"""

Created on Thu Nov 2 10:35:22 2023

@author: aishw

"""

import pandas as pd

import matplotlib.pyplot as plt

#Visualisation 1 (Line Plot): Seasonal rainfall in UK from 2000 to 2022/

#Max and Min seasonal rainfall in UK from 2000 to 2022.

def read\_and\_process\_data(file\_path):

# Read file into dataframe

df\_UK = pd.read\_excel(file\_path)

# Filter data for the years 2000-2022

df\_seasonal = pd.DataFrame(df\_UK[(df\_UK["year"] >= 2000) & (df\_UK["year"] <= 2022)],

columns=["year", "win", "spr", "sum", "aut"])

# Calculate max and min rainfall during the seasons over the years (2000-2022)

df\_seasonal["max"] = df\_seasonal[["win", "spr", "sum", "aut"]].max(axis=1)

df\_seasonal["min"] = df\_seasonal[["win", "spr", "sum", "aut"]].min(axis=1)

# Rename the columns

df\_seasonal.rename(columns={"year": "Year",

"win": "Winter",

"spr": "Spring",

"sum": "Summer",

"aut": "Autumn",

"max": "Maximum\_rainfall",

"min": "Minimum\_rainfall"}, inplace=True)

return df\_seasonal

def plot\_max\_min\_rainfall(df\_seasonal):

# Line plots

plt.figure(figsize=(30, 8))

# Plot the max and min rainfall during seasons with labels and customizing visualization

plt.subplot(1, 2, 1)

plt.plot(df\_seasonal["Year"], df\_seasonal["Maximum\_rainfall"], marker="o",

label="Maximum rainfall", color="green")

plt.plot(df\_seasonal["Year"], df\_seasonal["Minimum\_rainfall"], marker="o",

label="Minimum rainfall", color="red")

# Set title & labels and show the legend for lineplot1

plt.legend(title="Max and Min rainfall", borderpad=0.5, fontsize=9)

plt.title("Max and Min seasonal rainfall in UK")

plt.xlabel("Years")

plt.ylabel("Rainfall precipitation in millimetres (mm)")

plt.xticks(df\_seasonal["Year"])

def plot\_seasonal\_rainfall(df\_seasonal):

# Plot the four seasons with labels and customizing visualization

plt.subplot(1, 2, 2)

plt.plot(df\_seasonal["Year"], df\_seasonal["Winter"],

label="winter", color="green")

plt.plot(df\_seasonal["Year"], df\_seasonal["Spring"],

label="spring", color="red")

plt.plot(df\_seasonal["Year"], df\_seasonal["Summer"],

label="summer", color="orange")

plt.plot(df\_seasonal["Year"], df\_seasonal["Autumn"],

label="autumn", color="brown")

# Set title & labels and show the legend for lineplot2

plt.legend(title="Seasonal Rainfall", borderpad=0.5, fontsize=9)

plt.title("Seasonal rainfall in UK")

plt.xlabel("Years")

plt.ylabel("Rainfall precipitation in millimetres (mm)")

plt.xticks(df\_seasonal["Year"])

# Save as png

plt.savefig("UK\_seasonal\_rainfall.png")

# Show the plot

plt.show()

# Main program:

file\_path = "UK\_rainfall.xlsx"

df\_seasonal\_data = read\_and\_process\_data(file\_path)

print(df\_seasonal\_data)

plot\_max\_min\_rainfall(df\_seasonal\_data)

plot\_seasonal\_rainfall(df\_seasonal\_data)

#Visualisation 2(Pie chart): Ethnicity Percentage of Asians in England and Wales Regions

def read\_and\_print\_data(file\_path):

# Read file into dataframe

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

print(df\_diversity)

def plot\_ethnicity\_percentage(df\_diversity, ethnicity="Asian"):

# Filter data based on ethnicity

df\_ethnicity = pd.DataFrame(df\_diversity[df\_diversity["Ethnicity"] == ethnicity],

columns=["Ethnicity",

"Region",

"percentage of ethnic group"])

print(df\_ethnicity)

# Extract region and percentage of ethnic group

region = df\_ethnicity["Region"]

per\_ethnic = df\_ethnicity["percentage of ethnic group"]

# Pie Plot

plt.figure(figsize=(40, 20))

explode = (0,) \* len(region) # Set explode values

plt.pie(per\_ethnic, explode=explode, labels=region, startangle=45, autopct='%1.1f%%')

# Set title and show the legend

plt.legend(region, title="Region", borderpad=1, fontsize=11)

plt.title(f"Ethnicity Percentage of {ethnicity}s in England and Wales Regions",

fontsize=25)

# Save as png

plt.savefig(f"Ethnicity\_Percentage\_{ethnicity}.png")

# Show the plot

plt.show()

# Main program:

file\_path = "Diversity.csv"

read\_and\_print\_data(file\_path)

plot\_ethnicity\_percentage(pd.read\_csv(file\_path), ethnicity="Asian")

#Visualisation 3 (Bar chart): Average rainfall per year (2000 – 2022)

def read\_and\_process\_data(file\_path, start\_year, end\_year):

# Read file into dataframe

df\_UK = pd.read\_excel(file\_path)

# Filter data for the specified years

df\_yearly = pd.DataFrame(df\_UK[(df\_UK["year"] >= start\_year) & (df\_UK["year"] <= end\_year)],

columns=["year", "jan", "feb",

"mar", "apr", "may",

"jun", "jul", "aug",

"sep", "oct", "nov", "dec"])

# Calculate average precipitation per year

df\_yearly["Average"] = df\_yearly[["jan", "feb", "mar",

"apr", "may", "jun",

"jul", "aug", "sep",

"oct", "nov", "dec"]].mean(axis=1)

# Rename the columns

df\_yearly.rename(columns={"year": "Year",

"jan": "January",

"feb": "February",

"mar": "March",

"apr": "April",

"may": "May",

"jun": "June",

"jul": "July",

"aug": "August",

"sep": "September",

"oct": "October",

"nov": "November",

"dec": "December",

"Average": "Average"}, inplace=True)

return df\_yearly

def plot\_average\_rainfall(df\_yearly):

# Bar Plot

plt.figure(figsize=(20, 8))

plt.bar(df\_yearly["Year"], df\_yearly["Average"])

# Set title

plt.title("Average precipitation of rainfall per years (2000 - 2022)",

fontweight='bold', fontsize=10)

# Set labels and legend

plt.xlabel("Years (2000-2022)", fontweight='bold', fontsize=13)

plt.ylabel("Rainfall precipitation (mm)", fontweight='bold', fontsize=13)

plt.legend(df\_yearly["Year"], title="Max Average rainfall", borderpad=1, fontsize=11)

# Save as png

plt.savefig("Average\_rainfall\_per\_year.png")

# Show the plot

plt.show()

# Main program

file\_path = "UK\_rainfall.xlsx"

start\_year = 2000

end\_year = 2022

df\_yearly\_data = read\_and\_process\_data(file\_path, start\_year, end\_year)

print(df\_yearly\_data)

plot\_average\_rainfall(df\_yearly\_data)