• Assessment of Marginal Workers in Tamil Nadu- A Socioeconomic Analysis

TEAM NUMBER=01

VADDI HARHSA VARDHAN

(Team leader)

In [1]:
```
#importing the libraries in
python import pandas as pd
import matplotlib.pyplot as plt
#import seaborn for using
piechart import seaborn as sns
import numpy as np
```

Importing csv file data sets

In [2]:
```
# Load the dataset into a Pandas DataFrame
df = pd.read_csv('Downloads/DDW_B06SC_3300_State_TAMIL_NADU-2011.csv')
```

checking the data fully filled that is fully true

In [3]:
```
#checking the dataset given is null or
not df.isnull()
```

Out [3]:

	Table Code	State Code	District Code	Area Name	Total/ Rural/ Urban	Age group	Worked for 3 months or more but less than 6 months - Persons	Worked for 3 months or more but less than 6 months - Males	Worked for 3 months or more but less than 6 months - Females	Worked for less than 3 months - Persons	...	Industrial Category - N to O - Females	Industrial Category - P to Q - Persons	Industrial Category - P to Q - Males	Industrial Category - P to Q - Females	Industrial Category - R to U - HHI - Persons	Industri Catego - R U - H - Mal
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
589	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False
590	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False
591	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False
592	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False
593	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False

594 rows × 69 columns

Fetching and describe the data

In [4]:
```
l1=tuple([df['Worked for 3 months or more but less than 6 months - Females'],df['Worked for 3 months or more but less
th
```

In [5]:
```
l2=tuple([df['Industrial Category - N to O - Females'],df['Industrial Category - P to Q - Persons']])
```

In [6]:
```
df.describe()
```

Out [6]:

	Worked for 3 months or more but less than 6 months - Persons	Worked for 3 months or more but less than 6 months - Males	Worked for 3 months or more but less than 6 months - Females	Worked for less than 3 months - Persons	Worked for less than 3 months - Males	Worked for less than 3 months - Females	Industrial Category - A - Cultivators - Persons	Industrial Category - A - Cultivators - Males	Industrial Category - A - Cultivators - Females
count	5.940000e+02	594.000000	594.000000	594.000000	594.000000	594.000000	594.000000	594.000000	594.000000
mean	1.617277e+04	7932.700337	8240.067340	2981.629630	1338.289562	1643.340067	865.117845	466.424242	398.693603
std	7.607172e+04	36864.822704	39259.545337	13909.621137	6127.047670	7808.832522	4274.458077	2298.072295	1978.682322
min	0.000000e+00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	2.872500e+02	147.250000	144.000000	27.000000	14.250000	13.000000	9.000000	5.000000	4.000000
50%	2.225500e+03	1147.000000	1076.000000	430.000000	198.500000	213.000000	69.500000	35.500000	32.000000
75%	9.628500e+03	4770.500000	4887.500000	1775.250000	774.250000	946.500000	466.000000	244.250000	204.750000
max	1.200828e+06	589003.000000	611825.000000	221386.000000	99368.000000	122018.000000	64235.000000	34632.000000	29603.000000

8 rows × 63 columns

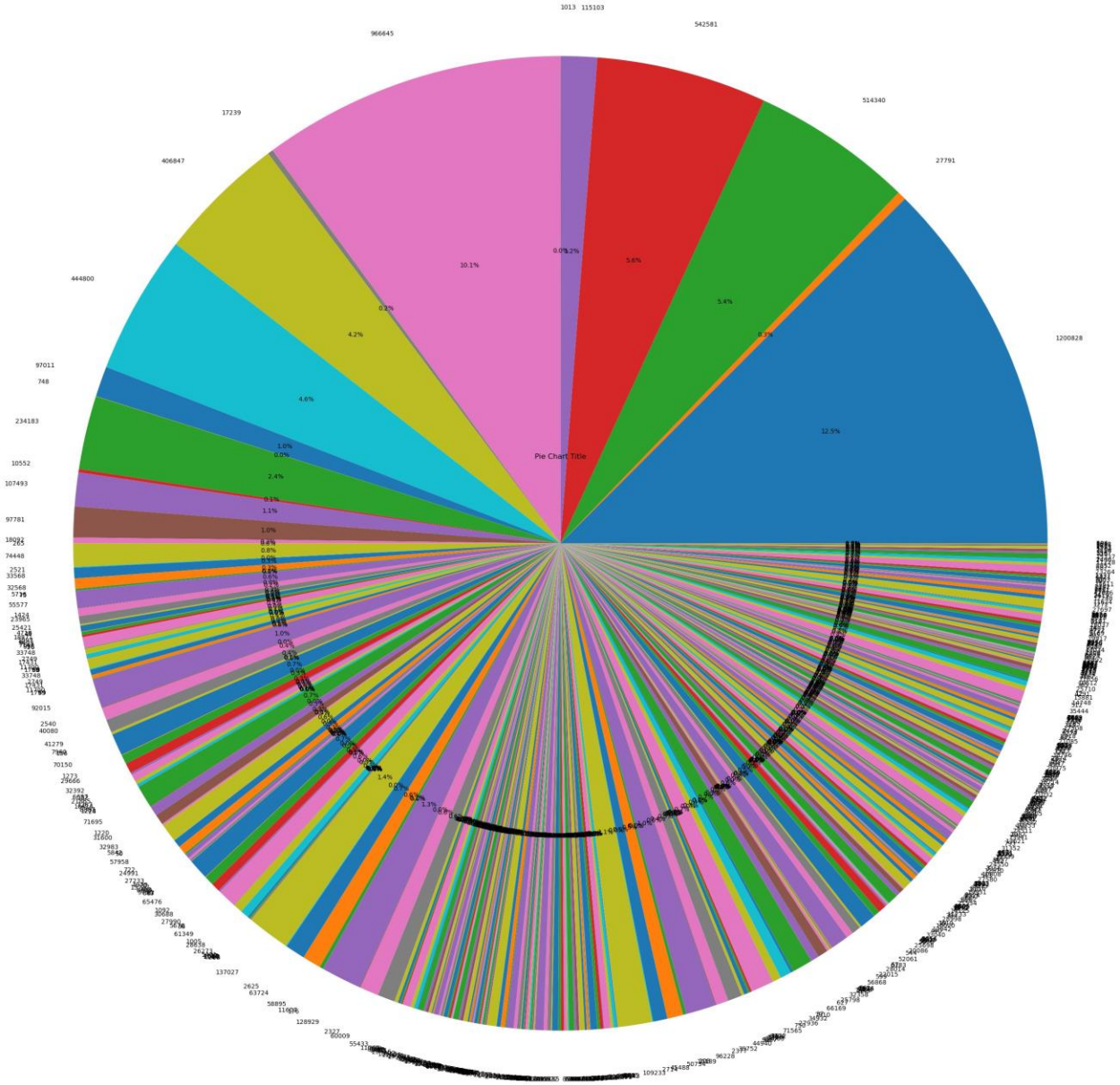
In [7]:
```
l1
```

Out [7]:
```
(0       611825
 1        13666
 2       254780
 3       290624
 4        52270
         ...
589         143
590        1631
591        1903
592         297
593           1
Name: Worked for 3 months or more but less than 6 months - Females, Length: 594, dtype:
int64,  589003
 1        14125
 2       259560
 3       251957
 4        62833
         ...
589         129
590        1654
591        1769
592         399
593           1
Name: Worked for 3 months or more but less than 6 months - Males, Length: 594, dtype:
int64)
```

In [8]:
```
l2
```

Out [8]:
```
(0       3565
 1         11
 2       1754
 3       1619
 4        175
         ...
589         0
590        20
591        33
592         0
593         0
Name: Industrial Category - N to O - Females, Length: 594, dtype:
int64,  11080
 1          122
 2         7536
 3         3205
 4          211
         ...
589          0
590         44
591         35
592          3
593          0
Name: Industrial Category - P to Q - Persons, Length: 594, dtype:
int64)
```

In [9]:
```
# assigning the csv data to variable of piechart
pie_chart_data= df['Worked for 3 months or more but less than 6 months -
Persons'] #asigning values to the pie chart
plt.pie(pie_chart_data, labels=df['Worked for 3 months or more but less than 6 months -  Persons'],
autopct='%1.1f%%',ra

plt.title('Pie Chart Title')
#printing the pie chart
plt.show()
```



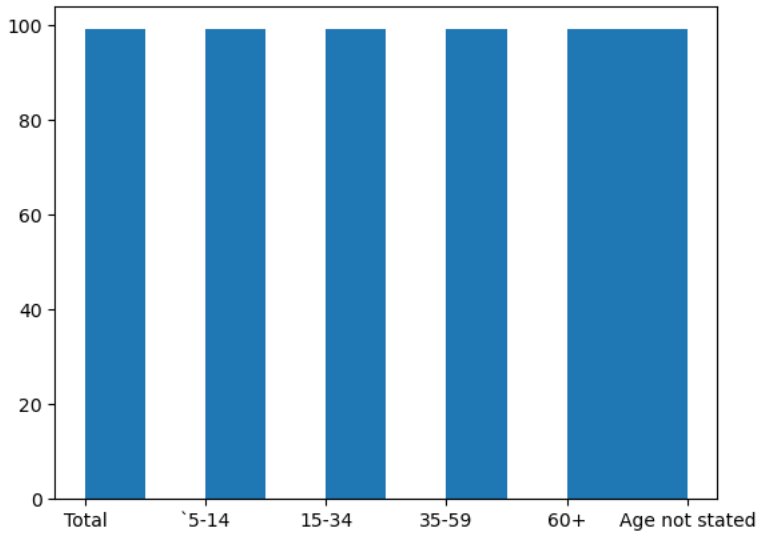
printing the pie chart using the given csv file data sets

In [10]:
```
price = df["Age group"]
```

visualizing the data sets column in the form of of histogram

In [11]:
```
plt.hist(price)
```

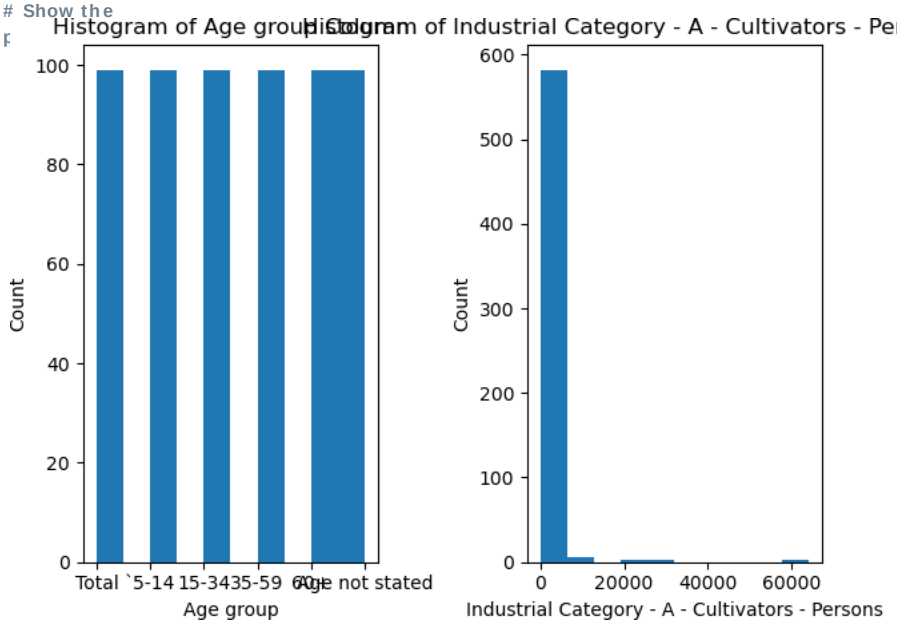
Out [11]:
```
(array([99.,  0., 99.,  0., 99.,  0., 99.,  0., 99., 99.]),
 array([0. , 0.5, 1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5, 5. ]),
 <BarContainer object of 10 artists>)
```



In [16]:

```
column_1 = df["Age group"]
column_2 = df["Industrial Category - A - Cultivators - Persons"]

# Create the histogram
fig, axs = plt.subplots(1, 2)

axs[0].hist(column_1)
axs[1].hist(column_2)

# Add a title and axis labels for each subplot
axs[0].set_title("Histogram of {}
Column".format(column_1.name)) axs[1].set_title("Histogram of
{} Column".format(column_2.name))

axs[0].set_xlabel(column_1.name)
axs[1].set_xlabel(column_2.name)

axs[0].set_ylabel("Count"
)
axs[1].set_ylabel("Count"
)

# Adjust the subplot
layout plt.tight_layout()
```

```
# Show the
p
```

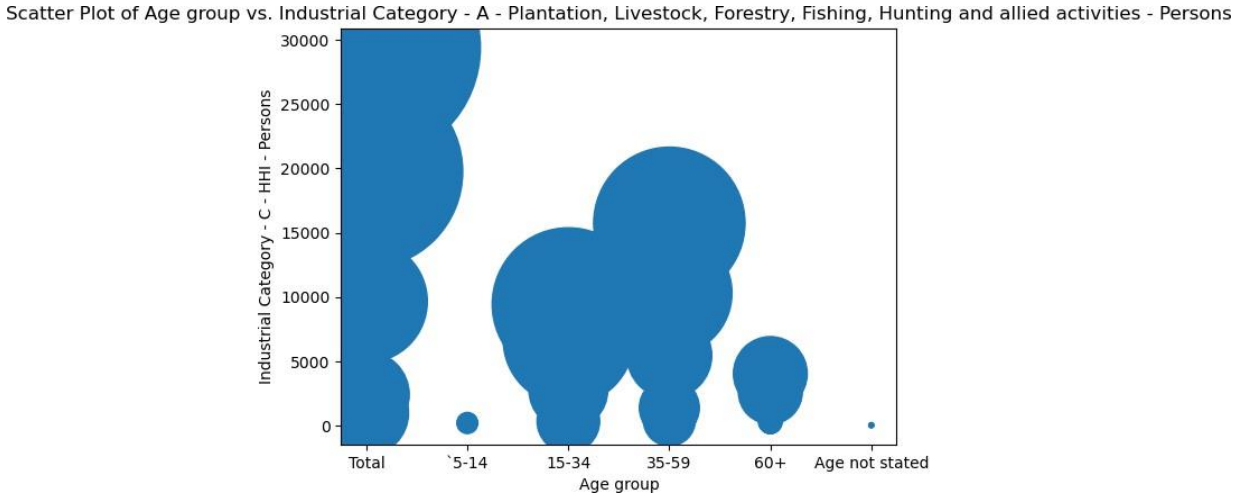


visualizing three different columns by using scatter plot

In [26]:

```
x_column = df["Age group"]
y_column = df["Industrial Category - A - Plantation, Livestock, Forestry, Fishing, Hunting and allied activities -
Perso z_column = df["Industrial Category - C - HHI - Persons"]
# Create the scatter plot
```

```
plt.scatter(x_column, y_column, z_column)

# Add a title and axis labels
plt.title("Scatter Plot of {} vs. {}".format(x_column.name, y_column.name,
z_column.name)) plt.xlabel(x_column.name)
plt.ylabel(y_column.name)
plt.ylabel(z_column.name)

# Show the
plot plt.show()
```



output for the scatter plot for the gib=ven three columns