"""

Libraries

"""

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

import seaborn as sns

from scipy.stats import skew

from scipy.stats import kurtosis

def dataExtraction(Filepath):

"""

Load a CSV file into a pandas DataFrame and create a reversed DataFrame.

Parameters:

- Filepath (str): The file path to the CSV file.

Returns:

- dfOrg (pd.DataFrame): Original DataFrame loaded from the CSV file.

- dataFrame (pd.DataFrame): Reversed DataFrame with 'Country Name'

and 'Time' columns swapped.

"""

df = pd.read\_csv(Filepath)

dfOrg = df.copy()

df[['Country Name' , 'Time']] = df[['Time' , 'Country Name']]

dataFrame = df.rename(columns = {'Country Name': 'Time' , 'Time': 'Country Name'})

return dfOrg , dataFrame

def histPlot():

# Plotting a skewness plot for the column using Seaborn

plt.figure(figsize = (8 , 6))

sns.histplot(df['Agricultural raw materials exports (% of merchandise exports) [TX.VAL.AGRI.ZS.UN]'] ,

kde = True , stat = "density" , color = "skyblue" , bins = 10)

plt.title('Skewness Plot for a Agricultural raw materials exports (% of merchandise exports) ' ,

fontsize = 17)

plt.xlabel('Skewness Plot for a Agricultural raw materials export Values')

plt.ylabel('Density')

plt.show()

def kdeplot():

"""

Create a histogram plot to visualize the distribution of a column in the DataFrame.

Parameters:

- df (pd.DataFrame): DataFrame containing the data to be plotted.

Returns:

None

"""

# Plotting a kurtosis plot for the column using Seaborn

plt.figure(figsize = (8 , 6))

sns.kdeplot(df['Average time to clear exports through customs (days) [IC.CUS.DURS.EX]'] ,

fill = True , color = "skyblue")

plt.title('Kurtosis Plot for Average time to clear exports through customs (days) [IC.CUS.DURS.EX]' ,

fontsize = 17)

plt.xlabel('Average time to clear exports through customs (days) [IC.CUS.DURS.EX]')

plt.ylabel('Density')

plt.show()

def linePlot():

"""

Create a line plot to visualize the trend of a variable over time for selected countries.

Parameters:

- df (pd.DataFrame): DataFrame containing data for multiple countries and years.

Returns:

None

"""

for country in ['Afghanistan' , 'Australia' , 'China' , 'Finland' , 'France']:

country\_data = df[df['Country Name'] == country]

plt.plot(country\_data['Time'] ,

country\_data['Insurance and financial services (% of commercial service exports) '

'[TX.VAL.INSF.ZS.WT]'] ,

label = country)

# Adding labels and title

plt.xlabel('Year')

plt.ylabel('Insurance and financial services ')

plt.title('Insurance and financial services Over the Years' , fontsize = 17)

# Adding legend

plt.legend()

# Display a grid

plt.grid(True)

plt.show()

def barGraph():

"""

Create a bar plot to visualize the trend of a variable over time

for multiple countries.

Parameters:

- barGraphData (pd.DataFrame): DataFrame containing data for multiple countries and years.

Returns:

None

"""

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

sns.barplot(x = 'Time', y = 'International tourism, receipts (% of total exports) '

'[ST.INT.RCPT.XP.ZS]',

hue = 'Country Name' , data = barGraphData)

plt.title('International tourism, receipts (% of total exports) Over Time (Bar Plot)' ,

fontsize = 17)

plt.xlabel('Year')

plt.ylabel('International tourism, receipts (% of total exports) ')

plt.legend(loc = 'upper left')

plt.show()

def pieGraph():

"""

Create a pie chart to visualize the distribution of a variable

across different categories.

Parameters:

- pieData (pd.DataFrame): DataFrame containing data for the

variable and corresponding categories.

Returns:

None

"""

# Plotting a pie chart using Matplotlib

plt.figure(figsize = (8 , 8))

plt.pie(pieData['Profit tax (% of commercial profits) [IC.TAX.PRFT.CP.ZS]'] ,

labels = pieData['Country Name'] , autopct = '%1.1f%%' , startangle = 140)

plt.title('Profit tax (% of commercial profits)' , fontsize = 17)

plt.show()

dfOrg , df = dataExtraction('NarendraData.csv')

print("Dataframe 1")

print(dfOrg.head())

print("Dataframe 2")

print(df.head())

"Statistical Methods"

df['Agricultural raw materials exports (% of merchandise exports) [TX.VAL.AGRI.ZS.UN]'] = \

pd.to\_numeric(df['Agricultural raw materials exports (% of merchandise exports) [TX.VAL.AGRI.ZS.UN]']

,errors = 'coerce')

df\_describe = df['Agricultural raw materials exports (% of merchandise exports) [TX.VAL.AGRI.ZS.UN]']\

.describe()

print("Describe" , df\_describe)

skewnessvalue = df['Agricultural raw materials exports (% of merchandise exports) [TX.VAL.AGRI.ZS.UN]']\

.skew()

print(skewnessvalue)

histPlot()

df['Average time to clear exports through customs (days) [IC.CUS.DURS.EX]'] = \

pd.to\_numeric(df['Average time to clear exports through customs (days) [IC.CUS.DURS.EX]'] ,

errors = 'coerce')

kurtosisval = df['Average time to clear exports through customs (days) [IC.CUS.DURS.EX]'].kurtosis()

print(kurtosisval)

kdeplot()

df['Insurance and financial services (% of commercial service exports) [TX.VAL.INSF.ZS.WT]'] = \

pd.to\_numeric(df['Insurance and financial services (% of commercial service exports) [TX.VAL.INSF.ZS.WT]'] ,

errors = 'coerce')

linePlot()

df['International tourism, receipts (% of total exports) [ST.INT.RCPT.XP.ZS]'] = \

pd.to\_numeric(df['International tourism, receipts (% of total exports) [ST.INT.RCPT.XP.ZS]'] ,

errors = 'coerce')

df['Time'] = pd.to\_numeric(df['Time'] , errors = 'coerce')

barGraphData = df[(df['Time'] >= 2015) & (df['Time'] <= 2020)]

# Plotting a bar graph for insurance percentage by country over time using Seaborn

barGraph()

df['Profit tax (% of commercial profits) [IC.TAX.PRFT.CP.ZS]'] = \

pd.to\_numeric(df['Profit tax (% of commercial profits) [IC.TAX.PRFT.CP.ZS]'] ,

errors = 'coerce')

pieData = df[df['Time'] == 2019]

pieGraph()