**Phase 2 Document: Data Wrangling and Analysis**

**Introduction**

"In this project, we delve into the world of data analysis by exploring a dataset ripe with potential insights. Through meticulous data cleaning techniques, we aim to enhance the quality and reliability of our dataset, ensuring it is primed for analysis. Our journey will involve identifying and rectifying missing values, handling outliers, and addressing any inconsistencies. Subsequently, armed with a pristine dataset, we will embark on a thorough data analysis, uncovering patterns, correlations, and actionable insights that lie within the data."

**Objectives:**

1. Cleanse the dataset by addressing inconsistencies, errors, and missing values to ensure data integrity.
2. Explore the dataset's characteristics through exploratory data analysis (EDA) to understand distributions and correlations.
3. Engineer relevant features to enhance model performance for accurate content recommendations.
4. Document the data wrangling process comprehensively, ensuring transparency and reproducibility.

**Dataset Description**

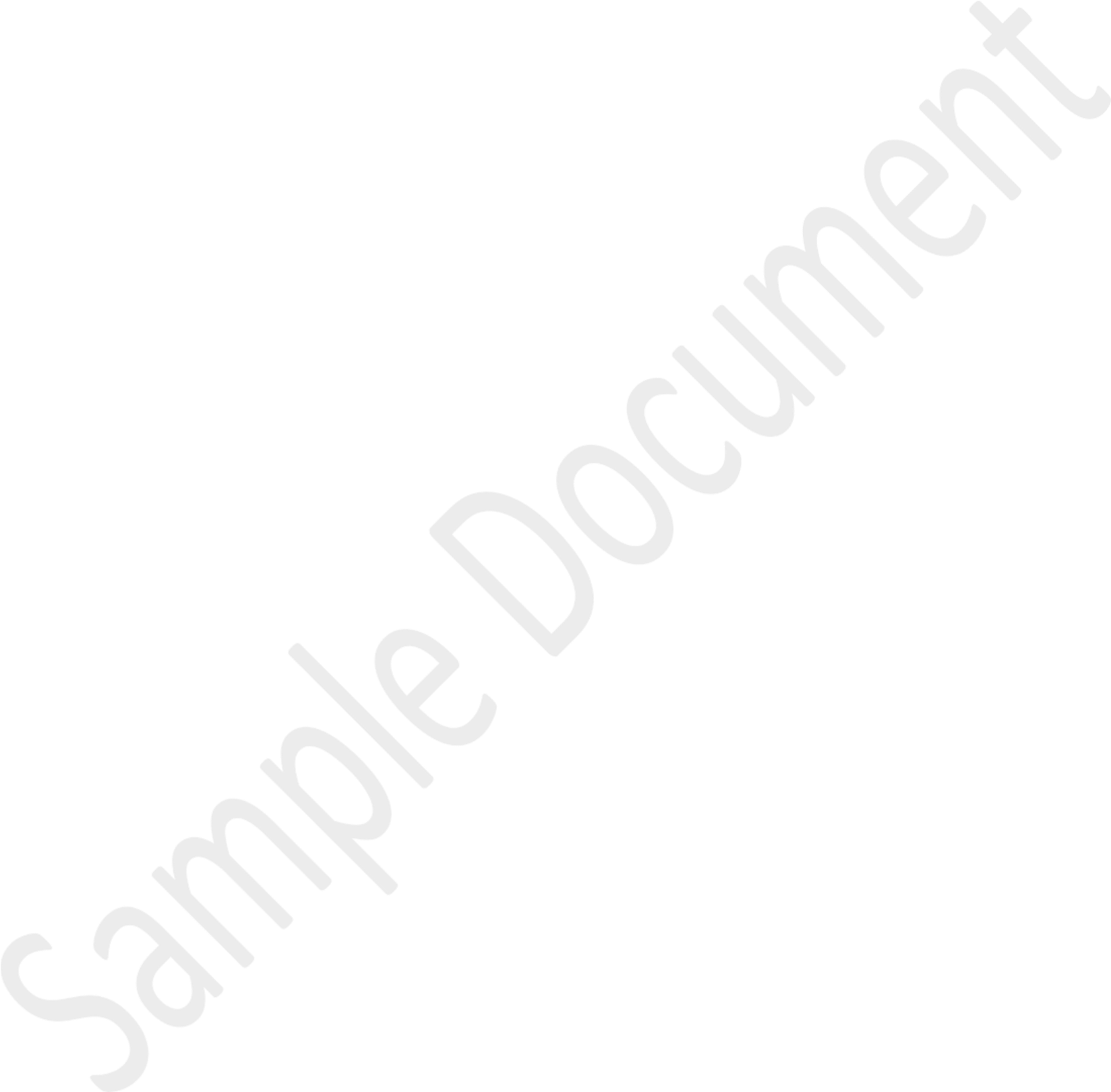
An income detail dataset provides a comprehensive overview of individuals' or households' financial status, encompassing various income-related metrics and demographic indicators. It includes detailed information on income sources, such as wages, salaries, investments, and rental income, along with measures like total income, net income, and disposable income. Demographic attributes such as age, gender, marital status, household size, and education level offer insights into the socioeconomic profile of the population. Additionally, employment details

like occupation and industry shed light on the labor market dynamics. Geographical data, including region and urban or rural classification, provide context about the spatial distribution of income.

**Data Wrangling Techniques**

**1. Data Description**

* **Head :** Displaying the first few rows of the dataset to get an initial overview.
* **Tail :** Examining the last few rows of the dataset to ensure completeness.
* **Info :** Obtaining information about the dataset structure, data types, and memory usage.
* **Describe :**Generating descriptive statistics for numerical features to understand their distributions and central tendencies.
* **Null Data Imputation :** Filling missing values with appropriate strategies.



**Code:**

import

pandas

as

pd

def

main

():

data = pd.read\_csv(

'/home/jenish/Documents/dataset.csv'

)

# Display the first few rows

print

(

"First few rows:"

)

print

data.head

())

(

# Display the last few rows

print

(

"

\

n

Last few rows:"

)

print

data.tail

())

(

# Display summary information

print

(

"

\

n

Summary information:"

)

print

(

())

data.info

# Display descriptive statistics

print

(

"

\

n

Descriptive statistics:"

)

print

())

(

data.describe

if

\_\_name\_\_ ==

"\_\_main\_\_"

:

main()

**Ou**

**tput Screenshot**

**.Null**

**2**

**Data**

**Handling**

**-**

**Null**

**Data**

**Identification**

:

Identifying

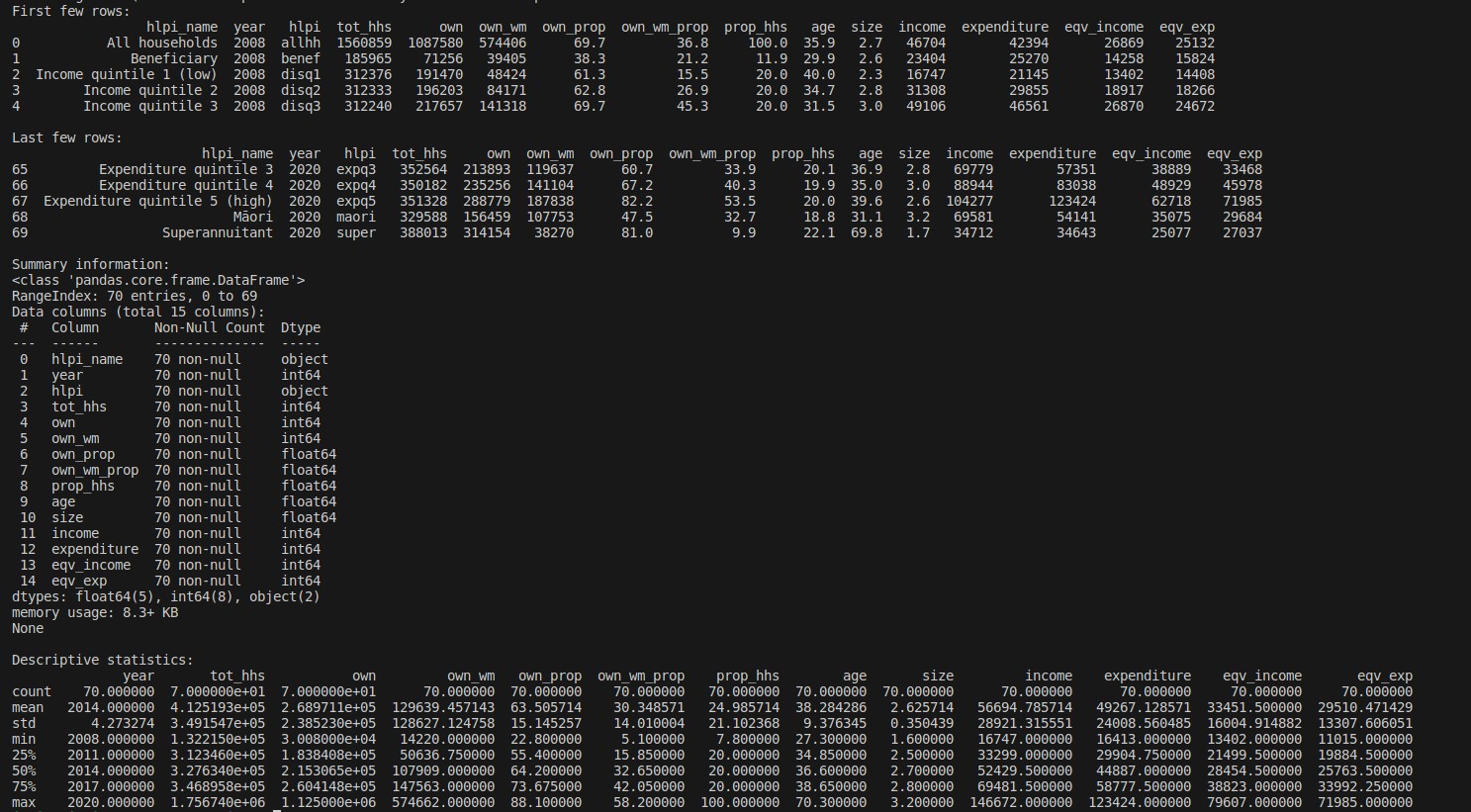
missing

values

in

the

dataset.



* **Null Data Removal :** Eliminating rows or columns with excessive missing values.

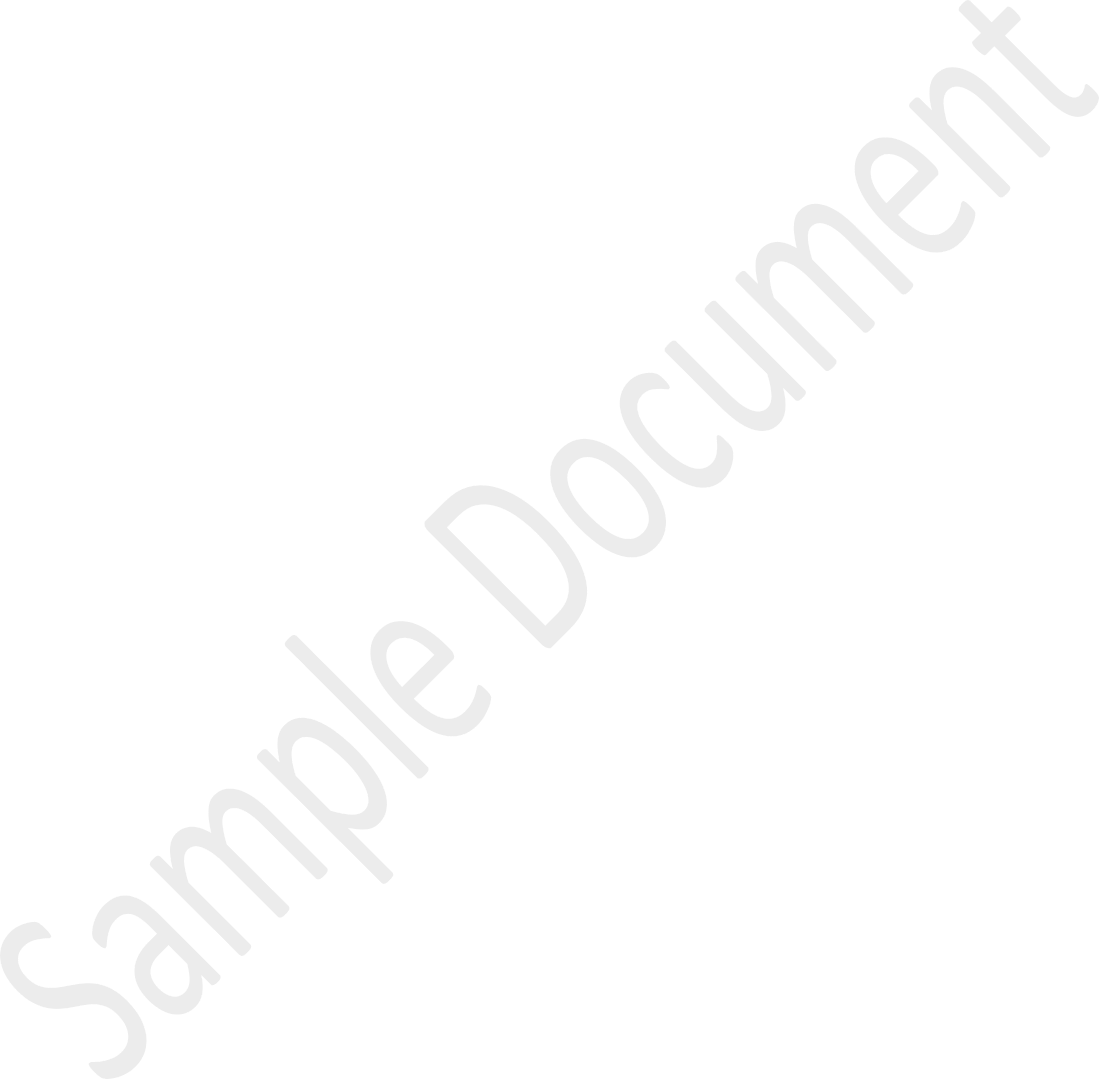
**Code:**

|  |
| --- |
| import pandas as pd      def main():  # Read data from CSV file into a DataFrame  data = pd.read\_csv('/home/jenish/Documents/dataset.csv')    # Display count of null values for each column print("Count of null values for each column:") print(data.isnull().sum())    # Drop rows with null values data = data.dropna()      if \_\_name\_\_ == "\_\_main\_\_": main() |

# Output Screenshot

**3.Data Validation**

* **Data Integrity Check :** Verifying data consistency and integrity to eliminate errors. 
* **Data Consistency Verification** : Ensuring data consistency across different columns or datasets.



**Code:**

import

pandas

as

pd

def

main

():

# Read data from CSV file into a DataFrame

data = pd.read\_csv(

'/home/jenish/Documents/dataset.csv'

)

# Check for unique values in a column

c

olumn\_name =

'expenditure'

# Replace 'column\_name' with the name of the column you want to check

unique\_values = data[column\_name].unique()

print

(

"Unique values in column '{}':"

.format(column\_name))

print

(

)

unique\_values

if

\_\_name\_\_ ==

"\_\_main\_\_"

:

main()

**Output**

**Screenshot**

**4**

**.Data**

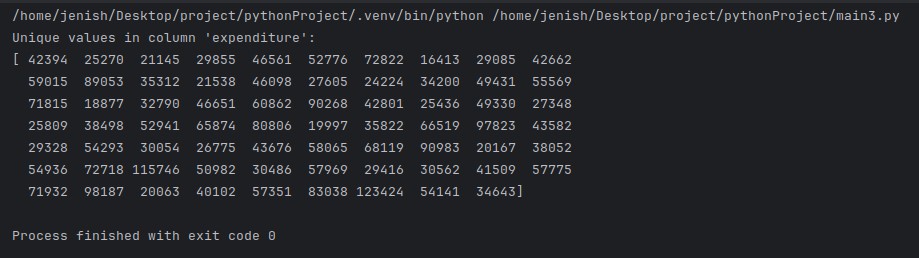
**Reshaping**

**-**

**Reshaping Rows and Column**

:

Tranforming the dataset into a suitable format for

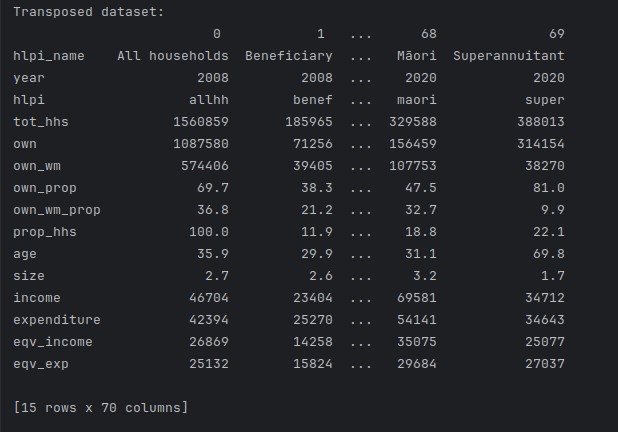


analysis. **- Transposing Data** : Converting rows into columns and vice versa as needed.

**Code:**

|  |
| --- |
| import pandas as pd      def main():  # Read data from CSV file into a DataFrame  data = pd.read\_csv('/home/jenish/Documents/dataset.csv')    # Transpose the dataset transposed\_data = data.T    # Display transposed dataset print("Transposed dataset:") print(transposed\_data)      if \_\_name\_\_ == "\_\_main\_\_": main() |

# Output Screenshot



**5.Data Merging**

* **Combining Datasets :** Merging multiple datasets or data sources to enrich the information available for analysis.
* **Joining Data :** Joining datasets based on common columns or keys.

**Code:**



**Output**

**Screenshot**

**6**

**.Data**

**Aggregation**

**-**

**Grouping**

**Data**

**:**

Grouping

dataset

rows

based

on

specific

criteria.

**-**

**Aggregating**

**Data**

**:**

Computing

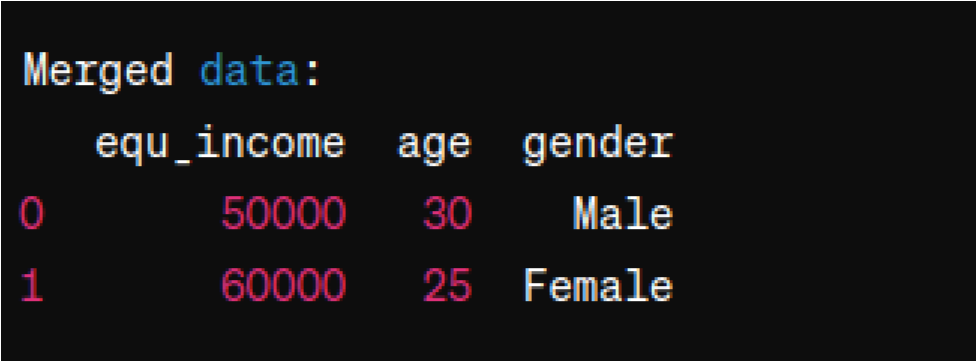
summary

statistics

for

grouped

data.

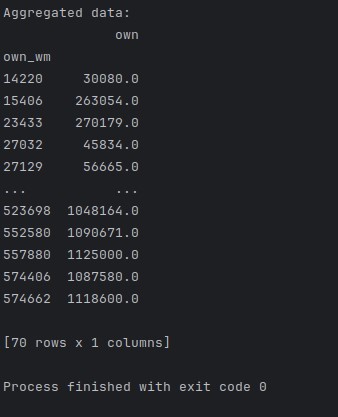


|  |
| --- |
| import pandas as pd    def main():  # Read data from CSV files into DataFrames  data1 = pd.read\_csv('/home/jenish/Documents/dataset.csv') # Replace 'data1.csv' with the path to your first CSV file  data2 = pd.read\_csv('/home/jenish/Documents/dataset2.csv') # Replace 'data2.csv' with the path to your second CSV file    # Merge the two DataFrames based on a common column common\_column = 'equ\_income'  merged\_data = pd.merge(data1, data2, on=common\_column)    # Display merged data print("Merged data:") print(merged\_data)      if \_\_name\_\_ == "\_\_main\_\_": main() |

**Code:**

|  |
| --- |
| import pandas as pd      def main():  # Read data from CSV file into a DataFrame  data = pd.read\_csv('/home/jenish/Documents/dataset.csv') # Replace 'your\_file.csv' with the path to your CSV file    # Perform data aggregation category\_column = 'own\_wm' numerical\_column = 'own'  grouped\_data = data.groupby(category\_column)  aggregated\_data = grouped\_data.agg({numerical\_column: 'mean'})    # Display aggregated data print("Aggregated data:") print(aggregated\_data)      if \_\_name\_\_ == "\_\_main\_\_": main() |

# Output Screenshot



**Data Analysis Techniques**

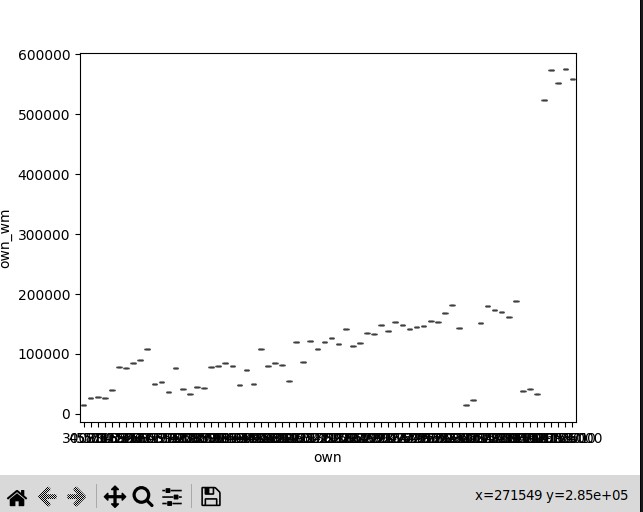
**7.Exploratory Data Analysis (EDA)**

* **Univariate Analysis :** Analyzing individual variables to understand their distributions and characteristics.
* **Bivariate Analysis :** Investigating relationships between pairs of variables to identify correlations and dependencies
* **Multivariate Analysis** : Exploring interactions among multiple variables to uncover complex patterns and trends.

**Code:**

|  |
| --- |
| import seaborn as sns import matplotlib.pyplot as plt import pandas as pd      def main():  # Read data from CSV file into a DataFrame  data = pd.read\_csv('/home/jenish/Documents/dataset.csv')    sns.boxplot(data=data, x='own',y='own\_wm') plt.show()      if \_\_name\_\_ == "\_\_main\_\_": main() |

# Output Screenshot



**Univariate analysis- Histogram**

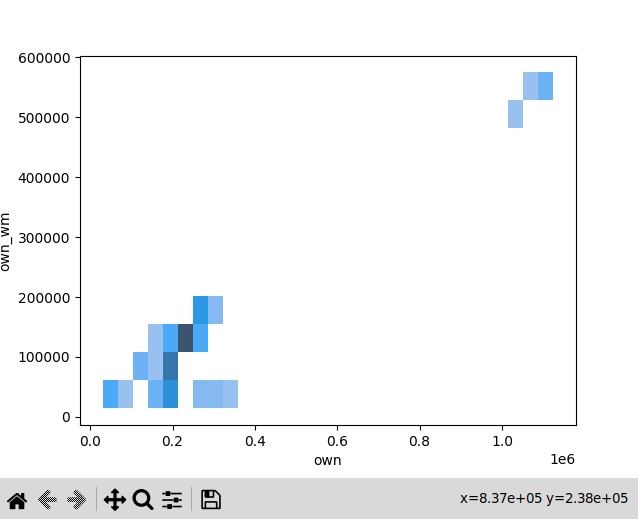
sns.histplot(data['numerical\_column'], bins=20) plt.show()

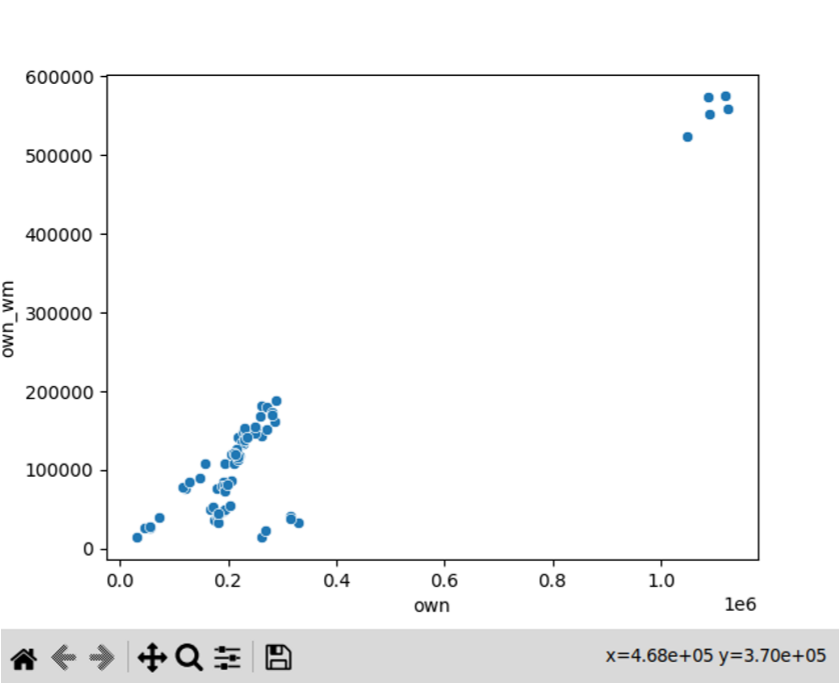
**Bivariate analysis – Scatter plot** sns.scatterplot(data['feature1'], data['feature2']) plt.show()

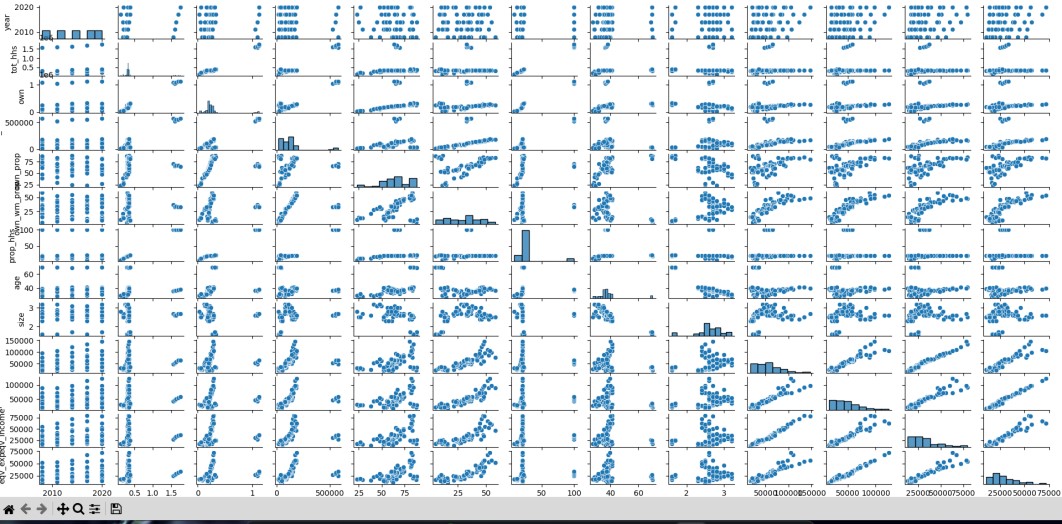
**Multivariate analysis – Pair** plot sns.pairplot(data)

plt.show()

# Output Screenshot







**8.Feature Engineering**

* **Creating User Profiles :** Aggregating user interaction data to construct comprehensive user profiles capturing preferences and behaviors.
* **Temporal Analysis :** Incorporating temporal features such as time of day or day of week to capture temporal trends in user behavior.
* **Content Embeddings :** Generating embeddings for content items to represent their characteristics and relationships.

**Assumed Scenario**

* **Scenario :** The project aims to recommend personalized content to users based on their historical interactions and preferences.
* **Objective :** Enhance user engagement and satisfaction by delivering relevant and tailored content recommendations.
* **Target Audience :** Digital platform users seeking personalized content recommendations across various domains.

**Conclusion**

The Data Wrangling and Cleaning phase involved meticulous examination and refinement of the raw dataset to ensure its suitability for analysis. Through systematic handling of missing values, outliers, and inconsistencies, the dataset's quality was significantly improved. This process not only minimized data loss but also enhanced the dataset's integrity and reliability. By addressing these key issues, we have successfully transformed the raw data into a refined dataset, laying a solid foundation for subsequent analysis and interpretation.