"""
Developer: Daniel Ivan Lewis
SL: Sebastian
11/26/2018
ISTA 131 Final Project: Disasters

Background: The Federal Emergency Management Agency (FEMA) is a part of the United States Department of Homeland
Security. FEMA works with distributing federal disaster aid funds to communities and individuals affected by disasters. Although FEMA was created in 1978, the FEMA data base traces back to 1954 with the first declared disasters and government funding information.

Summary: This python script analyzes FEMA disaster information and creates three separate visual plots representing the data pulled from the csv files.

Plot #1:
The first plot generated is a scatter plot of all of the FEMA declarations made
starting from 1953 to 2016 along with a regression line to analyze how the amount of FEMA declarations has changed throughout time.

Plot #2:
The second plot generated is a bar chart showcasing the average declaration types per year i,e, Hurricanes, Floods, etc.

Plot #3:
The third plot generated is a side-by-side box plot of all the recorded deaths available from 1954 to 2018. This showcases the average amount of deaths each year due to the following disaster types: Tornado, Flood, Hurricane and Fire. The box plots also identify the median and spread of the yearly deaths due to those specific disasters.
"""

```
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
import math

def get_disasters(fname = 'femaDeclarations.csv'):
    """
    Purpose: This function reads in the femaDeclarations csv file gathered from the official FEMA government website.
    The database lists all of the federal emergency declarations starting from 1954 to 2018 (Present Date). The data
    given includes declaration dates, federal assistance provided, disaster classifications, and county affected.
    Parameters: fname - A FEMA disaster csv file containing the information described above, set to
    'femaDeclarations.csv' as default)
    Returns: A FEMA disasters data frame containing information about all of the federal emergency declarations since
    1954 to 2018 (Present time).
    """
    return pd.read_csv(fname, header=0)

def get_data(fname):
    """
    Purpose: This function reads in a csv file passed in from The International Disaster Database created by Belgium's
    Centre for Research on the Epidemiology of Disasters (CRED), along with support from the World Health
Organization.
    The csv files passed in contain information about yearly deaths and funding information for disasters in the U.S.
    Parameters: fname - A csv file from the emdat database containing information about a specific disaster type from
```

```
    the U.S.
    Returns: A disaster Data Frame that contains yearly funding information and death information.
    """
    return pd.read_csv(fname, skiprows=[0], index_col=[0])

def get_hurricanes(df):
    """
    Purpose: This function generates the first plot showing the amount of FEMA declarations from the year 1954 to 2018
    (Present time). The plot also includes a line of regression to show how the amount of FEMA declarations have
changed
    throughout time. This shows us that over time there has been a slight increase in the amount of declarations
    through the years.
    Parameters: df - The disasters Data Frame generated from the get_disasters() function.
    Returns: None, Generates the first plot.
    """
    hurricanes = pd.Series(data=[0]*65, index=[i for i in range(1954, 2019)])  # Creates series for Hurricane data
    seen = []  # Empty list that will hold disaster number to make sure duplicates are removed
    for idx in df.index:
        year = int(df.loc[idx, 'declarationDate'].split('-')[0])  # Extracts year from date
        if df.loc[idx, 'incidentType'] == 'Hurricane' and df.loc[idx, 'disasterNumber'] not in seen:
            hurricanes.at[year] = hurricanes.loc[year] + 1  # Adds a count to the year
            seen.append(df.loc[idx, 'disasterNumber'])  # Adds the declaration number to seen list

    x = sm.add_constant(hurricanes.index)
    model = sm.OLS(hurricanes, x)
    line = model.fit()  # Creates a model from the hurricanes series created above
    slope, intercept = line.params[1], line.params[0]  # Gets slope and intercept used for regression line

    best_fit = [((slope * i) + intercept) for i in range(hurricanes.index[0], hurricanes.index[-1])]
    best_fit_srs = pd.Series(data=best_fit, index=[i for i in range(1955, 2019)])  # Creates regression line series

    # Code for generating first plot
    plt.figure(figsize=(14, 12))
    plt.plot(best_fit_srs, c='red')
    plt.title('FEMA Hurricane Declarations per Year', fontsize=24)
    plt.xlabel('Year (1954 - 2018)', fontsize=20)  # Sets X label
    plt.ylabel('Number of Declarations', fontsize=20)  # Sets Y label
    plt.scatter(x=hurricanes.index, y=hurricanes.values, c='blue')
    plt.xticks([i for i in range(1955, 2020, 5)])
    plt.tick_params(axis='both', which='major', labelsize=15)

def get_declaration_types(disasters):
    """
    Purpose: This function generates the second graph mentioned above. The bar chart displays the average declaration
    amounts for each classification type found in the disasters data frame. Classifications with an average less than
    0.5 are omitted from the bar chart, and all values are rounded up to the nearest whole integer. This gives an
    estimate of the disasters that are declared throughout an individual year in the U.S.
    Parameters: disasters - The disasters Data Frame generated from the get_disasters() function.
    Returns: None, Generates the second plot.
    """
    #  cols is list of all of the declaration types found in the data frame
    cols = ['Tornado', 'Flood', 'Hurricane', 'Fire', 'Severe Storm(s)', 'Drought',
            'Snow', 'Severe Ice Storm', 'Other', 'Volcano', 'Earthquake', 'Typhoon',
            'Toxic Substances', 'Dam/Levee Break', 'Freezing', 'Coastal Storm', 'Fishing Losses',
```

```
        'Mud/Landslide', 'Human Cause', 'Terrorist', 'Tsunami', 'Chemical']
    data = [[0]*22]*66  # Creates an empty 2d list that will be used to initialize new data frame
    index = [str(i) for i in range(1953, 2019)]  # Creates index for new data frame
    seen = []  # creates an empty list that will hold declaration numbers to avoid duplicates
    types = pd.DataFrame(data=data, columns=cols, index=index)  # Creates types data frame with info from above
    for idx in disasters.index:
        year = disasters.loc[idx, 'declarationDate'].split('-')[0]  # Extracts year from disasters data frame
        if disasters.loc[idx, "disasterNumber"] not in seen:
            types.at[year, disasters.loc[idx, "incidentType"]] += 1
            seen.append(disasters.loc[idx, "disasterNumber"])

    declaration_type_per_year = pd.Series(index=types.columns)  # Creates a new series that will hold averages
    for col in types.columns:
        declaration_type_per_year.at[col] = types.loc[:, col].mean()  # Gets average declaration amount from all years
                                                                      # spanning from 1953 to 2018
    # Code to generate plot
    plt.figure(figsize=(14, 12))
    plt.title('Average FEMA Declaration Classifications', fontsize=24)
    plt.xlabel('Classification Type', fontsize=20)
    plt.ylabel('Amount of Declarations', fontsize=20)
    plt.tick_params(axis='both', which='major', labelsize=10)
    for idx in declaration_type_per_year.index:
        if declaration_type_per_year.loc[idx] > .5:  # Removes values that are below .5
            plt.bar(x=idx, height=declaration_type_per_year.loc[idx].round(), color='blue')

def compare_deaths_by_disaster():
    """
    Purpose: This function generates the third plot mentioned above. The plot includes four side by side box plots of
    deaths recorded each year since 1953 to 2018 (Present time). Each box plot represents one of the following
    disasters: tornadoes, floods, hurricanes, and fires. The plot demonstrates the average deaths per year caused by
    each disaster as well as the spread and median that is generated from the data.
    Parameters: None.
    Returns: None, Generates the third plot.
    """
    # The following data frames are generated using the get_data() function
    us_tornadoes = get_data('us_tornadoes.csv')
    us_floods = get_data('us_floods.csv')
    us_hurricanes = get_data('us_hurricanes.csv')
    us_fires = get_data('us_fire.csv')

    # The following pandas series are created by slicing the Total Deaths column from each disaster data frame
    tor = us_tornadoes.loc[:, 'Total deaths'].dropna()
    floods = us_floods.loc[:, 'Total deaths'].dropna()
    hurr = us_hurricanes.loc[:, 'Total deaths'].dropna()
    fires = us_fires.loc[:, 'Total deaths'].dropna()

    # Code to generate plot
    plt.figure(figsize=(14, 12))
    plt.boxplot([tor, floods, hurr, fires], sym='', labels=['Tornado', 'Flood', 'Hurricane', 'Fire'], showmeans=True)
    plt.tick_params(axis='both', which='major', labelsize=15)
    plt.title('Average Yearly Deaths per Disaster', fontsize=24)
    plt.xlabel('Cause of Death', fontsize=20)
    plt.ylabel('Amount of Deaths', fontsize=20)
```

```
def main():
    plt.rcParams['figure.facecolor'] = 'lightgray'
    plt.rcParams['axes.facecolor'] = 'lightblue'
    disasters = get_disasters()
    get_hurricanes(disasters)
    get_declaration_types(disasters)
    compare_deaths_by_disaster()
    plt.show()

main()
```