Report on Data Visualization using Python and “matplotlib.pyplot”

## Introduction:

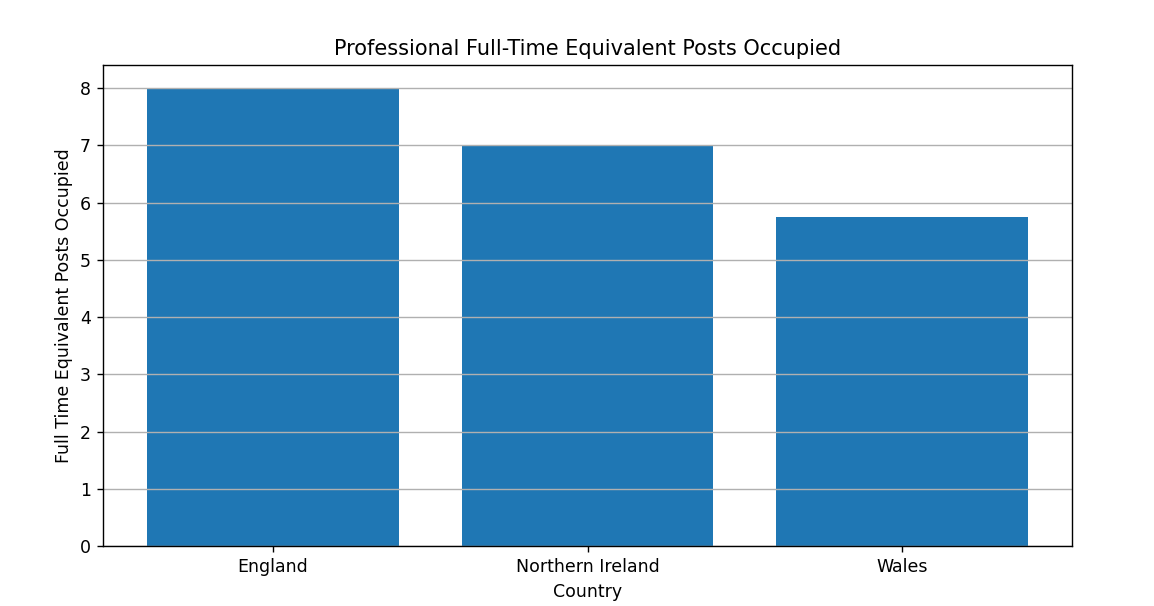
In this report, we present a data visualization analysis using Python and the “matplotlib.pyplot” library. The dataset used contains information related to establishments, interventions, and professional full-time equivalent posts across different countries sourced from World Bank Open Data.

## Code and Analysis:

The Python code provided reads the dataset into a Pandas DataFrame, and then it generates three different types of visualizations:

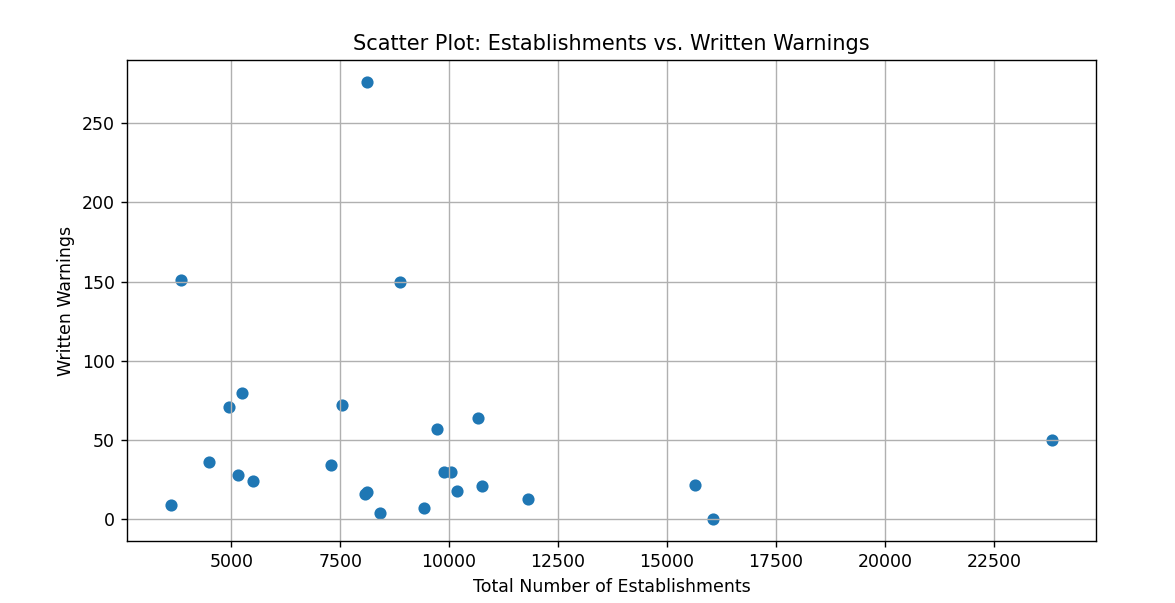
### 1. Scatter Plot:

The link between the overall number of establishments and the total number of establishments subject to written warnings was investigated using a scatter plot. Any possible correlations or patterns between these two variables can be found with the use of this visualization. To make the plot easier to read, grid lines were added and the title and axes were labeled for clarity.



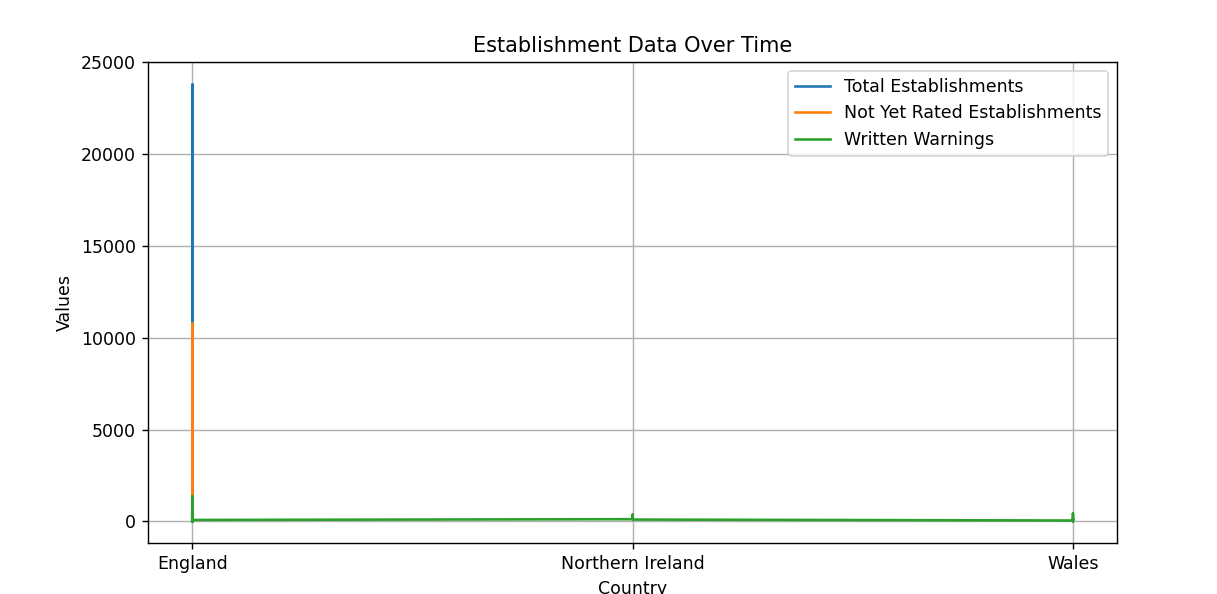
### 2. Bar Chart:

To show how many professional full-time equivalent positions are held by each nation, a bar chart was created. The workforces of each nation are clearly compared in this graphic. The plot has grid lines for readability, labeled axes, and a title.



### 3. Line Plot with Multiple Lines:

To see the evolution of three important variables over time, the total number of establishments (including those that haven't been evaluated for intervention), the number of establishments that haven't been rated for intervention, and the total number of establishments that have received written warnings, a line plot was made. We can see how these characteristics vary between countries and over time with this plot. For clarification, bespoke labels and a legend were added.



### Conclusion:

The code and visuals offer insightful information about the dataset. The line plot helps to identify countries with variances in these elements by illuminating changes in important establishment-related variables throughout time. The association between total establishments and those covered by written warnings is represented visually by the scatter plot, which may point problematic regions that require more research. A clear comparison of the workforces in various nations is shown in the bar chart.

Understanding complicated datasets requires data visualization, and Python's ‘matplotlib.pyplot’ module is an effective tool for producing a wide range of visualizations.

Data analysts and stakeholders can gain a better understanding of the dataset and its consequences by using these visualizations as a first step towards further in-depth data analysis and decision-making.

### Data Source:

<https://fsadata.github.io/local-authority-enforcement-monitoring-system-laems-data/data/2019-20-enforcement-data-food-standards.csv>

## Appendix:

import matplotlib.pyplot as plt

import pandas as pd

# Create a DataFrame from the data

df = pd.read\_csv("2019-20-enforcement-data-food-standards.csv")

# 1. Line Plot with Multiple Lines

def create\_line\_plot(dataframe, x\_col, y\_cols, labels, title):

plt.figure(figsize=(10, 5))

for i, y\_col in enumerate(y\_cols):

plt.plot(dataframe[x\_col], dataframe[y\_col], label=labels[i])

plt.xlabel(x\_col)

plt.ylabel('Values')

plt.title(title)

plt.legend()

plt.grid(True)

plt.show()

# Create a line plot with multiple lines

create\_line\_plot(df, 'Country', ['Totalnumberofestablishments(includingnotyetratedandoutside)(1)',

'Numberofestablishmentsnotyetratedforintervention(1)',

'TotalnumberofestablishmentssubjecttoWrittenwarnings'],

['Total Establishments', 'Not Yet Rated Establishments', 'Written Warnings'],

'Establishment Data Over Time')

# 2. Scatter Plot

def create\_scatter\_plot(dataframe, x\_col, y\_col, labels, title):

plt.figure(figsize=(10, 5))

plt.scatter(dataframe[x\_col], dataframe[y\_col])

plt.xlabel(labels[0])

plt.ylabel(labels[1])

plt.title(title)

plt.grid(True)

plt.show()

# Create a scatter plot

create\_scatter\_plot(df, 'Totalnumberofestablishments(includingnotyetratedandoutside)(1)',

'TotalnumberofestablishmentssubjecttoWrittenwarnings',

['Total Number of Establishments','Written Warnings'],

'Scatter Plot: Establishments vs. Written Warnings')

# 3. Bar Chart

def create\_bar\_chart(dataframe, x\_col, y\_col,labels, title):

plt.figure(figsize=(10, 5))

plt.bar(dataframe[x\_col], dataframe[y\_col])

plt.xlabel(labels[0])

plt.ylabel(labels[1])

plt.title(title)

plt.grid(axis='y')

plt.show()

# Create a bar chart

create\_bar\_chart(df, 'Country', 'ProfessionalFullTimeEquivalentPosts-occupied \*',

['Country', 'Full Time Equivalent Posts Occupied'],

'Professional Full-Time Equivalent Posts Occupied')

# Show the visualizations

plt.show()