**Assessment** **of** **Marginal** **Workers** **in** **TamilNadu**

**Phase 5:** Project Documentation & Submission

**Dataset Link:** https://tn.data.gov.in/catalog/marginal-workers-classified-age-industrial-category-and-sex-census-2011-india-and-states

**Objectives:**

* The primary objective of this project is to analyse the demographic characteristics of marginal workers in Tamil Nadu using the Census 2011 dataset.
* Specifically, we aim to identify and visualize age, industrial category, and area distribution among marginal workers.
* Gain insights into the overall socio-economic status of marginal workers in the region.

**Analysis** **Approach:**

* **Data Collection**
* Downloaded the dataset from the provided link
* Identify and gather relevant datasets containing information on marginal workers in Tamil Nadu.
* Ensure the datasets include variables such as age, gender, industrial category, and any other pertinent information.
* **Data Preprocessing**
* Cleaned and transformed the data for analysis.
* Cleaning and formatting the data to ensure accuracy and consistency.
* Handling missing values and outliers appropriately.
* **Data Analysis**
  + - Conducted Exploratory Data Analysis (EDA) to understand the dataset’s structure and identify key insights.
* **Data Visualization**
  + - Based on the analysis, select appropriate visualization types for each aspect of the demographic characteristics.
    - Utilize python libraries like Matplotlib and Seaborn to create visualization
* **Interpretation**
  + - Interpreted the findings from the analysis and visualizations.

**Visualization** **Types:**

* **Bar charts** for visualizing the distribution of marginal workers by age groups, industrial categories, area and gender.
* **Pie charts** to show the gender distribution among marginal workers.
* **Histograms** to visualize the age distribution.

**Code** **Implementation:**

#import necessary libararies

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

#use the 'pd.read\_csv()' function to load dataset into a pandas

df=pd.read\_csv(r"C:\Users\fathi\OneDrive\Documents\dataset.csv")

#explore the loaded dataset to get an overview of its structure and contents:

#display the first rows of the dataframe

print(df.head())

Table Code State Code District Code Area Name Total/ Rural/ Urban \

0 B0806SC `33 `000 State - TAMIL NADU Total

1 B0806SC `33 `000 State - TAMIL NADU Total

2 B0806SC `33 `000 State - TAMIL NADU Total

3 B0806SC `33 `000 State - TAMIL NADU Total

4 B0806SC `33 `000 State - TAMIL NADU Total

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 16823

4 3671

[5 rows x 69 columns]

#display basic information about the dataframe

print(df.info())

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 594 entries, 0 to 593

Data columns (total 69 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Table Code 594 non-null object

1 State Code 594 non-null object

2 District Code 594 non-null object

3 Area Name 594 non-null object

4 Total/ Rural/ Urban 594 non-null object

5 Age group 594 non-null object

6 Worked for 3 months or more but less than 6 months - Persons 594 non-null int64

7 Worked for 3 months or more but less than 6 months - Males 594 non-null int64

8 Worked for 3 months or more but less than 6 months - Females 594 non-null int64

9 Worked for less than 3 months - Persons 594 non-null int64

10 Worked for less than 3 months - Males 594 non-null int64

11 Worked for less than 3 months - Females 594 non-null int64

12 Industrial Category - A - Cultivators - Persons 594 non-null int64

13 Industrial Category - A - Cultivators - Males 594 non-null int64

14 Industrial Category - A - Cultivators - Females 594 non-null int64

15 Industrial Category - A - Agricultural labourers - Persons 594 non-null int64

16 Industrial Category - A - Agricultural labourers - Males 594 non-null int64

17 Industrial Category - A - Agricultural labourers - Females 594 non-null int64

18 Industrial Category - A - Plantation, Livestock, Forestry, Fishing, Hunting and allied activities - Persons 594 non-null int64

19 Industrial Category - A - Plantation, Livestock, Forestry, Fishing, Hunting and allied activities - Males 594 non-null int64

...

68 Industrial Category - R to U - Non HHI - Females 594 non-null int64

dtypes: int64(63), object(6)

#display summary statistics

print(df.describe())

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

count 5.940000e+02

mean 1.617277e+04

std 7.607172e+04

min 0.000000e+00

25% 2.872500e+02

50% 2.225500e+03

75% 9.628500e+03

max 1.200828e+06

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

count 594.000000

mean 7932.700337

std 36864.822704

min 0.000000

25% 147.250000

50% 1147.000000

75% 4770.500000

max 589003.000000

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

count 594.000000

mean 8240.067340

std 39259.545337

min 0.000000

...

75% 500.000000

max 66287.000000

[8 rows x 63 columns]

#data preprocessing

#check for missing values

print(df.isnull().sum())

Table Code 0

State Code 0

District Code 0

Area Name 0

Total/ Rural/ Urban 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: 69, dtype: int64

#handle missing values

df=df.dropna()

#remove duplicates

df=df.drop\_duplicates()

#convert data types

#df['column\_name']=df['column\_name'].astype('desired\_type')

#save the dataframe to a new CSV file

df.to\_csv(r'C:\Users\fathi\OneDrive\Documents\cleaned\_dataset.csv',index=False)

#demographic analysis

#Calculate the distribution of marginal workers based on age, industrial category, and area

#group by age, industrial category, and area, and calculate the counts

age\_distribution=df.groupby('Age group').size()

industrial\_category\_distribution=df.groupby('Industrial Category - A - Cultivators - Persons').size()

area\_distribution=df.groupby('Area Name').size()

#Data Visualization

#Age Distribution

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

sns.barplot(x=age\_distribution.index,y=age\_distribution.values)

plt.title('Age Distribution of Marginal Workers')

plt.xlabel('Age')

plt.ylabel('Count')

plt.xticks(rotation=45)

plt.show()



#industriaal Category distribution

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

sns.barplot(x=industrial\_category\_distribution.index,y=industrial\_category\_distribution.values)

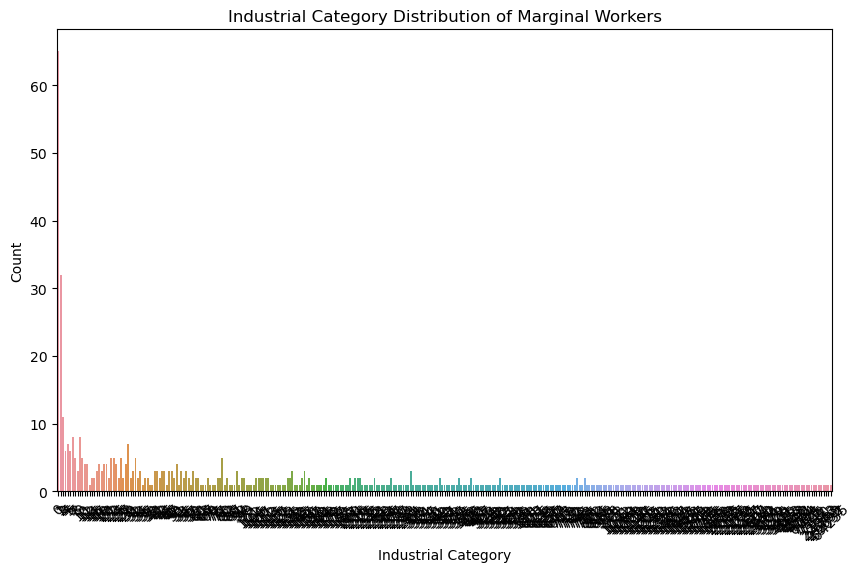
plt.title('Industrial Category Distribution of Marginal Workers')

plt.xlabel('Industrial Category')

plt.ylabel('Count')

plt.xticks(rotation=45)

plt.show()



#Area Distribution

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

sns.barplot(x=area\_distribution.index,y=area\_distribution.values)

plt.title('Area Distribution of Marginal Workers')

plt.xlabel('Area')

plt.ylabel('Count')

plt.show()

