Marginal Workers in Tamil Nadu

INTRODUCTION

To work with data on marginal workers in Tamil Nadu, including district code, state, area, age, and working periods, Python provides a powerful toolkit. By following these steps, you will gain insights from your dataset and uncover valuable information to drive informed decisions. This guide will walk you through the process.

Data Preparation

Format and Structure

Examine the dataset's format and structure, discussing the specific variables and their meanings. Gain an understanding of the information captured, including district code, state, area, age, and working periods. Properly comprehending the dataset's structure is crucial for accurate analysis.

Importing into Python

Learn how to import the dataset into Python, leveraging the pandas library to efficiently handle large datasets. Familiarize yourself with the necessary syntax and functions to read the CSV file, ensuring the data is ready for analysis. Master the art of seamless data integration.

Data Manipulation

"Data never sleeps, and neither should your analysis. Dive into the world of data manipulation and cleaning to ensure your dataset is pristine. In this section, we will explore techniques to handle missing values, outliers, and inconsistencies with precision and elegance."

Data Visualization

Data Extraction

First, let's extract and load the dataset into a Pandas DataFrame. Make sure to have the Pandas library installed Unleash the power of visual storytelling by incorporating captivating visualizations into your analysis. Learn how to create bar charts, scatter plots, and interactive maps to bring your data to life. Engage your audience and make meaningful connections.

Data Preprocessing

Now, we'll preprocess the data to prepare it for visualization. Preprocessing can include cleaning, filtering, and aggregating the data.Leave a lasting impact with your data-driven conclusions We group the data based on age and working periods, then create a stacked bar chart to visualize the count of workers in different age groups for each working period.

- The legend and bbox_to_anchor parameters are used to position the legend outside the plot to avoid overlapping with the bars.

IBM COGNOS

IBM Cognos is a business intelligence and performance management software suite that allows organizations to perform data analysis, reporting, and dashboarding. It offers a range of tools and features to conduct data analysis. Here's an overview of how you can perform data analysis in IBM Cogno

IBM Cognos provides a comprehensive platform for data analysis and reporting. Users can create a wide range of reports and dashboards to gain insights from their data and make informed business decisions. It also supports collaboration and sharing of insights with other team members and stakeholders.

Data Access

- Connect to Data Sources: IBM Cognos can connect to various data sources, including databases, spreadsheets, and web services.
- Data Import: Import data from these sources into Cognos for analysis.

Data Exploration

- Create Reports: Build interactive reports using a drag-and-drop interface.
- Ad-Hoc Querying: Allow users to create ad-hoc queries to explore data.
- Data Exploration: Drill down, pivot, and filter data to gain insights.

Data Analysis

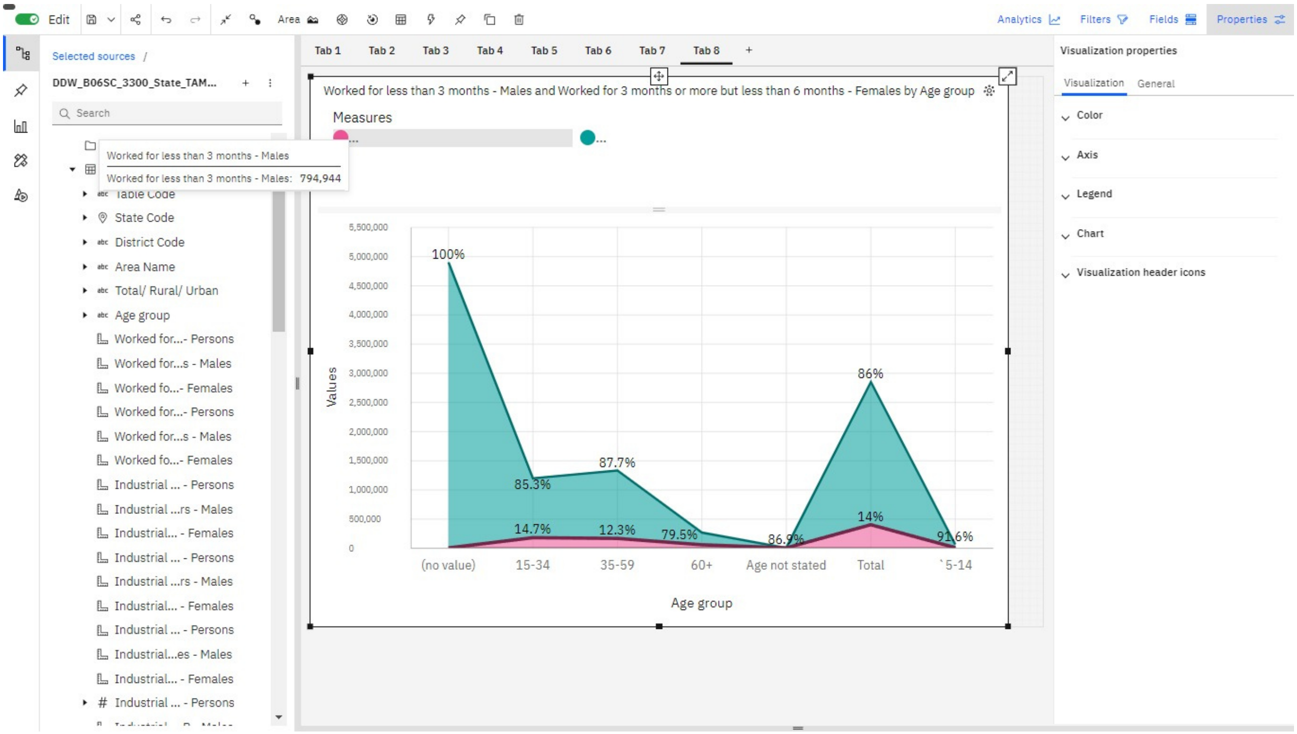
- Advanced Analytics: Utilize advanced analytics features to perform predictive modeling and statistical analysis.
- Data Mining: Discover patterns and trends in data.
- Scenario Analysis: Analyze "what-if" scenarios to understand the impact of changes.

Performance Monitoring and Optimization

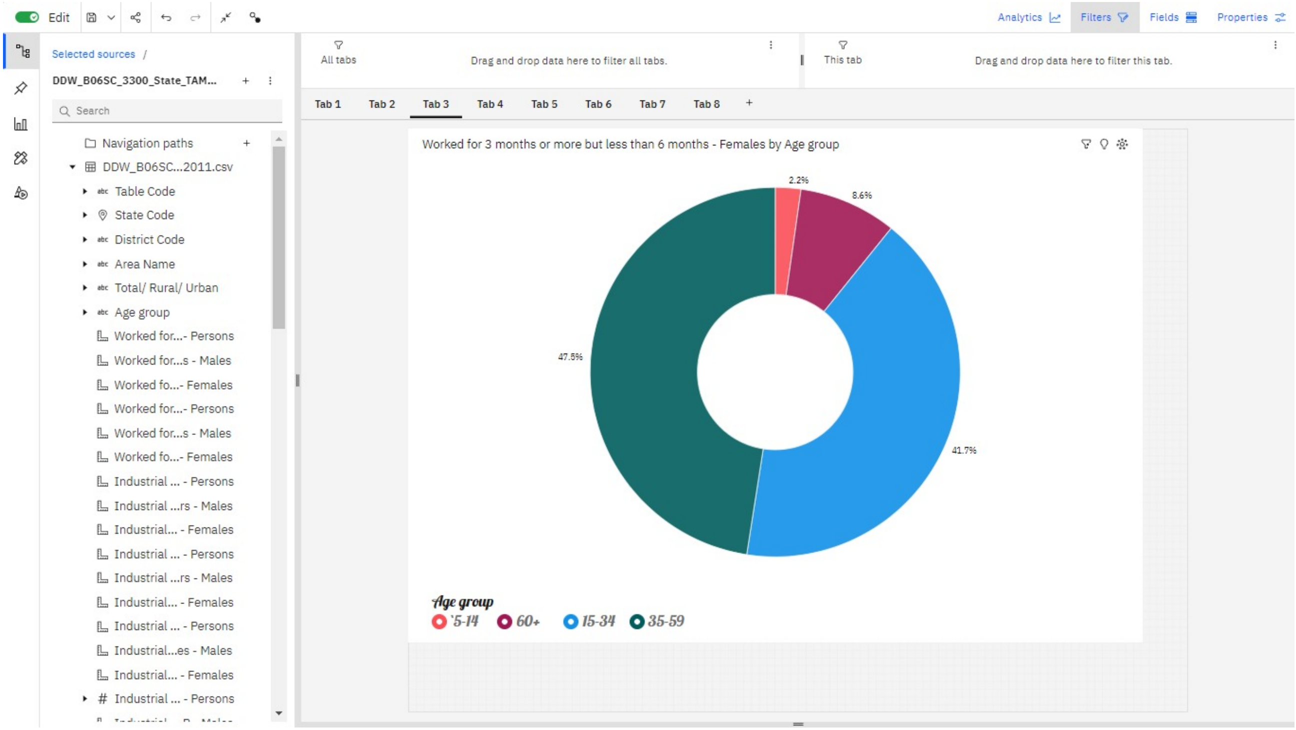
- Monitor Report Performance: Analyze and optimize the performance of reports and queries.
- Resource Management: Allocate resources to ensure smooth operation.

IBM COGNOS DASHBOARD

The dashboard of the given data source file has been visualized using the tool IBM COGNOS for the more client pickup manner



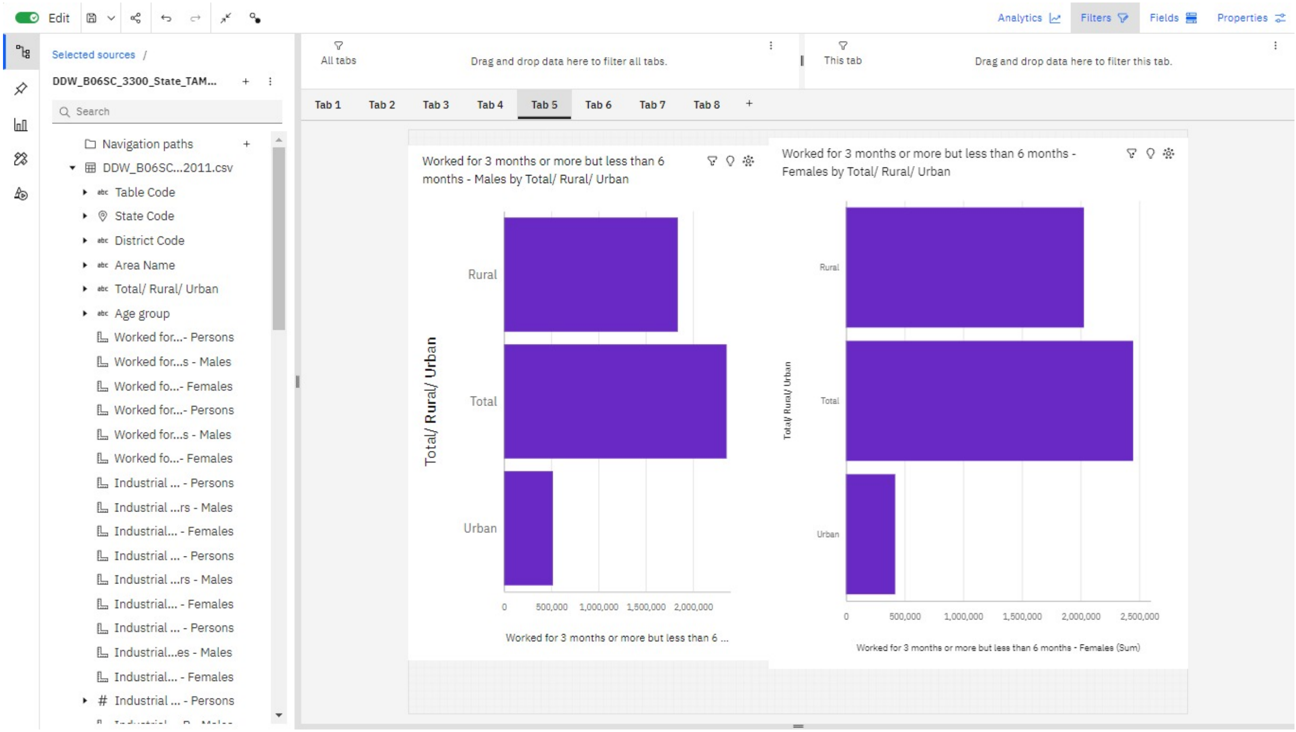
DATA VISUALITION IN CHART WHICH SHOWN DATASOURCE BY USING DATASET



Advanced Analysis

python in data analytics

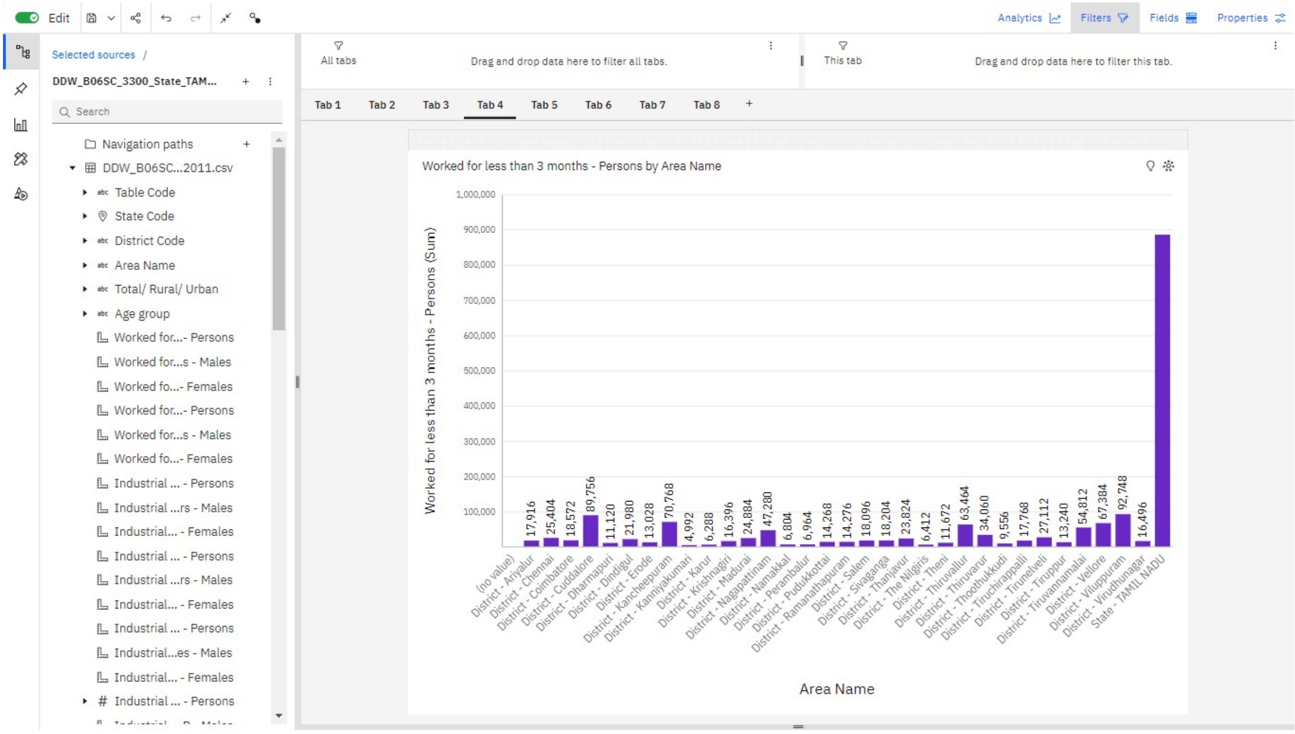
Leverage the power of Python to perform advanced statistical analysis, machine learning, and data manipulation

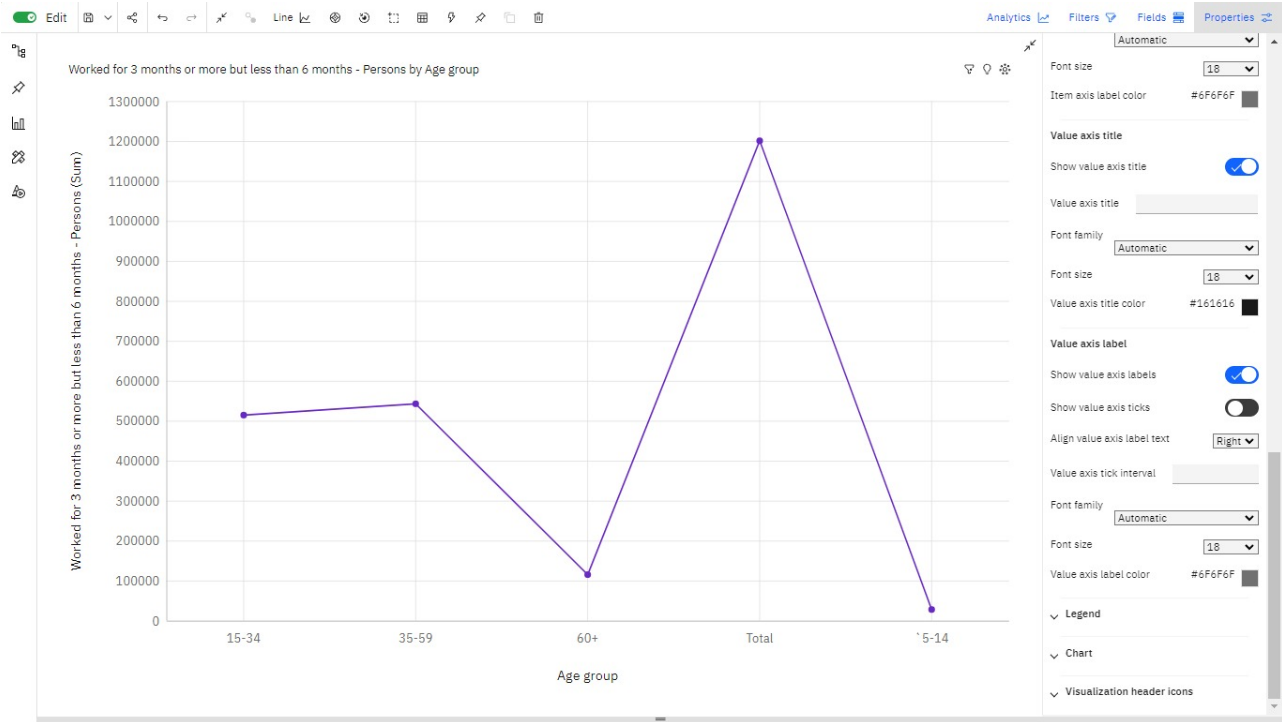


DATA SCIENCE IN PYTHON

1. **Data Collection**:
   - Gather data from various sources, including databases, web scraping, APIs, and data files (e.g., CSV, Excel, JSON).
2. **Data Cleaning and Preprocessing**:
   - Handle missing values, outliers, and inconsistencies in the data.
   - Convert data into a structured format for analysis.
3. **Data Exploration and Analysis**:
   - Use libraries like NumPy, Pandas, and Matplotlib for basic data exploration and visualization.
   - Calculate summary statistics, create plots, and identify patterns in the data.
4. **Data Visualization**:
   - Create visualizations to better understand the data and communicate insights. Libraries like Matplotlib, Seaborn, and Plotly are commonly used for this purpose.

5. **Feature Engineering**:
   - Select relevant features (columns) from the dataset.
   - Create new features, normalize data, and transform variables as needed.
6. **Model Building**:
   - Use machine learning libraries such as Scikit-Learn and TensorFlow to build predictive models.
   - Select the appropriate algorithms and techniques for your problem.
7. **Model Training and Validation**:
   - Split the data into training and testing sets.
   - Train machine learning models on the training data and validate their performance on the test data.
8. **Model Evaluation**:
   - Use various metrics (e.g., accuracy, precision, recall, F1-score, ROC AUC) to evaluate the model's performance.
   - Tune hyperparameters and adjust the model as needed.
9. **Model Deployment**:
   - Deploy the trained model for real-world applications, which can include web applications, APIs, or embedded systems.
10. **Data Communication**:
    - Communicate your findings and insights effectively through reports, dashboards, and data storytelling.





Automatic

Font size 18

Item axis label color #6F6F6F

Value axis title

Show value axis title

Value axis title

Font family Automatic

Font size 18

Value axis title color #161616

Value axis label

Show value axis labels

Show value axis ticks

Align value axis label text Right

Value axis tick interval

Font family Automatic

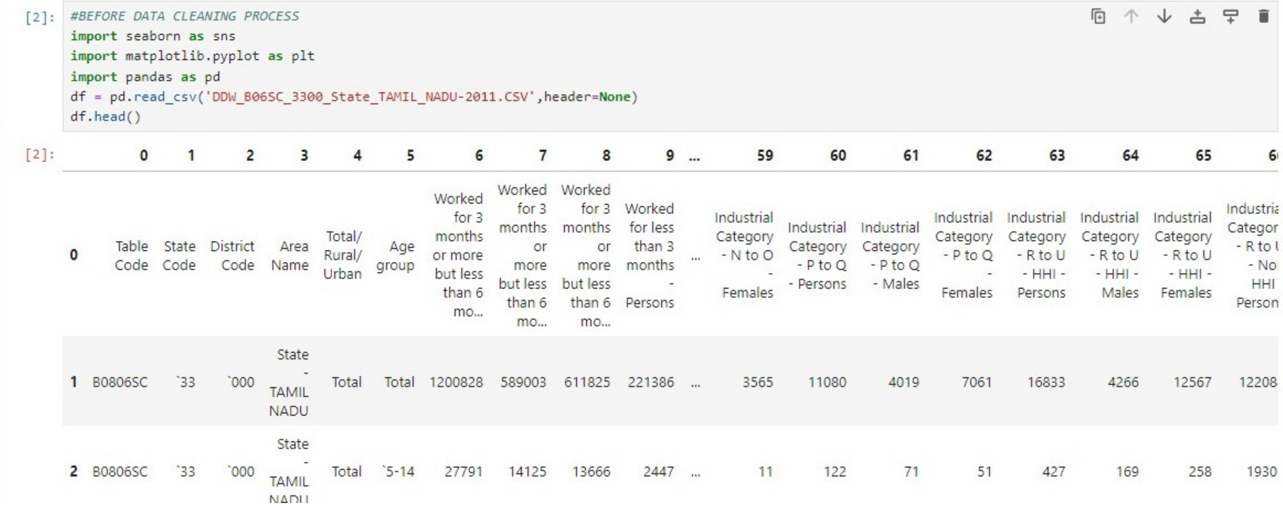
Font size 18

Value axis label color #6F6F6F

DATA PREPROCESSING

Data preprocessing is a fundamental step in the data analysis pipeline. . The primary purpose of data preprocessing is to improve the quality, consistency, and relevance of the data, making it easier to extract meaningful insights or to train accurate models.

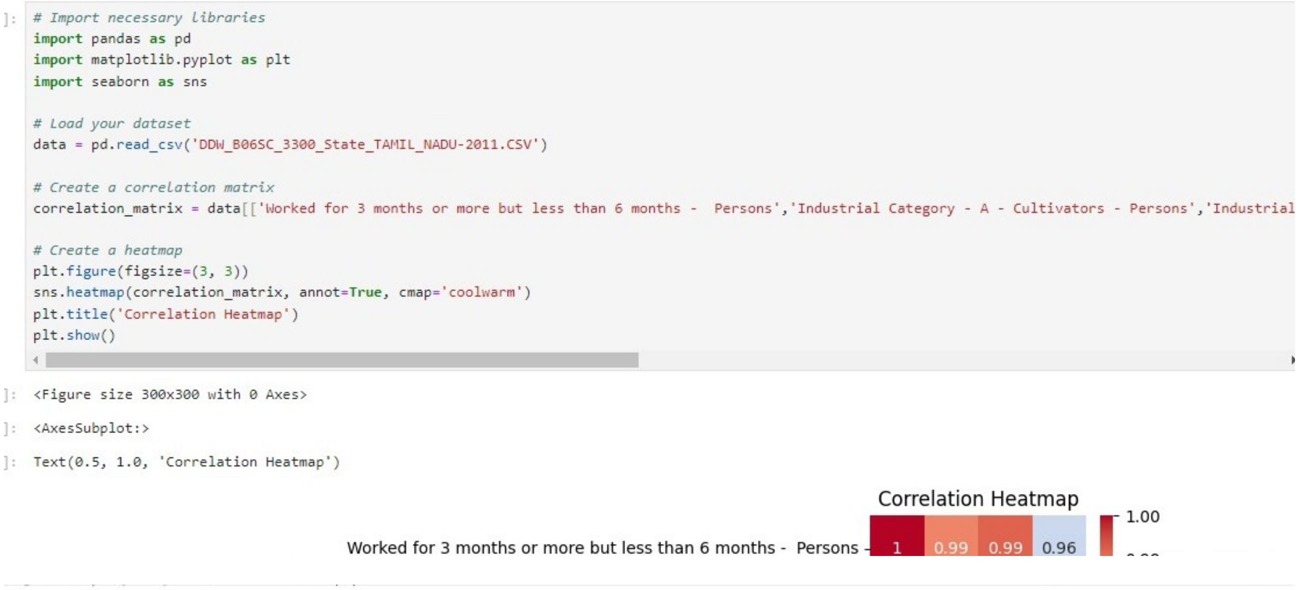
DATA CLEANING



```
[2]: #BEFORE DATA CLEANING PROCESS
     import seaborn as sns
     import matplotlib.pyplot as plt
     import pandas as pd
     df = pd.read_csv('DDW_B06SC_3300_State_TAMIL_NADU-2011.CSV',header=None)
     df.head()
```

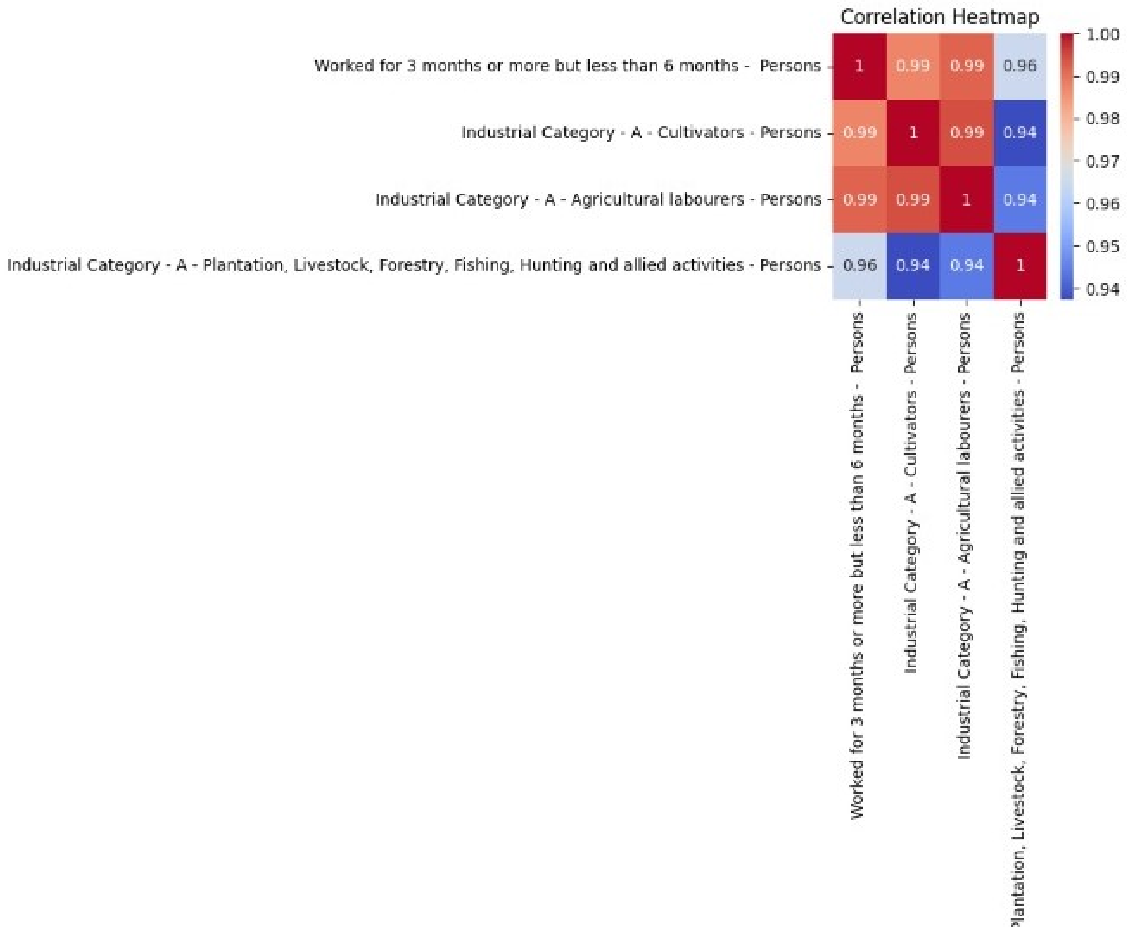
[2]:

	0	1	2	3	4	5	6	7	8	9	...	59	60	61	62	63	64	65	6
0	Table Code	State Code	District Code	Area Name	Total/ Rural/ Urban	Age group	Worked for 3 months or more but less than 6 mo...	Worked for 3 months or more but less than 6 mo...	Worked for 3 months or more but less than 6 mo...	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	Industrial Category - R to U - HHI - Males	Industrial Category - R to U - HHI - Females	Industria Categor - R to I - No HHI Person
1	B0806SC	`33	`000	State - TAMIL NADU	Total	Total	1200828	589003	611825	221386	...	3565	11080	4019	7061	16833	4266	12567	12208
2	B0806SC	`33	`000	State - TAMIL NADU	Total	`5-14	27791	14125	13666	2447	...	11	122	71	51	427	169	258	1930



```
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load your dataset
data = pd.read_csv('DDW_B06SC_3300_State_TAMIL_NADU-2011.CSV')

# Create a correlation matrix
correlation_matrix = data[['Worked for 3 months or more but less than 6 months -  Persons','Industrial Category - A - Cultivators - Persons','Industrial

# Create a heatmap
plt.figure(figsize=(3, 3))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

```
<Figure size 300x300 with 0 Axes>

<AxesSubplot:>

Text(0.5, 1.0, 'Correlation Heatmap')
```



CONCLUSION

In this analysis, we have examined a dataset pertaining to marginal workers in the state of Tamil Nadu. Our goal was to gain a better understanding of the characteristics of these workers, their distribution across districts, and their working patterns. Here are the main conclusions drawn from our analysis: