import time

import pandas as pd

import numpy as np

CITY\_DATA = { 'chicago': 'chicago.csv',

'new york city': 'new\_york\_city.csv',

'washington': 'washington.csv' }

def get\_filters():

"""

Asks user to specify a city, month, and day to analyze.

Returns:

(str) city - name of the city to analyze

(str) month - name of the month to filter by, or "all" to apply no month filter

(str) day - name of the day of week to filter by, or "all" to apply no day filter

"""

print('Hello! Let\'s explore some US bikeshare data!')

# get user input for city (chicago, new york city, washington). HINT: Use a while loop to handle invalid inputs

city =set(city.lower() for city in ("Chicago", "New York", "Washington"))

while True:

answer = input("Which City?")

if answer.lower() in city:

break

print("sorry, that is no city, kindly try again")

# get user input for month (all, january, february, ... , june)

month = set(month.lower() for month in ("January", "February", "March", "April",

"May", "June", "July", "August", "September", "October", "November", "December"))

while True:

answer = input("Which Month? ")

if answer.lower() in month:

break

print("Sorry, didn't recognize your answer, try again")

# get user input for day of week (all, monday, tuesday, ... sunday)

day = set(day.lower() for day in ("All", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"))

while True:

answer = input("which day? or all day?")

if answer.lower() in day:

break

print("sorry kindly try again")

print('-'\*40)

return city, month, day

def load\_data(city, month, day):

"""

Loads data for the specified city and filters by month and day if applicable.

Args:

(str) city - name of the city to analyze

(str) month - name of the month to filter by, or "all" to apply no month filter

(str) day - name of the day of week to filter by, or "all" to apply no day filter

Returns:

df - Pandas DataFrame containing city data filtered by month and day

"""

# load data file into a dataframe

df = pd.read\_csv(CITY\_DATA[city])

# convert the Start Time column to datetime

df['Start Time'] = pd.to\_datetime(df['Start Time'])

# produce a column called month and weekday

df['month'] = df['Start Time'].dt.month

df['day'] = df['Start Time'].dt.weekday\_name

if month !='all':

months = ['january', 'february', 'march', 'april', 'may', 'june','july','august','september','october','november','december']

month = months.index(month) + 1

df = df[df['month'] == month]

if day !='all':

df=df[df['day']==day.title()]

return df

def time\_stats(df):

"""Displays statistics on the most frequent times of travel."""

print('\nCalculating The Most Frequent Times of Travel...\n')

start\_time = time.time()

# display the most common month

most\_common\_month = df["month"].mode()[0]

print(most\_common\_month)

# display the most common day of week

common\_day = df["day"].mode()[0]

print(common\_day)

# display the most common start hour

common\_hour = df["hour"].mode()[0]

print(common\_hour)

print("\nThis took %s seconds." % (time.time() - start\_time))

print('-'\*40)

def station\_stats(df):

"""Displays statistics on the most popular stations and trip."""

print('\nCalculating The Most Popular Stations and Trip...\n')

start\_time = time.time()

# display most commonly used start station

com\_start = df["Start Station"].mode()[0]

print(com\_start)

# display most commonly used end station

com\_end = df["End Station"].mode()[0]

print(com\_end)

# display most frequent combination of start station and end station trip

com\_combi = df["End Station"] + df["Start Station"].mode()[0]

print("\nThis took %s seconds." % (time.time() - start\_time))

print('-'\*40)

def trip\_duration\_stats(df):

"""Displays statistics on the total and average trip duration."""

print('\nCalculating Trip Duration...\n')

start\_time = time.time()

# display total travel time

travel\_time = df["Trip Duration"].sum()

print (travel\_time)

# display mean travel time

mean\_time = df[" Trip Duration"].mean()

print (mean\_time)

print("\nThis took %s seconds." % (time.time() - start\_time))

print('-'\*40)

def user\_stats(df):

"""Displays statistics on bikeshare users."""

print('\nCalculating User Stats...\n')

start\_time = time.time()

# Display counts of user types

count\_user = df["User Type"].value\_counts()

print(count\_user)

# Display counts of gender

count\_gender = df["Gender"].value\_counts()

print(count\_gender)

# Display earliest, most recent, and most common year of birth

early\_birth = df["Birth Year"].max()

recent\_birth = df["Birth Year"].min()

common\_birth = df["Birth Year"].mode()[0]

print (early\_birth, recent\_birth, common\_birth)

print("\nThis took %s seconds." % (time.time() - start\_time))

print('-'\*40)

def main():

while True:

city, month, day = get\_filters()

df = load\_data(city, month, day)

time\_stats(df)

station\_stats(df)

trip\_duration\_stats(df)

user\_stats(df)

restart = input('\nWould you like to restart? Enter yes or no.\n')

if restart.lower() != 'yes':

break

if \_\_name\_\_ == "\_\_main\_\_":

main()