Perform the following operation on Autompg.csv of XYZ Custom Cars company using Pandas

In [4]:
```
#Importing libraries
import numpy as np
import pandas as pd
```

Read data from an existing file

In [10]:
```
import pandas as pd
import numpy as np
df = pd.read_csv('auto_mpg.csv')
df.head()
```

Out[10]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name
0	18.0	8	307.0	130.0	3504	12.0	70	usa	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693	11.5	70	usa	buick skylark 320
2	18.0	8	318.0	150.0	3436	11.0	70	usa	plymouth satellite
3	16.0	8	304.0	150.0	3433	12.0	70	usa	amc rebel sst
4	17.0	8	302.0	140.0	3449	10.5	70	usa	ford torino

Engineers at XYZ Custom Cars want to know how many cars are Fuel efficient

MPG > 29, Horsepower < 93.5,

Weight < 2500

In [7]:
```
df.loc[(df['mpg'] > 29) & (df['horsepower'] < 93.5) & (df['weight'] < 2500)]
```

Out[7]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name
51	30.0	4	79.0	70.0	2074	19.5	71	europe	peugeot 304
52	30.0	4	88.0	76.0	2065	14.5	71	europe	fiat 124b
53	31.0	4	71.0	65.0	1773	19.0	71	japan	toyota corolla 1200
54	35.0	4	72.0	69.0	1613	18.0	71	japan	datsun 1200
129	31.0	4	79.0	67.0	1950	19.0	74	japan	datsun b210
...	...	...	...	...	...	...	...	...	...
384	32.0	4	91.0	67.0	1965	15.7	82	japan	honda civic (auto)
385	38.0	4	91.0	67.0	1995	16.2	82	japan	datsun 310 gx
391	36.0	4	135.0	84.0	2370	13.0	82	usa	dodge charger 2.2
394	44.0	4	97.0	52.0	2130	24.6	82	europe	vw pickup
395	32.0	4	135.0	84.0	2295	11.6	82	usa	dodge rampage

81 rows × 9 columns

Engineers at XYZ Custom Cars want to know how many cars are Muscle cars

Displacement >262, Horsepower > 126, Weight in range[2800, 3600]

In [8]:
```
df.loc[(df['displacement'] > 262) & (df['horsepower'] > 126) & (df['weight'] >=2800) & (df['weight'] <= 3600)]
```

Out[8]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name
0	18.0	8	307.0	130.0	3504	12.0	70	usa	chevrolet chevelle malibu
2	18.0	8	318.0	150.0	3436	11.0	70	usa	plymouth satellite
3	16.0	8	304.0	150.0	3433	12.0	70	usa	amc rebel sst
4	17.0	8	302.0	140.0	3449	10.5	70	usa	ford torino
10	15.0	8	383.0	170.0	3563	10.0	70	usa	dodge challenger se
13	14.0	8	455.0	225.0	3086	10.0	70	usa	buick estate wagon (sw)
121	15.0	8	318.0	150.0	3399	11.0	73	usa	dodge dart custom
166	13.0	8	302.0	129.0	3169	12.0	75	usa	ford mustang ii
251	20.2	8	302.0	139.0	3570	12.8	78	usa	mercury monarch ghia
262	19.2	8	305.0	145.0	3425	13.2	78	usa	chevrolet monte carlo landau
264	18.1	8	302.0	139.0	3205	11.2	78	usa	ford futura

Engineers at XYZ Custom Cars want to know how many cars are SUVs

Horsepower > 140 , Weight > 4500

In [9]:
```
df.loc[(df['horsepower'] > 140) & (df['weight'] >=4500)]
```

Out[9]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name
25	10.0	8	360.0	215.0	4615	14.0	70	usa	ford f250
28	9.0	8	304.0	193.0	4732	18.5	70	usa	hi 1200d
42	12.0	8	383.0	180.0	4955	11.5	71	usa	dodge monaco (sw)
43	13.0	8	400.0	170.0	4746	12.0	71	usa	ford country squire (sw)
44	13.0	8	400.0	175.0	5140	12.0	71	usa	pontiac safari (sw)
67	11.0	8	429.0	208.0	4633	11.0	72	usa	mercury marquis
68	13.0	8	350.0	155.0	4502	13.5	72	usa	buick lesabre custom
90	12.0	8	429.0	198.0	4952	11.5	73	usa	mercury marquis brougham
94	13.0	8	440.0	215.0	4735	11.0	73	usa	chrysler new yorker brougham
95	12.0	8	455.0	225.0	4951	11.0	73	usa	buick electra 225 custom
103	11.0	8	400.0	150.0	4997	14.0	73	usa	chevrolet impala
104	12.0	8	400.0	167.0	4906	12.5	73	usa	ford country
105	13.0	8	360.0	170.0	4654	13.0	73	usa	plymouth custom suburb
137	13.0	8	350.0	150.0	4699	14.5	74	usa	buick century luxus (sw)
156	16.0	8	400.0	170.0	4668	11.5	75	usa	pontiac catalina
159	14.0	8	351.0	148.0	4657	13.5	75	usa	ford ltd

Engineers at XYZ Custom Cars want to know how many cars are Racecars

Weight <2223, acceleration > 17

In [61]:
```
df.loc[(df['acceleration'] > 17) & (df['weight'] < 2223)]
```

Out[61]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name
19	26.0	4	97.0	46.0	1835	20.5	70	europe	volkswagen 1131 deluxe sedan
51	30.0	4	79.0	70.0	2074	19.5	71	europe	peugeot 304
53	31.0	4	71.0	65.0	1773	19.0	71	japan	toyota corolla 1200
54	35.0	4	72.0	69.0	1613	18.0	71	japan	datsun 1200
55	27.0	4	97.0	60.0	1834	19.0	71	europe	volkswagen model 111
56	26.0	4	91.0	70.0	1955	20.5	71	usa	plymouth cricket
79	26.0	4	96.0	69.0	2189	18.0	72	europe	renault 12 (sw)
102	26.0	4	97.0	46.0	1950	21.0	73	europe	volkswagen super beetle
117	29.0	4	68.0	49.0	1867	19.5	73	europe	fiat 128
129	31.0	4	79.0	67.0	1950	19.0	74	japan	datsun b210
131	32.0	4	71.0	65.0	1836	21.0	74	japan	toyota corolla 1200
145	32.0	4	83.0	61.0	2003	19.0	74	japan	datsun 710
181	33.0	4	91.0	53.0	1795	17.5	75	japan	honda civic cvcc
195	29.0	4	85.0	52.0	2035	22.2	76	usa	chevrolet chevette
196	24.5	4	98.0	60.0	2164	22.1	76	usa	chevrolet woody
198	33.0	4	91.0	53.0	1795	17.4	76	japan	honda civic
216	31.5	4	98.0	68.0	2045	18.5	77	japan	honda accord cvcc
218	36.0	4	79.0	58.0	1825	18.6	77	europe	renault 5 gtl
244	43.1	4	90.0	48.0	1985	21.5	78	europe	volkswagen rabbit custom diesel
246	32.8	4	78.0	52.0	1985	19.4	78	japan	mazda glc deluxe
247	39.4	4	85.0	70.0	2070	18.6	78	japan	datsun b210 gx
303	31.8	4	85.0	65.0	2020	19.2	79	japan	datsun 210
310	38.1	4	89.0	60.0	1968	18.8	80	japan	toyota corolla tercel
322	46.6	4	86.0	65.0	2110	17.9	80	japan	mazda glc
324	40.8	4	85.0	65.0	2110	19.2	80	japan	datsun 210
325	44.3	4	90.0	48.0	2085	21.7	80	europe	vw rabbit c (diesel)
331	33.8	4	97.0	67.0	2145	18.0	80	japan	subaru dl
346	32.3	4	97.0	67.0	2065	17.8	81	japan	subaru
347	37.0	4	85.0	65.0	1975	19.4	81	japan	datsun 210 mpg
348	37.7	4	89.0	62.0	2050	17.3	81	japan	toyota tercel
376	37.0	4	91.0	68.0	2025	18.2	82	japan	mazda glc custom l
377	31.0	4	91.0	68.0	1970	17.6	82	japan	mazda glc custom
379	36.0	4	98.0	70.0	2125	17.3	82	usa	mercury lynx l
394	44.0	4	97.0	52.0	2130	24.6	82	europe	vw pickup

XYZ Custom cars want the data sorted according to the number of cylinders.

In [63]:
```
df.sort_values(by = 'cylinders')
```

Out[63]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name
243	21.5	3	80.0	110.0	2720	13.5	77	japan	mazda rx-4
334	23.7	3	70.0	100.0	2420	12.5	80	japan	mazda rx-7 gs
111	18.0	3	70.0	90.0	2124	13.5	73	japan	maxda rx3
71	19.0	3	70.0	97.0	2330	13.5	72	japan	mazda rx2 coupe
237	30.5	4	98.0	63.0	2051	17.0	77	usa	chevrolet chevette
...	...	...	...	...	...	...	...	...	...
86	14.0	8	304.0	150.0	3672	11.5	73	usa	amc matador
85	13.0	8	350.0	175.0	4100	13.0	73	usa	buick century 350
285	17.0	8	305.0	130.0	3840	15.4	79	usa	chevrolet caprice classic
92	13.0	8	351.0	158.0	4363	13.0	73	usa	ford ltd
0	18.0	8	307.0	130.0	3504	12.0	70	usa	chevrolet chevelle malibu

392 rows × 9 columns

There is a requirement in which the cars that have lowest acceleration must be assessed. It is also to be checked that which cars have higher horsepower despite having lower acceleration.

In [64]:
```
df.sort_values(['acceleration', 'horsepower'], ascending = (1,0))
```

Out[64]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name
11	14.0	8	340.0	160.0	3609	8.0	70	usa	plymouth 'cuda 340
7	14.0	8	440.0	215.0	4312	8.5	70	usa	plymouth fury iii
9	15.0	8	390.0	190.0	3850	8.5	70	usa	amc ambassador dpl
6	14.0	8	454.0	220.0	4354	9.0	70	usa	chevrolet impala
116	16.0	8	400.0	230.0	4278	9.5	73	usa	pontiac grand prix
...	...	...	...	...	...	...	...	...	...
195	29.0	4	85.0	52.0	2035	22.2	76	usa	chevrolet chevette
59	23.0	4	97.0	54.0	2254	23.5	72	europe	volkswagen type 3
326	43.4	4	90.0	48.0	2335	23.7	80	europe	vw dasher (diesel)
394	44.0	4	97.0	52.0	2130	24.6	82	europe	vw pickup
299	27.2	4	141.0	71.0	3190	24.8	79	europe	peugeot 504

392 rows × 9 columns

The board of XYZ custom cars wants to know about minimum and maximum sum, mean and median of all the numerical columns

In [65]:
```
#Using list comprehension to get the numerical columns
list1 = [col for col in df.columns if df[col].dtype in ['float', 'int64']]
df[list1].agg(['min', 'max','sum','mean','median'])
```

Out[65]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year
min	9.000000	3.000000	68.00000	46.000000	1.613000e+03	8.000000	70.000000
max	46.600000	8.000000	455.00000	230.000000	5.140000e+03	24.800000	82.000000
sum	9190.800000	2145.000000	76209.50000	40952.000000	1.167213e+06	6092.200000	29784.000000
mean	23.445918	5.471939	194.41199	104.469388	2.977584e+03	15.541327	75.979592
median	22.750000	4.000000	151.00000	93.500000	2.803500e+03	15.500000	76.000000

XYZ custom cars want to know the number of cars manufactured in each year.

In [72]:
```
df.groupby(['model_year']).count()['horsepower'] #all values will be same, so display any one column
```

Out[72]:
```
model_year
70    29
71    27
72    28
73    40
74    26
75    30
76    34
77    28
78    36
79    29
80    27
81    28
82    30
Name: horsepower, dtype: int64
```

The engineers at XYZ Custom Cars want to know about the relationship between model year and acceleration of cars.

In [74]:
```
df.groupby(['model_year']).mean()[['acceleration']]
```

Out[74]:

	acceleration
model_year	
70	12.948276
71	15.000000
72	15.125000
73	14.312500
74	16.173077
75	16.050000
76	15.941176
77	15.435714
78	15.805556
79	15.813793
80	17.018519
81	16.325000
82	16.510000

The engineers at XYZ Custom Cars want to know the frequency

distribution of different number of cylinders across different years.

In [75]:
```
pd.crosstab(df['model_year'], df['cylinders'])
```

Out[75]:

model_year	3	4	5	6	8
70	0	7	0	4	18
71	0	12	0	8	7
72	1	14	0	0	13
73	1	11	0	8	20
74	0	15	0	6	5
75	0	12	0	12	6
76	0	15	0	10	9
77	1	14	0	5	8
78	0	17	1	12	6
79	0	12	1	6	10
80	1	23	1	2	0
81	0	20	0	7	1
82	0	27	0	3	0

The engineers at XYZ custom cars want to know the mean of all the numerical attributes of cars for each year

In [76]:
```
pivot1 = pd.pivot_table(df, index = 'model_year', aggfunc=np.mean)
pivot1
```

Out[76]:

model_year	acceleration	cylinders	displacement	horsepower	mpg	weight
70	12.948276	6.758621	281.413793	147.827586	17.689655	3372.793103
71	15.000000	5.629630	213.888889	107.037037	21.111111	3030.592593
72	15.125000	5.821429	218.375000	120.178571	18.714286	3237.714286
73	14.312500	6.375000	256.875000	130.475000	17.100000	3419.025000
74	16.173077	5.230769	170.653846	94.230769	22.769231	2878.038462
75	16.050000	5.600000	205.533333	101.066667	20.266667	3176.800000
76	15.941176	5.647059	197.794118	101.117647	21.573529	3078.735294
77	15.435714	5.464286	191.392857	105.071429	23.375000	2997.357143
78	15.805556	5.361111	177.805556	99.694444	24.061111	2861.805556
79	15.813793	5.827586	206.689655	101.206897	25.093103	3055.344828
80	17.018519	4.148148	116.074074	77.481481	33.803704	2441.592593
81	16.325000	4.642857	136.571429	81.035714	30.185714	2530.178571
82	16.510000	4.200000	128.133333	81.466667	32.000000	2434.166667