Python- Pandas Library

Dataset: Motor Trend Car Road Tests 32 observations on 11 (numeric) variables. The data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973--74 models).

Dataset Import

Here we are going to import our dataset.

In [1]:
```
#import pandas library
import pandas as pd
#raed dataset and store into a dataframe
cars=pd.read_csv("mtcars2.csv")
#print
cars
```

Out[1]:

	S.No	Unnamed: 1	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	1	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	2	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	3	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	4	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	5	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
5	6	Valiant	18.1	6	225.0	105	2.76	3.460	NaN	1	0	3	1
6	7	Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
7	8	Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
8	9	Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
9	10	Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
10	11	Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
11	12	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
12	13	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
13	14	Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
14	15	Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
15	16	Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
16	17	Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
17	18	Fiat 128	32.4	4	78.7	66	4.08	2.200	NaN	1	1	4	1
18	19	Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
19	20	Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1

	S.No	Unnamed: 1	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
20	21	Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
21	22	Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
22	23	AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
23	24	Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
24	25	Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
25	26	Fiat X1-9	27.3	4	79.0	66	4.08	1.935	NaN	1	1	4	1
26	27	Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
27	28	Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
28	29	Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
29	30	Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
30	31	Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
31	32	Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

In [58]:
```
#to be removed before recording---to convert the mpg column to str--
cars.mpg = cars.mpg.astype(str)
```

Analysis

After importing the dataset we are all set to perform different analysis.

In [53]:
```
#check the type
type(cars)
```

Out[53]:
```
pandas.core.frame.DataFrame
```

In [3]:
```
#view only the first five records
cars.head()
```

Out[3]:

	S.No	Unnamed: 1	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	1	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	2	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	3	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	4	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	5	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2

In [4]:
```
#view only the first ten records
cars.head(10)
```

Out[4]:

	S.No	Unnamed: 1	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	1	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	2	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	3	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	4	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	5	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
5	6	Valiant	18.1	6	225.0	105	2.76	3.460	NaN	1	0	3	1
6	7	Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
7	8	Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
8	9	Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
9	10	Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4

In [5]:

```
#view only the last five records
cars.tail()
```

Out[5]:

	S.No	Unnamed: 1	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
27	28	Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.9	1	1	5	2
28	29	Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.5	0	1	5	4
29	30	Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.5	0	1	5	6
30	31	Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.6	0	1	5	8
31	32	Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.6	1	1	4	2

In [6]:

```
#view only the last ten records
cars.tail(10)
```

Out[6]:

	S.No	Unnamed: 1	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
22	23	AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
23	24	Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
24	25	Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
25	26	Fiat X1-9	27.3	4	79.0	66	4.08	1.935	NaN	1	1	4	1
26	27	Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
27	28	Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
28	29	Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
29	30	Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
30	31	Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
31	32	Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

In [7]:
```
#view number of rows and columns in the dataframe
cars.shape
```

Out[7]:
```
(32, 13)
```

In [8]:
```
#print a concise summary of the columns
cars.info(null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 13 columns):
S.No          32 non-null int64
Unnamed: 1    32 non-null object
mpg           32 non-null object
cyl           32 non-null int64
disp          32 non-null float64
hp            32 non-null int64
drat          32 non-null float64
wt            32 non-null float64
qsec          29 non-null float64
vs            32 non-null int64
am            32 non-null int64
gear          32 non-null int64
carb          32 non-null int64
dtypes: float64(4), int64(7), object(2)
memory usage: 3.3+ KB
```

In [9]:
```
#mean
cars.mean()
```

Out[9]:
```
S.No     16.500000
cyl       6.187500
disp    230.721875
hp      146.687500
drat      3.596563
wt        3.217250
qsec     17.674828
vs        0.437500
am        0.406250
gear      3.687500
carb      2.812500
dtype: float64
```

In [10]:
```
#median
cars.median()
```

Out[10]:
```
S.No     16.500
mpg      19.200
cyl       6.000
disp    196.300
hp      123.000
drat      3.695
wt        3.325
qsec     17.420
vs        0.000
am        0.000
gear      4.000
```

```
carb      2.000
dtype: float64
```

In [11]:
```
#standard deviation
cars.std()
```

Out[11]:
```
S.No      9.380832
cyl       1.785922
disp    123.938694
hp       68.562868
drat      0.534679
wt        0.978457
qsec      1.780394
vs        0.504016
am        0.498991
gear      0.737804
carb      1.615200
dtype: float64
```

In [12]:
```
#maximum of each attribute
cars.max()
```

Out[12]:
```
S.No                  32
Unnamed: 1    Volvo 142E
mpg                 33.9
cyl                    8
disp                 472
hp                   335
drat                4.93
wt                 5.424
qsec                22.9
vs                     1
am                     1
gear                   5
carb                   8
dtype: object
```

In [13]:
```
#minimum of each attribute
cars.min()
```

Out[13]:
```
S.No                   1
Unnamed: 1    AMC Javelin
mpg                  10.4
cyl                     4
disp                 71.1
hp                     52
drat                 2.76
wt                  1.513
qsec                 14.5
vs                      0
am                      0
gear                    3
carb                    1
dtype: object
```

In [2]:
```
#number of non-null records in each column
cars.count()
```

Out[2]:
```
S.No          32
```

```
Unnamed: 1    32
mpg           32
cyl           32
disp          32
hp            32
drat          32
wt            32
qsec          29
vs            32
am            32
gear          32
carb          32
dtype: int64
```

In [3]:
```
#descriptive statistics summary
cars.describe()
```

Out[3]:

	S.No	mpg	cyl	disp	hp	drat	wt	qsec	
count	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000	29.000000	32.000
mean	16.500000	20.090625	6.187500	230.721875	146.687500	3.596563	3.217250	17.674828	0.437
std	9.380832	6.026948	1.785922	123.938694	68.562868	0.534679	0.978457	1.780394	0.504
min	1.000000	10.400000	4.000000	71.100000	52.000000	2.760000	1.513000	14.500000	0.000
25%	8.750000	15.425000	4.000000	120.825000	96.500000	3.080000	2.581250	16.870000	0.000
50%	16.500000	19.200000	6.000000	196.300000	123.000000	3.695000	3.325000	17.420000	0.000
75%	24.250000	22.800000	8.000000	326.000000	180.000000	3.920000	3.610000	18.600000	1.000
max	32.000000	33.900000	8.000000	472.000000	335.000000	4.930000	5.424000	22.900000	1.000

Cleaning

After analysing we have realized that our dataset is not perfect, some column names are irrelevant, there are a few missing data, some column has important data but cant perform some operation because they are of string type. So now with the help of PAndas we will be cleaning or in other words we will be making our data perfect to perform further operations.

In [62]:
```
#rename column
cars=cars.rename(columns={'Unnamed: 1':'model'})
cars
```

Out[62]:

	S.No	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	1	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	2	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	3	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	4	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	5	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2

	S.No	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
5	6	Valiant	18.1	6	225.0	105	2.76	3.460	NaN	1	0	3	1
6	7	Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
7	8	Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
8	9	Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
9	10	Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
10	11	Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
11	12	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
12	13	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
13	14	Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
14	15	Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
15	16	Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
16	17	Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
17	18	Fiat 128	32.4	4	78.7	66	4.08	2.200	NaN	1	1	4	1
18	19	Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
19	20	Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
20	21	Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
21	22	Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
22	23	AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
23	24	Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
24	25	Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
25	26	Fiat X1-9	27.3	4	79.0	66	4.08	1.935	NaN	1	1	4	1
26	27	Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
27	28	Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
28	29	Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
29	30	Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
30	31	Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
31	32	Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

In [63]:

```
#Fill the null values with mean of the column
cars.qsec=cars.qsec.fillna(cars.qsec.mean())
cars
```

Out[63]:

	S.No	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	1	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.460000	0	1	4	4
1	2	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.020000	0	1	4	4

	S.No	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
2	3	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.610000	1	1	4	1
3	4	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.440000	1	0	3	1
4	5	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.020000	0	0	3	2
5	6	Valiant	18.1	6	225.0	105	2.76	3.460	17.674828	1	0	3	1
6	7	Duster 360	14.3	8	360.0	245	3.21	3.570	15.840000	0	0	3	4
7	8	Merc 240D	24.4	4	146.7	62	3.69	3.190	20.000000	1	0	4	2
8	9	Merc 230	22.8	4	140.8	95	3.92	3.150	22.900000	1	0	4	2
9	10	Merc 280	19.2	6	167.6	123	3.92	3.440	18.300000	1	0	4	4
10	11	Merc 280C	17.8	6	167.6	123	3.92	3.440	18.900000	1	0	4	4
11	12	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.400000	0	0	3	3
12	13	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.600000	0	0	3	3
13	14	Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.000000	0	0	3	3
14	15	Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.980000	0	0	3	4
15	16	Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.820000	0	0	3	4
16	17	Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.420000	0	0	3	4
17	18	Fiat 128	32.4	4	78.7	66	4.08	2.200	17.674828	1	1	4	1
18	19	Honda Civic	30.4	4	75.7	52	4.93	1.615	18.520000	1	1	4	2
19	20	Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.900000	1	1	4	1
20	21	Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.010000	1	0	3	1
21	22	Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.870000	0	0	3	2
22	23	AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.300000	0	0	3	2
23	24	Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.410000	0	0	3	4
24	25	Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.050000	0	0	3	2
25	26	Fiat X1-9	27.3	4	79.0	66	4.08	1.935	17.674828	1	1	4	1
26	27	Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.700000	0	1	5	2
27	28	Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.900000	1	1	5	2
28	29	Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.500000	0	1	5	4
29	30	Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.500000	0	1	5	6
30	31	Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.600000	0	1	5	8
31	32	Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.600000	1	1	4	2

In [18]:

```
#drop unwanted column
cars = cars.drop(columns=['S.No'])
cars
```

Out[18]:

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.460000	0	1	4	4
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.020000	0	1	4	4
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.610000	1	1	4	1
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.440000	1	0	3	1
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.020000	0	0	3	2
5	Valiant	18.1	6	225.0	105	2.76	3.460	17.674828	1	0	3	1
6	Duster 360	14.3	8	360.0	245	3.21	3.570	15.840000	0	0	3	4
7	Merc 240D	24.4	4	146.7	62	3.69	3.190	20.000000	1	0	4	2
8	Merc 230	22.8	4	140.8	95	3.92	3.150	22.900000	1	0	4	2
9	Merc 280	19.2	6	167.6	123	3.92	3.440	18.300000	1	0	4	4
10	Merc 280C	17.8	6	167.6	123	3.92	3.440	18.900000	1	0	4	4
11	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.400000	0	0	3	3
12	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.600000	0	0	3	3
13	Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.000000	0	0	3	3
14	Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.980000	0	0	3	4
15	Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.820000	0	0	3	4
16	Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.420000	0	0	3	4
17	Fiat 128	32.4	4	78.7	66	4.08	2.200	17.674828	1	1	4	1
18	Honda Civic	30.4	4	75.7	52	4.93	1.615	18.520000	1	1	4	2
19	Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.900000	1	1	4	1
20	Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.010000	1	0	3	1
21	Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.870000	0	0	3	2
22	AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.300000	0	0	3	2
23	Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.410000	0	0	3	4
24	Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.050000	0	0	3	2
25	Fiat X1-9	27.3	4	79.0	66	4.08	1.935	17.674828	1	1	4	1
26	Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.700000	0	1	5	2
27	Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.900000	1	1	5	2
28	Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.500000	0	1	5	4
29	Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.500000	0	1	5	6
30	Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.600000	0	1	5	8
31	Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.600000	1	1	4	2

In [64]:

```
#find correlation matrix
```

```
df=cars[['mpg','cyl','disp','hp','drat','wt','qsec','vs','am','gear','carb']].corr()
df
#you can see mpg is of string type so we can't perform correlation.
```

Out[64]:

	cyl	disp	hp	drat	wt	qsec	vs	am	gear
cyl	1.000000	0.902033	0.832447	-0.699938	0.782496	-0.548775	-0.810812	-0.522607	-0.492687
disp	0.902033	1.000000	0.790949	-0.710214	0.887980	-0.385207	-0.710416	-0.591227	-0.555569
hp	0.832447	0.790949	1.000000	-0.448759	0.658748	-0.650674	-0.723097	-0.243204	-0.125704
drat	-0.699938	-0.710214	-0.448759	1.000000	-0.712441	0.120175	0.440278	0.712711	0.699610
wt	0.782496	0.887980	0.658748	-0.712441	1.000000	-0.130362	-0.554916	-0.692495	-0.583287
qsec	-0.548775	-0.385207	-0.650674	0.120175	-0.130362	1.000000	0.667873	-0.271763	-0.203784
vs	-0.810812	-0.710416	-0.723097	0.440278	-0.554916	0.667873	1.000000	0.168345	0.206023
am	-0.522607	-0.591227	-0.243204	0.712711	-0.692495	-0.271763	0.168345	1.000000	0.794059
gear	-0.492687	-0.555569	-0.125704	0.699610	-0.583287	-0.203784	0.206023	0.794059	1.000000
carb	0.526988	0.394977	0.749812	-0.090790	0.427606	-0.573987	-0.569607	0.057534	0.274073

In [66]:

```
#So change mpg from string to float to perform correlation
cars.mpg = cars.mpg.astype(float)
#see the change
cars.info(null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 13 columns):
S.No     32 non-null int64
model    32 non-null object
mpg      32 non-null float64
cyl      32 non-null int64
disp     32 non-null float64
hp       32 non-null int64
drat     32 non-null float64
wt       32 non-null float64
qsec     32 non-null float64
vs       32 non-null int64
am       32 non-null int64
gear     32 non-null int64
carb     32 non-null int64
dtypes: float64(5), int64(7), object(1)
memory usage: 3.3+ KB
```

In [7]:

```
#Again find the correlation including mpg
df=cars[['mpg', 'cyl','disp','hp','drat','wt','qsec','vs','am','gear','carb']].corr()
df
```

Out[7]:

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am
mpg	1.000000	-0.852162	-0.847551	-0.776168	0.681172	-0.867659	0.401294	0.664039	0.599832
cyl	-0.852162	1.000000	0.902033	0.832447	-0.699938	0.782496	-0.579882	-0.810812	-0.522607

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	
disp	-0.847551	0.902033	1.000000	0.790949	-0.710214	0.887980	-0.406922	-0.710416	-0.591227	-
hp	-0.776168	0.832447	0.790949	1.000000	-0.448759	0.658748	-0.690114	-0.723097	-0.243204	-
drat	0.681172	-0.699938	-0.710214	-0.448759	1.000000	-0.712441	0.128973	0.440278	0.712711	
wt	-0.867659	0.782496	0.887980	0.658748	-0.712441	1.000000	-0.137198	-0.554916	-0.692495	-
qsec	0.401294	-0.579882	-0.406922	-0.690114	0.128973	-0.137198	1.000000	0.717275	-0.288955	-
vs	0.664039	-0.810812	-0.710416	-0.723097	0.440278	-0.554916	0.717275	1.000000	0.168345	
am	0.599832	-0.522607	-0.591227	-0.243204	0.712711	-0.692495	-0.288955	0.168345	1.000000	
gear	0.480285	-0.492687	-0.555569	-0.125704	0.699610	-0.583287	-0.207942	0.206023	0.794059	
carb	-0.550925	0.526988	0.394977	0.749812	-0.090790	0.427606	-0.616964	-0.569607	0.057534	

Now from the correlation we can see lowest correlation is between "am" and "carb". So when we will taking "carb" as reference for any analysis we can drop "am", since it will not add any change or value to the analysis. In the same way we can drop "carb" column while taking "am" as reference for any analysis.

Manipluation

Sometimes we don't need the whole dataset for analysis and cut them in small chunks, sometimes we need to arrange them in an certain order(ascending or descending), sometimes we may need to set a certain value to a column or apply a function to a column, sometmes we may also need to apply filters to our dataset---all of these comes under Pandas Data Manipulation.

Indexing by position

In [69]:
```
#view hp column only
cars.iloc[:,4]
```

Out[69]:
```
0     160.0
1     160.0
2     108.0
3     258.0
4     360.0
5     225.0
6     360.0
7     146.7
8     140.8
9     167.6
10    167.6
11    275.8
12    275.8
13    275.8
14    472.0
15    460.0
16    440.0
17     78.7
```

```
18     75.7
19     71.1
20    120.1
21    318.0
22    304.0
23    350.0
24    400.0
25     79.0
26    120.3
27     95.1
28    351.0
29    145.0
30    301.0
31    121.0
Name: disp, dtype: float64
```

In [70]:
```
#first five records of hp column
cars.iloc[0:5,4]
```

Out[70]:
```
0    160.0
1    160.0
2    108.0
3    258.0
4    360.0
Name: disp, dtype: float64
```

In [25]:
```
#all rows, all columns
cars.iloc[:,:]
```

Out[25]:

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.460000	0	1	4	4
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.020000	0	1	4	4
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.610000	1	1	4	1
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.440000	1	0	3	1
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.020000	0	0	3	2
5	Valiant	18.1	6	225.0	105	2.76	3.460	17.674828	1	0	3	1
6	Duster 360	14.3	8	360.0	245	3.21	3.570	15.840000	0	0	3	4
7	Merc 240D	24.4	4	146.7	62	3.69	3.190	20.000000	1	0	4	2
8	Merc 230	22.8	4	140.8	95	3.92	3.150	22.900000	1	0	4	2
9	Merc 280	19.2	6	167.6	123	3.92	3.440	18.300000	1	0	4	4
10	Merc 280C	17.8	6	167.6	123	3.92	3.440	18.900000	1	0	4	4
11	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.400000	0	0	3	3
12	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.600000	0	0	3	3
13	Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.000000	0	0	3	3
14	Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.980000	0	0	3	4
15	Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.820000	0	0	3	4
16	Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.420000	0	0	3	4

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
17	Fiat 128	32.4	4	78.7	66	4.08	2.200	17.674828	1	1	4	1
18	Honda Civic	30.4	4	75.7	52	4.93	1.615	18.520000	1	1	4	2
19	Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.900000	1	1	4	1
20	Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.010000	1	0	3	1
21	Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.870000	0	0	3	2
22	AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.300000	0	0	3	2
23	Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.410000	0	0	3	4
24	Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.050000	0	0	3	2
25	Fiat X1-9	27.3	4	79.0	66	4.08	1.935	17.674828	1	1	4	1
26	Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.700000	0	1	5	2
27	Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.900000	1	1	5	2
28	Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.500000	0	1	5	4
29	Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.500000	0	1	5	6
30	Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.600000	0	1	5	8
31	Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.600000	1	1	4	2

In [26]:

```
#for attributes from hp to carb see all the records from index 6
cars.iloc[6:,4:]
```

Out[26]:

	hp	drat	wt	qsec	vs	am	gear	carb
6	245	3.21	3.570	15.840000	0	0	3	4
7	62	3.69	3.190	20.000000	1	0	4	2
8	95	3.92	3.150	22.900000	1	0	4	2
9	123	3.92	3.440	18.300000	1	0	4	4
10	123	3.92	3.440	18.900000	1	0	4	4
11	180	3.07	4.070	17.400000	0	0	3	3
12	180	3.07	3.730	17.600000	0	0	3	3
13	180	3.07	3.780	18.000000	0	0	3	3
14	205	2.93	5.250	17.980000	0	0	3	4
15	215	3.00	5.424	17.820000	0	0	3	4
16	230	3.23	5.345	17.420000	0	0	3	4
17	66	4.08	2.200	17.674828	1	1	4	1
18	52	4.93	1.615	18.520000	1	1	4	2
19	65	4.22	1.835	19.900000	1	1	4	1

	hp	drat	wt	qsec	vs	am	gear	carb
20	97	3.70	2.465	20.010000	1	0	3	1
21	150	2.76	3.520	16.870000	0	0	3	2
22	150	3.15	3.435	17.300000	0	0	3	2
23	245	3.73	3.840	15.410000	0	0	3	4
24	175	3.08	3.845	17.050000	0	0	3	2
25	66	4.08	1.935	17.674828	1	1	4	1
26	91	4.43	2.140	16.700000	0	1	5	2
27	113	3.77	1.513	16.900000	1	1	5	2
28	264	4.22	3.170	14.500000	0	1	5	4
29	175	3.62	2.770	15.500000	0	1	5	6
30	335	3.54	3.570	14.600000	0	1	5	8
31	109	4.11	2.780	18.600000	1	1	4	2

In [73]:
```
#Now we want to look at all the rows and only the first column
cars.iloc[:,1]
```

Out[73]:
```
0              Mazda RX4
1          Mazda RX4 Wag
2             Datsun 710
3         Hornet 4 Drive
4      Hornet Sportabout
5                Valiant
6             Duster 360
7              Merc 240D
8               Merc 230
9               Merc 280
10             Merc 280C
11            Merc 450SE
12            Merc 450SL
13           Merc 450SLC
14    Cadillac Fleetwood
15   Lincoln Continental
16     Chrysler Imperial
17              Fiat 128
18           Honda Civic
19        Toyota Corolla
20         Toyota Corona
21      Dodge Challenger
22           AMC Javelin
23            Camaro Z28
24      Pontiac Firebird
25             Fiat X1-9
26         Porsche 914-2
27          Lotus Europa
28        Ford Pantera L
29          Ferrari Dino
30         Maserati Bora
31            Volvo 142E
Name: model, dtype: object
```

indexing by label

In [75]:
```
#see all the record of mpg column
cars.loc[:,"mpg"]
```

Out[75]:
```
0     21.0
1     21.0
2     22.8
3     21.4
4     18.7
5     18.1
6     14.3
7     24.4
8     22.8
9     19.2
10    17.8
11    16.4
12    17.3
13    15.2
14    10.4
15    10.4
16    14.7
17    32.4
18    30.4
19    33.9
20    21.5
21    15.5
22    15.2
23    13.3
24    19.2
25    27.3
26    26.0
27    30.4
28    15.8
29    19.7
30    15.0
31    21.4
Name: mpg, dtype: float64
```

In [78]:
```
#display the records from index 0 to index 6 from mpg column
cars.loc[:6,"mpg"]
```

Out[78]:
```
0    21.0
1    21.0
2    22.8
3    21.4
4    18.7
5    18.1
6    14.3
Name: mpg, dtype: float64
```

In [77]:
```
#see the first 7 records from mpg to qsec column
cars.loc[:6,"mpg":"qsec"]
```

Out[77]:

	mpg	cyl	disp	hp	drat	wt	qsec
0	21.0	6	160.0	110	3.90	2.620	16.460000
1	21.0	6	160.0	110	3.90	2.875	17.020000

	mpg	cyl	disp	hp	drat	wt	qsec
2	22.8	4	108.0	93	3.85	2.320	18.610000
3	21.4	6	258.0	110	3.08	3.215	19.440000
4	18.7	8	360.0	175	3.15	3.440	17.020000
5	18.1	6	225.0	105	2.76	3.460	17.674828
6	14.3	8	360.0	245	3.21	3.570	15.840000

In [31]:
```
#set value 1 to column 'am'
cars['am'] = 1
cars
```

Out[31]:

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.460000	0	1	4	4
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.020000	0	1	4	4
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.610000	1	1	4	1
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.440000	1	1	3	1
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.020000	0	1	3	2
5	Valiant	18.1	6	225.0	105	2.76	3.460	17.674828	1	1	3	1
6	Duster 360	14.3	8	360.0	245	3.21	3.570	15.840000	0	1	3	4
7	Merc 240D	24.4	4	146.7	62	3.69	3.190	20.000000	1	1	4	2
8	Merc 230	22.8	4	140.8	95	3.92	3.150	22.900000	1	1	4	2
9	Merc 280	19.2	6	167.6	123	3.92	3.440	18.300000	1	1	4	4
10	Merc 280C	17.8	6	167.6	123	3.92	3.440	18.900000	1	1	4	4
11	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.400000	0	1	3	3
12	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.600000	0	1	3	3
13	Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.000000	0	1	3	3
14	Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.980000	0	1	3	4
15	Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.820000	0	1	3	4
16	Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.420000	0	1	3	4
17	Fiat 128	32.4	4	78.7	66	4.08	2.200	17.674828	1	1	4	1
18	Honda Civic	30.4	4	75.7	52	4.93	1.615	18.520000	1	1	4	2
19	Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.900000	1	1	4	1
20	Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.010000	1	1	3	1
21	Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.870000	0	1	3	2
22	AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.300000	0	1	3	2
23	Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.410000	0	1	3	4

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
24	Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.050000	0	1	3	2
25	Fiat X1-9	27.3	4	79.0	66	4.08	1.935	17.674828	1	1	4	1
26	Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.700000	0	1	5	2
27	Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.900000	1	1	5	2
28	Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.500000	0	1	5	4
29	Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.500000	0	1	5	6
30	Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.600000	0	1	5	8
31	Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.600000	1	1	4	2

In [80]:

```
#double up records in 'am' using lambda fxn
f = lambda x: x*2
cars['am']= cars['am'].apply(f)
cars
```

Out[80]:

	S.No	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	1	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.460000	0	2	4	4
1	2	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.020000	0	2	4	4
2	3	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.610000	1	2	4	1
3	4	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.440000	1	0	3	1
4	5	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.020000	0	0	3	2
5	6	Valiant	18.1	6	225.0	105	2.76	3.460	17.674828	1	0	3	1
6	7	Duster 360	14.3	8	360.0	245	3.21	3.570	15.840000	0	0	3	4
7	8	Merc 240D	24.4	4	146.7	62	3.69	3.190	20.000000	1	0	4	2
8	9	Merc 230	22.8	4	140.8	95	3.92	3.150	22.900000	1	0	4	2
9	10	Merc 280	19.2	6	167.6	123	3.92	3.440	18.300000	1	0	4	4
10	11	Merc 280C	17.8	6	167.6	123	3.92	3.440	18.900000	1	0	4	4
11	12	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.400000	0	0	3	3
12	13	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.600000	0	0	3	3
13	14	Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.000000	0	0	3	3
14	15	Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.980000	0	0	3	4
15	16	Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.820000	0	0	3	4
16	17	Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.420000	0	0	3	4
17	18	Fiat 128	32.4	4	78.7	66	4.08	2.200	17.674828	1	2	4	1
18	19	Honda Civic	30.4	4	75.7	52	4.93	1.615	18.520000	1	2	4	2
19	20	Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.900000	1	2	4	1

	S.No	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
20	21	Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.010000	1	0	3	1
21	22	Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.870000	0	0	3	2
22	23	AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.300000	0	0	3	2
23	24	Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.410000	0	0	3	4
24	25	Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.050000	0	0	3	2
25	26	Fiat X1-9	27.3	4	79.0	66	4.08	1.935	17.674828	1	2	4	1
26	27	Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.700000	0	2	5	2
27	28	Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.900000	1	2	5	2
28	29	Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.500000	0	2	5	4
29	30	Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.500000	0	2	5	6
30	31	Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.600000	0	2	5	8
31	32	Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.600000	1	2	4	2

Sorting

In [33]:
```
#sorting cyl column ascending order
cars.sort_values(by='cyl')
```

Out[33]:

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
31	Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.600000	1	2	4	2
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.610000	1	2	4	1
27	Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.900000	1	2	5	2
26	Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.700000	0	2	5	2
25	Fiat X1-9	27.3	4	79.0	66	4.08	1.935	17.674828	1	2	4	1
20	Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.010000	1	2	3	1
7	Merc 240D	24.4	4	146.7	62	3.69	3.190	20.000000	1	2	4	2
8	Merc 230	22.8	4	140.8	95	3.92	3.150	22.900000	1	2	4	2
19	Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.900000	1	2	4	1
18	Honda Civic	30.4	4	75.7	52	4.93	1.615	18.520000	1	2	4	2
17	Fiat 128	32.4	4	78.7	66	4.08	2.200	17.674828	1	2	4	1
29	Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.500000	0	2	5	6
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.460000	0	2	4	4
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.020000	0	2	4	4
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.440000	1	2	3	1

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
10	Merc 280C	17.8	6	167.6	123	3.92	3.440	18.900000	1	2	4	4
9	Merc 280	19.2	6	167.6	123	3.92	3.440	18.300000	1	2	4	4
5	Valiant	18.1	6	225.0	105	2.76	3.460	17.674828	1	2	3	1
13	Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.000000	0	2	3	3
28	Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.500000	0	2	5	4
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.020000	0	2	3	2
24	Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.050000	0	2	3	2
23	Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.410000	0	2	3	4
22	AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.300000	0	2	3	2
21	Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.870000	0	2	3	2
6	Duster 360	14.3	8	360.0	245	3.21	3.570	15.840000	0	2	3	4
11	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.400000	0	2	3	3
16	Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.420000	0	2	3	4
30	Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.600000	0	2	5	8
14	Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.980000	0	2	3	4
12	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.600000	0	2	3	3
15	Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.820000	0	2	3	4

In [79]:

```
#sort cyl in descending order
cars.sort_values(by='cyl', ascending=False)
```

Out[79]:

	S.No	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
16	17	Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.420000	0	0	3	4
30	31	Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.600000	0	1	5	8
4	5	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.020000	0	0	3	2
28	29	Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.500000	0	1	5	4
6	7	Duster 360	14.3	8	360.0	245	3.21	3.570	15.840000	0	0	3	4
24	25	Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.050000	0	0	3	2
23	24	Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.410000	0	0	3	4
22	23	AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.300000	0	0	3	2
21	22	Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.870000	0	0	3	2
11	12	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.400000	0	0	3	3
12	13	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.600000	0	0	3	3
13	14	Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.000000	0	0	3	3

	S.No	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
14	15	Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.980000	0	0	3	4
15	16	Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.820000	0	0	3	4
1	2	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.020000	0	1	4	4
29	30	Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.500000	0	1	5	6
0	1	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.460000	0	1	4	4
10	11	Merc 280C	17.8	6	167.6	123	3.92	3.440	18.900000	1	0	4	4
9	10	Merc 280	19.2	6	167.6	123	3.92	3.440	18.300000	1	0	4	4
5	6	Valiant	18.1	6	225.0	105	2.76	3.460	17.674828	1	0	3	1
3	4	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.440000	1	0	3	1
17	18	Fiat 128	32.4	4	78.7	66	4.08	2.200	17.674828	1	1	4	1
18	19	Honda Civic	30.4	4	75.7	52	4.93	1.615	18.520000	1	1	4	2
19	20	Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.900000	1	1	4	1
20	21	Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.010000	1	0	3	1
8	9	Merc 230	22.8	4	140.8	95	3.92	3.150	22.900000	1	0	4	2
7	8	Merc 240D	24.4	4	146.7	62	3.69	3.190	20.000000	1	0	4	2
25	26	Fiat X1-9	27.3	4	79.0	66	4.08	1.935	17.674828	1	1	4	1
26	27	Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.700000	0	1	5	2
27	28	Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.900000	1	1	5	2
2	3	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.610000	1	1	4	1
31	32	Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.600000	1	1	4	2

Filtering

In [35]:
```
#filter records with more than 6 cylinders
cars['cyl'] > 6
```

Out[35]:
```
0     False
1     False
2     False
3     False
4      True
5     False
6      True
7     False
8     False
9     False
10    False
11     True
12     True
13     True
14     True
15     True
```

```
16     True
17    False
18    False
19    False
20    False
21     True
22     True
23     True
24     True
25    False
26    False
27    False
28     True
29    False
30     True
31    False
Name: cyl, dtype: bool
```

In [36]:

```
#filter records with more than 6 cylinders
filter1 = cars['cyl'] > 6
#apply filter to dataframe
filtered_new = cars[filter1]
#view filtered dataframe
filtered_new
```

Out[36]:

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	2	3	2
6	Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	2	3	4
11	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	2	3	3
12	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	2	3	3
13	Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	2	3	3
14	Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	2	3	4
15	Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	2	3	4
16	Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	2	3	4
21	Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	2	3	2
22	AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	2	3	2
23	Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	2	3	4
24	Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	2	3	2
28	Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	2	5	4
30	Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	2	5	8

In [37]:

```
#filter records with more than 6 cyl and hp more than 300
filter2 = (cars["cyl"] > 6) & (cars["hp"] > 300)
#apply filter to dataframe
filtered_review = cars[filter2]
#display filtered data
filtered_review
```

Out[37]:

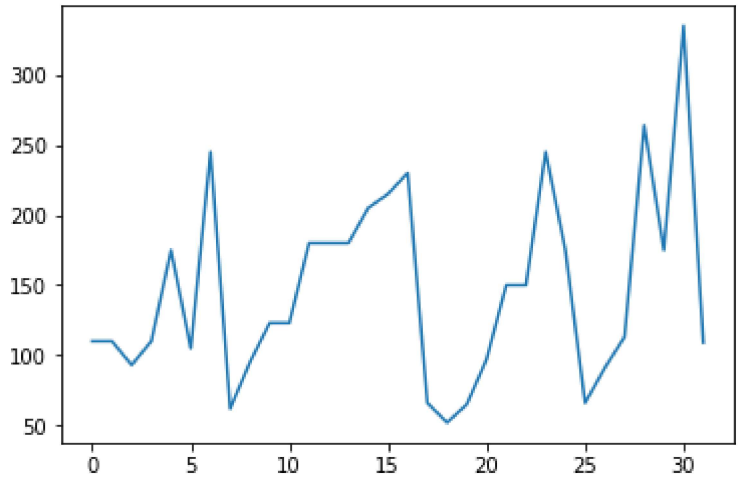
	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
30	Maserati Bora	15.0	8	301.0	335	3.54	3.57	14.6	0	2	5	8

Data Visualization

In [38]:

```
#import matplotlib
import matplotlib.pyplot  as plt
%matplotlib inline
#see how hp varies with each car with line plot
y1 = cars['hp']
x  = range(32)
plt.plot(x,y1)
```

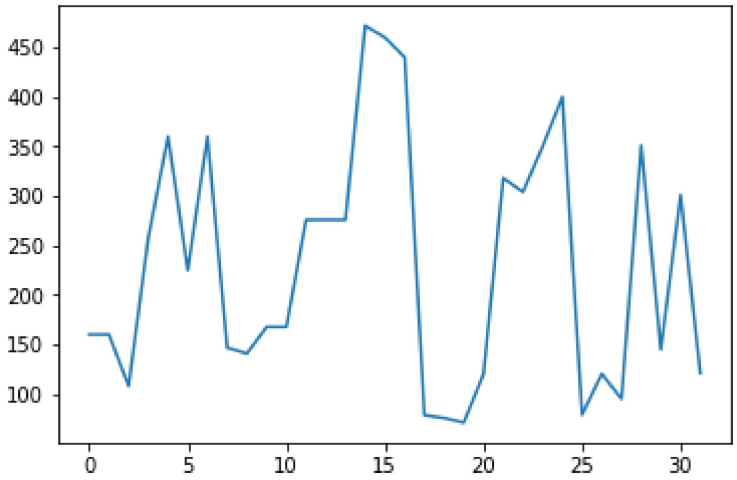
Out[38]: [<matplotlib.lines.Line2D at 0x1cbfac2cba8>]



In [39]:

```
#import matplotlib
import matplotlib.pyplot  as plt
%matplotlib inline
#see how hp varies with each car with line plot
y2 = cars['disp']
x  = range(32)
plt.plot(x,y2)
```

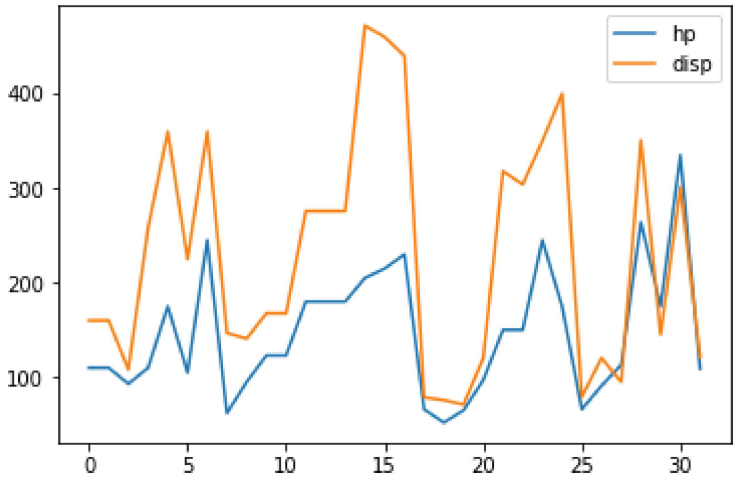
Out[39]: [<matplotlib.lines.Line2D at 0x1cbfacc37b8>]



In [4]:

```
#import matplotlib
import matplotlib.pyplot  as plt
%matplotlib inline
y1 = cars['hp']
y2 = cars['disp']
#see how both hp and disp varies
x  = range(32)
plt.plot(x,y1)
plt.plot(x,y2)
plt.legend()
```

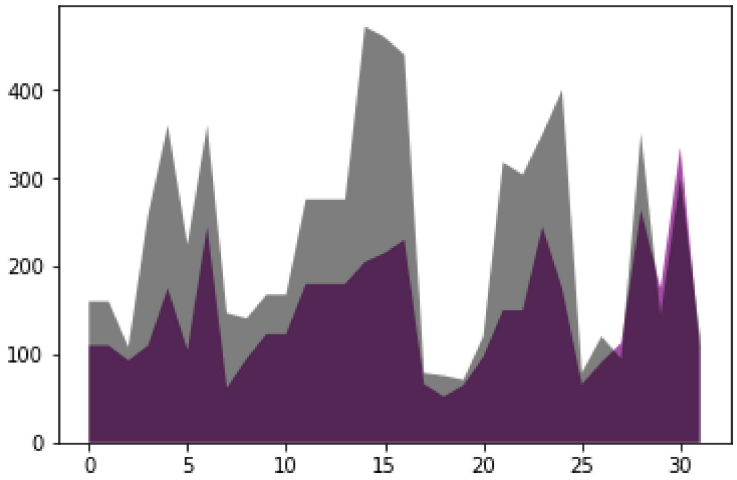
Out[4]: <matplotlib.legend.Legend at 0x2da9cd1b470>



In [6]:

```
#import matplotlib
import matplotlib.pyplot  as plt
%matplotlib inline
y1 = cars['hp']
y2 = cars['disp']
x  = range(32)
#area plot of hp and disp
plt.stackplot(x,y1,colors = 'purple', alpha = 0.7)
plt.stackplot(x,y2,colors = 'black', alpha = 0.5)
```

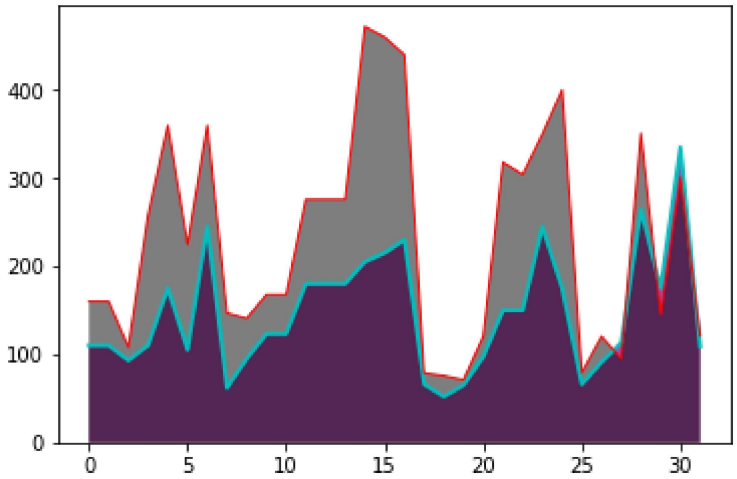
Out[6]: [<matplotlib.collections.PolyCollection at 0x2da9ce11940>]



In [107…

```
import matplotlib.pyplot  as plt
%matplotlib inline
y1 = cars['hp']
y2 = cars['disp']
x  = range(32)
#plot both line plot and area plot to see the margin
plt.plot(x,y1, linewidth = 2.0, color = 'c')
plt.stackplot(x,y1,colors = 'purple', alpha = 0.7)
plt.plot(x,y2, linewidth = 1.0, color = 'r')
plt.stackplot(x,y2,colors = 'black', alpha = 0.5)
```

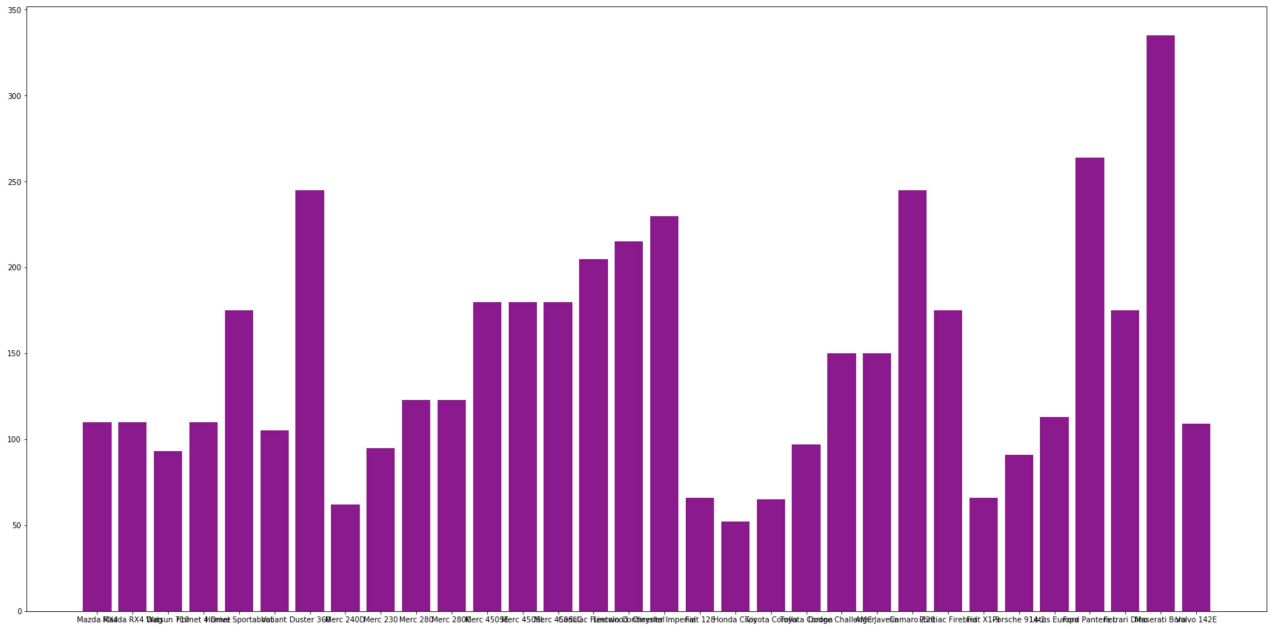
Out[107… [<matplotlib.collections.PolyCollection at 0x1cbfe64bd30>]



In [121…

```
#import matplotlib
import matplotlib.pyplot  as plt
%matplotlib inline
y = cars['hp']
x  = range(32)
#model to list
x1 = cars['model'].tolist()
#adding figure to adjust figsize
fig = plt.figure(figsize = (30,15))
#see how hp changes with bar plot
plt.bar(x1,y,color="purple", alpha=0.9)
```

Out[121… <BarContainer object of 32 artists>



In [117…

```
#import matplotlib
import matplotlib.pyplot  as plt
%matplotlib inline
y = cars['hp']
x  = range(32)
x1 = cars['model'].tolist()
fig = plt.figure(figsize = (17,10))
#to avoid the overlapping issue plot horizontal bar plot
plt.barh(x1,y, color="purple", alpha=0.8)
```

Out[117… <BarContainer object of 32 artists>