

```
In [58]: import pandas as pd
import requests
from bs4 import BeautifulSoup
```

```
In [59]: # URL to scrape
url = "https://www.thekennelclub.org.uk/search/breeds-a-to-z/"
```

```
In [60]: # Requesting the URL content
response = requests.get(url)
response_content = response.content
```

```
In [61]: # Parsing the content with BeautifulSoup
soup = BeautifulSoup(response_content, 'html.parser')
```

```
In [62]: # Extracting breed names
breed_names = soup.find_all("strong", class_="m-breed-card__title")
names = [n.text.strip() for n in breed_names]
```

```
In [63]: # Extracting breed characteristics
all_characteristics = soup.find_all("dd", class_="m-breed-summary__value")
characteristics = {
    'size': [all_characteristics[x*10].text.strip() for x in range(len(names))],
    'exercise': [all_characteristics[x*10+1].text.strip() for x in range(len(names))],
    'home_size': [all_characteristics[x*10+2].text.strip() for x in range(len(names))],
    'grooming': [all_characteristics[x*10+3].text.strip() for x in range(len(names))],
    'coat_length': [all_characteristics[x*10+4].text.strip() for x in range(len(names))],
    'sheds': [all_characteristics[x*10+5].text.strip() for x in range(len(names))],
    'lifespan': [all_characteristics[x*10+6].text.strip() for x in range(len(names))],
    'vulnerability': [all_characteristics[x*10+7].text.strip() for x in range(len(names))],
    'town_country': [all_characteristics[x*10+8].text.strip() for x in range(len(names))],
    'garden_size': [all_characteristics[x*10+9].text.strip() for x in range(len(names))]
```

```
In [ ]: all_characteristics
```

```
In [65]: # Extracting breed groups
breed_groups = soup.find_all("div", class_="m-breed-card__category")
groups = [bg.text.strip() for bg in breed_groups]
```

```
In [ ]: breed_groups
```

```
In [ ]: groups
```

```
In [68]: len(groups)
```

```
Out[68]: 223
```

```
In [69]: # Creating a DataFrame from the extracted data
dog_df = pd.DataFrame({
    'Breed Name': names,
    'Breed Group': groups,
    **characteristics
})
```

```
In [ ]: dog_df
```

```

In [71]: # Saving the DataFrame to an Excel file
dog_df.to_excel("C:/Users/Lenovo/Documents/Python Scripts/Dog_Breeds.xlsx", index=False)

In [72]: import pandas as pd
import matplotlib.pyplot as plt

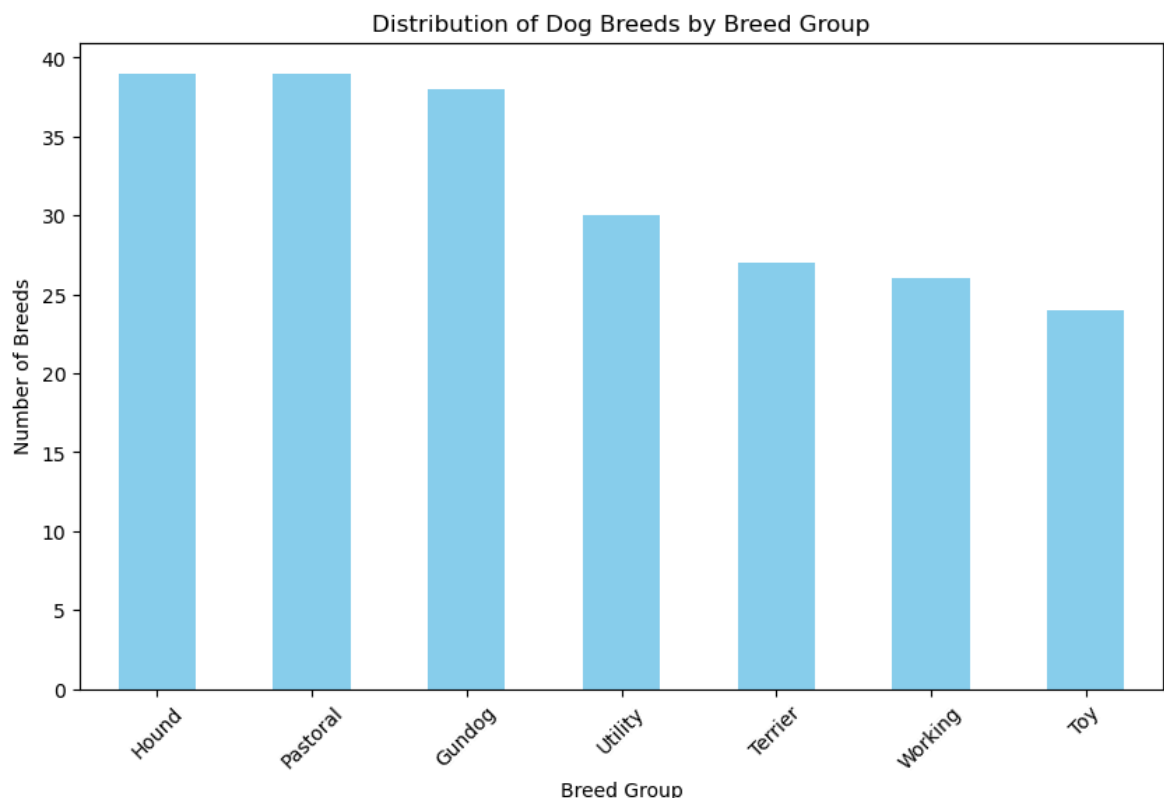
In [ ]: # Load the dataset
dog_df = pd.read_excel("C:/Users/Lenovo/Documents/Python Scripts/Dog_Breeds.xlsx")

# Display the first few rows of the dataset
print(dog_df.head())

In [74]: # Count the number of breeds in each breed group
breed_group_counts = dog_df['Breed Group'].value_counts()

# Plot the distribution
plt.figure(figsize=(10, 6))
breed_group_counts.plot(kind='bar', color='skyblue')
plt.title('Distribution of Dog Breeds by Breed Group')
plt.xlabel('Breed Group')
plt.ylabel('Number of Breeds')
plt.xticks(rotation=45)
plt.show()

```



```

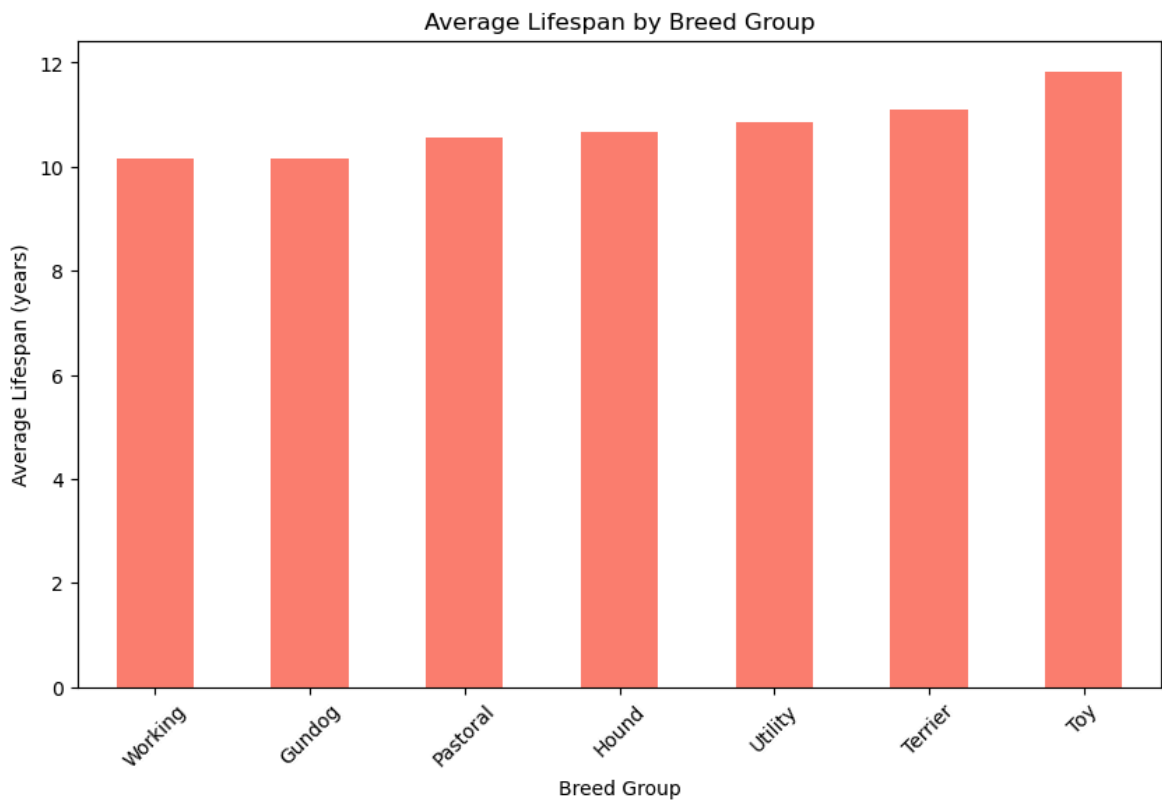
In [75]: # Convert Lifespan to numeric values (assuming the format is "X - Y years")
dog_df['lifespan'] = dog_df['lifespan'].str.extract('(\d+)').astype(int)

# Calculate the average lifespan for each breed group
avg_lifespan_by_group = dog_df.groupby('Breed Group')['lifespan'].mean().sort_values()

# Plot the average lifespan by breed group
plt.figure(figsize=(10, 6))
avg_lifespan_by_group.plot(kind='bar', color='salmon')
plt.title('Average Lifespan by Breed Group')

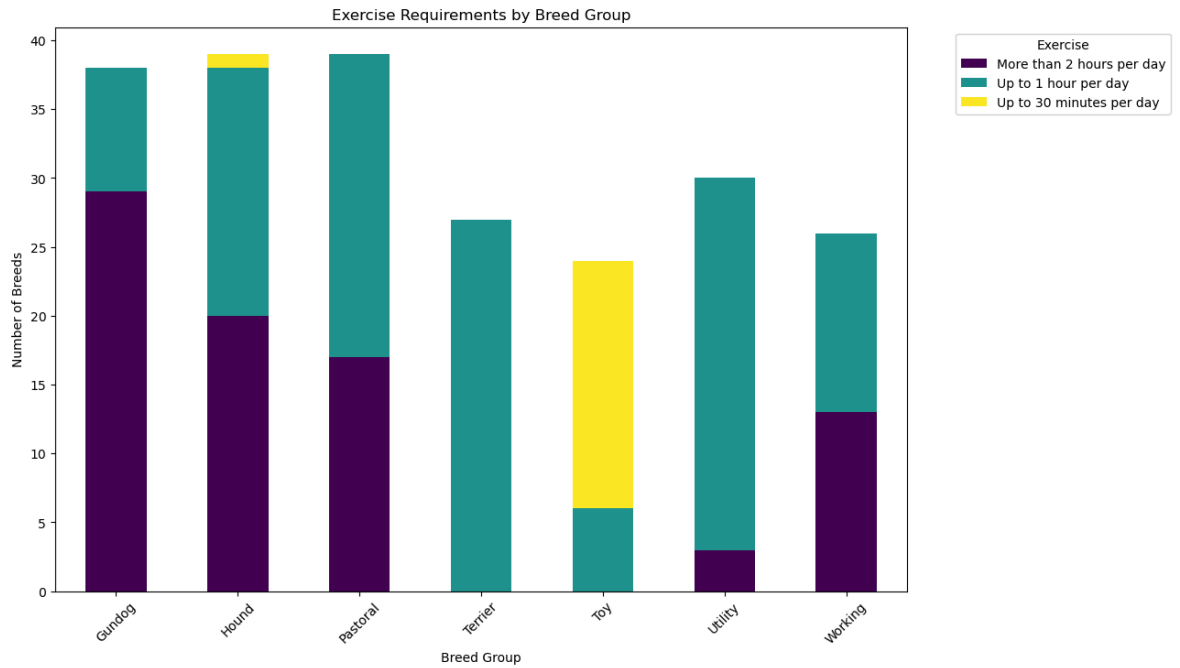
```

```
plt.xlabel('Breed Group')
plt.ylabel('Average Lifespan (years)')
plt.xticks(rotation=45)
plt.show()
```



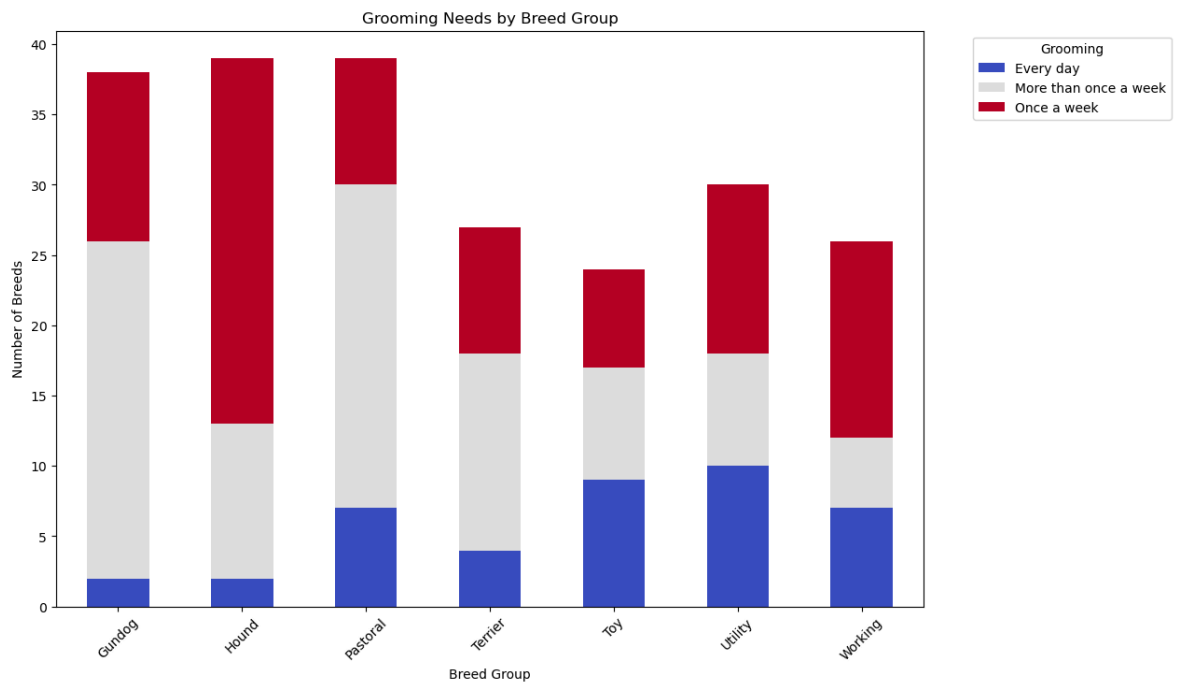
```
In [76]: # Create a crosstab of exercise requirements by breed group
exercise_crosstab = pd.crosstab(dog_df['Breed Group'], dog_df['exercise'])

# Plot the crosstab as a stacked bar chart
exercise_crosstab.plot(kind='bar', stacked=True, figsize=(12, 8), colormap='viridis')
plt.title('Exercise Requirements by Breed Group')
plt.xlabel('Breed Group')
plt.ylabel('Number of Breeds')
plt.xticks(rotation=45)
plt.legend(title='Exercise', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



```
In [77]: # Create a crosstab of grooming needs by breed group
grooming_crosstab = pd.crosstab(dog_df['Breed Group'], dog_df['grooming'])

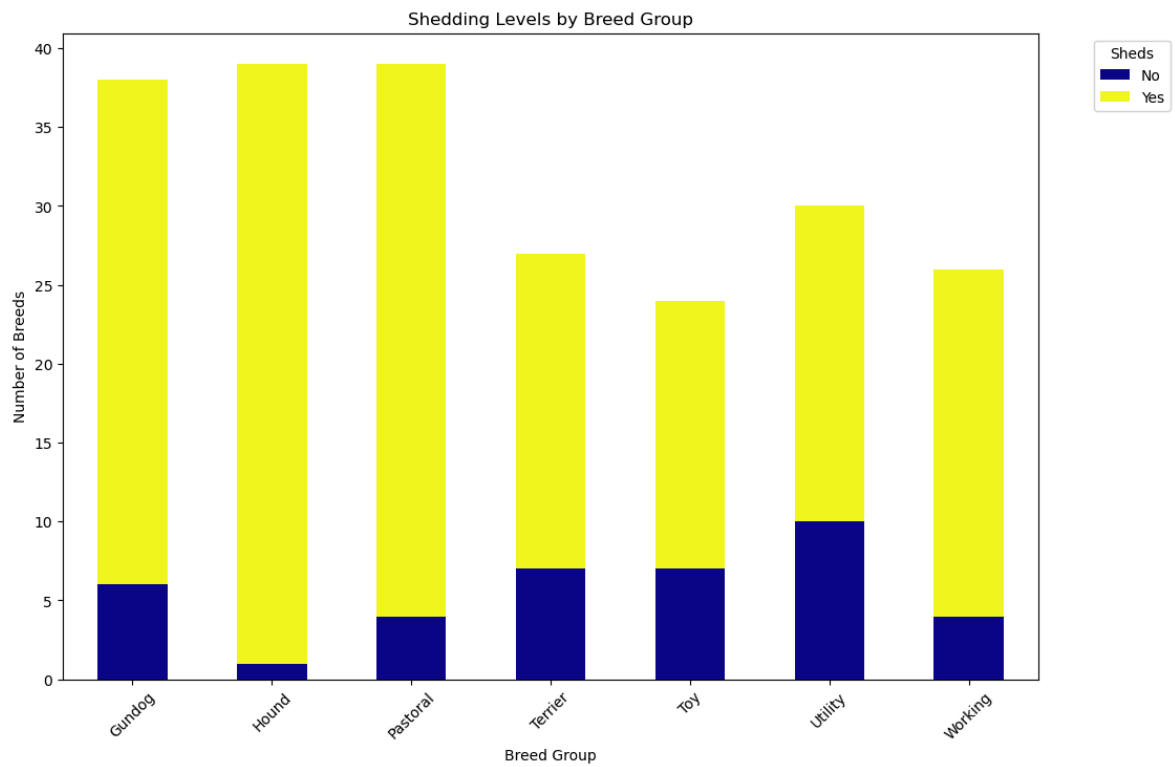
# Plot the crosstab as a stacked bar chart
grooming_crosstab.plot(kind='bar', stacked=True, figsize=(12, 8), colormap='cool')
plt.title('Grooming Needs by Breed Group')
plt.xlabel('Breed Group')
plt.ylabel('Number of Breeds')
plt.xticks(rotation=45)
plt.legend(title='Grooming', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



```
In [78]: # Create a crosstab of shedding levels by breed group
sheds_crosstab = pd.crosstab(dog_df['Breed Group'], dog_df['sheds'])

# Plot the crosstab as a stacked bar chart
sheds_crosstab.plot(kind='bar', stacked=True, figsize=(12, 8), colormap='plasma')
plt.title('Shedding Levels by Breed Group')
```

```
plt.xlabel('Breed Group')
plt.ylabel('Number of Breeds')
plt.xticks(rotation=45)
plt.legend(title='Sheds', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



In []: