Programming for Data Science for Enterprise Nanodegree Program

project 2

**Explore US Bike Share Data**

By Nehal Salah Mostafa Hasib

* In this project, I used python codes to explore a data related to bike share systems for three major cities in the United States—Chicago, New York City, and Washington this by writing python code to import the data from the 3 files (chicago.csv, new\_york\_city.csv,washington.csv) and answer interesting questions that I have about it by computing descriptive statistics as follows .
* **#1 Popular times of travel (i.e., occurs most often in the start time)**
  1. Which is the most common month preferable for travelling for the 3 cities?
  2. What is the most common day of week for the 3 cities for the 3 cities ?
  3. What is the most common most common hour of day for the 3 cities ?

**#2 Popular stations and trip**

1. What is the most common start station for the 3 cities?
2. What is the most common end station for the 3 cities?
3. What is the most common trip from start to end for the 3 cities(most frequent combination of start station and end station)?

**#3 Trip duration**

1. What is the total travel time for travelers for the 3 cities؟
2. What is the average travel time for travelers for the 3 cities ?

**#4 User info (counts of each user type , ,common recent data of birth)**

A)What is the user type that prevail travelers in New york , Chicago cities and Washington ?

B ) What is the most common gender that prefer traveling ?

C ) What is the most common recent year of birth for the travelers in both New york and Chicago cities?

* The first solution when I make the project using the option 2 using my personal local machine , First using the code that I use all the following frame works :

Import time

import pandas as pd

import numpy as np

import datetime as dt

**# load data file into a dataframe**

city\_data = pd.read\_csv (r’/c/Users/computec/Desktop/project2/chicago.csv, /c/Users/computec/Desktop/project2/Washington.csv, /c/Users/computec/Desktop/project2/new\_york\_city.csv ')

**another solution**

chicago = pd.read\_csv (r’/c/Users/computec/Desktop/project2/chicago.csv)

new york city **=** pd.read\_csv(r’/c/Users/computec/Desktop/project2/new\_york\_city.csv)

washington **=** pd.read\_csv(r’/c/Users/computec/Desktop/project2/ washington.csv)

**# create the dictionary city\_data to group the imported cities .csv**

city\_data = { 'chicago': 'chicago.csv', 'new york city': 'new\_york\_city.csv',

'washington': 'washington.csv' }

# **creating a DataFrame manually from a dictionary city\_data** , then passing the dictionary city\_data to the pd.DataFrame() function.

dataframes = []

for city in city\_data:

cities =dataframes.append(pd.read\_csv(chicago))

#**selecting data filters, city, month, day**

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

"""

Specifiying 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

"""

def get\_filters():

**cities** = ['Chicago','New York','Washington']

**months** = ['January','Feburary','March','April','May','June','All']

**days\_of\_week** = ['Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday','All']

**# selecting for city (Chicago, New York, Washington).**

**""" selecting the city that we want to analyze from the cities list that have been filtered"""**

while True:

city = input('\n"Chicago\", \"New York\" or \"Washington\" \n')

if city.title() in cities:

print('\n {}.'.format(city.title()))

break

**# selecting month (January - June or All).**

**""" selecting the month or all monthes that we want to analyze from the cities list that have been filtered"""**

while True:

month = input('\nDo you want to search for a specific month? You can pick data from January to June. If you need data for all months, just enter All. \n')

if month.title() in months:

print('\n {}.'.format(month.title()))

break

**# selecting day of week (Sunday - Saturday or All)**

**""" selecting the day of week that we want to analyze from the cities list that have been filtered"""**

while True:

day = input('\n'Monday\','Tuesday\','Wednesday\','Thursday\','Friday\','Saturday\','Sunday\','All'\n')

if day.title() in days\_of\_week:

print('\n {}.'.format(day.title()))

break

#  **Loading data for analysis based on city, month, and day filters**

**"""creating data frame for each city"""**

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

df = pd.read\_csv.cities(city)

**#1 Popular times of travel (i.e., occurs most often in the start time(Which is the most common month preferable for travelling for the 3 cities?)**

**# Converting the Start Time column to datetime**

**"""selecting an object (start time column ) from the data frame (df) and converting it to date time using date time function** **"""**

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

**# Extracting month and day from Start Time column to create a new columns**

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

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

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

**#Month filter - using the index of the month from list to get corresponding int**

if month != 'all':

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

month = months.index(month) + 1

**# Created new dataframe for month by filtering the month**

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

**# the most common day of week**

# filter by day of week if applicable

if day != 'all':

**"""Created new dataframe for day””’**

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

return df

**"""Displaying the statistics for the Most frequent time to travel"""**

def time\_stats(df):

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

start\_time = time.time()

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

**# Most common month**

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

Common\_month = df['month'].mode()[0]

*“””find the most popular month””’*

print( 'Most common month: ', common \_ month )

**# Most common day of week**

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

*“””find the most common day “””*

common\_day = df['day\_of\_week'].mode()[0]

print( 'Most common day : ', common\_ *day* )

**# Most common hour**

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

|  |
| --- |
| common\_hour = df['hour'].mode()[0] |

print('Most popular hour: ', common\_hour)

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

    print('-'\*40)

**#2 Popular stations and trip**

**# Most popular stations and trip**

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()

**# Most common start station**

df['common\_start'] = df['Start Station']

"""display most commonly used start station"""

|  |
| --- |
| common\_Start = df['common\_start'].mode()[0] |
| print ('The most common start station is: ', common\_Start) |
|  |
| **# C # Common end station** |
| df['common\_end'] = df['End Station'] |
| common\_end = df['common\_end'].mode()[0] |
| print('The most common end station is: ', common\_end) |
| **# Frequent start and end station** |
| common\_trip = df['common\_start'] + ' to ' + df['common\_end'] |
| print ('The most common trip is: ', common\_trip.mode()[0])  **another solution**   """display most frequent combination of start station and end station trip"""      common\_trip = (df['Start Station']+df['End Station']).mode()[0]      print(common\_trip) |
| print ("\nThis took %s seconds." % (time.time() - start\_time)) |
| print ('-'\*40)  **#3 Trip duration** |

**# Trip duration function**

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

def trip\_duration\_stats(df):

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

start\_time = time.time()

**# Trip start time**

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

**# Trip end time**

trip\_end = pd.to\_datetime(df['End Time'])

**# Total trip time - new column**

df['Trip Total Time'] = trip\_start - trip\_end

**# Adding total trip**

total\_time = df['Trip Total Time'].sum()

print("The total amount of time for a trip is: " + str(total\_time))

**Average travel time**

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

print("The average time of a trip is: " + str(mean\_time))

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

print('-'\*40)

**another solution**

**# display total travel time**

total\_travel = df['Trip Duration'].sum()

**# display mean travel time**

mean\_travel = df['Trip Duration'].mean()

print("Total travel time: ", total\_travel)

print("mean travel time: ", mean\_travel)

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

print('-'\*40)

**#4 User info (counts of each user type , ,common recent data of birth)**

**# User data Function**

def user\_stats(df):

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

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

start\_time = time.time()

**# Count user type**

user\_type = df['User Type'].value\_counts()

**# Count gender**

gender = df['Gender'].value\_counts()

**# Earliest and most common year of birth**

earliest\_year = df.sort\_values('Birth Year').iloc[0]

common\_year = df['Birth Year'].mode() [0]

**another solution**

df =df.sort\_values(by='birth year',descending=True)

return. df

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

    earliest\_birthdate = df['Birth Year'].min()

    most\_recent\_birthdate = df['Birth Year'].max()

    most\_common\_birthdate = df['Birth Year'].mode()

print('Count of user types: ', user\_type)

print('Count of gender: ', gender)

print('Oldest person to rent: ', earliest\_year['Birth Year'])

print('Most common birth year: ', common\_year)

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

 print('-'\*40)

**# Additional data**

def display\_data(df):

view\_data = input('Would you like to see the next five rows of data? Yes or No: ')

view\_data = view\_data.lower()

if view\_data == 'yes':

print(df.head(5))

display\_data(df)

else:

exit(print("Thank you for Exploring US Bikeshare Data"))

# Main Funtion - Call all the funtions

def main() -> object:

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)

display\_data(df)

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

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

break

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

main()

**these links that I used and there are a lot that used**

<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.sort_values.html>

<https://www.javatpoint.com/python-filter-function>

<https://www.geeksforgeeks.org/python-pandas-dataframe-describe-method/>

<https://www.tutorialspoint.com/python_pandas/python_pandas_descriptive_statistics.htm>

<https://review.udacity.com/#!/rubrics/1379/view>

<https://www.motivateco.com/>