import pandas as pd

main\_data = pd.read\_csv(r'C:\Users\Mohamed\Downloads\result data\updated\_final\_dataset\_for\_Project2')

output\_data = pd.read\_csv(r'C:\Users\Mohamed\Downloads\result data\scan\_results\_quick (1).csv')

merged\_df = pd.merge(output\_data, main\_data, on=['Port', 'Protocol'], how='inner')

merged\_df  
  
  
  
  
  
import seaborn as sns

import matplotlib.pyplot as plt

# Load data

main\_data = pd.read\_csv(r'C:\Users\Mohamed\Downloads\result data\updated\_final\_dataset\_for\_Project2')

output\_data = pd.read\_csv(r'C:\Users\Mohamed\Downloads\result data\scan\_results\_quick (1).csv')

# Merge datasets

merged\_df = pd.merge(output\_data, main\_data, on=['Port', 'Protocol'], how='inner')

merged\_df

# Function to plot risk distribution without "Other"

def plot\_risk\_distribution(merged\_df, ax):

category\_counts = merged\_df['Risk'].value\_counts()

# Plot without "Other"

category\_counts.plot(kind='pie', autopct='%1.1f%%', startangle=90, colors=['#98FF98', '#FFDAB9','#FF6F61','#4DB8B1'], ax=ax)

ax.set\_title('Port Risks')

ax.set\_ylabel('')

# Function to plot risk levels

def plot\_risk\_levels(merged\_df, ax):

risk\_counts = merged\_df['Risk'].value\_counts()

risk\_counts.plot(kind='bar', color=['red', 'orange', 'yellow', 'green', 'blue', 'purple'], ax=ax)

ax.set\_title('Distribution of Risk Levels', fontsize=16)

ax.set\_xlabel('Risk Levels', fontsize=14)

ax.set\_ylabel('Number of Ports', fontsize=14)

ax.set\_xticklabels(ax.get\_xticklabels(), rotation=45)

ax.grid(axis='y', linestyle='--', alpha=0.7)

# Function to plot transport protocol

def plot\_transport\_protocol(merged\_df, ax):

risk\_counts = merged\_df['Protocol'].value\_counts()

risk\_counts.plot(kind='bar', color=['blue', 'purple','yellow', 'green'], ax=ax)

ax.set\_title('Transport Protocol', fontsize=16)

ax.set\_xlabel('Transport Protocol', fontsize=14)

ax.set\_ylabel('Number of Ports', fontsize=14)

ax.set\_xticklabels(ax.get\_xticklabels(), rotation=45)

ax.grid(axis='y', linestyle='--', alpha=0.7)

# Function to plot risk by protocol

def plot\_risk\_by\_protocol(merged\_df, ax):

grouped\_data = merged\_df.groupby(['Protocol', 'Risk']).size().unstack(fill\_value=0)

grouped\_data.plot(kind='bar', figsize=(12, 8), colormap='viridis', ax=ax)

ax.set\_title('Risk Levels by Transport Protocol', fontsize=16)

ax.set\_xlabel('Protocol', fontsize=14)

ax.set\_ylabel('Count', fontsize=14)

ax.legend(title='Risk Level', fontsize=12)

ax.grid(axis='y', linestyle='--', alpha=0.7)

# Function to plot DDoS risk distribution without "Other"

def plot\_ddos\_risk(merged\_df, ax):

category\_counts = merged\_df['DDoS Risk'].value\_counts()

# Plot without "Other"

category\_counts.plot(kind='pie', autopct='%1.1f%%', startangle=90, colors=['#4DB8B1', '#FF6F61'], ax=ax)

ax.set\_title('DDoS Risk')

ax.set\_ylabel('')

# Function to plot scatter plot for Port vs Protocol

def plot\_scatter\_port\_vs\_protocol(merged\_df, ax):

sns.scatterplot(x='Port', y='Protocol', hue='DDoS Risk', data=merged\_df, palette='Set1', s=100, edgecolor='w', alpha=0.7, ax=ax)

ax.set\_title('Scatter Plot: DDoS Risk vs Port and Protocol', fontsize=16)

ax.set\_xlabel('Port', fontsize=14)

ax.set\_ylabel('Protocol', fontsize=14)

# Create the plots

fig, axs = plt.subplots(2, 3, figsize=(18, 12))

plot\_risk\_distribution(merged\_df, axs[0, 0])

plot\_risk\_levels(merged\_df, axs[0, 1])

plot\_transport\_protocol(merged\_df, axs[0, 2])

plot\_risk\_by\_protocol(merged\_df, axs[1, 0])

plot\_ddos\_risk(merged\_df, axs[1, 1])

plot\_scatter\_port\_vs\_protocol(merged\_df, axs[1, 2])

plt.tight\_layout()

plt.show()