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

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

# Show basic info
print("Dataset Info:")
print(df.info())
print("\nFirst 5 Rows:")
print(df.head())

# Count of each label
label_counts = df['label'].value_counts()
print("\nLabel Distribution:")
print(label_counts)

# Plot label distribution
plt.figure(figsize=(8, 5))
label_counts.plot(kind='bar', color='skyblue')
plt.title("Label Distribution")
plt.xlabel("Label")
plt.ylabel("Count")
plt.xticks(rotation=0)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

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

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

# Basic dataset overview
print("□ Dataset Info:")
print(df.info())
print("\n□ First 5 Rows:")
print(df.head())

# Check for missing values
print("\n? Missing Values:")
print(df.isnull().sum())

# Unique labels and their counts
print("\n Label Distribution:")
label_counts = df['label'].value_counts()
```

```
print(label_counts)

# Plot label distribution
plt.figure(figsize=(8, 5))
sns.countplot(data=df, x='label', palette='Set2')
plt.title('Label Distribution')
plt.xlabel('Label')
plt.ylabel('Count')
plt.grid(axis='y', linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()

# Check for duplicates
print("\n Duplicate Rows:")
duplicate_rows = df[df.duplicated()]
print(duplicate_rows)

# Check unique filenames and labels
print("\n Unique Filenames:", df['filename'].nunique())
print(" Unique Labels:", df['label'].nunique())

# Analyze filename patterns (e.g., based on subfolders or naming)
df['subfolder'] = df['filename'].apply(lambda x: x.split('/')[0])
print("\n Image Subfolder Distribution:")
print(df['subfolder'].value_counts())

# Visualize images per subfolder
plt.figure(figsize=(8, 5))
sns.countplot(data=df, x='subfolder',
order=df['subfolder'].value_counts().index, palette='coolwarm')
plt.title('Image Count by Subfolder')
plt.xlabel('Subfolder')
plt.ylabel('Count')
plt.grid(axis='y', linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()
```