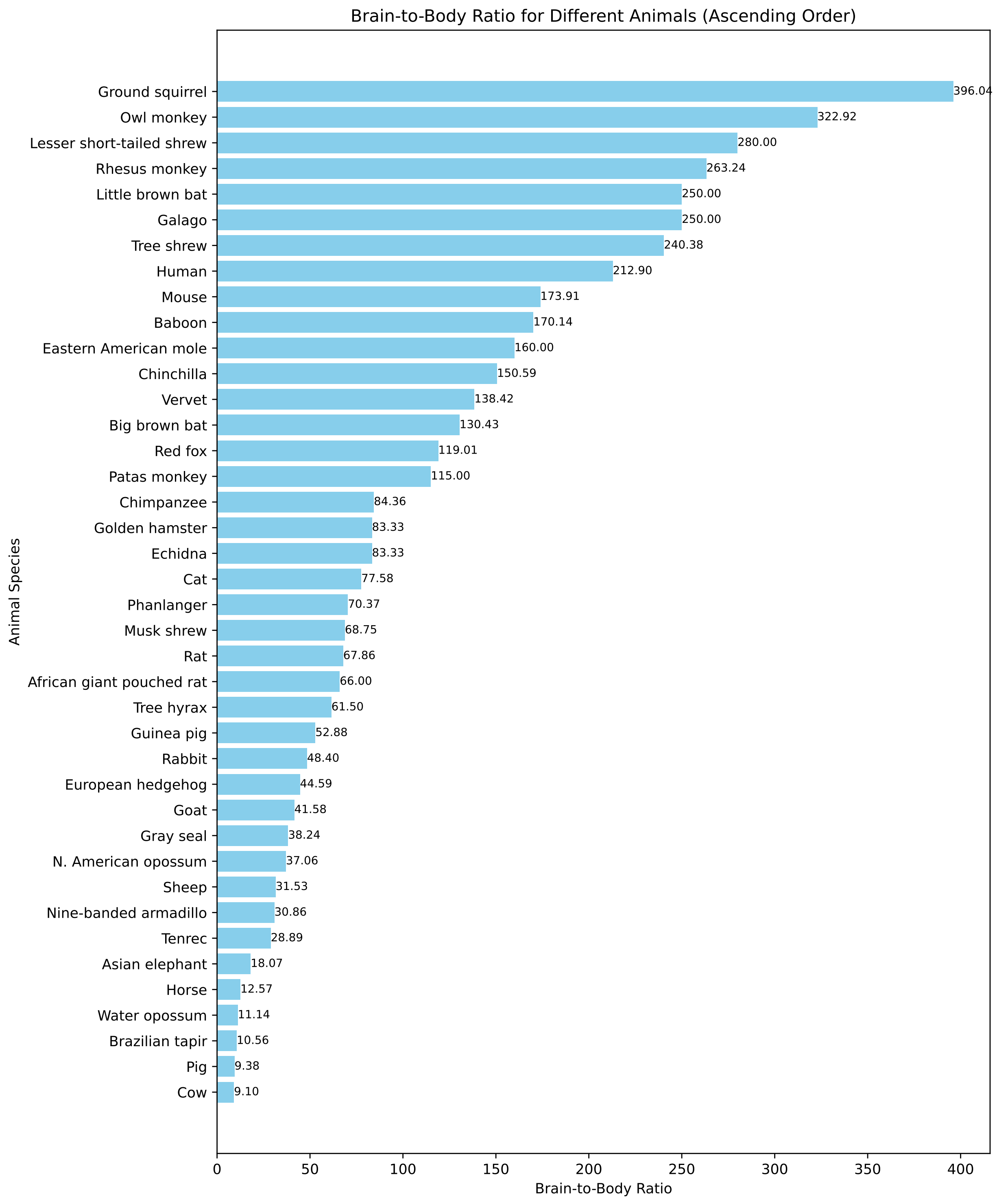
```python
# Bar plot for Brain-to-Body Ratio
plt.figure(figsize=(10, 6))
bars = plt.bar(animals['species'], animals['Brain-to-Body Ratio'], color='skyblue')

# Add text labels for each bar
for bar, ratio in zip(bars, animals['Brain-to-Body Ratio']):
    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(), f'{ratio:.2f}',
             ha='center', va='bottom', fontsize=8)

plt.title('Brain-to-Body Ratio for Different Animals')
plt.xlabel('Animal Species')
plt.ylabel('Brain-to-Body Ratio')
plt.xticks(rotation=45, ha='right')  # Rotate x-axis labels for better visibility
```

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



```python
# Calculate brain-to-body weight ratio
animals['Brain-to-Body Ratio'] = animals['Brain Weight (g)'] / animals['Body Weight (kg)'] * 10

# Sort DataFrame by Brain-to-Body Ratio in ascending order
animals_sorted = animals.sort_values(by='Brain-to-Body Ratio')

# Horizontal bar plot for Brain-to-Body Ratio
plt.figure(figsize=(10, 12))
```

```
bars = plt.barh(animals_sorted['species'], animals_sorted['Brain-to-Body Ratio'], color='skyblue')

# Add text labels for each bar
for bar, ratio in zip(bars, animals_sorted['Brain-to-Body Ratio']):
    plt.text(bar.get_width(), bar.get_y() + bar.get_height() / 2, f'{ratio:.2f}',
             ha='left', va='center', fontsize=8)

plt.title('Brain-to-Body Ratio for Different Animals (Ascending Order)')
plt.xlabel('Brain-to-Body Ratio')
plt.ylabel('Animal Species')

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



Conclusion

This concludes our exploration and visualization of the ‘SleepInMammals’ dataset using Python.