# -\*- coding: utf-8 -\*-

"""

Created on Tue Nov 8 00:12:50 2023

@author: Sanghavi rangineni

"""

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

# Load the dataset

crime\_data = pd.read\_csv("US\_Crime\_Rates\_1960\_2014.csv")

# Display the first few rows of the dataframe

crime\_data.head()

#1. multiple line chart

def plot\_crime\_trends(df):

"""

Let's visualize some crime data! We'll make a groovy line chart to show how different

crime types have changed over time. It'll be like a colorful history lesson.

"""

# Time to break out the crayons! We'll make a rainbow of lines.

colors = ['blue', 'green', 'red', 'cyan', 'magenta', 'yellow', 'black', 'orange', 'purple']

# Give our canvas some space so it's not a messy pile-up of lines

plt.figure(figsize=(14, 7))

# drawing a line for each crime

crime\_types = ['Violent', 'Property', 'Murder', 'Forcible\_Rape', 'Robbery',

'Aggravated\_assault', 'Burglary', 'Larceny\_Theft', 'Vehicle\_Theft']

for i, crime in enumerate(crime\_types):

plt.plot(df['Year'], df[crime], marker='o', color=colors[i], label=crime)

# Add some descriptive labels

plt.title('Crime Trends in the US (1960-2010)', fontsize=16)

plt.xlabel('Year', fontsize=12)

plt.ylabel('Number of Crimes', fontsize=12)

# A grid helps organize all the data into tidy little boxes

plt.grid(True)

# And a legend is like a map key - it explains what each line means

plt.legend()

# The moment we've been waiting for...reveal the chart!

plt.show()

# Now let's call our function with the crime data.

plot\_crime\_trends(crime\_data)

# 2.bar chart

def plot\_murders\_first\_decade(df):

"""

Let's take a trip back in time to the groovy 1960s

and check out the murder rates in the US!

We'll make a rockin' bar chart to compare the numbers year-by-year.

"""

# Grab just the first 10 rows

first\_ten\_years = df.head(10)

# Time to set up our canvas - make it big and bold!

plt.figure(figsize=(14, 7))

# Paint those bars pink to make a statement!

plt.bar(first\_ten\_years['Year'], first\_ten\_years['Murder'], color='pink')

# Give our chart a far out title

plt.title('Number of Murders in the US (1960-1969)', fontsize=16)

# Label the axes so they're clear as day

plt.xlabel('Year', fontsize=12)

plt.ylabel('Number of Murders', fontsize=12)

# Add some gridlines for a fine-looking background

plt.grid(axis='y')

# Align the years so this chart stays neat and tidy

plt.xticks(first\_ten\_years['Year'])

# And now, the moment we've been waiting for - cue the chart!

plt.show()

plot\_murders\_first\_decade(crime\_data)

# 3. Pie chart

def plot\_crime\_pie\_chart(df, year, threshold=1):

"""

Pie chart to visualize the proportion of different crimes in a given year.

Combines smaller slices into an 'Other' category and uses a legend instead of labels.

"""

# Isolate the data for the chosen year

year\_data = df[df['Year'] == year]

# Keep just the crime stats, ditch the other columns

crime\_stats = year\_data.drop(columns=['Year', 'Total', 'Population']).iloc[0]

# Group smaller categories into "Other"

small\_categories = crime\_stats[crime\_stats < threshold].sum()

crime\_stats = crime\_stats[crime\_stats >= threshold]

crime\_stats['Other'] = small\_categories

# Define a color palette

colors = plt.cm.Paired(np.arange(len(crime\_stats))/len(crime\_stats))

# Calculate the explode values for smaller slices to separate them a bit

explode\_values = [0.1 if value < threshold else 0 for value in crime\_stats]

# Crafting our pie chart

plt.figure(figsize=(12, 8))

wedges, texts, autotexts = plt.pie(

crime\_stats,

autopct='%1.1f%%',

colors=colors,

startangle=140,

explode=explode\_values

)

# We'll put the legend outside of the plot

plt.legend(wedges, crime\_stats.index, title="Crime Types", loc="center left", bbox\_to\_anchor=(1, 0, 0.5, 1))

# Ensuring our pie looks as good as it tastes with an equal aspect ratio

plt.axis('equal')

# Adding a clear title to our chart

plt.title(f'Crime Distribution in the US, {year}', fontsize=16)

# Show off the finished product!

plt.show()

# Testing pie chart with the year 1965

plot\_crime\_pie\_chart(crime\_data, 1965)