**1. Distribution of Public Key Lengths**

This visualization shows the distribution of public key lengths across the dataset, which is crucial for assessing the strength of encryption.

python

Copy code

import pandas as pd

import matplotlib.pyplot as plt

# Load the dataset

file\_path = 'your\_file\_path.csv' # Replace with your actual file path

df = pd.read\_csv(file\_path)

# Plot the distribution of public key lengths

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

df['public\_key\_length'].plot(kind='hist', bins=20, color='skyblue', edgecolor='black')

plt.title('Distribution of Public Key Lengths')

plt.xlabel('Public Key Length (bits)')

plt.ylabel('Frequency')

plt.grid(True)

plt.show()

**2. Validity Period Distribution**

This visualization displays the distribution of certificate validity periods, which helps in understanding adherence to security best practices.

python

Copy code

import pandas as pd

import matplotlib.pyplot as plt

import datetime

# Load the dataset

file\_path = 'your\_file\_path.csv' # Replace with your actual file path

df = pd.read\_csv(file\_path)

# Convert validity\_start and validity\_end to datetime

df['validity\_start'] = pd.to\_datetime(df['validity\_start'], format='%d-%b-%y')

df['validity\_end'] = pd.to\_datetime(df['validity\_end'], format='%d-%b-%y')

# Calculate the validity period in days

df['validity\_period'] = (df['validity\_end'] - df['validity\_start']).dt.days

# Plot the distribution of validity periods

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

df['validity\_period'].plot(kind='hist', bins=20, color='green', edgecolor='black')

plt.title('Distribution of Certificate Validity Periods')

plt.xlabel('Validity Period (days)')

plt.ylabel('Frequency')

plt.grid(True)

plt.show()

To see how many websites use each TLS version directly, you can modify the code slightly to print out the counts before plotting them. Here's an updated version of the code that includes this step:

python

Copy code

import pandas as pd

import matplotlib.pyplot as plt

# Load the dataset from the file

file\_path = 'your\_file\_path.csv' # Replace with your actual file path

df = pd.read\_csv(file\_path)

# Extract the TLS version

def extract\_tls\_version(encryption\_str):

if isinstance(encryption\_str, str):

if 'TLSv1.3' in encryption\_str:

return 'TLS 1.3'

elif 'TLSv1.2' in encryption\_str:

return 'TLS 1.2'

else:

return 'Other'

return 'Unknown'

# Apply the function to create a new column

df['tls\_version'] = df['encryption'].apply(extract\_tls\_version)

# Count the number of websites using each TLS version

tls\_counts = df['tls\_version'].value\_counts()

# Print the counts

print("Counts of websites using each TLS version:")

print(tls\_counts)

# Visualize the counts of websites using TLSv1.3 and TLSv1.2

# Filter only for TLS 1.3 and TLS 1.2

tls\_counts\_filtered = tls\_counts.loc[['TLS 1.3', 'TLS 1.2']]

# Create a bar chart

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

tls\_counts\_filtered.plot(kind='bar', color=['blue', 'orange'])

plt.title('Number of Websites Using TLSv1.3 and TLSv1.2')

plt.xlabel('TLS Version')

plt.ylabel('Number of Websites')

plt.xticks(rotation=0)

plt.show()  
  
  
  
  
**Updated Code for Pie Chart with Top 15 Issuers:**

python

Copy code

import pandas as pd

import matplotlib.pyplot as plt

# Load the dataset directly from the CSV

df = pd.read\_csv('your\_file\_path.csv') # Replace with your actual file path

# Count the number of certificates issued by each issuer

issuer\_counts = df['issuer'].value\_counts()

# Limit to the top 15 issuers and group the rest into 'Other'

top\_issuers = issuer\_counts.head(15)

other\_issuers = pd.Series(issuer\_counts[15:].sum(), index=['Other'])

issuer\_counts\_limited = pd.concat([top\_issuers, other\_issuers])

# Plot the pie chart for issuer market distribution

plt.figure(figsize=(12, 10)) # Increase the figure size to accommodate long names

issuer\_counts\_limited.plot(kind='pie', autopct='%1.1f%%', startangle=140, colors=plt.cm.Paired.colors)

# Adjust title and layout

plt.title('Market Distribution of Top 15 Certificate Issuers', pad=20)

plt.ylabel('') # Remove the default y-label to clean up the chart

# Prevent labels from getting cut off

plt.tight\_layout()

# Show the plot

plt.show()

**Explanation:**

1. **Top 15 Issuers**: The code now selects the top 15 issuers and groups the rest into an "Other" category.
2. **Figure Size**: The figure size is increased to 12x10 to provide more space for longer names and ensure they are fully visible.
3. **Tight Layout**: plt.tight\_layout() is added to prevent labels from getting cut off by automatically adjusting the spacing.

**Summary:**

This code should produce a more readable pie chart with the top 15 certificate issuers. The larger figure size and tight layout adjustments help ensure that the issuer names are fully displayed and not cut off.

**Cipher Suite Trends**import pandas as pd  
import matplotlib.pyplot as plt  
  
# Load the original dataset  
df = pd.read\_csv('domains\_without\_errors.csv')  
  
# Function to extract encryption details (same as before)  
def extract\_encryption\_details(encryption\_str):  
 if isinstance(encryption\_str, str):  
 encryption\_parts = encryption\_str.replace('(', '').replace(')', '').split(', ')  
 cipher\_suite = encryption\_parts[0]  
 return cipher\_suite  
 return None  
  
# Apply the function to create a new 'cipher\_suite' column  
df['cipher\_suite'] = df['encryption'].apply(extract\_encryption\_details)  
  
# Identify the top 10 categories by frequency  
top\_categories = df['category'].value\_counts().head(10).index  
  
# Filter the dataset to include only the top 10 categories  
df\_top\_categories = df[df['category'].isin(top\_categories)]  
  
# Group by category and cipher\_suite and count occurrences  
cipher\_suite\_by\_category = df\_top\_categories.groupby(['category', 'cipher\_suite']).size().unstack(fill\_value=0)  
  
# Plotting the cipher suite trends by category as a stacked bar chart with subtle colors  
plt.figure(figsize=(14, 8))  
cipher\_suite\_by\_category.plot(kind='bar', stacked=True, figsize=(14, 8), colormap='Set3')  
plt.title('Cipher Suite Trends by Top 10 Website Categories')  
plt.xlabel('Website Category')  
plt.ylabel('Number of Websites')  
plt.xticks(rotation=45, ha='right')  
plt.legend(title='Cipher Suite', bbox\_to\_anchor=(1.05, 1), loc='upper left')  
plt.tight\_layout()  
plt.show()