```
In [3]:    1  import pandas as pd
           2  from scipy.stats import zscore,norm
           3  import matplotlib.pyplot as plt
           4
```

Task1: To calculate household probabilities of cellular

```
In [20]:   1  avg=70
           2  std=11.351
           3  x1=55
           4  x2=80
           5  zscore1=(x1-70)/11.351
           6  zscore2=(x2-70)/11.351
           7  zscore3=(0-70)/11.351
           8  zscore4=(40-70)/11.351
           9  p1= norm.cdf(zscore1)
          10  p2=norm.cdf(zscore2)
          11  p3=norm.cdf(zscore3)
          12  p4=norm.cdf(zscore4)
          13  prob=(1-p1)-(1-p2)
          14  prob2=(1-p3)-(1-p4)
          15  print("Probability of finding between 55USD and 8USD",round(prob,2))
          16  print("Probability of finding less than 40USD",round(prob2,5))
```

```
Probability of finding between 55USD and 8USD 0.72
Probability of finding less than 40USD 0.00411
```

```
In [22]:   1  smart=pd.read_csv(r"K:\Desktop\NIIT\tables\DS1_C5_S5_SmartCarRiding_Data_Practice.csv")
           2  smart
```
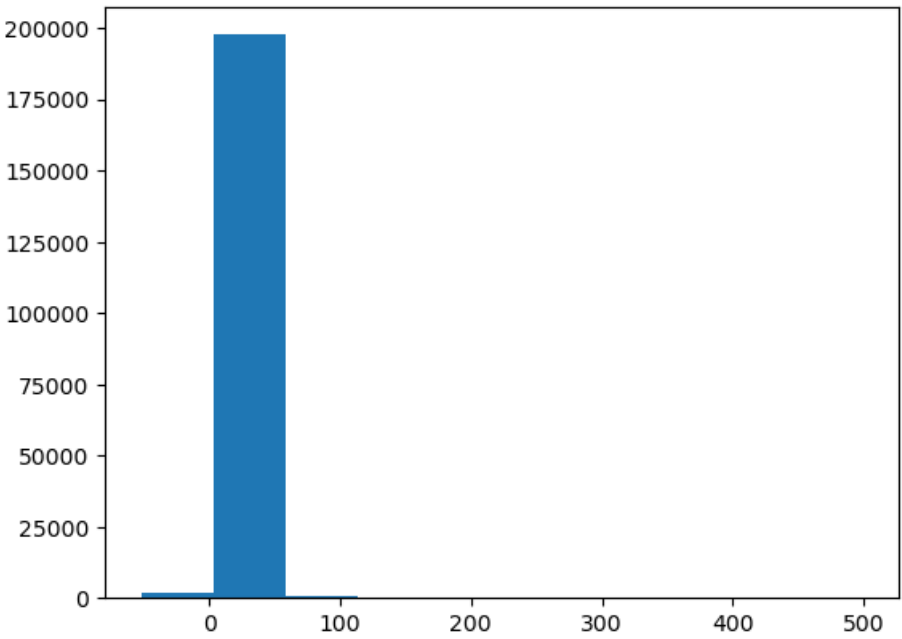
Out[22]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
0	24238194	2015-05-07 19:52:06.0000003	7.5	2015-05-07 19:52:06 UTC	-73.999817	40.738354	-73.999512	40.723217	1
1	27835199	2009-07-17 20:04:56.0000002	7.7	2009-07-17 20:04:56 UTC	-73.994355	40.728225	-73.994710	40.750325	1
2	44984355	2009-08-24 21:45:00.00000061	12.9	2009-08-24 21:45:00 UTC	-74.005043	40.740770	-73.962565	40.772647	1
3	25894730	2009-06-26 08:22:21.0000001	5.3	2009-06-26 08:22:21 UTC	-73.976124	40.790844	-73.965316	40.803349	3
4	17610152	2014-08-28 17:47:00.000000188	16.0	2014-08-28 17:47:00 UTC	-73.925023	40.744085	-73.973082	40.761247	5
...	...	...	...	...	...	...	...	...	...
199995	42598914	2012-10-28 10:49:00.00000053	3.0	2012-10-28 10:49:00 UTC	-73.987042	40.739367	-73.986525	40.740297	1
199996	16382965	2014-03-14 01:09:00.0000008	7.5	2014-03-14 01:09:00 UTC	-73.984722	40.736837	-74.006672	40.739620	1
199997	27804658	2009-06-29 00:42:00.00000078	30.9	2009-06-29 00:42:00 UTC	-73.986017	40.756487	-73.858957	40.692588	2
199998	20259894	2015-05-20 14:56:25.0000004	14.5	2015-05-20 14:56:25 UTC	-73.997124	40.725452	-73.983215	40.695415	1
199999	11951496	2010-05-15 04:08:00.00000076	14.1	2010-05-15 04:08:00 UTC	-73.984395	40.720077	-73.985508	40.768793	1

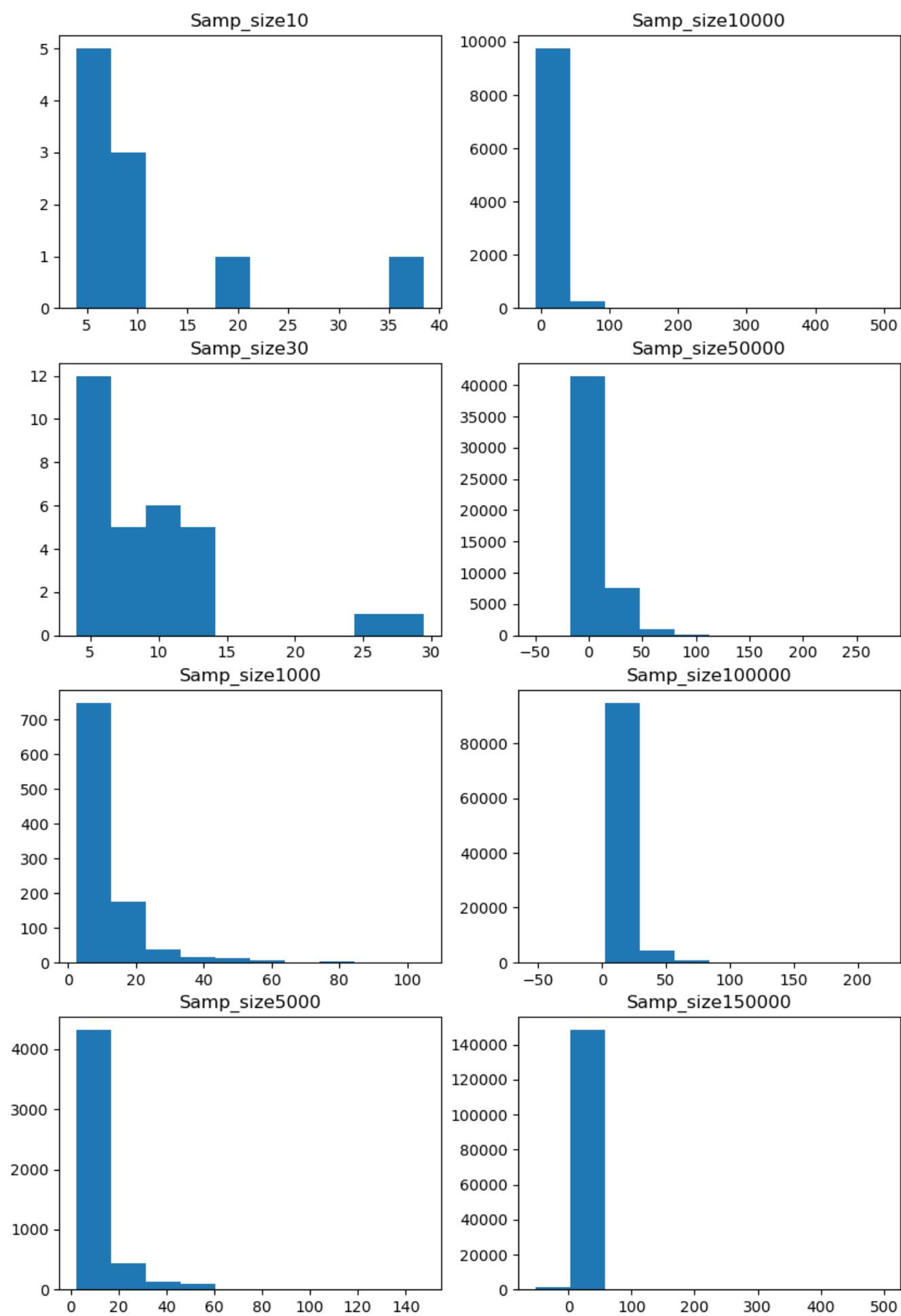200000 rows × 9 columns

Data preprocessing

```
In [31]:   1  plt.hist(smart.fare_amount);
```

```
1  samp=[10,30,1000,5000,10000,50000,100000,150000]
2  fig,ax=plt.subplots(4,2,figsize=(10,15))
3  j=0
4  k=0
5  for i in samp:
6      garb=smart["fare_amount"].sample(i,ignore_index=True)
7      ax[j,k].hist(garb)
8      ax[j,k].set_title("Samp_size"+str(i))
9      j+=1
10     if(j==4):
11         k+=1
12         j=0
13
```


To extract data using random sampling with replace=True

```
1  samp=[10,30,1000,5000,10000,50000,100000,150000]
2  fig,ax=plt.subplots(4,2,figsize=(10,15))
3  j=0
4  k=0
5  for i in samp:
6      garb=smart["fare_amount"].sample(i,ignore_index=True)
7      ax[j,k].hist(garb)
8      ax[j,k].set_title("Samp_size"+str(i))
9      j+=1
10     if(j==4):
11         k+=1
12         j=0
```

```
In [57]:  1  samples=pd.DataFrame()
          2  cols=[("Sample_"+str(i)) for i in range(0,21)]
          3  for i in range(0,21):
          4      samples[cols[i]]=smart["fare_amount"].sample(200,replace=True,ignore_index=True)
          5  samples
          6
```

Out[57]:

	Sample_0	Sample_1	Sample_2	Sample_3	Sample_4	Sample_5	Sample_6	Sample_7	Sample_8	Sample_9	...	Sample_11	Sample_12	Sample_13	Sample_14	Sample_15	Sample_1
0	4.5	5.7	11.5	9.50	4.5	6.1	12.0	5.3	6.0	8.1	...	7.0	30.50	4.50	7.50	24.1	10.
1	7.5	3.5	18.0	9.50	9.0	4.5	23.5	3.3	17.0	9.0	...	11.7	9.70	6.10	4.10	3.7	6.
2	4.5	5.5	7.5	6.00	7.5	2.9	29.7	10.1	11.5	4.0	...	9.3	9.70	5.00	8.50	12.9	8.
3	6.5	11.3	6.9	6.90	4.5	4.9	8.1	17.5	4.5	14.1	...	11.7	13.50	15.30	57.33	6.9	8.
4	8.5	3.5	4.9	4.90	6.5	4.1	5.7	7.3	8.1	10.5	...	4.9	33.54	5.00	11.50	5.7	30.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
195	7.0	6.1	8.5	6.50	18.1	16.0	4.5	6.5	8.1	13.5	...	5.3	18.50	34.80	19.70	5.0	10.
196	4.1	12.1	15.5	3.70	3.7	6.9	13.7	5.7	6.5	9.5	...	5.0	8.10	6.00	4.10	6.9	5.
197	6.0	8.9	37.5	4.50	7.7	3.7	10.1	7.0	11.3	6.1	...	6.5	6.90	5.50	29.30	7.0	5.
198	16.9	3.3	6.5	42.83	17.0	7.0	4.9	13.3	3.7	6.0	...	6.5	15.50	10.90	48.33	13.3	6.
199	4.9	4.1	7.3	8.00	8.1	12.5	10.1	4.5	12.5	11.5	...	6.5	7.30	85.16	19.00	8.5	10.

```
In [ ]:  1
```