**ASSESSMENT OF MARGINAL WORKER IN TAMILNADU-A SOCIOECONOMIC ANALYSIS**

**PHASE 3:** Development Part 2

**TOPIC:** In this part you will continue building your project.

Perform the demographic analysis

Calculating the distribution of marginal workers based on age, industrial category, and sex using data aggregation and manipulation.

Creating visualizations using data visualization libraries (e.g., Matplotlib, Seaborn).

**Introduction:**

• Socioeconomic analysis is a multidisciplinary approach that examines the complex interplay between social and economic factors within a given society or community. It aims to understand and assess the impact of economic policies, practices, and disparities on the well-being and quality of life of individuals and groups.

• Socioeconomic analysis relies heavily on data collection, statistical analysis, and research. Researchers often use surveys, censuses, and other data sources to gather information that informs their analysis.

**Dataset Overview:**

Marginal workers dataset offers a comprehensive collection of data points, capturing a wide array of socioeconomic variables. This dataset encompasses factors such as income, education, employment, healthcare, demographic information, and more. By leveraging this dataset, we aim to gain valuable insights into the socioeconomic well-being and challenges faced by the target population.

**Given Dataset:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| District Code | Area Name | Total/  Rural  / Urban | .... | Industrial  Category - R  to U  - Non HHI –  Persons | Industrial  Category –  R to U  - Non HHI  - Males | Industrial Category - R to U - Non HHI - Females |
| `000 | State –  TAMIL NADU | Total | .... | 122088 | 55801 | 66287 |
| `000 | State –  TAMIL NADU | Total | .... | 19305 | 9774 | 9531 |
| `000 | State –  TAMIL NADU | Total | ... | 68929 | 32803 | 36126 |
| .  .  .  .  .  . | .  .  .  .  .  . | .  .  .  .  .  . | ....  ....  ....  ....  ....  .... | .  .  .  .  .  . | .  .  .  .  .  . | .  .  .  .  .  . |
| `633 | District - Tiruppur | Urban | .... | 279 | 103 | 176 |
| `633 | District - Tiruppur | Urban | .... | 81 | 35 | 46 |
| `633 | District - Tiruppur | Urban | .... | 0 | 0 | 0 |

595 rows x 69 columns

**Overview of the process:**

Performing demographic analysis and calculating the distribution of marginal workers based on age, industrial category, and sex often involving aggregation and manipulation data. After that creating visualization using python libraries .

1.Load and Prepare Data:

Load the dataset using Pandas and prepare it for analysis.

2.Data Aggregation:

Use Pandas to aggregate the data based on age, industrial category, and sex. You can use functions like ‘groupby’ and ‘pivot\_table’ to aggregate the data as needed.

3.Calculate Distribution:

Calculate the distribution of marginal workers in each category. This involves counting the number of workers in each age group, industrial category, and sex category.

4.Create Visualizations:

Visualize the distribution using data visualization libraries like Matplotlib or Seaborn.

**1. Load the dataset:**

Load your dataset into a Pandas. You can typically find house price datasets in CSV format, but you can adapt this code to other formats as needed.

Program:

data = pd.read\_csv(' https://tn.data.gov.in/catalog/marginal-workers-classified-age-industrial-category-and-sex-census-2011-india-and-states.csv ')

print(data.read())

**2. Data Aggregation:**

Aggregation in pandas provides various functions that perform a mathematical or logical operation on our dataset and returns a summary of that function. Aggregation can be used to get a summary of columns in our dataset like getting sum, minimum, maximum, etc. from a particular column of our dataset.

Program:

# Aggregate data based on a specific column (e.g., 'Category') and calculate the sum of another column (e.g., 'Value’)

print("\nAggregated Data:")

print(data.describe())

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

#Use Pandas to aggregate the data based on age, industrial category, and sex. You can use functions like ‘groupby’ and ‘pivot\_table’ to aggregate the data as needed.

print(data.groupby('Area Name')['State Code'].sum().reset\_index())

pivot\_table = pd.pivot\_table(data, values='Worked for 3 months or more but less than 6 months - Persons', index='Table Code', columns='Age group', aggfunc='sum', fill\_value=0)

print(pivot\_table)

We use the pd.pivot\_table function to create a pivot table. The values parameter is set to 'Worked for 3 months or more but less than 6 months - Persons', the index parameter is set to 'Table Code', and the columns parameter is set to 'Age group'. We use the 'sum' aggregation function to calculate the 'Worked for 3 months or more but less than 6 months - Persons' for each combination of 'Table Code' and 'Industrial Category'.

The fill\_value parameter is set to 0, which means that missing combinations will have a value of 0 in the pivot table.

**Program:**

**In:**

import pandas as pd

import numpy as np

import plotly.express as px

import matplotlib.pyplot as plt

data = pd.read\_csv('DDW\_B06SC\_3300\_State\_TAMIL\_NADU-2011 (1).csv')

print(data.head())

print("\nAggregated Data:")

print(data.describe())

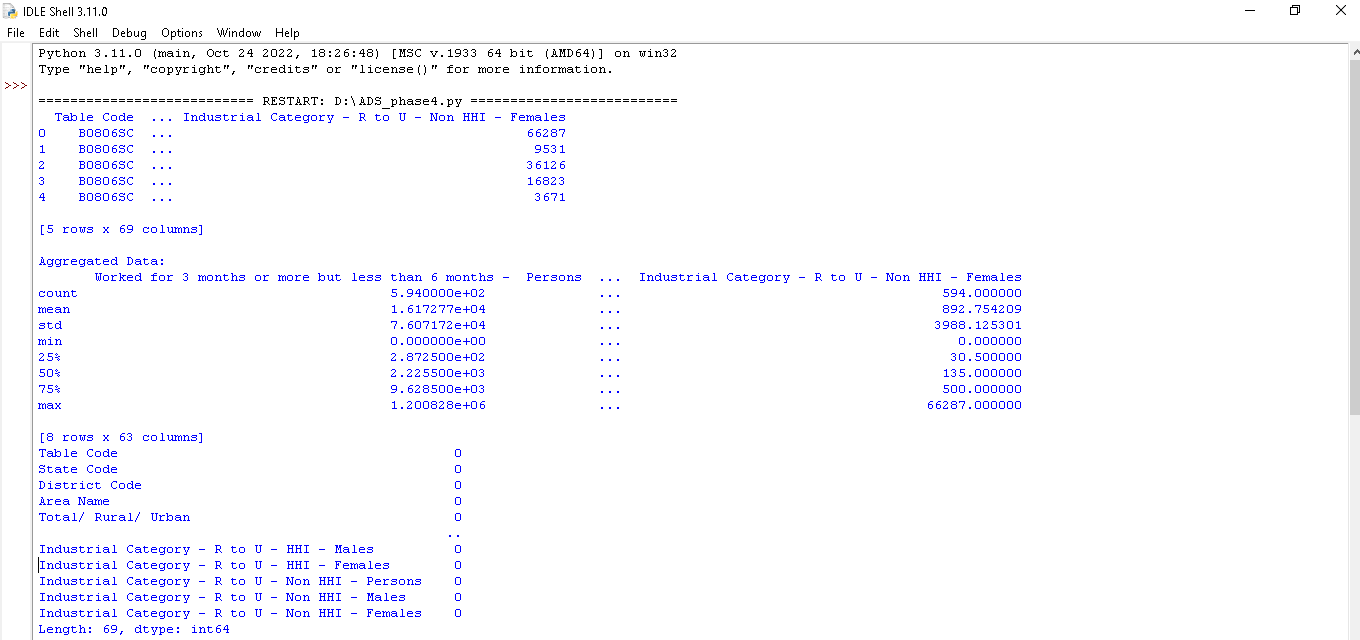
print(data.isnull().sum())

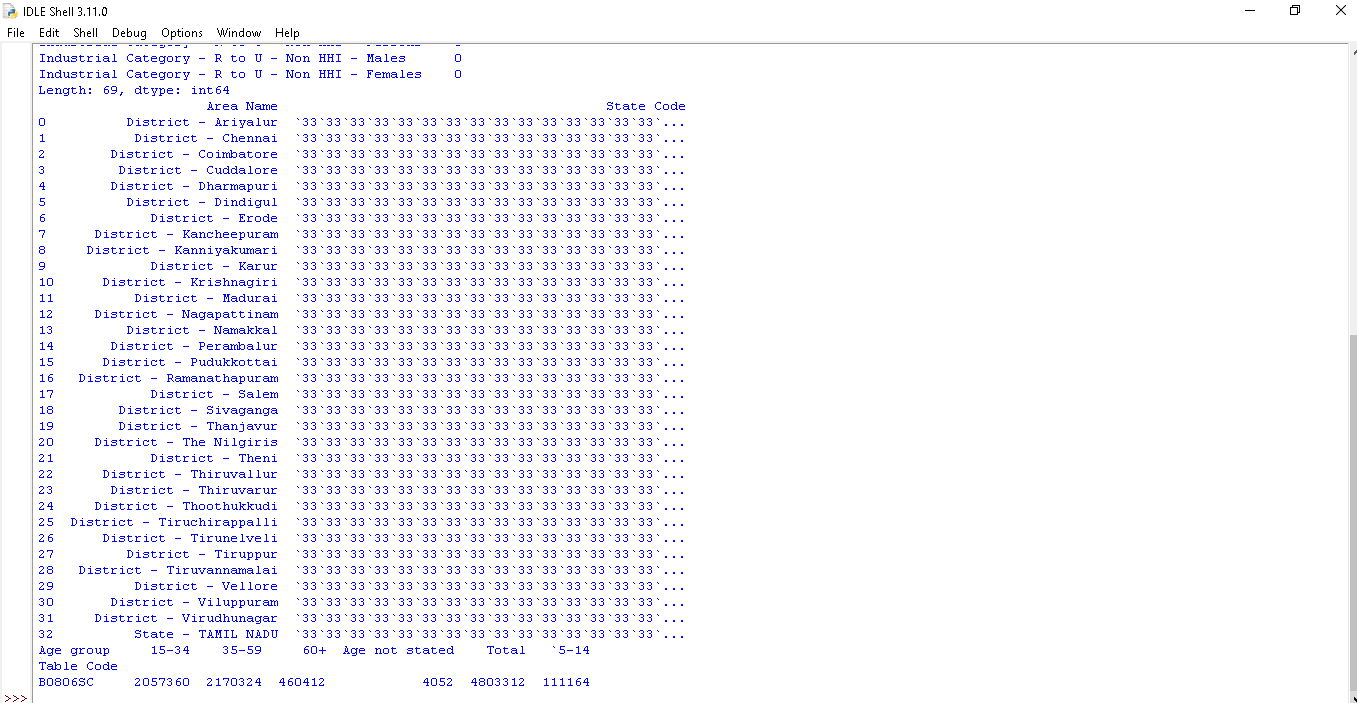
print(data.groupby('Area Name')['State Code'].sum().reset\_index())

pivot\_table = pd.pivot\_table(data, values='Worked for 3 months or more but less than 6 months - Persons', index='Table Code', columns='Age group', aggfunc='sum', fill\_value=0)

print(pivot\_table)

**Out:**



****

**3. Data Distribution:**

**1.** **Univariate Data Distribution:**

To explore the distribution of individual variables, consider the following methods:

* Histograms: Create histograms to visualize the distribution of numerical data using df['column'].hist().
* Box Plots: Use box plots to visualize the distribution and identify outliers with df.boxplot(column='column').
* Kernel Density Estimation (KDE) Plots: Create KDE plots for a smooth estimation of the data distribution with sns.kdeplot(df['column']) (if using Seaborn).

**2. Bivariate Data Distribution:**

* Analyzing relationships between variables:
* Scatter Plots: Create scatter plots to visualize relationships between two numerical variables with df.plot.scatter(x='x\_column', y='y\_column').
* Pair Plots: Use pair plots (if using Seaborn) to visualize bivariate relationships for multiple variables with sns.pairplot(df).
* Correlation Matrix: Calculate and visualize the correlation between variables using df.corr() and a heatmap (if using Seaborn) for better visualization.

**3. Categorical Data Distribution:**

For categorical variables:

* Bar Plots: Create bar plots to visualize the distribution of categorical data using df['column'].value\_counts().plot(kind='bar').
* Count Plots: Use count plots (if using Seaborn) to visualize the count of each category with sns.countplot(x='column', data=df).

**4. Probability Distribution Functions:**

* To understand the probability distribution of numerical variables:
* Kernel Density Estimation (KDE) Plots: Visualize the PDF using sns.kdeplot(df['column']PDF):

**5. Visualize Data Distribution:**

* To create more advanced visualizations, use Matplotlib and Seaborn. Customize plots by specifying titles, labels, and colors to communicate the distribution more effectively.

**6. Interpret the Results:**

* Based on your visualizations and data exploration, make interpretations about the data distribution, outliers, correlations, and any insights you've gained.
* Data distribution analysis is essential for understanding the characteristics of your data, which can inform further data processing, modeling, and decision-making.

**Program:**

**In:**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Load and prepare data

data = pd.read\_csv('DDW\_B06SC\_3300\_State\_TAMIL\_NADU-2011 (1).csv')

# Data Aggregation

# Assuming 'age', 'industrial\_category', and 'sex' are columns in your dataset

aggregated\_data = data.groupby(['Age group', 'Industrial Category - A - Plantation, Livestock, Forestry, Fishing, Hunting and allied activities - Persons', 'Total/ Rural/ Urban']).size().reset\_index(name='count')

# Create Visualizations

# You can use various types of plots depending on your analysis needs

# Bar chart for distribution of marginal workers by age group

plt.figure(figsize=(12, 6))

sns.barplot(x='Age group', y='count', data=aggregated\_data, hue='Total/ Rural/ Urban')

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

plt.xlabel('Age group')

plt.ylabel('Count')

plt.xticks(rotation=45)

plt.show()

**Out:**

