```
#Georgia Sugisandhea/535230080
#Mengimport data dari google drive
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
#Untuk membaca dataset yang sudah di import
import numpy as np
```

```
#Mendefinisikan path data yang dipakai
data_path = "/content/drive/MyDrive/Big Data/Praktikum 02/ClassicCars.csv"
```

```
#Mendefinisikan tipe tipe data yang dimasukkan karena numpy tidak dapat membaca file dengan tipe data yang berbeda
types = ['U20', 'U10', 'U5', 'U20', 'U3', 'f4', 'f4', 'f4', 'f4', 'U10', 'i4', 'i4', 'i4', 'i4', 'i4']

#Setelah di definisikan, numpy membaca dataset
data = np.genfromtxt(data_path, dtype=types, delimiter=',', names=True)
```

```
#Mencantumkan path data file yang digunakan
data_path = "Praktikum02/ClassicCars.csv"
```

```
#Untuk mengambil nama nama kolom dari dataset
data.dtype.names

#Untuk mengambil bentuk array dan ukuran dari dataset.
#Output dari coding ini menunjukkan bahwa ada 205 entri informasi data dalam dataset tersebut.
print("The Shape of the array ", data.shape)
```

```
The Shape of the array  (205,)
```

```
#Untuk output data-data di kolom yang bernama 'make'
print(data['make'])

#Untuk mengambil data-data yang ada di kolom bernama 'bodystyle'
data['bodystyle']
```

```
 'isuzu' 'isuzu' 'isuzu' 'isuzu' 'jaguar' 'jaguar' 'jaguar' 'mazda'
 'mazda' 'mazda' 'mazda' 'mazda' 'mazda' 'mazda' 'mazda' 'mazda' 'mazda'
 'mazda' 'mazda' 'mazda' 'mazda' 'mazda' 'mazda' 'mazda' 'mercedes-benz'
 'mercedes-benz' 'mercedes-benz' 'mercedes-benz' 'mercedes-benz'
 'mercedes-benz' 'mercedes-benz' 'mercedes-benz' 'mercury' 'mitsubishi'
 'mitsubishi' 'mitsubishi' 'mitsubishi' 'mitsubishi' 'mitsubishi'
 'mitsubishi' 'mitsubishi' 'mitsubishi' 'mitsubishi' 'mitsubishi'
 'mitsubishi' 'mitsubishi' 'nissan' 'nissan' 'nissan' 'nissan' 'nissan'
 'nissan' 'nissan' 'nissan' 'nissan' 'nissan' 'nissan' 'nissan' 'nissan'
 'nissan' 'nissan' 'nissan' 'nissan' 'nissan' 'peugot' 'peugot' 'peugot'
 'peugot' 'peugot' 'peugot' 'peugot' 'peugot' 'peugot' 'peugot' 'peugot'
 'plymouth' 'plymouth' 'plymouth' 'plymouth' 'plymouth' 'plymouth'
 'plymouth' 'porsche' 'porsche' 'porsche' 'porsche' 'porsche' 'renault'
 'renault' 'saab' 'saab' 'saab' 'saab' 'saab' 'saab' 'subaru' 'subaru'
 'subaru' 'subaru' 'subaru' 'subaru' 'subaru' 'subaru' 'subaru' 'subaru'
 'subaru' 'subaru' 'toyota' 'toyota' 'toyota' 'toyota' 'toyota' 'toyota'
 'toyota' 'toyota' 'toyota' 'toyota' 'toyota' 'toyota' 'toyota' 'toyota'
 'toyota' 'toyota' 'toyota' 'toyota' 'toyota' 'toyota' 'toyota' 'toyota'
 'toyota' 'toyota' 'toyota' 'toyota' 'toyota' 'toyota' 'toyota' 'toyota'
 'toyota' 'toyota' 'volkswagen' 'volkswagen' 'volkswagen' 'volkswagen'
 'volkswagen' 'volkswagen' 'volkswagen' 'volkswagen' 'volkswagen'
 'volkswagen' 'volkswagen' 'volkswagen' 'volvo' 'volvo' 'volvo' 'volvo'
 'volvo' 'volvo' 'volvo' 'volvo' 'volvo' 'volvo' 'volvo']
array(['convertible', 'convertible', 'hatchback', 'sedan', 'sedan',
       'sedan', 'sedan', 'wagon', 'sedan', 'hatchback', 'sedan', 'sedan',
       'sedan', 'sedan', 'sedan', 'sedan', 'sedan', 'sedan', 'hatchback',
       'hatchback', 'sedan', 'hatchback', 'hatchback', 'hatchback',
       'hatchback', 'sedan', 'sedan', 'sedan', 'wagon', 'hatchback',
       'hatchback', 'hatchback', 'hatchback', 'hatchback', 'hatchback',
       'sedan', 'wagon', 'hatchback', 'hatchback', 'sedan', 'sedan',
       'sedan', 'sedan', 'sedan', 'sedan', 'sedan', 'hatchback', 'sedan',
       'sedan', 'sedan', 'hatchback', 'hatchback', 'hatchback', 'sedan',
       'sedan', 'hatchback', 'hatchback', 'hatchback', 'hatchback',
```

```
       'sedan', 'sedan', 'sedan', 'sedan', 'sedan', 'wagon', 'sedan',
       'hatchback', 'sedan', 'wagon', 'hardtop', 'hatchback', 'sedan',
       'sedan', 'wagon', 'sedan', 'hatchback', 'hatchback', 'hatchback',
       'sedan', 'sedan', 'wagon', 'wagon', 'sedan', 'sedan', 'wagon',
       'wagon', 'sedan', 'sedan', 'sedan', 'hatchback', 'hatchback',
       'hatchback', 'sedan', 'sedan', 'wagon', 'hatchback', 'hatchback',
       'hardtop', 'hardtop', 'convertible', 'hatchback', 'wagon',
       'hatchback', 'hatchback', 'sedan', 'hatchback', 'sedan',
       'hatchback', 'sedan', 'hatchback', 'hatchback', 'hatchback',
       'sedan', 'sedan', 'sedan', 'sedan', 'sedan', 'wagon', 'wagon',
       'wagon', 'wagon', 'hatchback', 'hatchback', 'hatchback', 'wagon',
       'wagon', 'wagon', 'sedan', 'hatchback', 'sedan', 'hatchback',
       'sedan', 'hatchback', 'sedan', 'sedan', 'hatchback', 'sedan',
       'hatchback', 'hardtop', 'hardtop', 'hatchback', 'hardtop',
       'hatchback', 'convertible', 'sedan', 'sedan', 'hatchback', 'sedan',
       'hatchback', 'hatchback', 'hatchback', 'sedan', 'wagon', 'sedan',
       'sedan', 'sedan', 'sedan', 'sedan', 'sedan', 'sedan',
       'convertible', 'hatchback', 'sedan', 'sedan', 'wagon', 'sedan',
       'wagon', 'sedan', 'wagon', 'sedan', 'wagon', 'sedan', 'sedan',
       'sedan', 'sedan', 'sedan'], dtype='<U20')
```

```
#Mencari data terkecil dari data-data di kolom bernama 'price'
#dan memasukkan ke variable min
min=np.min(data['price'])

#Mencari data terbesar dari data-data di kolom bernama 'price'
#dan memasukkan ke variable max
max=np.max(data['price'])

print('Min car price:', min) #Mencetak variable min
print('Max xar price:', max) #Mencetak variable max
print('Range: ', max-min) #Mencetak selisih data terbesar dan terkecil dengan menguranginya
```

```
Min car price: 5118
Max xar price: 45400
Range:  40282
```

```
#Mencetak data terkecil dari data kolom bernama 'price' dengan function numpy np.min
print("Min car price:", np.min(data['price']))

#Mencetak data terbesar dari data kolom bernama 'price' dengan function numpy np.max
print("Max car price:", np.max(data['price']))

#Mencetak rata-rata/mean dari data kolom bernama 'price' dengan function numpy np.mean
print("Mean car price:", np.mean(data['price']))

#Mencetak media dari data kolom bernama 'price' dengan function numpy np.media
print("Median car price:", np.median(data['price']))
```

```
Min car price: 5118
Max car price: 45400
Mean car price: 13300.239024390245
Median car price: 10345.0
```

```
#Menggunakan numpy functions untuk membuat fungsi untuk menghitung standar deviasi
def stdUsingNumpyOnly(prices):
  return np.sqrt(np.sum(np.power(np.subtract(prices, np.mean(prices)),2)/len(prices)))


#Menggunakan numpy functions untuk membuat fungsi untuk menghitung standar deviasi dengan cara lain
def stdImplementation(prices):
  meanPrice = np.mean(prices)
  priceDiffSq = [np.power(price-meanPrice, 2) for price in prices]
  priceDiffAvg = np.sum(priceDiffSq)/len(prices)
  return np.sqrt(priceDiffAvg)


print("Standard deviation using only numpy functions: ", stdUsingNumpyOnly(data['price']))
print("Standard deviation by implementing std function: ", stdImplementation(data['price']))
print("Standard deviation using Numpy's std function:", np.std(data['price']))
```

```
Standard deviation using only numpy functions:  7969.54140103854
Standard deviation by implementing std function:  7969.54140103854
Standard deviation using Numpy's std function: 7969.54140103854
```

```
#Cara menghitung volume mobil dalam data set mobil yang digunakan
carsVolume = np.multiply(np.multiply(data['length'], data['height']), data['width'])
#Cara ke2
carsVolume = data['length']* data['height']* data['width']
#Cara ke3
carsVolume = np.prod(np.vstack([data['length'], data['height'], data['width']]), axis=0)

#Dari hasil penghitungan volume mobil tersebut, mencari volume mobil yang paling maksimal
maxVolume = np.max(carsVolume)

#dan juga volume yang terkecil
minVolume = np.min(carsVolume)

#Mencari mobil yang punya volume terbesar dengan maxVolume
carWithMaxVolume = np.argmax(carsVolume)

#Mencari mobil yang punya volume terkecil dengan minVolume
carWithMinVolume = np.argmin(carsVolume)

#Mencetak hasil mobil dengan volume terbesar dan volumenya
print("Max volume:", maxVolume, " belongs to car ", data[carWithMaxVolume])

#Mencetak hasil mobil dengan volume terkecil dan volumenya
print("Min volume:", minVolume, " belongs to car ", data[carWithMinVolume])
```

```
Max volume: 846007.7  belongs to car  ('mercedes-benz', 'gas', 'four', 'sedan', 'rwd', 120.9, 208.1, 71.7, 56.7, 'eight', 308, 184, 14,
Min volume: 452643.2  belongs to car  ('chevrolet', 'gas', 'two', 'hatchback', 'fwd', 88.4, 141.1, 60.3, 53.2, 'three', 61, 48, 47, 53,
```

```
#Cara mengkali 2 value dengan function numpy np.multiply
par1 = 10
par2 = 2
np.multiply(par1, par2)
```

```
20
```

```
#Untuk mencetak banyaknya jenis bodystyles yang ada dalam dataset yang dipakai
print("Unique bodystyles: ", np.unique(data['bodystyle']))
```

```
Unique bodystyles:  ['convertible' 'hardtop' 'hatchback' 'sedan' 'wagon']
```

```
#Untuk mengambil banyaknya merek mobil yang ada di dataset dengan kolom "make"
uniqueCarMakes = np.unique(data['make'])

#dan mencetak banyaknya data dalam variable uniqueCarMakes yang isinya merek mobil yang ada dan apa saja
print("There are ", len(uniqueCarMakes), "unique car makes which are:", uniqueCarMakes)
```

```
There are  22 unique car makes which are: ['alfa-romero' 'audi' 'bmw' 'chevrolet' 'dodge' 'honda' 'isuzu' 'jaguar'
 'mazda' 'mercedes-benz' 'mercury' 'mitsubishi' 'nissan' 'peugot'
 'plymouth' 'porsche' 'renault' 'saab' 'subaru' 'toyota' 'volkswagen'
 'volvo']
```

```
#untuk mengurangi data di kolom "highwaympg" dan "citympg" di masing masing baris dan di assign ke fuelDiff
fuelDiff = np.subtract(data['highwaympg'], data['citympg'])

#mencari data fuelDiff yang paling kecil
minFuelDiff = np.min(fuelDiff)

#mencari data fuelDiff yang paling besar
maxFuelDiff = np.max(fuelDiff)

#mencari mobil yang punya fuelDiff paling kecil
carWithMinFuelDiff = np.argmin(fuelDiff)

#mencari mobil yang punya fuelDiff yang paling besar
carWithMaxFuelDiff = np.argmax(fuelDiff)

#mencetak data merek mobil, enginesize mobil, horsepower mobil, dan fuelDifference, untuk mobil yang fuelDifference terkecil...
print("A %s with engine size=%d and horsepower=%d has the minimum fuel difference (%d) when driven in the city and the highway"
      % (data['make'][carWithMinFuelDiff], data["enginesize"][carWithMinFuelDiff], data['horsepower'][carWithMinFuelDiff], minFuelDiff))

#dan juga terbesar
print("A %s with engine size=%d and horsepower=%d has the minimum fuel difference (%d) when driven in the city and the highway"
      % (data['make'][carWithMaxFuelDiff], data["enginesize"][carWithMaxFuelDiff], data['horsepower'][carWithMaxFuelDiff], maxFuelDiff))
```

```
A peugot with engine size=152 and horsepower=95 has the minimum fuel difference (0) when driven in the city and the highway
A porsche with engine size=203 and horsepower=288 has the minimum fuel difference (11) when driven in the city and the highway
```

```
#mencari jenis jenis merek dari kolom data "make" dan mencari mana yang memiliki jumlah mobil terbanyak
makes,counts = np.unique(data['make'], return_counts=True)
maxCarsSameMake = np.argmax(counts)
make = makes[maxCarsSameMake]

#mencetak hasil merek yang memiliki mobil terbanyak dan jumlahnya
print("The company has %d %scars " % (np.max(counts), make))
```

```
The company has 32 toyotacars
```

```
#Mencari mobil yang termasuk kedalam mobil dengan base ban yang besar yaitu lebih dari 100
carsWithLargeWheelBase = np.count_nonzero(data['wheelbase']>100)

#mencetak hasil pencarian tersebut
print("There are %d cars whose wheel base is greater than 100" %(carsWithLargeWheelBase))

#Mencari mobil yang memiliki base ban lebih kecil dari 88.6
np.count_nonzero(data['wheelbase']<88.6)

#Mencari mobil yang memiliki base ban sebesar 110
np.count_nonzero(data['wheelbase']==110)
```

```
There are 63 cars whose wheel base is greater than 100
3
```

```
#Mencari mobil yang punya bodystyle berupa convertible dan berharga kurang dari <£15000
cheapConvertibles = data[(data['bodystyle']=="convertible") & (data['price']<15000)]

#Mencetak hasil pencarian
print("Details of convertible that cost less than £15000:\n", cheapConvertibles)
```

```
Details of convertible that cost less than £15000:
 [('alfa-romero', 'gas', 'two', 'convertible', 'rwd', 88.6, 168.8, 64.1, 48.8, 'four', 130, 111, 21, 27, 13495)
 ('volkswagen', 'gas', 'two', 'convertible', 'fwd', 94.5, 159.3, 64.2, 55.6, 'four', 109,  90, 24, 29, 11595)]
```

```
#Mencetak rentang interkuartil dari semua data harga di dataset
Q3P = np.percentile(data['price'], 75) #Mencari kuartil ke 3 terlebih dahulu
Q1P = np.percentile(data['price'], 25) #Mencari kuartil pertama
IQRP = Q3P - Q1P #Mengurangi kuartil ke 3 dan 1 untuk mendapatkan rentang interkuartil
print('Price IQR', IQRP) #Mencetak hasil perhitungan
```

```
Price IQR 8715.0
```

```
#Perhitungan percentile range ke 50 untuk data horsepower dari data semua mobil yang memiliki nilai yang sama dengan kuartil 50
percentile50HP = np.percentile(data['horsepower'], 50)

print('Horsepower 50th percentile:', percentile50HP) #Mencetak hasil perhitungan
```