MPG Cars

Introduction:

The following exercise utilizes data from UC Irvine Machine Learning Repository

Step 1. Import the necessary libraries

In [1]:
```
import pandas as pd
import numpy as np
```

Step 2. Import the first dataset cars1 and cars2.

Step 3. Assign each to a variable called cars1 and cars2

In [2]:
```
cars1 = pd.read_csv("https://raw.githubusercontent.com/guipsamora/pandas_exercises/master/05_Merge/Auto_MPG/cars1.csv")
cars2 = pd.read_csv("https://raw.githubusercontent.com/guipsamora/pandas_exercises/master/05_Merge/Auto_MPG/cars2.csv")

cars1.head()
```

Out[2]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model	origin	car	Unnamed: 9	Unnamed: 10	Unnamed: 11	Unnamed: 12	Unnamed: 13
0	18.0	8	307	130	3504	12.0	70	1	chevrolet chevelle malibu	NaN	NaN	NaN	NaN	NaN
1	15.0	8	350	165	3693	11.5	70	1	buick skylark 320	NaN	NaN	NaN	NaN	NaN
2	18.0	8	318	150	3436	11.0	70	1	plymouth satellite	NaN	NaN	NaN	NaN	NaN
3	16.0	8	304	150	3433	12.0	70	1	amc rebel sst	NaN	NaN	NaN	NaN	NaN
4	17.0	8	302	140	3449	10.5	70	1	ford torino	NaN	NaN	NaN	NaN	NaN

In [3]:
```
cars2.head()
```

Out[3]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model	origin	car
0	33.0	4	91	53	1795	17.4	76	3	honda civic
1	20.0	6	225	100	3651	17.7	76	1	dodge aspen se
2	18.0	6	250	78	3574	21.0	76	1	ford granada ghia
3	18.5	6	250	110	3645	16.2	76	1	pontiac ventura sj
4	17.5	6	258	95	3193	17.8	76	1	amc pacer d/l

Step 4. Oops, it seems our first dataset has some unnamed blank columns, fix cars1

In [4]:
```
cars1 = cars1.loc[:, "mpg":"car"]
cars1
```

Out[4]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model	origin	car
0	18.0	8	307	130	3504	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350	165	3693	11.5	70	1	buick skylark 320
2	18.0	8	318	150	3436	11.0	70	1	plymouth satellite
3	16.0	8	304	150	3433	12.0	70	1	amc rebel sst
4	17.0	8	302	140	3449	10.5	70	1	ford torino
...	...	...	...	...	...	...	...	...	...
193	24.0	6	200	81	3012	17.6	76	1	ford maverick
194	22.5	6	232	90	3085	17.6	76	1	amc hornet
195	29.0	4	85	52	2035	22.2	76	1	chevrolet chevette
196	24.5	4	98	60	2164	22.1	76	1	chevrolet woody
197	29.0	4	90	70	1937	14.2	76	2	vw rabbit

198 rows × 9 columns

Step 5. What is the number of observations in each dataset?

In [5]:
```
print(cars1.shape)
print(cars2.shape)
```

```
(198, 9)
(200, 9)
```

Step 6. Join cars1 and cars2 into a single DataFrame called cars

In [6]:
```
cars = cars1.append(cars2)
cars
```

Out[6]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model	origin	car
0	18.0	8	307	130	3504	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350	165	3693	11.5	70	1	buick skylark 320
2	18.0	8	318	150	3436	11.0	70	1	plymouth satellite
3	16.0	8	304	150	3433	12.0	70	1	amc rebel sst
4	17.0	8	302	140	3449	10.5	70	1	ford torino
...	...	...	...	...	...	...	...	...	...
195	27.0	4	140	86	2790	15.6	82	1	ford mustang gl
196	44.0	4	97	52	2130	24.6	82	2	vw pickup
197	32.0	4	135	84	2295	11.6	82	1	dodge rampage
198	28.0	4	120	79	2625	18.6	82	1	ford ranger
199	31.0	4	119	82	2720	19.4	82	1	chevy s-10

398 rows × 9 columns

Step 7. Oops, there is a column missing, called owners. Create a random number Series from 15,000 to 73,000.

In [7]:
```
nr_owners = np.random.randint(15000, high=73001, size=398, dtype='l')
nr_owners
```

Out[7]:
```
array([30525, 42200, 55227, 58546, 68479, 46555, 63460, 42288, 61176,
       37464, 45030, 67503, 55131, 52144, 72258, 71430, 16610, 37934,
       67884, 58826, 63500, 42924, 48257, 47405, 41088, 45705, 68985,
       38555, 40859, 35781, 36360, 67756, 55028, 51200, 43203, 70239,
       52881, 64663, 26067, 22688, 59333, 48269, 30234, 70102, 47950,
       54315, 16257, 33162, 30436, 35288, 67851, 71727, 52205, 35253,
       61291, 35355, 21343, 67010, 63737, 20988, 48947, 69145, 29869,
       43263, 26777, 59125, 42520, 63205, 56326, 44769, 32521, 27970,
       48779, 16393, 63834, 47875, 27367, 66531, 25074, 42942, 15665,
       22971, 39009, 56406, 71838, 57182, 30622, 60486, 45810, 16994,
       15430, 17746, 20102, 27589, 66335, 25595, 67786, 53674, 33959,
       62957, 23499, 21548, 69628, 52190, 19871, 66070, 34409, 30368,
       43501, 37014, 69691, 44348, 17814, 39261, 49885, 36171, 19278,
       37278, 72956, 43058, 56212, 71035, 55101, 23662, 27675, 53404,
       40951, 63214, 18621, 15077, 53333, 64701, 16483, 28733, 61420,
       47366, 56745, 40398, 16291, 33459, 46483, 64363, 47703, 64544,
       22980, 53993, 33816, 43136, 27874, 25422, 36899, 36808, 54998,
       48358, 61363, 15820, 68581, 59561, 62922, 19290, 18639, 48391,
       50405, 18203, 54520, 27092, 56496, 53918, 28083, 30197, 30899,
       42681, 62732, 40756, 39525, 63504, 30436, 59804, 62877, 52349,
       52454, 21086, 55297, 57027, 33410, 59281, 45312, 64095, 68049,
       46302, 23832, 25796, 16058, 45103, 50340, 49184, 19974, 40705,
       68561, 16044, 35949, 47261, 60579, 16486, 39078, 69061, 36035,
       37282, 36859, 62692, 24110, 51166, 19563, 64164, 66293, 24229,
       29934, 52496, 57916, 44853, 35516, 32617, 66332, 56322, 66039,
       63013, 28931, 66425, 24384, 60857, 21401, 46637, 15167, 37332,
       45438, 19034, 24098, 59555, 63354, 53302, 44282, 51849, 19107,
       50255, 60344, 65608, 63229, 57272, 71264, 50406, 50199, 26468,
       51046, 37222, 57078, 53948, 63884, 15677, 31173, 42470, 40349,
       56142, 45969, 50042, 56565, 52865, 57004, 69079, 22312, 17581,
       66851, 51147, 50660, 58206, 26604, 23101, 60552, 23371, 37379,
       25401, 54923, 66203, 37527, 15818, 60948, 59789, 33304, 47421,
       22646, 42928, 45181, 15047, 41861, 58976, 68103, 41195, 40596,
       55550, 22802, 39413, 24785, 33281, 49832, 27486, 63285, 72272,
       34238, 41348, 69021, 49235, 20510, 47405, 38065, 71659, 26608,
       19891, 22993, 43212, 34738, 57478, 61121, 27454, 40186, 20705,
       21460, 32600, 19277, 42862, 60248, 61034, 38099, 41750, 49126,
       40546, 46570, 34813, 64566, 53986, 53550, 72306, 54176, 18322,
       32830, 43438, 57022, 61889, 38390, 66215, 25410, 18928, 35794,
       48181, 27805, 52873, 40365, 60161, 46237, 50684, 54350, 36443,
       45699, 45597, 25387, 50467, 72954, 59692, 68554, 32890, 45049,
       66624, 44144, 25430, 63316, 32179, 22055, 33237, 20135, 61799,
       54699, 34802, 59057, 37513, 42198, 70890, 34531, 56901, 21020,
       59803, 26486, 71352, 38207, 45772, 71968, 15853, 53383, 19043,
       69183, 70301])
```

Step 8. Add the column owners to cars

In [8]:
```
cars['owners'] = nr_owners
cars.tail()
```

Out[8]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model	origin	car	owners
195	27.0	4	140	86	2790	15.6	82	1	ford mustang gl	15853
196	44.0	4	97	52	2130	24.6	82	2	vw pickup	53383
197	32.0	4	135	84	2295	11.6	82	1	dodge rampage	19043
198	28.0	4	120	79	2625	18.6	82	1	ford ranger	69183
199	31.0	4	119	82	2720	19.4	82	1	chevy s-10	70301

In [ ]:
```

```