**Name: Umer Khalid Reg No: 2022605**

**Analysis of Traffic Accidents and Petrol Prices**

**Introduction:**

This report presents an analysis of traffic accidents and petrol prices, focusing on data sourced from two separate CSV files. The first dataset contains information on traffic accidents in Pakistan, while the second dataset includes petrol prices worldwide. The analysis aims to provide insights into the frequency of traffic accidents in different provinces of Pakistan and compare petrol prices across various countries, as well as comparing petrol prices in different currencies, specifically USD and PKR (Pakistan Rupee).

**1. Analysis of Traffic Accidents in Pakistan:**

The analysis of traffic accidents in Pakistan involved processing a dataset containing information on accidents in different provinces. Initially, the dataset was loaded into a Pandas DataFrame, and unnecessary rows were skipped to ensure data integrity. The dataset was then filtered to exclude entries not related to specific provinces, and the total number of accidents, fatalities, and injuries were calculated for each province. Additionally, the analysis identified the provinces with the highest and lowest numbers of accidents, fatalities, and injuries, providing valuable insights into traffic accident trends across Pakistan.

**Code:**

*#import library*

import pandas as pd

print('2022605')

*# Define column names because name of first column was missing*

column\_names = ['Location', 'Year', 'Total number of accidents', 'Fatal accidents', 'Non-fatal accidents', 'Killed', 'Injured', 'Vehicles involved']

*# Read the CSV file, skipping the first 3 rows because they were unnecessary and creating problems, and using the defined column names*

df = pd.read\_csv(r'D:/Semester 4/TDS/Pakistani\_Traffic\_Accidents (1).csv', skiprows=3, names=column\_names)

*# Filtering out rows not containing "pakistan" in the specified column 'Location' because it was not needed in province trends*

filtered\_data = df[~df['Location'].str.contains("pakistan", case=False)]

*# Grouping by 'Location' and calculating the sum of accidents*

grouped\_data = filtered\_data.groupby('Location')['Total number of accidents'].sum()

*# Sorting the grouped DataFrame of 'Total number of accidents' in descending order*

sorted\_data = grouped\_data.sort\_values(ascending=False)

*#Getting only top row*

top\_rows = sorted\_data.head(1)

print('Province with most accidents is')

print(top\_rows)

*# Sorting the grouped DataFrame of 'Total number of accidents' in ascending order*

sorted\_data1 = grouped\_data.sort\_values(ascending=True)

*#Getting only top row*

top\_rows1 = sorted\_data1.head(1)

print('Province with least accidents is')

print(top\_rows1)

*#Grouping by 'Location' and calculating total number of kills*

grouped\_data1 = filtered\_data.groupby('Location')['Killed'].sum()

*# Sorting the grouped DataFrame of number of kills in descending order*

sorted\_data2 = grouped\_data1.sort\_values(ascending=False)

*#Getting only top row*

top\_rows2 = sorted\_data2.head(1)

print('Province with most kills is')

print(top\_rows2)

*# Sorting the grouped DataFrame of number of kills in ascending order*

sorted\_data3 = grouped\_data1.sort\_values(ascending=True)

*#Getting only top row*

top\_rows3 = sorted\_data3.head(1)

print('Province with least kills is')

print(top\_rows3)

*#Grouping by 'Location' and calculating total number of injuries*

grouped\_data2 = filtered\_data.groupby('Location')['Injured'].sum()

*# Sorting the grouped DataFrame of number of injuries in descending order*

sorted\_data4 = grouped\_data2.sort\_values(ascending=False)

*#Getting only top row*

top\_rows4 = sorted\_data4.head(1)

print('Province with most injuries is')

print(top\_rows4)

*# Sorting the grouped DataFrame of number of injuries in ascending order*

sorted\_data5 = grouped\_data2.sort\_values(ascending=True)

*#Getting only top row*

top\_rows5 = sorted\_data5.head(1)

print('Province with least injuries is')

print(top\_rows5)

*# Filtering out rows containing "pakistan" in the specified column 'Location' because only it was needed in this trends*

filtered\_data1 = df[df['Location'].str.contains("pakistan", case=False, regex=False)]

*#Grouping by 'Year' and calculating total number of accidents*

grouped\_data3 = filtered\_data1.groupby('Year')['Total number of accidents'].sum()

*# Sorting the grouped DataFrame of 'Total number of accidents' in descending order*

sorted\_data6 = grouped\_data3.sort\_values(ascending=False)

*#Getting only top row*

top\_rows6 = sorted\_data6.head(1)

print('Year with most accidents is')

print(top\_rows6)

*# Sorting the grouped DataFrame of 'Total number of accidents' in ascending order*

sorted\_data7 = grouped\_data3.sort\_values(ascending=True)

*#Getting only top row*

top\_rows7 = sorted\_data7.head(1)

print('Year with least accidents is')

print(top\_rows7)

**Output:**

**PS D:\Semester 4\TDS> python -u "d:\Semester 4\TDS\Assignment1Q1.py"**

**2022605**

**Province with most accidents is**

**Location**

**Punjab 49354**

**Name: Total number of accidents, dtype: int64**

**Province with least accidents is**

**Location**

**Islamabad 1640**

**Name: Total number of accidents, dtype: int64**

**Province with most kills is**

**Location**

**Punjab 29978**

**Name: Killed, dtype: int64**

**Province with least kills is**

**Location**

**Islamabad 931**

**Name: Killed, dtype: int64**

**Province with most injuries is**

**Location**

**Punjab 57103**

**Name: Injured, dtype: int64**

**Province with least injuries is**

**Location**

**Islamabad 1197**

**Name: Injured, dtype: int64**

**Year with most accidents is**

**Year**

**2017-18 11121**

**Name: Total number of accidents, dtype: int64**

**Year with least accidents is**

**Year**

**2014-15\* 7865**

**Name: Total number of accidents, dtype: int64**

**PS D:\Semester 4\TDS>**

**2. Comparing Petrol Prices Across Different Countries:**

The analysis of petrol prices worldwide utilized a dataset containing information on petrol prices across various countries. The dataset was sorted by petrol prices per gallon in descending order to identify countries with the highest petrol prices. A bar graph was generated to visualize the top N countries with the highest petrol prices, providing a clear comparison of petrol prices across different regions.

**3. Comparing Petrol Prices After Changing Currency from USD to PKR:**

Furthermore, the comparison of petrol prices in different currencies, specifically USD and PKR, focused on analyzing petrol prices in Pakistan. The dataset was filtered to include only data related to Pakistan, and petrol prices in PKR and USD were plotted side by side on a bar graph. This comparison highlighted the impact of currency fluctuations on petrol prices and provided valuable insights into petrol price dynamics in Pakistan.

**Code:**

*#import libraries*

import pandas as pd

import matplotlib.pyplot as plt

*# Read the CSV file*

df = pd.read\_csv(r'D:/Semester 4/TDS/Petrol\_Gas\_Prices\_Worldwide.csv', encoding='latin-1')

print('2022605')

*#Comparing petrol prices across different countries using bar graph*

*# Sort the dataset by petrol prices per gallon in decending order so we can know countries with highest petrol prices*

df\_sorted = df.sort\_values(by='Price Per Gallon (USD)', ascending=False)

*# Plot the top N countries petrol prices in the list because there were too many countries and graph was becoming messy*

top\_n = 20     *# Change the number as needed*

*# plot the petrol prices*

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

plt.bar(df\_sorted['Country'][:top\_n], df\_sorted['Price Per Gallon (USD)'][:top\_n], color='blue')

plt.xlabel('Countries')

plt.ylabel('Petrol Price per Gallon(USD)')

plt.title('Petrol Prices Across Countries')

plt.xticks(rotation=90)  *# Rotate x-axis labels for better readability*

plt.tight\_layout()  *# Adjust layout to prevent clipping of labels*

plt.show()

*# Comparing petrol price after changing of currency from USD to PKR*

*# Filtering Pakistan because we only need Pakistan for this comparison*

df\_filtered = df[df['Country'].isin(['Pakistan'])]

*#plot petrol price in different currencies*

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

plt.bar(df\_filtered['Country'] + ' (PKR)', df\_filtered['Price Per Liter (PKR)'], color='blue', label='Petrol Price (PKR)')

plt.bar(df\_filtered['Country'] + ' (USD)', df\_filtered['Price Per Liter (USD)'], color='red', label='Petrol Price (USD)')

plt.xlabel('Country')

plt.ylabel('Petrol Price')

plt.title('Petrol Prices Comparison (PKR vs USD)')

plt.xticks(rotation=90)  *# Rotate x-axis labels for better readability*

plt.legend()

plt.tight\_layout()  *# Adjust layout to prevent clipping of labels*

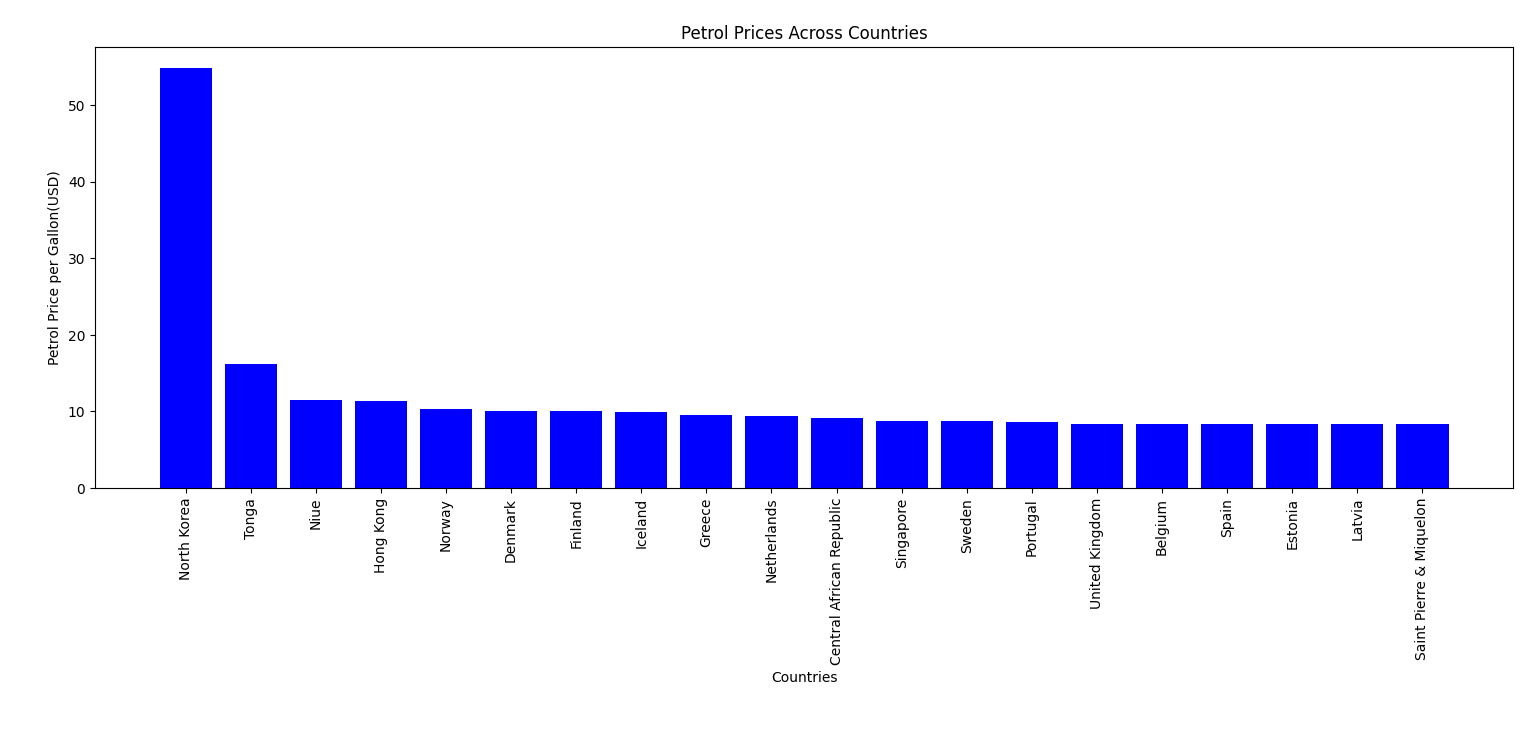
plt.show()

**Output:**

**PS D:\Semester 4\TDS> python -u "d:\Semester 4\TDS\Assignment1Q2.py"**

**2022605**

**PS D:\Semester 4\TDS>**

****

**A blue square with red lines

Description automatically generated**

**Conclusion:**

The combined analysis of traffic accidents and petrol prices offers valuable insights into two critical aspects of transportation and energy economics. The analysis of traffic accidents provides policymakers and stakeholders with valuable information to develop strategies for improving road safety and reducing accident-related injuries and fatalities. Additionally, the comparison of petrol prices worldwide and the analysis of petrol prices in different currencies offer insights into global fuel price trends and their implications for consumers and policymakers.

**Recommendations:**

Based on the findings of the analysis, it is recommended to implement targeted interventions to improve road safety in provinces with high accident rates. Additionally, policymakers may consider measures to address the impact of high petrol prices on consumers, such as implementing fuel subsidies or promoting alternative energy sources. Furthermore, ongoing monitoring of petrol prices and traffic accident trends is essential to inform evidence-based decision-making and policy development in the transportation and energy sectors.

**References:**

Pandas Documentation: https://pandas.pydata.org/pandas-docs/stable/index.html

Matplotlib Documentation: https://matplotlib.org/stable/contents.html