#!/usr/bin/env python3

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

"""

Created on Thu Nov 9 21:49:54 2023

@author: kamalib

"""

#importing libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

#Read the file and print

Walmart\_Sales= pd.read\_csv('/Users/kamalib/Downloads/WALMART\_SALES\_DATA-1.csv')

print(Walmart\_Sales)

#defining the functions

#The graph

def plot\_line\_graph(shop1\_data, shop2\_data, x\_label, y\_label, graph\_title,

legend\_labels):

# To plot the figure

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

# Plot the two shops with labels

plt.plot(shop1\_data, label=legend\_labels[0])

plt.plot(shop2\_data, label=legend\_labels[1])

# To plot xlabel and ylabel

plt.xlabel(x\_label)

plt.ylabel(y\_label)

# Title for the line graph

plt.title(graph\_title)

# To plot the legend

plt.legend()

# To show the plot

plt.show()

#Inference:

#The chart denotes the comparison of stores with various factors

#The factors taken into consideration are Fuel price and Unemployment

#The pie chart

def plot\_pie\_chart(data, labels, title):

# To plot the pie chart

plt.pie(data, labels=labels)

# Title for the pie chart

plt.title(title)

# To plot the legend

plt.legend(loc=1)

# To show the plot

plt.show()

#Inference:

#The pie chart denotes the count of purchase on holidays and non-holidays

#The purchase on holidays is more than non-holidays.

#The bar chart

def plot\_bar\_chart(data, x\_column, height\_column, x\_label, y\_label, title):

# Plot bar chart

plt.bar(x=data[x\_column], height=data[height\_column])

# To plot xlabel and ylabel

plt.xlabel(x\_label)

plt.ylabel(y\_label)

# Title for the bar chart

plt.title(title)

# To show the plot

plt.show()

#Inference:

#The chart denotes the comparison of average sales of holiday week and non-holiday week

#It is observed that the average sales of holiday week is more compared to non-holiday week

#The line chart

def plot\_sales\_and\_factors(data\_path):

# Read the file

Walmart\_Sales = pd.read\_csv(data\_path)

print(Walmart\_Sales)

# Grouping columns together

salesdata = Walmart\_Sales.groupby(['Holiday\_Flag']).agg({'Weekly\_Sales': ['sum']})

salesdata1 = Walmart\_Sales.groupby(['Holiday\_Flag']).agg({'Fuel\_Price': ['sum']})

salesdata2 = Walmart\_Sales.groupby(['Holiday\_Flag']).agg({'Unemployment': ['sum']})

# To plot figure

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

salesdata1[('Fuel\_Price', 'sum')].plot(label='Fuel\_Price Sum')

salesdata2[('Unemployment', 'sum')].plot(label='Unemployment Sum')

# To plot the legend

plt.legend()

# To show the plot

plt.show()

#Inference:

#The chart denotes the comparison of average sales of holiday week and non-holiday week

#It is observed that the average sales of holiday week is more compared to non-holiday week

#Main Program

#grouping fuel\_price together

shop = Walmart\_Sales.groupby(['Store']).agg({'Fuel\_Price':['mean','max',

'sum']})

shop[:5]

#grouping Unemployment

shop2 = Walmart\_Sales.groupby(['Store']).agg({'Unemployment':['sum']})

##grouping columns together

salesdata = Walmart\_Sales.groupby(['Holiday\_Flag']).agg({'Weekly\_Sales':

['sum']})

salesdata[:5]

salesdata1 = Walmart\_Sales.groupby(['Holiday\_Flag']).agg({'Fuel\_Price':

['sum']})

salesdata1[:5]

salesdata2 = Walmart\_Sales.groupby(['Holiday\_Flag']).agg({'Unemployment':

['sum']})

salesdata2[:5]

Walmart\_Sales.head()

Walmart\_Sales.shape

#Extracting the columns and print

Walmart\_Sales.isnull().sum()

Walmart\_Sales.info()

#To set the date according to month & year

Walmart\_Sales['Month'] = pd.DatetimeIndex(Walmart\_Sales['Date']).month

Walmart\_Sales['Year'] = pd.DatetimeIndex(Walmart\_Sales['Date']).year

Walmart\_Sales.head()

#To calculate the sales usings counts

Holiday = Walmart\_Sales['Holiday\_Flag'].value\_counts().index

Holiday\_Count = Walmart\_Sales['Holiday\_Flag'].value\_counts().values

#grouping columns together

avg\_sales = Walmart\_Sales.groupby('Holiday\_Flag')['Weekly\_Sales'].mean().reset\_index()

#Renaming the columns

avg\_sales.rename(columns={'Weekly\_Sales': 'Average\_Sales'}, inplace=True)

#Replacing the columns

avg\_sales['Holiday\_Flag'] = avg\_sales['Holiday\_Flag'].replace({0: 'Non-Holiday Week', 1: 'Special Holiday Week'})

#plots

plot\_line\_graph(shop['Fuel\_Price'], shop2['Unemployment'], 'Store', 'Factors', 'Fuel vs Unemployment', ['Shop 1', 'Shop 2'])

plot\_pie\_chart(Holiday\_Count, ['Holiday', 'Non-Holiday'], 'Holiday & Non-Holiday')

plot\_bar\_chart(avg\_sales, 'Holiday\_Flag', 'Average\_Sales', 'Holiday Week', 'Avg Sales', 'Average sale on Non-holiday and Special holiday')

plot\_sales\_and\_factors('/Users/kamalib/Downloads/WALMART\_SALES\_DATA-1.csv')