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

"""

Created on Sun Feb 25 21:55:00 2018

@author: Goodwill Mumvenge

"

"""

##There are several approaches to this, but i will present only two

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

import csv

#load data file using pandas

data\_file= pd.read\_csv('myscript.csv',delimiter=',')

Gender, Age, Spend = [], [], []# headings of the data

print (data\_file[:,0])# this prints all the data

#Alternatively, we can load data using numpy, though pandas is faster than numpy

data\_file= np.loadtxt('myscript.txt', delimiter=',')# just an alternative

Gender, Age, Spend = [], [], []# headings of the data

print (data\_file[:,0])# this prints all the data

Gender\_col= data\_file['Gender']

print(Gender\_col)# This prints the gender column only

#Test data type of each row

Age= data\_file['Age']

print(Age[:])# Print all ages

avg\_age\_col= np.mean(Age)# Calculates the average age of all buyers

max\_age\_col= np.max(Age)#calculates the maximum age of buyers

min\_age\_col= np.min(Age)#calculates the minimum ageof buyers

print(avg\_age\_col)

print(max\_age\_col)

print(min\_age\_col)

Spend= data\_file['Spend']

print(Spend[:])

avg\_spend\_col= np.mean(Spend)# Calcultes the average Spend of all buyers

max\_spend\_col= np.max(Spend)#calculates the maximum spend value

min\_spend\_col= np.min(Spend)#calculates the minimum spendvalue

print(avg\_spend\_col)

print(max\_spend\_col)

print(min\_spend\_col)

#having calculated the average, i now proceed to create a new table with above average spenders

my\_data= [Age<=90,Spend>=avg\_spend\_col, Gender\_col]

result= pd.concat(my\_data,axis=1)

print(result)

plt.scatter(Age, Spend,color='r')#Plot the scatter graph

plt.scatter(Gender\_col, Spend,color='b')

plt.show()

#I can import data from an SQL file and run queries in python as follows

import sqlite3

my\_data2= sqlite3.connect("C:\\Users\\goodwill\\Desktop\\buyers.db")

query= "SELECT\* FROM buyers;"

results= pd.read\_sql(query, my\_data2)

print (results.head())

#we can then run the usual SQL queries and manipulate the data

### IF THE DATA IS TOO BIG, WE CAN USE THE FOLLOWING CODE

def getstuff(filename, criterion):

with open(filename, "rb") as csvfile:

datareader = csv.reader(csvfile)

count = 0

for row in datareader:

if row[3] in ("column header", criterion):

yield row

count += 1

elif count < 2:

continue

else:

return

def getdata(filename, criteria):

for criterion in criteria:

for row in getstuff(filename, criterion):

yield row

##if the data is too big, we can also process it in chunks as follows

file\_name= ('C:/Users/Goodwill Mumvenge/desktop/myscript.csv')

chunksize = 10 \*\* 10

for chunk in pd.read\_csv(file\_name.csv, chunksize=chunksize):

print (chunk[0:2])# This will print two the chunks of similar size