"""

RUN: ipython bikeshare.py

REFERENCES:

https://stackoverflow.com/questions/29652264/why-is-my-python-function-not-being-executed

https://classroom.udacity.com/nanodegrees/nd104/parts/53470233-d93c-4a31-a59f-11388272fe6b/modules/0f8a717f-4ac2-49d7-9ac4-15ae692793fa/lessons/ee7d089a-4a92-4e5d-96d2-bb256fae28e9/concepts/b05491a6-fd04-4889-8736-df78744b3615

https://www.delftstack.com/howto/python/mode-of-list-in-python

https://www.studytonight.com/python-howtos/how-to-get-month-name-from-month-number-in-python

https://www.alpharithms.com/python-datetime-weekday-name-482616/

https://stackoverflow.com/questions/287871/how-to-print-colored-text-to-the-terminal

"""

import time

import pandas as pd

import numpy as np

import calendar

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

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

'washington': 'washington.csv' }

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

city = ''

month = ''

day = ''

#def get\_filters(city,month,day):

city = ''

while city not in ('chicago', 'new york city', 'washington'):

city = input("Please select the city name from the following options : chicago, new york city, washington.\n").lower()

if city in ('chicago','new york city','washington'):

break

month = ''

while month not in ('all','january', 'february','march', 'april', 'may', 'june',’july’, ‘august’,’september’, ‘october’, ‘november’, ‘december’):

month = input("\nPlease select the month from the following options : all,january,february,march,april,may,june,july, august,september,october,november, december.\n").lower()

if month in ('all','january', 'february','march', 'april', 'may', 'june',’july’, ‘august’,’september’, ‘october’, ‘november’, ‘december’):

break

day = ''

while day not in ('all','monday', 'tuesday','wednesday', 'thursday', 'friday', 'saturday', 'sunday'):

day = input("\nPlease select the day from the following options : all,monday,tuesday,wednesday,thursday,friday,saturday.\n").lower()

if day in ('all','monday', 'tuesday','wednesday', 'thursday', 'friday', 'saturday', 'sunday'):

break

#get\_filters(city,month,day)

print('-'\*50)

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

# load data file into a dataframe

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

#print (df.head())

# convert the Start Time column to datetime

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

# extract month and day of week from Start Time to create new columns

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

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

# filter by month if applicable

if month != 'all':

# use the index of the months list to get the corresponding int

months = ['january', 'february', 'march', 'april', 'may', 'june']

month = months.index(month) + 1

# filter by month to create the new dataframe

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

# filter by day of week if applicable

if day != 'all':

# filter by day of week to create the new dataframe

df = df[df['day\_of\_week'] == day.title()]

#print (day.title())

#print (df.head())

return df

load\_data(city, month, day)

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

start\_time = time.time()

# TO DO: display the most common month

# load data file into a dataframe

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

# convert the Start Time column to datetime

df['Start Time'] = pd.to\_datetime(df['Start Time'], format='%Y%m%d', errors='ignore')

#print(df['Start Time'] )

# extract month from the Start Time column to create a popular month column

df['popular\_month'] = pd.DatetimeIndex(df['Start Time']).month

#print(df['popular\_month'])

popular\_month = df['popular\_month'].value\_counts().idxmax()

print('The most frequent month: ', '\033[91m' + calendar.month\_name[popular\_month] + '\033[0m' )

# TO DO: display the most common day of week

Common\_Day\_of\_week = df['day\_of\_week'].mode()[0]

print ('The most common day of the week: ', '\033[91m' + Common\_Day\_of\_week + '\033[0m' )

# TO DO: display the most common start hour

# extract hour from the Start Time column to create an hour column

df['hour'] = pd.DatetimeIndex(df['Start Time']).hour

#print(df['hour'])

popular\_hour = df['hour'].value\_counts().idxmax()

print('The most frequent Start Hour: ', '\033[91m' + str(popular\_hour) + '\033[0m')

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

print('-'\*50)

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

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

start\_time = time.time()

# TO DO: display most commonly used start station

Common\_used\_station = df['Start Station'].mode()[0]

print ('The most commonly used start station: ', '\033[91m' + Common\_used\_station + '\033[0m')

# TO DO: display most commonly used end station

Common\_used\_end\_station = df['End Station'].mode()[0]

print ('The most commonly used end station: ', '\033[91m' + Common\_used\_end\_station + '\033[0m')

# TO DO: display most frequent combination of start station and end station trip

Start\_end = (df['Start Station'] + ' AND ' + df['End Station']).mode()[0]

print ('The most commonly used Start and end station: ', '\033[91m' + Start\_end + '\033[0m')

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

print('-'\*50)

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

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

start\_time = time.time()

# TO DO: display total travel time

TotalTravel\_Time = df['Trip Duration'].sum()

print ('The total travel time: ', '\033[91m' + str(TotalTravel\_Time) + '\033[0m')

# TO DO: display mean travel time

Mean\_Travel\_Time = df['Trip Duration'].mean()

print ('The mean travel time: ', '\033[91m' + str(Mean\_Travel\_Time) + '\033[0m')

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

print('-'\*50)

display\_rows = ''

row\_count = 0

while display\_rows not in ('yes', 'no'):

display\_rows = input("Would you like to see 5 rows of data. Please state yes or no.\n").lower()

if display\_rows == ('no'):

break

elif display\_rows == 'yes':

print(df.iloc[row\_count:row\_count + 5])

row\_count = row\_count + 5

display\_rows = ''

continue

print('Have a good one')