

# DSL1\_C5\_S3\_PRACTICE

In [7]:

```
import pandas as pd
import numpy as np
import statistics as st
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [9]:

```
car = pd.read_csv('E:\Aishwarya official\Aishwarya Data Scince\course 5\DS1_C5_S3_Mtcars_Data\car
```

Out[9]:

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	210	3.90	2.620	16.46	0	1	4	4
1	Mazda RX4 Wag	21.0	6	160.0	210	3.90	2.875	17.02	0	1	4	4
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	Hornet 4 Drive	21.4	6	258.0	210	3.08	3.215	19.44	1	0	3	1
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
5	Valiant	18.1	6	225.0	205	2.76	3.460	20.22	1	0	3	1
6	Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
7	Merc 240D	24.4	4	146.7	330	3.69	3.190	20.00	1	0	4	2
8	Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
9	Merc 280	19.2	6	167.6	223	3.92	3.440	18.30	1	0	4	4
10	Merc 280C	17.8	6	167.6	223	3.92	3.440	18.90	1	0	4	4
11	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
12	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
13	Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
14	Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
15	Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
16	Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
17	Fiat 128	32.4	4	78.7	340	4.08	2.200	19.47	1	1	4	1
18	Honda Civic	30.4	4	75.7	280	4.93	1.615	18.52	1	1	4	2
19	Toyota Corolla	33.9	4	71.1	180	4.22	1.835	19.90	1	1	4	1
20	Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
21	Dodge Challenger	15.5	8	318.0	250	2.76	3.520	16.87	0	0	3	2
22	AMC Javelin	15.2	8	304.0	180	3.15	3.435	17.30	0	0	3	2
23	Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
24	Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
25	Fiat X1-9	27.3	4	79.0	290	4.08	1.935	18.90	1	1	4	1
26	Porsche 914-2	26.0	4	120.3	320	4.43	2.140	16.70	0	1	5	2
27	Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
28	Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
29	Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
30	Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
31	Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

## Task 1.1 : sampling and distribution

In [11]:

```
random_sampling = car.sample(20, ignore_index=True)
random_sampling
```

Out[11]:

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
1	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
2	Mazda RX4	21.0	6	160.0	210	3.90	2.620	16.46	0	1	4	4
3	Toyota Corolla	33.9	4	71.1	180	4.22	1.835	19.90	1	1	4	1
4	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
5	Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
6	Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
7	Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2
8	Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
9	Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
10	Fiat X1-9	27.3	4	79.0	290	4.08	1.935	18.90	1	1	4	1
11	Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
12	Valiant	18.1	6	225.0	205	2.76	3.460	20.22	1	0	3	1
13	Merc 280C	17.8	6	167.6	223	3.92	3.440	18.90	1	0	4	4
14	Hornet 4 Drive	21.4	6	258.0	210	3.08	3.215	19.44	1	0	3	1
15	Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
16	Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
17	Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
18	Porsche 914-2	26.0	4	120.3	320	4.43	2.140	16.70	0	1	5	2
19	Honda Civic	30.4	4	75.7	280	4.93	1.615	18.52	1	1	4	2

## Task 3.1.2 : random sample with replacement

In [14]:

```
random_sam = car.sample(20, replace = True ,random_state =1)
random_sam
```

Out[14]:

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
5	Valiant	18.1	6	225.0	205	2.76	3.460	20.22	1	0	3	1
11	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
12	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
8	Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
31	Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2
9	Merc 280	19.2	6	167.6	223	3.92	3.440	18.30	1	0	4	4
11	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
5	Valiant	18.1	6	225.0	205	2.76	3.460	20.22	1	0	3	1
15	Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
0	Mazda RX4	21.0	6	160.0	210	3.90	2.620	16.46	0	1	4	4
16	Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
1	Mazda RX4 Wag	21.0	6	160.0	210	3.90	2.875	17.02	0	1	4	4
12	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
7	Merc 240D	24.4	4	146.7	330	3.69	3.190	20.00	1	0	4	2
13	Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
28	Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
6	Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
25	Fiat X1-9	27.3	4	79.0	290	4.08	1.935	18.90	1	1	4	1
18	Honda Civic	30.4	4	75.7	280	4.93	1.615	18.52	1	1	4	2
20	Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1

## Task 3.1.3:-Systematic Sampling

In [22]:

```
sys = np.arange(0, len(car), step=2)
car.iloc[sys]
```

Out[22]:

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	210	3.90	2.620	16.46	0	1	4	4
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
6	Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
8	Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
10	Merc 280C	17.8	6	167.6	223	3.92	3.440	18.90	1	0	4	4
12	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
14	Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
16	Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
18	Honda Civic	30.4	4	75.7	280	4.93	1.615	18.52	1	1	4	2
20	Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
22	AMC Javelin	15.2	8	304.0	180	3.15	3.435	17.30	0	0	3	2
24	Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
26	Porsche 914-2	26.0	4	120.3	320	4.43	2.140	16.70	0	1	5	2
28	Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
30	Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8

## Task 3.1.4 : Stratified Sampling

In [43]:

```

v = car.groupby('vs').get_group(1)
samp_v=v.sample(10,replace= True, ignore_index=True)
stratified_samp = pd.DataFrame()
stratified_samp = stratified_samp.append(samp_v, ignore_index=True)

s = car.groupby('vs').get_group(0)
samp_s = s.sample(10,replace= True , ignore_index =True)
stratified_samp = stratified_samp.append(samp_s, ignore_index=True)
stratified_samp

```

Out[43]:

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2
1	Fiat 128	32.4	4	78.7	340	4.08	2.200	19.47	1	1	4	1
2	Merc 280C	17.8	6	167.6	223	3.92	3.440	18.90	1	0	4	4
3	Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
4	Hornet 4 Drive	21.4	6	258.0	210	3.08	3.215	19.44	1	0	3	1
5	Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
6	Hornet 4 Drive	21.4	6	258.0	210	3.08	3.215	19.44	1	0	3	1
7	Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
8	Fiat X1-9	27.3	4	79.0	290	4.08	1.935	18.90	1	1	4	1
9	Toyota Corolla	33.9	4	71.1	180	4.22	1.835	19.90	1	1	4	1
10	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
11	Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
12	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
13	Mazda RX4 Wag	21.0	6	160.0	210	3.90	2.875	17.02	0	1	4	4
14	Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
15	Dodge Challenger	15.5	8	318.0	250	2.76	3.520	16.87	0	0	3	2
16	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
17	Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
18	Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
19	Porsche 914-2	26.0	4	120.3	320	4.43	2.140	16.70	0	1	5	2

## Task 3.1.5 : Cluster Sampling

In [50]:

```

clust_samp=pd.DataFrame()
samp_v=car.groupby('vs').get_group(1).sample(frac=0.7, replace=True, ignore_index=True, random_state=123)
samp_s=car.groupby('vs').get_group(0).sample(frac=0.6, replace=True, ignore_index=True, random_state=123)
clust_samp=pd.concat([clust_samp, samp_s, samp_v], ignore_index=True)
clust_samp

```

Out[50]:

	model	mpg	cyl	displacement	hp	drat	wt	qsec	vs	am	gear	carb
0	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
1	AMC Javelin	15.2	8	304.0	180	3.15	3.435	17.30	0	0	3	2
2	Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
3	Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
4	Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
5	AMC Javelin	15.2	8	304.0	180	3.15	3.435	17.30	0	0	3	2
6	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
7	Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
8	Mazda RX4	21.0	6	160.0	210	3.90	2.620	16.46	0	1	4	4
9	Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
10	Mazda RX4 Wag	21.0	6	160.0	210	3.90	2.875	17.02	0	1	4	4
11	Merc 280	19.2	6	167.6	223	3.92	3.440	18.30	1	0	4	4
12	Fiat X1-9	27.3	4	79.0	290	4.08	1.935	18.90	1	1	4	1
13	Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
14	Honda Civic	30.4	4	75.7	280	4.93	1.615	18.52	1	1	4	2
15	Toyota Corolla	33.9	4	71.1	180	4.22	1.835	19.90	1	1	4	1
16	Fiat X1-9	27.3	4	79.0	290	4.08	1.935	18.90	1	1	4	1
17	Merc 280	19.2	6	167.6	223	3.92	3.440	18.30	1	0	4	4
18	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
19	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
20	Hornet 4 Drive	21.4	6	258.0	210	3.08	3.215	19.44	1	0	3	1

## Task 3.2.1

In [57]:

```

hp = car['hp']
samp1 = hp.sample(n=10, replace=True, random_state=123)
samp2 = hp.sample(n=20, replace=True, random_state=123)
samp3 = hp.sample(n=50, replace=True, random_state=123)

```

In [63]:

```
tables = [hp,samp1,samp2,samp3]
mean=[]
median=[]
std=[]
for i in tables:
    mean.append(i.mean())
    median.append(i.median())
    std.append(i.std())
pd.DataFrame([mean, median , std], columns=['Hp_pop','samp_10','samp_20','samp_30'], index=
```

Out[63]:

	Hp_pop	samp_10	samp_20	samp_30
mean	211.312500	228.800000	230.700000	216.340000
std	210.000000	234.000000	219.000000	210.000000
median	68.513449	93.156738	80.294261	73.292428

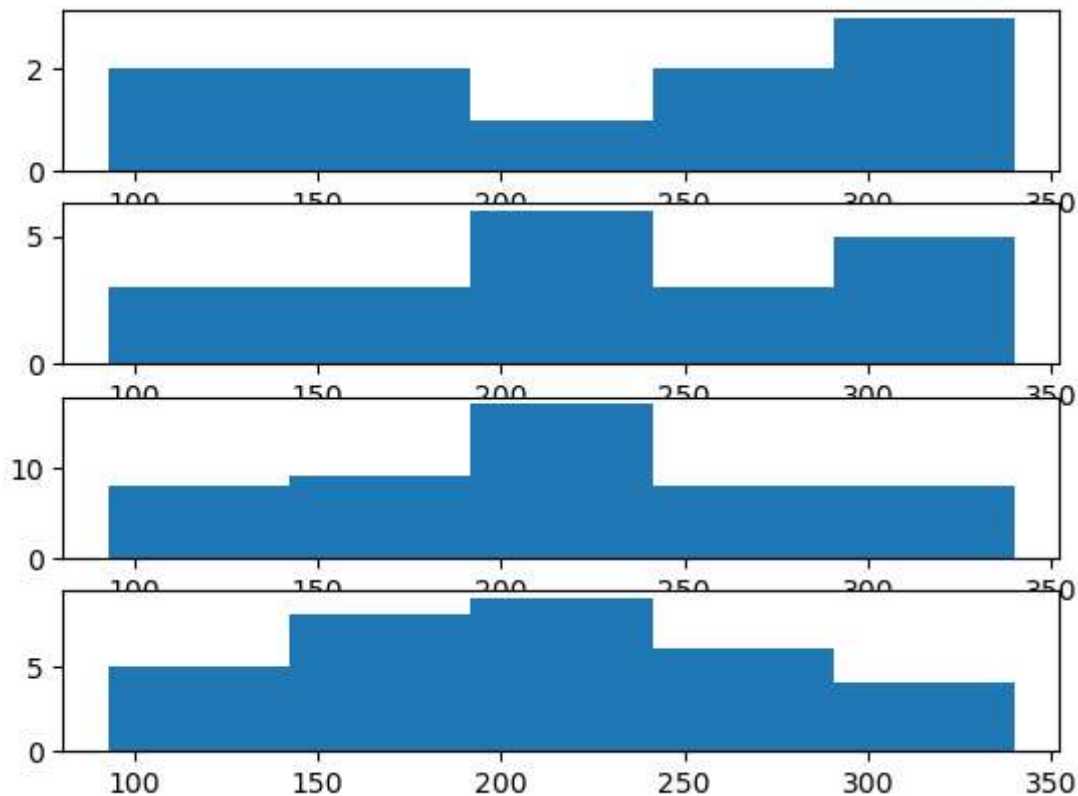


In [65]:

```
fig,ax=plt.subplots(4,1)
ax[0].hist(samp1,bins=5)
ax[1].hist(samp2,bins=5)
ax[2].hist(samp3,bins=5)
ax[3].hist(hp,bins=5)
```

Out[65]:

```
(array([5., 8., 9., 6., 4.]),
 array([ 93. , 142.4, 191.8, 241.2, 290.6, 340. ]),
 <BarContainer object of 5 artists>)
```



Interpretation:- different samples are unevenly distributed and more data gives accurate predictions