In [1]:

```
import pandas as pd
```

In [3]:

```
df = pd.read_csv('/home/prannoys21/Downloads/network
assignment/images_assign4/forced_atlas_2/data.csv')
```

In [4]:

```
df.head()
```

Out[4]:

	Id	Label	timeset	1	2	3	4	5	modularity_class	weighted indegree	wei outd
0	1026	1026	<[1246290100000.0, 1246309000000.0]; [12463414...	94	28	151	<[1246290100000.0, 1246291000000.0, 1]; [12462...	<[1246290100000.0, 1246291000000.0, 2]; [12462...	12	0	34
1	1029	1029	<[1246353100000.0, 1246354000000.0]; [12463594...	61	28	344	<[1246353100000.0, 1246354000000.0, 1]; [12463...	<[1246353100000.0, 1246354000000.0, 3]; [12463...	14	0	99
2	1032	1032	<[1246260400000.0, 1246268500000.0]; [12462883...	108	45	297	<[1246260400000.0, 1246261300000.0, 5]; [12462...	<[1246260400000.0, 1246261300000.0, 12]; [1246...	3	1	71
3	1033	1033	<[1246281100000.0, 1246297300000.0]; [12463117...	89	60	649	<[1246281100000.0, 1246282000000.0, 2]; [12462...	<[1246281100000.0, 1246282000000.0, 2]; [12462...	12	2	207
4	1035	1035	<[1246442200000.0, 1246468300000.0]>	29	44	265	<[1246442200000.0, 1246443100000.0, 3]; [12464...	<[1246442200000.0, 1246443100000.0, 3]; [12464...	13	1	92

In [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 113 entries, 0 to 112
Data columns (total 17 columns):
Id                            113 non-null int64
Label                         113 non-null int64
timeset                       113 non-null object
1                             113 non-null int64
2                             113 non-null int64
3                             113 non-null int64
4                             113 non-null object
5                             113 non-null object
modularity_class              113 non-null int64
weighted indegree             113 non-null int64
weighted outdegree            113 non-null int64
Weighted Degree               113 non-null int64
Eccentricity                  113 non-null int64
closnesscentrality            113 non-null float64
harmonicclosnesscentrality    113 non-null float64
betweenesscentrality          113 non-null float64
eigencentrality               113 non-null float64
dtypes: float64(4), int64(10), object(3)
memory usage: 15.1+ KB
```

In [6]:

```
df.describe()
```

Out[6]:

	Id	Label	1	2	3	modularity_class	weighted indegree	weighted outdegree	Weigh Deg
count	113.000000	113.000000	113.000000	113.000000	113.000000	113.000000	113.000000	113.000000	113.000(
mean	1141.433628	1141.433628	91.504425	38.283186	421.911504	8.522124	41.159292	41.159292	82.3185
std	78.869566	78.869566	34.354279	18.382576	597.324310	4.690601	38.350723	37.575297	50.9909
min	1026.000000	1026.000000	4.000000	1.000000	2.000000	0.000000	0.000000	0.000000	2.000000
25%	1085.000000	1085.000000	70.000000	25.000000	115.000000	5.000000	11.000000	17.000000	44.0000(
50%	1131.000000	1131.000000	98.000000	35.000000	285.000000	9.000000	29.000000	32.000000	76.0000(
75%	1187.000000	1187.000000	114.000000	48.000000	535.000000	12.000000	63.000000	52.000000	103.000(
max	1360.000000	1360.000000	206.000000	97.000000	4672.000000	16.000000	180.000000	207.000000	275.000(

Though we can see the average values in above table, lets individually calculate them

In [7]:

```
df['weighted indegree'].mean()
```

Out[7]:

```
41.15929203539823
```

In [8]:

```
df['Weighted Degree'].mean()
```

Out[8]:

```
82.31858407079646
```

In [9]:

```
df['modularity_class'].mean()
```

Out[9]:

```
8.52212389380531
```

In [10]:

```
df['betweenesscentrality'].mean()
```

Out[10]:

```
33.5044247964602
```

In [11]:

```
df['eigencentrality'].mean()
```

Out[11]:

```
0.1163550353982301
```