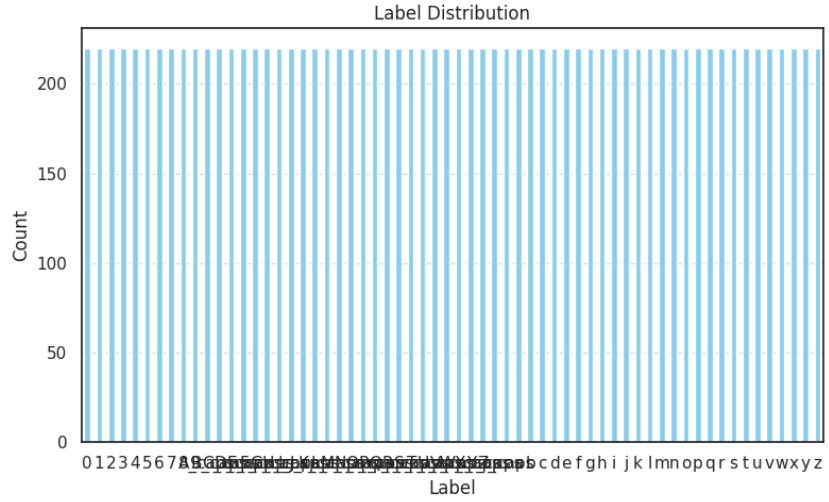
```
import pandas as pd
import matplotlib.pyplot as plt

# 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()
```

```
Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13640 entries, 0 to 13639
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   filename  13640 non-null  object
 1   label     13640 non-null  object
dtypes: object(2)
memory usage: 213.3+ KB
None

First 5 Rows:
                  filename label
0  0/0.032.1.augmented.png     0
1  0/0.053.1.augmented.png     0
2  0/0.052.4.augmented.png     0
3  0/0.049.3.augmented.png     0
4  0/0.042.3.augmented.png     0

Label Distribution:
label
0    220
1    220
2    220
3    220
4    220
    ...
v    220
w    220
x    220
y    220
z    220
Name: count, Length: 62, dtype: int64
```



Explain code image_labels.csv

1 to 10 of 13640 entries Filter

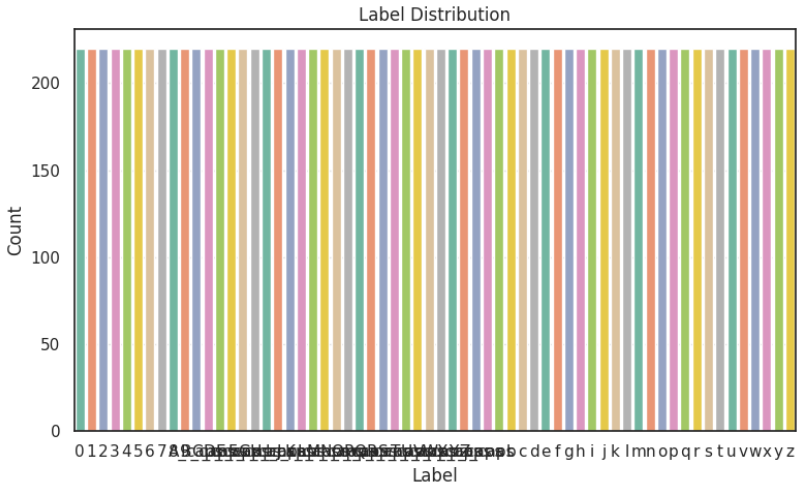
filename	label
0/0.032.1.augmented.png	0
0/0.053.1.augmented.png	0
0/0.052.4.augmented.png	0
0/0.049.3.augmented.png	0
0/0.042.3.augmented.png	0
0/0.053.2.augmented.png	0
0/0.040.5.augmented.png	0
0/0.006.1.augmented.png	0
0/0.022.2.augmented.png	0
0/0.049.2.augmented.png	0

Show 10 per page

1 2 10 100 1000 1300 1360 1364

```
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,
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()
```

```
w    220
x    220
y    220
z    220
Name: count, Length: 62, dtype: int64
<ipython-input-20-e754218d0709>:25: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in

  sns.countplot(data=df, x='label', palette='Set2')
```




```
  Duplicate Rows:
Empty DataFrame
Columns: [filename, label]
Index: []

📁 Unique Filenames: 13640
🏷️ Unique Labels: 62

📂 Image Subfolder Distribution:
subfolder
0    220
1    220
2    220
3    220
4    220
    ...
v    220
w    220
x    220
y    220
z    220
Name: count, Length: 62, dtype: int64
<ipython-input-20-e754218d0709>:49: FutureWarning:
```