ASSESSMENT OF MARGINAL WORKERS IN TAMIL NADU –

A SOCIOECONOMIC ANALYSIS

INTRODUCTION:

Demographic characteristics of marginal workers in Tamil Nadu provide important insights into the composition of this labour force and are significant for various reasons, including economic and social policy planning.

Performing a socioeconomic analysis of marginal workers based on age, industrial category, and sex is essential for understanding the demographics and patterns within this group. By delving into these factors, we can gain valuable insights into the challenges they face and develop targeted solutions. Utilizing visualizations like bar charts, pie charts, and heatmaps will not only enhance the comprehension of complex data but also facilitate effective communication of the findings to a wider audience. Through this analysis, we can identify trends, disparities, and areas for improvement, enabling informed decision-making and policy formulation.

• Age : The age distribution of marginal workers can indicate the youthfulness or aging of this labour force. A higher proportion of young marginal workers may reflect the need for skill development and employment opportunities for the youth.

• Sex : Understanding the gender balance among marginal workers is essential for addressing gender-related labour market disparities.

• Industrial Category : The industrial category or the sector in which the marginal workers work provides an insight about their economy.

OBJECTIVE :

To perform a socioeconomic analysis of the demographic characteristics of marginal workers based on age, industrial category, and sex and create visualizations such as bar charts, pie charts, or heatmaps to represent the distribution across different categories.

DESIGN THINKING :

EMPATHISE :

Understand the problem by conducting survey among marginal workers in Tamil Nadu understanding their experience and challenges related to age, industrial category, and sex-based disparities.

DEFINE :

Synthesise the data and insights gathered empathy phase to identify the specific demographic characteristics (age, industrial category, and sex).

IDEATE :

Approach a diverse team to gather data and insights through internet and social medias.

PROTOTYPE :

Use Data Visualisation tools such as bar charts, pie charts and heat maps to represent the gathered data.

TEST :

Implement the created prototype with a small group of marginal workers to gather feedback from them.

IMPLEMENT :

Implement this solution in collaborating with relevant agencies to ensure successful execution.

CLUSTERING ANALYSIS :

Objective :

To identify patterns among different industrial categories and age groups among marginal workers based on demographics.

Data Collection and Preparation :

Data Source:

https://tn.data.gov.in/resource/marginal-workers-classified-age-industrial-category-and-sex-scheduled-caste-2011-tamil

Data Description:

The attributes included in the dataset are age, gender, and industrial category.

Choosing the Number of Clusters (K):

Using the Elbow method we assign number of clusters, choose an appropriate K value based on the Elbow Method

Hierarchal Clustering Algorithm:

Hierarchical clustering is a type of clustering algorithm used in data analysis and machine learning to create a hierarchy of clusters. It organizes marginal workers data into a hierarchy of clusters, forming a tree-like structure called a dendrogram.

Initialisation:

Initialize N cluster. Each data point starts as a single cluster.

We use complete linkage method to compute the distance between clusters as the maximum distance between their data points. Euclidean distance is used as the distance metric in this clustering process.

Implementation :

Software and Libraries :

The software and libraries used for the analysis:

Python

Pandas

Numpy

Matplotlib

Scipy

Seaborn

Code:

import pandas as pd

from sklearn.cluster import KMeans

import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler

# Load the CSV file into a Pandas DataFrame

file\_path = '/content/DDW\_B06SC\_3300\_State\_TAMIL\_NADU-2011.csv'

data = pd.read\_csv(file\_path)

# Select the columns you want to use for clustering

selected\_columns = data.columns[7:-1]

# Create a new DataFrame with the selected columns

selected\_data = data[selected\_columns]

# Normalize the data (important for K-Means)

scaler = StandardScaler()

normalized\_data = scaler.fit\_transform(selected\_data)

# Determine the optimal number of clusters (K) using the Elbow method

inertia = []

k\_values = range(1, 6)  # Reducing the range to avoid the error

for k in k\_values:

    kmeans = KMeans(n\_clusters=k, random\_state=42)

    kmeans.fit(normalized\_data)

    inertia.append(kmeans.inertia\_)

# Plot the Elbow method graph

plt.figure(figsize=(8, 4))

plt.plot(k\_values, inertia, marker='o', linestyle='--')

plt.title('Elbow Method for Optimal K')

plt.xlabel('Number of Clusters (K)')

plt.ylabel('Inertia (Within-cluster Sum of Squares)')

plt.xticks(k\_values)

plt.grid()

plt.tight\_layout()

plt.show()

# Based on the Elbow method, choose an appropriate K value (number of clusters)

# You can visually inspect the graph and select a value where the inertia starts to level off.

# Perform K-Means clustering with the chosen K value

k = 3  # You can adjust this value based on the Elbow method result

kmeans = KMeans(n\_clusters=k, random\_state=42)

clusters = kmeans.fit\_predict(normalized\_data)

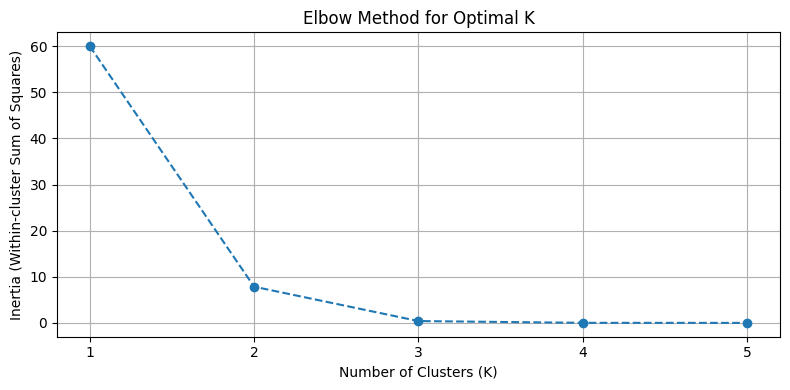
# Add the cluster labels to the DataFrame

data['Cluster'] = clusters

# Now, you have assigned clusters to each row in your dataset.

# You can further analyze and visualize the clustering results as needed.

# For example, you can create scatter plots or other visualizations to see how data points in different clusters are distributed.



import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from scipy.cluster import hierarchy

from scipy.spatial.distance import pdist

# Load the CSV file into a Pandas DataFrame

file\_path = '/content/DDW\_B06SC\_3300\_State\_TAMIL\_NADU-2011.csv'

data = pd.read\_csv(file\_path)

# Select the columns you want to use for clustering

# In this case, you may want to use the columns representing industrial categories and age groups.

selected\_columns = data.columns[7:-1]  # Adjust the column range as needed

# Create a new DataFrame with the selected columns

selected\_data = data[selected\_columns]

# Normalize the data (important for hierarchical clustering)

normalized\_data = (selected\_data - selected\_data.mean()) / selected\_data.std()

# Calculate the pairwise distances between data points

distances = pdist(normalized\_data, metric='euclidean')

# Perform hierarchical clustering

linkage\_matrix = hierarchy.linkage(distances, method='ward')

# Create a cluster diagram (dendrogram)

plt.figure(figsize=(10, 6))

dendrogram = hierarchy.dendrogram(linkage\_matrix, labels=data['Age group'].tolist(), orientation='top')

plt.title('Hierarchical Clustering Dendrogram')

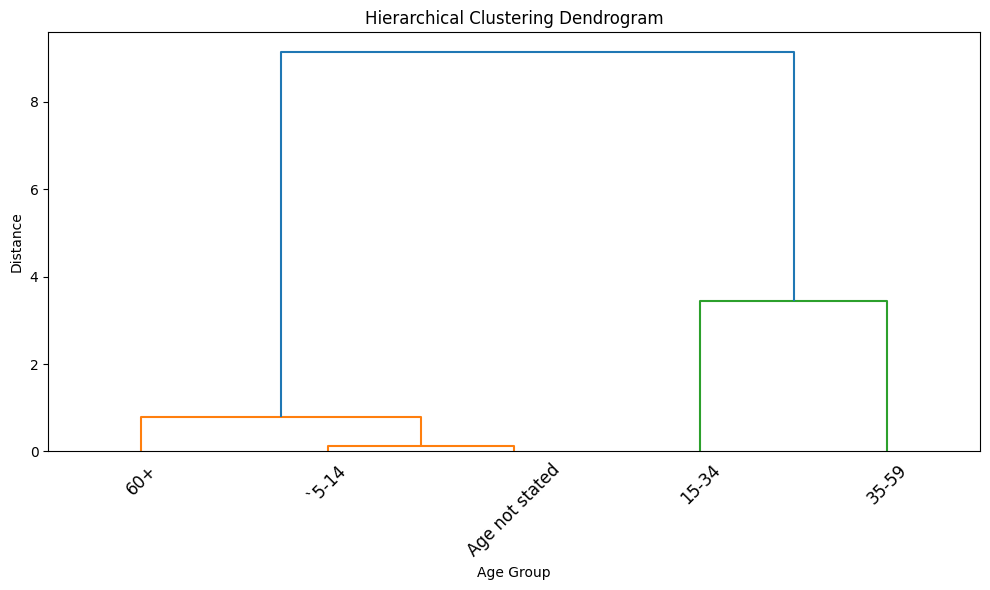
plt.xlabel('Age Group')

plt.ylabel('Distance')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from scipy.cluster import hierarchy

from scipy.spatial.distance import pdist

# Load the CSV file into a Pandas DataFrame

file\_path = '/content/DDW\_B06SC\_3300\_State\_TAMIL\_NADU-2011.csv'

data = pd.read\_csv(file\_path)

# Select the columns you want to use for clustering

# In this case, you may want to use the columns representing industrial categories and age groups.

selected\_columns = data.columns[7:-1]  # Adjust the column range as needed

# Create a new DataFrame with the selected columns

selected\_data = data[selected\_columns]

# Normalize the data (important for hierarchical clustering)

normalized\_data = (selected\_data - selected\_data.mean()) / selected\_data.std()

# Calculate the pairwise distances between data points

distances = pdist(normalized\_data, metric='euclidean')

# Perform Agglomerative hierarchical clustering

agglomerative\_linkage = hierarchy.linkage(distances, method='ward')

# Create an Agglomerative cluster diagram (dendrogram)

plt.figure(figsize=(10, 6))

dendrogram\_agg = hierarchy.dendrogram(agglomerative\_linkage, labels=data['Age group'].tolist(), orientation='top')

plt.title('Agglomerative Hierarchical Clustering Dendrogram --bottom-up approach--')

plt.xlabel('Age Group')

plt.ylabel('Distance')

plt.xticks(rotation=45)

plt.tight\_layout()

# Perform Divisive hierarchical clustering (top-down approach)

# You can use the fcluster function with t parameter to set the desired number of clusters.

from scipy.cluster.hierarchy import fcluster

k = 3  # Specify the number of clusters for Divisive clustering

divisive\_clusters = fcluster(agglomerative\_linkage, k, criterion='maxclust')

# Add the divisive cluster labels to the DataFrame

data['Divisive Cluster'] = divisive\_clusters

# Now, you can analyze and visualize the divisive clusters as needed.

# Show the Divisive clustering results

print("Divisive Clusters:")

print(data[['Age group', 'Divisive Cluster']])

# Plot the Divisive clusters

plt.figure(figsize=(8, 4))

plt.scatter(data['Divisive Cluster'], data['Age group'], c=data['Divisive Cluster'], cmap='rainbow')

plt.title('Divisive Clustering---top-down approach---')

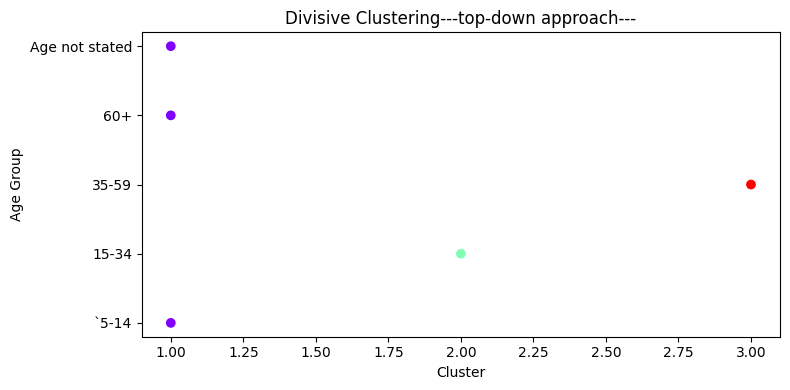
plt.xlabel('Cluster')

plt.ylabel('Age Group')

plt.tight\_layout()

plt.show()





PREPROCESSING AND LOADING DATASET :

* Data Validation : Loading and preprocessing allow us to check the data for errors, inconsistencies, and missing values. It's essential to ensure that our dataset is accurate and reliable for analysis.
* Data Cleaning : Preprocessing helps us to clean and fix issues in the data, such as removing duplicates, handling missing values, and dealing with outliers. Clean data leads to more robust and reliable results.
* Understanding the Data : Loading data allows you to examine its structure, dimensions, and contents. Preprocessing helps in summarizing and visualizing data, which is crucial for gaining insights and understanding its characteristics.
* Reducing Resource Usage : Preprocessing can involve reducing the memory or storage requirements of the dataset, making it more manageable for analysis.

Importing libraries :

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

Code :

import pandas as pd

# Load the dataset into a Pandas DataFrame

file\_path = 'DDW\_B06SC\_3300\_State\_TAMIL\_NADU-2011 (1).csv'

data = pd.read\_csv(file\_path)

# Display the first few rows of the dataset to get an overview

print(data.head())

# Check the data types of each column

print(data.dtypes)

# Summary statistics of the dataset

print(data.describe())

# Check for missing values

print(data.isnull().sum())

# If needed, you can perform additional preprocessing such as data cleaning, handling missing values, or renaming columns.

# Ensure that numeric columns are of the appropriate data type (int or float)

numeric\_columns = data.columns[1:]

data[numeric\_columns] = data[numeric\_columns].apply(pd.to\_numeric)

# Now, the dataset is loaded, and you can proceed with your data analysis and visualization.

Age group Worked for 3 months or more but less than 6 months - Persons \

0 Total 1200828

1 `5-14 27791

2 15-34 514340

3 35-59 542581

4 60+ 115103

Worked for 3 months or more but less than 6 months - Males \

0 589003

1 14125

2 259560

3 251957

4 62833

Worked for 3 months or more but less than 6 months - Females \

0 611825

1 13666

2 254780

3 290624

4 52270

Worked for less than 3 months - Persons \

0 221386

1 2447

2 92423

3 99202

4 27165

Worked for less than 3 months - Males \

0 99368

1 1247

2 43892

3 40691

4 13465

Worked for less than 3 months - Females \

0 122018

1 1200

2 48531

3 58511

4 13700

Industrial Category - A - Cultivators - Persons \

0 64235

1 1710

2 24863

3 29692

4 7930

Industrial Category - A - Cultivators - Males \

0 34632

1 825

2 12711

3 15927

4 5151

Industrial Category - A - Cultivators - Females ... \

0 29603 ...

1 885 ...

2 12152 ...

3 13765 ...

4 2779 ...

Industrial Category - N to O - Females \

0 3565

1 11

2 1754

3 1619

4 175

Industrial Category - P to Q - Persons \

0 11080

1 122

2 7536

3 3205

4 211

Industrial Category - P to Q - Males \

0 4019

1 71

2 2718

3 1131

4 93

Industrial Category - P to Q - Females \

0 7061

1 51

2 4818

3 2074

4 118

Industrial Category - R to U - HHI - Persons \

0 16833

1 427

2 8346

3 6591

4 1457

Industrial Category - R to U - HHI - Males \

0 4266

1 169

2 2127

3 1487

4 483

Industrial Category - R to U - HHI - Females \

0 12567

1 258

2 6219

3 5104

4 974

Industrial Category - R to U - Non HHI - Persons \

0 122088

1 19305

2 68929

3 26498

4 7065

Industrial Category - R to U - Non HHI - Males \

0 55801

1 9774

2 32803

3 9675

4 3394

Industrial Category - R to U - Non HHI - Females

0 66287

1 9531

2 36126

3 16823

4 3671

[5 rows x 64 columns]

Age group object

Worked for 3 months or more but less than 6 months - Persons int64

Worked for 3 months or more but less than 6 months - Males int64

Worked for 3 months or more but less than 6 months - Females int64

Worked for less than 3 months - Persons int64

...

Industrial Category - R to U - HHI - Males int64

Industrial Category - R to U - HHI - Females int64

Industrial Category - R to U - Non HHI - Persons int64

Industrial Category - R to U - Non HHI - Males int64

Industrial Category - R to U - Non HHI - Females int64

Length: 64, dtype: object

Worked for 3 months or more but less than 6 months - Persons \

count 6.000000e+00

mean 4.002760e+05

std 4.590478e+05

min 1.013000e+03

25% 4.961900e+04

50% 3.147215e+05

75% 5.355208e+05

max 1.200828e+06

Worked for 3 months or more but less than 6 months - Males \

count 6.000000

mean 196334.333333

std 223894.178166

min 528.000000

25% 26302.000000

50% 157395.000000

75% 257659.250000

max 589003.000000

Worked for 3 months or more but less than 6 months - Females \

count 6.000000

mean 203941.666667

std 235402.173170

min 485.000000

25% 23317.000000

50% 153525.000000

75% 281663.000000

max 611825.000000

Worked for less than 3 months - Persons \

count 6.000000

mean 73795.333333

std 84219.057978

min 149.000000

25% 8626.500000

50% 59794.000000

75% 97507.250000

max 221386.000000

Worked for less than 3 months - Males \

count 6.000000

mean 33122.666667

std 37567.179424

min 73.000000

25% 4301.500000

50% 27078.000000

75% 43091.750000

max 99368.000000

Worked for less than 3 months - Females \

count 6.000000

mean 40672.666667

std 46756.236588

min 76.000000

25% 4325.000000

50% 31115.500000

75% 56016.000000

max 122018.000000

Industrial Category - A - Cultivators - Persons \

count 6.000000

mean 21411.666667

std 24252.722080

min 40.000000

25% 3265.000000

50% 16396.500000

75% 28484.750000

max 64235.000000

Industrial Category - A - Cultivators - Males \

count 6.000000

mean 11544.000000

std 12978.785829

min 18.000000

25% 1906.500000

50% 8931.000000

75% 15123.000000

max 34632.000000

Industrial Category - A - Cultivators - Females \

count 6.000000

mean 9867.666667

std 11293.940104

min 22.000000

25% 1358.500000

50% 7465.500000

75% 13361.750000

max 29603.000000

Industrial Category - A - Agricultural labourers - Persons ... \

count 6.00000 ...

mean 302584.00000 ...

std 348616.58324 ...

min 557.00000 ...

25% 31129.75000 ...

50% 225372.50000 ...

75% 423894.00000 ...

max 907752.00000 ...

Industrial Category - N to O - Females \

count 6.000000

mean 1188.333333

std 1411.727122

min 6.000000

25% 52.000000

50% 897.000000

75% 1720.250000

max 3565.000000

Industrial Category - P to Q - Persons \

count 6.000000

mean 3693.333333

std 4648.545909

min 6.000000

25% 144.250000

50% 1708.000000

75% 6453.250000

max 11080.000000

Industrial Category - P to Q - Males \

count 6.000000

mean 1339.666667

std 1677.172104

min 6.000000

25% 76.500000

50% 612.000000

75% 2321.250000

max 4019.000000

Industrial Category - P to Q - Females \

count 6.000000

mean 2353.666667

std 2971.543886

min 0.000000

25% 67.750000

50% 1096.000000

75% 4132.000000

max 7061.000000

Industrial Category - R to U - HHI - Persons \

count 6.000000

mean 5611.000000

std 6478.460959

min 12.000000

25% 684.500000

50% 4024.000000

75% 7907.250000

max 16833.000000

Industrial Category - R to U - HHI - Males \

count 6.000000

mean 1422.000000

std 1616.378669

min 0.000000

25% 247.500000

50% 985.000000

75% 1967.000000

max 4266.000000

Industrial Category - R to U - HHI - Females \

count 6.000000

mean 4189.000000

std 4865.908425

min 12.000000

25% 437.000000

50% 3039.000000

75% 5940.250000

max 12567.000000

Industrial Category - R to U - Non HHI - Persons \

count 6.000000

mean 40696.000000

std 46571.506791

min 291.000000

25% 10125.000000

50% 22901.500000

75% 58321.250000

max 122088.000000

Industrial Category - R to U - Non HHI - Males \

count 6.000000

mean 18600.333333

std 21515.733645

min 155.000000

25% 4964.250000

50% 9724.500000

75% 27045.750000

max 55801.000000

Industrial Category - R to U - Non HHI - Females

count 6.000000

mean 22095.666667

std 25128.037565

min 136.000000

25% 5136.000000

50% 13177.000000

75% 31300.250000

max 66287.000000

[8 rows x 63 columns]

Age group 0

Worked for 3 months or more but less than 6 months - Persons 0

Worked for 3 months or more but less than 6 months - Males 0

Worked for 3 months or more but less than 6 months - Females 0

Worked for less than 3 months - Persons 0

..

Industrial Category - R to U - HHI - Males 0

Industrial Category - R to U - HHI - Females 0

Industrial Category - R to U - Non HHI - Persons 0

Industrial Category - R to U - Non HHI - Males 0

Industrial Category - R to U - Non HHI - Females 0

Length: 64, dtype: int64

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Print the column names in your dataset

print(data.columns)

# Load the dataset into a Pandas DataFrame

file\_path = 'DDW\_B06SC\_3300\_State\_TAMIL\_NADU-2011 (1).csv'

data = pd.read\_csv(file\_path)

Index(['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',

'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',

'Industrial Category - A - Agricultural labourers - Persons',

'Industrial Category - A - Agricultural labourers - Males',

'Industrial Category - A - Agricultural labourers - Females',

'Industrial Category - A - Plantation, Livestock, Forestry, Fishing, Hunting and allied activities - Persons',

'Industrial Category - A - Plantation, Livestock, Forestry, Fishing, Hunting and allied activities - Males',

'Industrial Category - A - Plantation, Livestock, Forestry, Fishing, Hunting and allied activities - Females',

'Industrial Category - B - Persons', 'Industrial Category - B - Males',

'Industrial Category - B - Females',

'Industrial Category - C - HHI - Persons',

'Industrial Category - C - HHI - Males',

'Industrial Category - C - HHI - Females',

'Industrial Category - C - Non HHI - Persons',

'Industrial Category - C - Non HHI - Males',

'Industrial Category - C - Non HHI - Females',

'Industrial Category - D & E - Persons',

'Industrial Category - D & E - Males',

'Industrial Category - D & E - Females',

'Industrial Category - F - Persons', 'Industrial Category - F - Males',

'Industrial Category - F - Females',

'Industrial Category - G - HHI - Persons',

'Industrial Category - G - HHI - Males',

'Industrial Category - G - HHI - Females',

'Industrial Category - G - Non HHI - Persons',

'Industrial Category - G - Non HHI - Males',

'Industrial Category - G - Non HHI - Females',

'Industrial Category - H - Persons', 'Industrial Category - H - Males',

'Industrial Category - H - Females',

'Industrial Category - I - Persons', 'Industrial Category - I - Males',

'Industrial Category - I - Females',

'Industrial Category - J - HHI - Persons',

'Industrial Category - J - HHI - Males',

'Industrial Category - J - HHI - Females',

'Industrial Category - J - Non HHI - Persons',

'Industrial Category - J - Non HHI - Males',

'Industrial Category - J - Non HHI - Females',

'Industrial Category - K to M - Persons',

'Industrial Category - K to M - Males',

'Industrial Category - K to M - Females',

'Industrial Category - N to O - Persons',

'Industrial Category - N to O - Males',

'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',

'Industrial Category - R to U - HHI - Males',

'Industrial Category - R to U - HHI - Females',

'Industrial Category - R to U - Non HHI - Persons',

'Industrial Category - R to U - Non HHI - Males',

'Industrial Category - R to U - Non HHI - Females'],

dtype='object')

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Load the dataset into a Pandas DataFrame

file\_path = 'DDW\_B06SC\_3300\_State\_TAMIL\_NADU-2011 (1).csv'

data = pd.read\_csv(file\_path)

# Example 1: Bar Chart

# Create a bar chart to visualize the total number of marginal workers in different age groups.

age\_groups = data['Age group']

total\_workers\_3\_6\_months = data['Worked for 3 months or more but less than 6 months - Males']

plt.figure(figsize=(10, 6))

plt.bar(age\_groups, total\_workers\_3\_6\_months)

plt.title('Total Number of Marginal Workers Working 3-6 Months by Age Group')

plt.xlabel('Age Group')

plt.ylabel('Total Count')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()

# Example 2: Pie Chart

# Create a pie chart to show the distribution of marginal workers by age group.

plt.figure(figsize=(6, 6))

plt.pie(total\_workers\_3\_6\_months, labels=age\_groups, autopct='%1.1f%%', startangle=140)

plt.title('Distribution of Marginal Workers Working 3-6 Months by Age Group')

plt.axis('equal')

plt.show()

# Example 3: Heatmap

# Create a heatmap to visualize the correlation between age groups and the total number of workers.

# Select relevant columns for correlation analysis

correlation\_data = data[['Age group', 'Worked for 3 months or more but less than 6 months - Males']]

# Convert age group to a numerical representation

age\_group\_mapping = {age\_group: index for index, age\_group in enumerate(age\_groups)}

correlation\_data['Age group'] = correlation\_data['Age group'].map(age\_group\_mapping)

plt.figure(figsize=(8, 6))

sns.heatmap(correlation\_data.corr(), annot=True, cmap='coolwarm', linewidths=0.5)

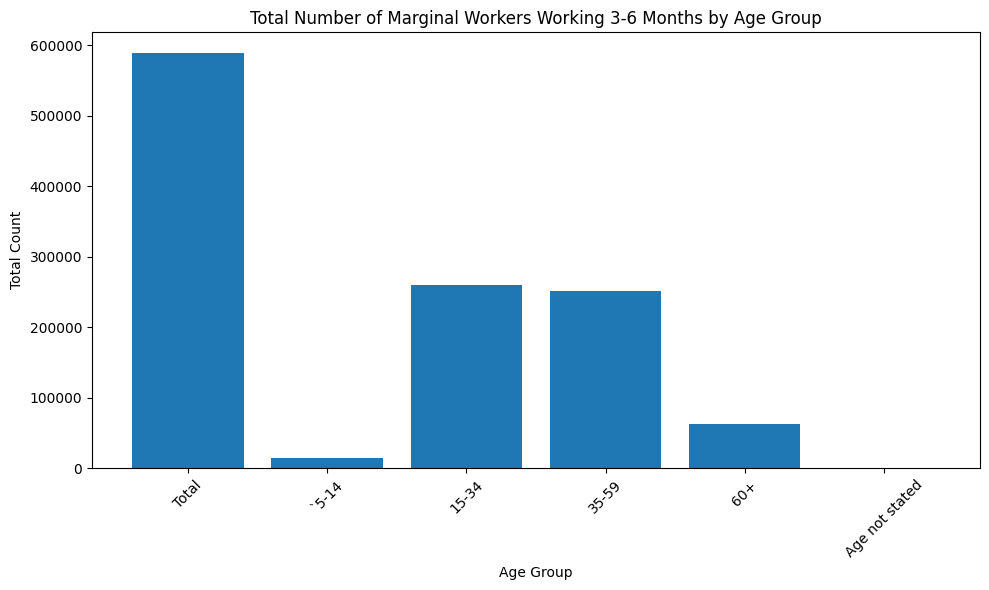
plt.title('Correlation Heatmap: Age Group vs. Workers (3-6 Months)')

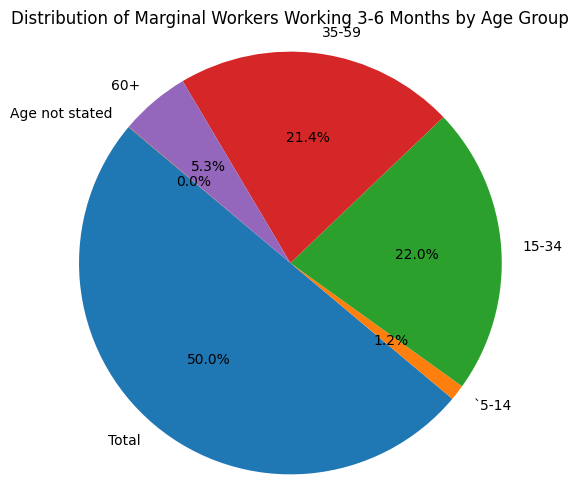
plt.xlabel('Age Group')

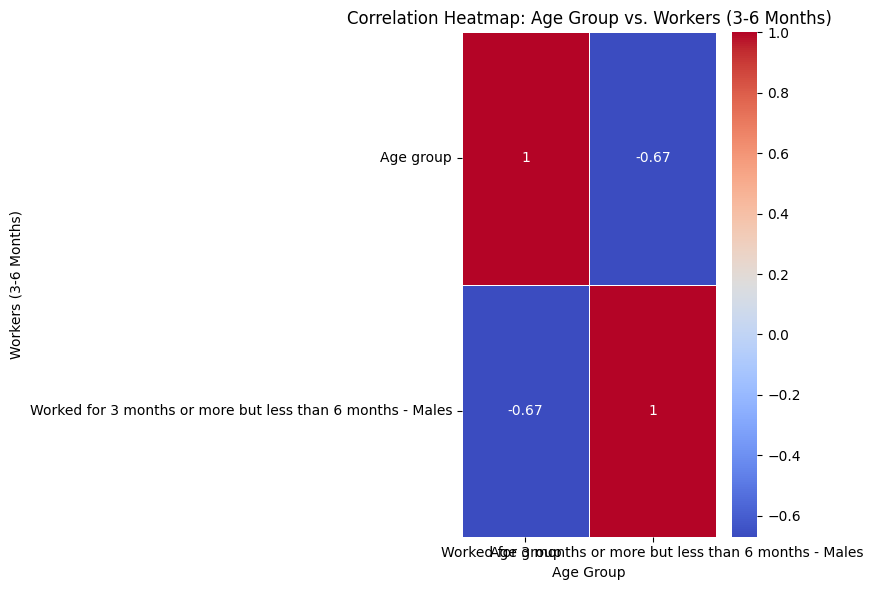
plt.ylabel('Workers (3-6 Months)')

plt.tight\_layout()

plt.show()







VISUALIZATION :

To perform a demographic analysis and create visualizations of the marginal workers in Tamil Nadu based on their industrial category, age and sex, the data for this analysis is obtained from the Census of India 2011, which provides information of the marginal workers classified by various criteria, such as age, sex and industry. The data is aggregated and manipulated using spreadsheet(Excel) and Python programming language to calculate the percentage and number of marginal workers in different age groups and industrial category. The results are then visualized using data visualization libraries(matplotlib and seaborn) to create charts and graphs that show the patterns and trends of marginal workers in Tamil Nadu. This analysis provides some recommendations for further research and policy interventions.

1. Visualization of Cultivators :

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Load the dataset into a Pandas DataFrame

file\_path = '/content/DDW\_B06SC\_3300\_State\_TAMIL\_NADU-2011.csv'

data = pd.read\_csv(file\_path)

# Example 1: Bar Chart

# Create a bar chart to visualize the total number of marginal workers in different age groups.

age\_groups = data['Age group']

Industrial\_category= data['Cultivators']

plt.figure(figsize=(5, 5))

plt.bar(age\_groups, Industrial\_category)

plt.title('Cultivators')

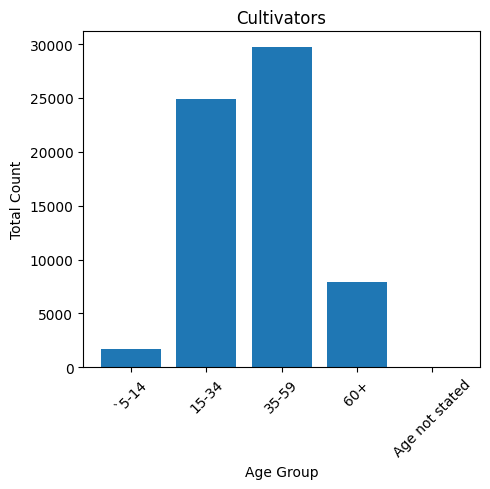
plt.xlabel('Age Group')

plt.ylabel('Total Count')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



The industrial category(Cultivators) consists most workers belonging to 35-59 age group.

1. Visualization of Agricultural laborers :

age\_groups = data['Age group']

Industrial\_category= data['Agricultural labourers']

plt.figure(figsize=(5, 5))

plt.bar(age\_groups, Industrial\_category)

plt.title('Agricultural labourers')

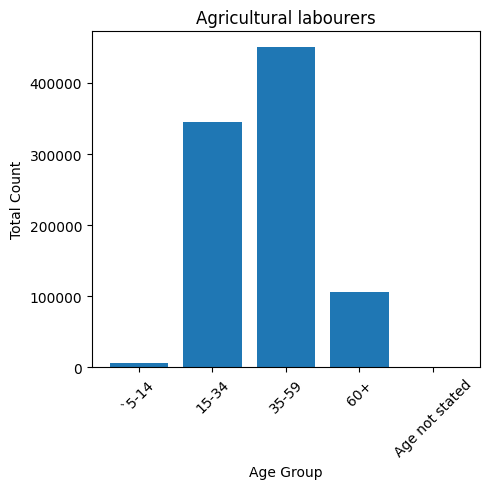
plt.xlabel('Age Group')

plt.ylabel('Total Count')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



The industrial category(Agricultural laborer’s) consists most workers belonging to 35-59 age group.

1. Visualization of Hunters and allied activists :

age\_groups = data['Age group']

Industrial\_category= data['Hunting and allied activities']

plt.figure(figsize=(5, 5))

plt.bar(age\_groups, Industrial\_category)

plt.title('Hunting and allied activities')

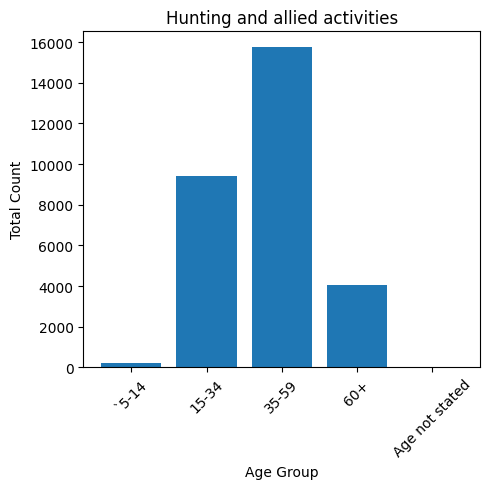
plt.xlabel('Age Group')

plt.ylabel('Total Count')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



The industrial category(Hunting and allied activities) consists most workers belonging to 35-59 age group.

1. Visualization of Industrial Category B Workers :

age\_groups = data['Age group']

Industrial\_category= data['Industrial Category - B ']

plt.figure(figsize=(5, 5))

plt.bar(age\_groups, Industrial\_category)

plt.title('Industrial Category - B ')

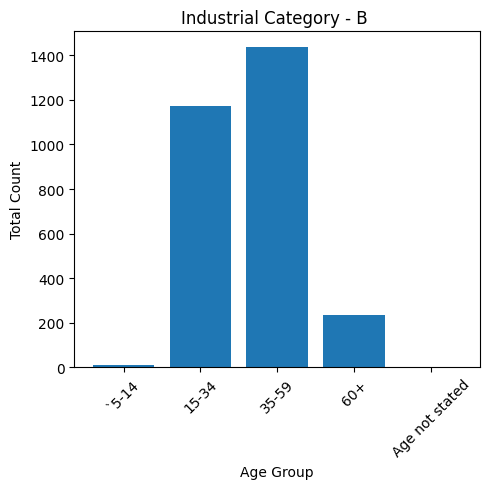
plt.xlabel('Age Group')

plt.ylabel('Total Count')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



The industrial category(Industrial Category – B) consists most workers belonging to 35-59 age group.

1. Visualization of Category – C – HHI Workers :

age\_groups = data['Age group']

Industrial\_category= data['Category - C - HHI']

plt.figure(figsize=(5, 5))

plt.bar(age\_groups, Industrial\_category)

plt.title('Category - C - HHI')

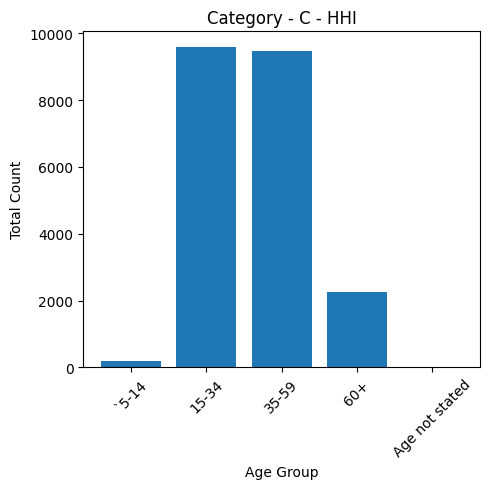
plt.xlabel('Age Group')

plt.ylabel('Total Count')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



The industrial category(Category – C - HHI) consists most workers belonging to 15-34 age group.

1. Visualization of Category – C – Non HHI Workers :

age\_groups = data['Age group']

Industrial\_category= data['Category - C - Non HHI']

plt.figure(figsize=(5, 5))

plt.bar(age\_groups, Industrial\_category)

plt.title('Category - C - Non HHI')

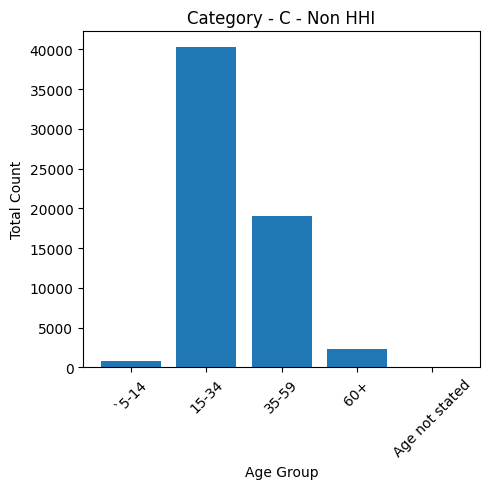
plt.xlabel('Age Group')

plt.ylabel('Total Count')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



The industrial category(Category – C – Non HHI) consists most workers belonging to 15-34 age group.

1. Visualization of Category – D&E Workers :

age\_groups = data['Age group']

Industrial\_category= data['Category - D & E']

plt.figure(figsize=(5, 5))

plt.bar(age\_groups, Industrial\_category)

plt.title('Category - D & E')

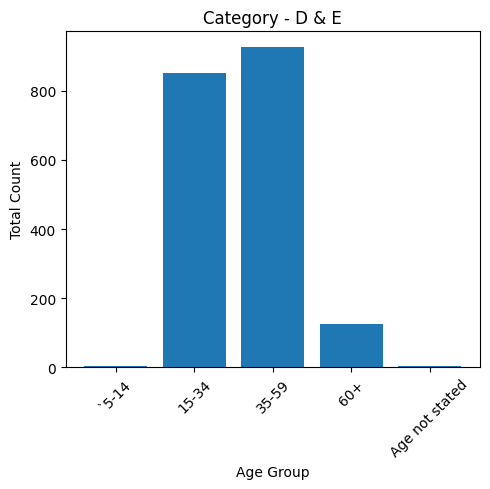
plt.xlabel('Age Group')

plt.ylabel('Total Count')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



The industrial category(Category – D&E­) consists most workers belonging to 35-59 age group.

1. Visualization of Category – F Workers :

age\_groups = data['Age group']

Industrial\_category= data['Category - F']

plt.figure(figsize=(5, 5))

plt.bar(age\_groups, Industrial\_category)

plt.title('Category - F')

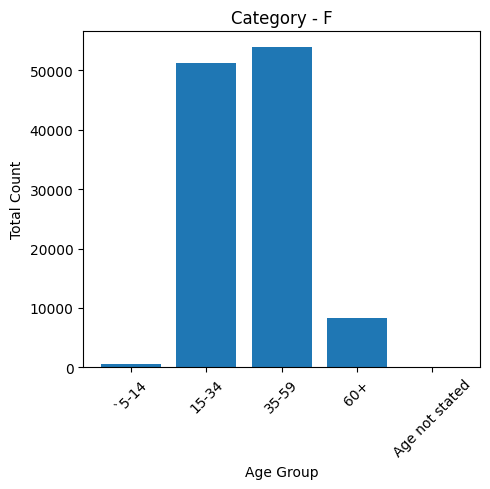
plt.xlabel('Age Group')

plt.ylabel('Total Count')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



The industrial category(Category - F) consists most workers belonging to 35-59 age group.

1. Visualization of Category – G – Non HHI Workers :

age\_groups = data['Age group']

Industrial\_category= data['Category - G - Non HHI']

plt.figure(figsize=(5, 5))

plt.bar(age\_groups, Industrial\_category)

plt.title('Category - G - Non HHI')

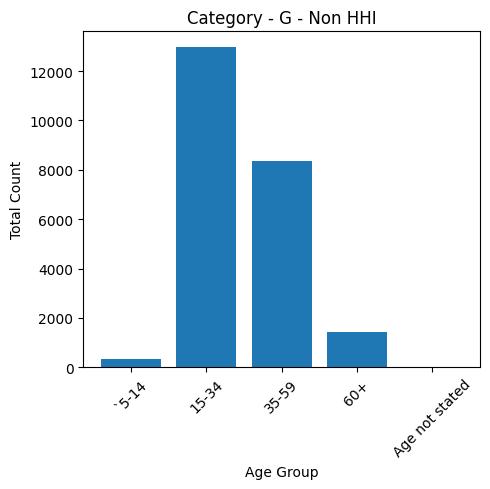
plt.xlabel('Age Group')

plt.ylabel('Total Count')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



The industrial category(Category - G – Non HHI) consists most workers belonging to 15-34 age group.

1. Visualization of Category – H Workers :

age\_groups = data['Age group']

Industrial\_category= data['Category - H']

plt.figure(figsize=(5, 5))

plt.bar(age\_groups, Industrial\_category)

plt.title('Category - H')

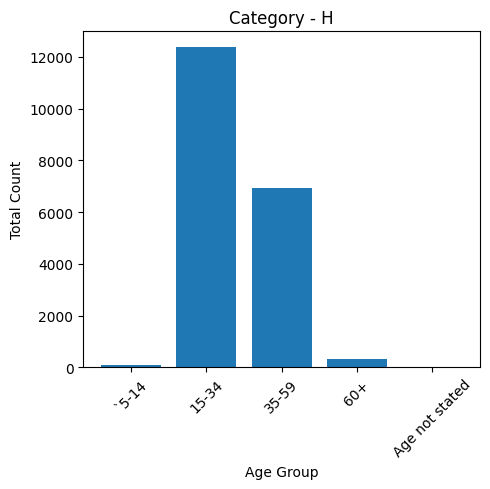
plt.xlabel('Age Group')

plt.ylabel('Total Count')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



The industrial category(Category - H) consists most workers belonging to 15-34 age group.

1. Visualization of Category – I Workers :

age\_groups = data['Age group']

Industrial\_category= data['Category - I']

plt.figure(figsize=(5, 5))

plt.bar(age\_groups, Industrial\_category)

plt.title('Category - I')

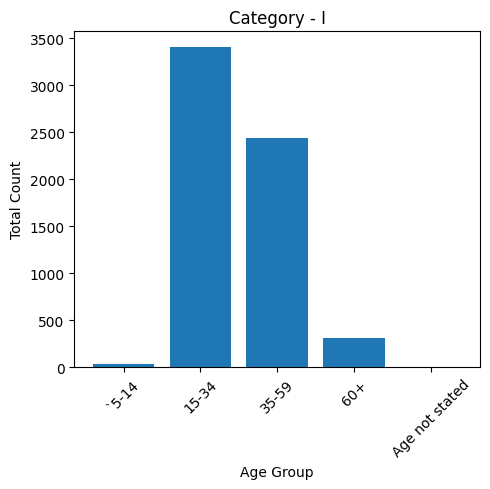
plt.xlabel('Age Group')

plt.ylabel('Total Count')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



The industrial category(Category - I) consists most workers belonging to 15-34 age group.

1. Visualization of Category – J – HHI Workers :

age\_groups = data['Age group']

Industrial\_category= data['Category - J - HHI']

plt.figure(figsize=(5, 5))

plt.bar(age\_groups, Industrial\_category)

plt.title('Category - J - HHI')

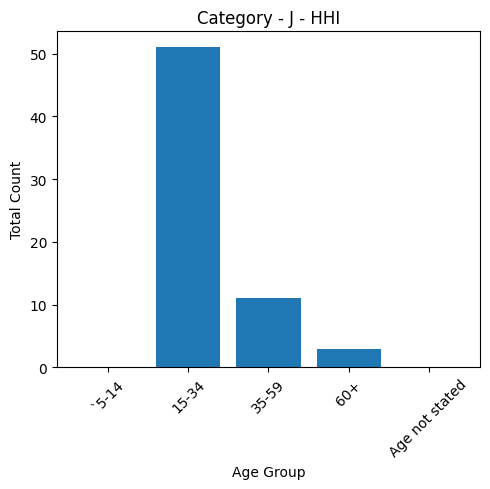
plt.xlabel('Age Group')

plt.ylabel('Total Count')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



The industrial category(Category – J - HHI) consists most workers belonging to 15-34 age group.

1. Visualization of Category – J – Non HHI Workers :

age\_groups = data['Age group']

Industrial\_category= data['Category - J - Non HHI']

plt.figure(figsize=(5, 5))

plt.bar(age\_groups, Industrial\_category)

plt.title('Category - J - Non HHI')

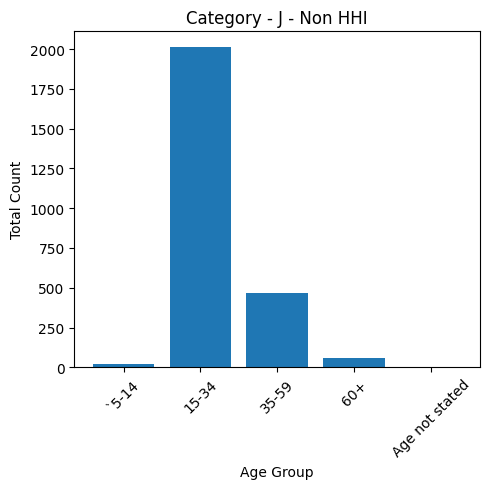
plt.xlabel('Age Group')

plt.ylabel('Total Count')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



The industrial category(Category – J - Non HHI) consists most workers belonging to 15-34 age group.

1. Visualization of Category – K to M Workers :

age\_groups = data['Age group']

Industrial\_category= data['Category - K to M']

plt.figure(figsize=(5, 5))

plt.bar(age\_groups, Industrial\_category)

plt.title('Category - K to M  ')

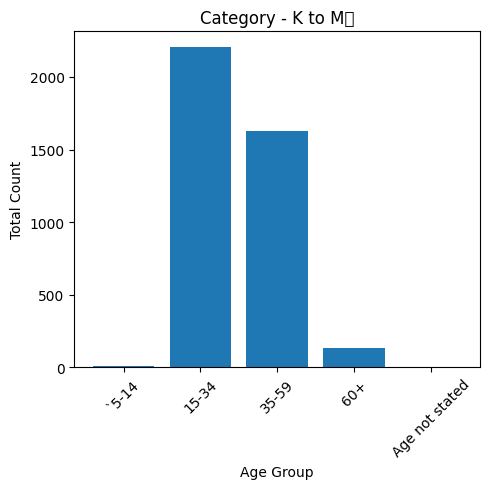
plt.xlabel('Age Group')

plt.ylabel('Total Count')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



The industrial category(Category - K to M) consists most workers belonging to 15-34 age group.

1. Visualization of Category – N to O Workers :

age\_groups = data['Age group']

Industrial\_category= data['Category - N to O']

plt.figure(figsize=(5, 5))

plt.bar(age\_groups, Industrial\_category)

plt.title('Category - N to O')

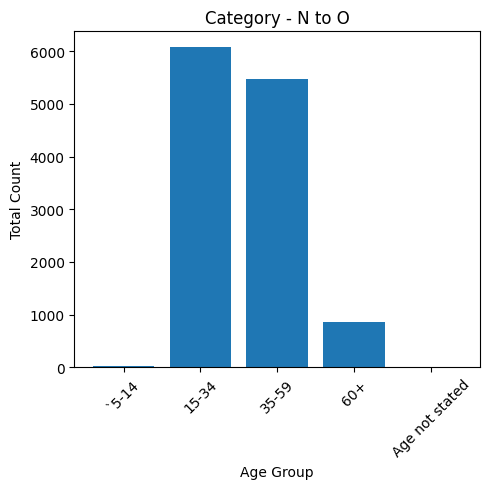
plt.xlabel('Age Group')

plt.ylabel('Total Count')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



The industrial category(Category – N to O) consists most workers belonging to 15-34 age group.

1. Visualization of Category – P to Q Workers :

age\_groups = data['Age group']

Industrial\_category= data['Category - P to Q']

plt.figure(figsize=(5, 5))

plt.bar(age\_groups, Industrial\_category)

plt.title('Category - P to Q')

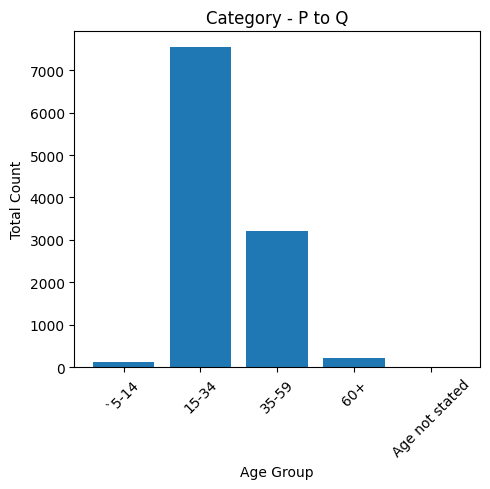
plt.xlabel('Age Group')

plt.ylabel('Total Count')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



The industrial category(Category – P to Q) consists most workers belonging to 15-34 age group.

1. Visualization of Category – R to U HHI Workers :

age\_groups = data['Age group']

Industrial\_category= data['Category - R to U - HHI']

plt.figure(figsize=(5, 5))

plt.bar(age\_groups, Industrial\_category)

plt.title('Category - R to U - HHI')

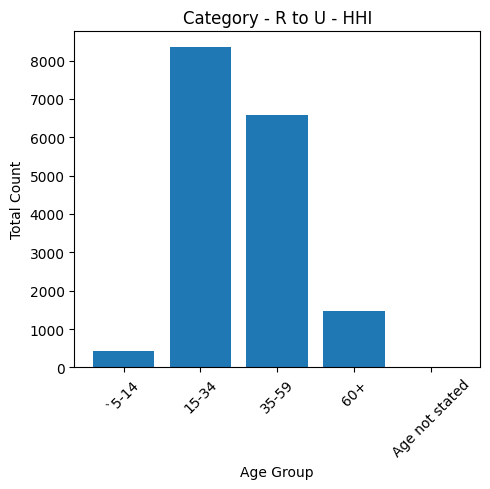
plt.xlabel('Age Group')

plt.ylabel('Total Count')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



The industrial category(Category – R to U - HHI) consists most workers belonging to 15-34 age group.

1. Visualization of Category – R to U Non HHI Workers :

age\_groups = data['Age group']

Industrial\_category= data['Category - R to U - Non HHI']

plt.figure(figsize=(5, 5))

plt.bar(age\_groups, Industrial\_category)

plt.title('Category - R to U - Non HHI')

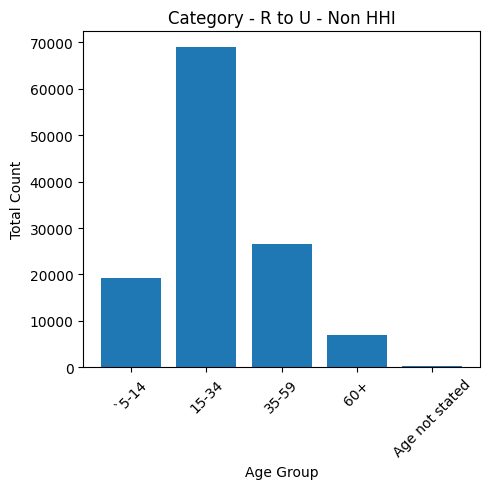
plt.xlabel('Age Group')

plt.ylabel('Total Count')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()



The industrial category(Category – R to U - Non HHI) consists most workers belonging to 15-34 age group.

1. Visualization of Different Industrial Categories Workers :

#import the libraries

import pandas as pd

import matplotlib.pyplot as plt

#providing the dataset to visualise

data = pd.read\_csv('/content/industrial\_category.csv')

categories = data['Industrial categories']

values = data['total number of workers']

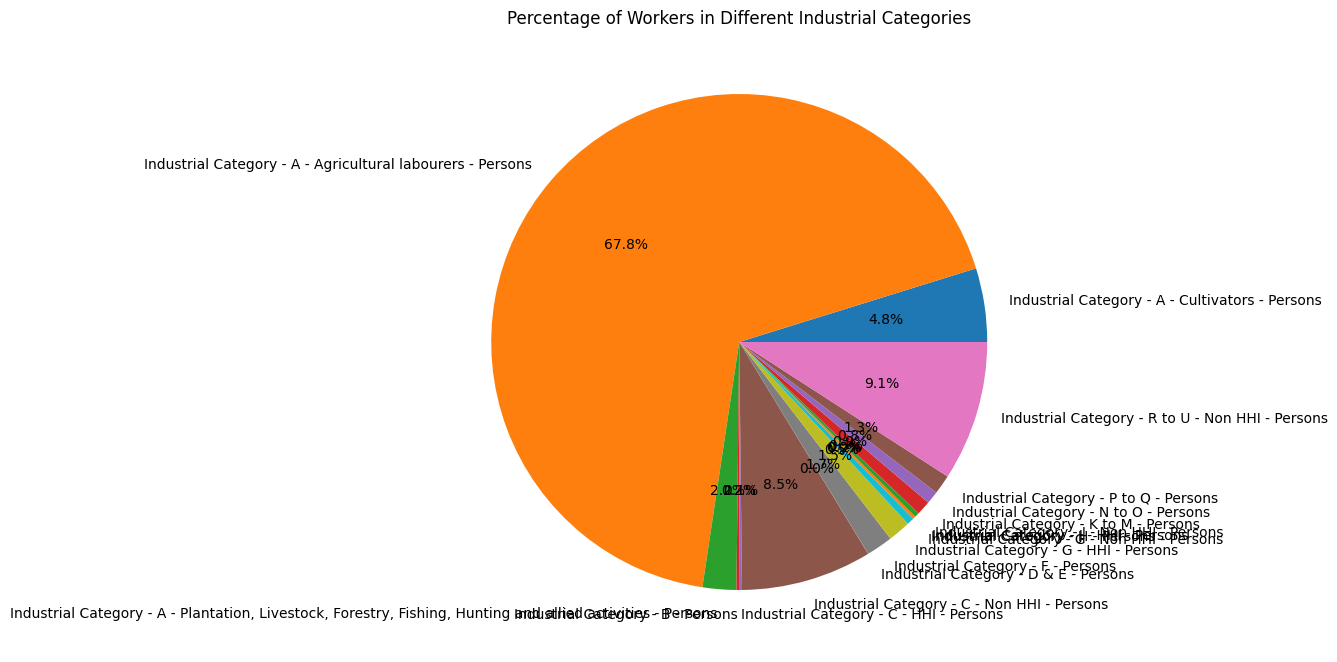
#plot the pie chart

plt.figure(figsize=(8, 18))

plt.pie(values, labels=categories, autopct='%1.1f%%', startangle=0)

plt.title('Percentage of Workers in Different Industrial Categories')

plt.show()



The above visualization shows that 67.8% of workers are Agricultural laborer’s which is the majority industry in Tamil Nadu.

1. Visualization of Different Age group Workers :

#import the libraries

import pandas as pd

import matplotlib.pyplot as plt

#providing the dataset to visualise

data = pd.read\_csv('/content/dataset.csv')

categories = data['Age group']

values = data['sumation']

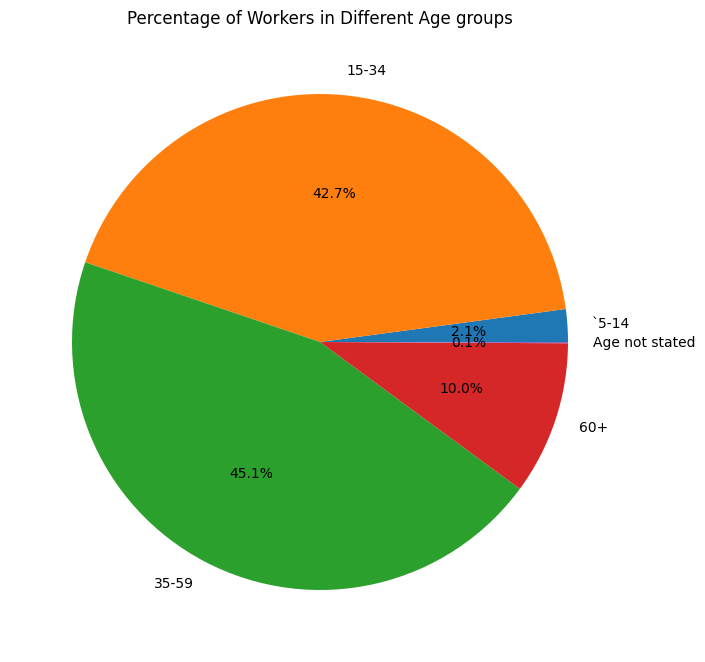
#plot the pie chart

plt.figure(figsize=(8, 18))

plt.pie(values, labels=categories, autopct='%1.1f%%', startangle=0)

plt.title('Percentage of Workers in Different Age groups’)

plt.show()



The above visualization shows that the workers under the age group of 35-59 are the majority workers in Tamil Nadu(45.1%) followed by the workers under the age group 15-34 (42.7%).

CONCLUSION :

Conducting a socioeconomic analysis of marginal workers by considering demographic factors such as age, industrial category, and sex provides valuable insights into their workforce dynamics. Through the use of bar charts, pie charts, and heatmaps, a comprehensive understanding of the distribution across these categories emerges. These visual representations not only enhance data interpretation but also facilitate informed decision-making for policymakers, employers, and social organizations. By recognizing patterns and disparities, targeted interventions can be designed to improve the conditions of marginal workers, fostering a more inclusive and equitable society.