**MEASURING ENERGY CONSUMPTION**

**Sanjay Kumar V** (2021503046**)**

**Ragul N** (2021503542)

**Achyut Prasad DC** (2021503002)

**Santhosh Kumar S** (2021503302)

**Phase 4: Development Part 2**

# PROBLEM STATEMENT:

The problem at hand is to create an automated system that measures energy consumption, analyzes the data, and provides visualizations for informed decision-making. This solution aims to enhance efficiency, accuracy, and ease of understanding in managing energy consumption across various sectors.

# ABSTRACT:

The second phase of the development project focuses on data analysis and visualization of an energy consumption dataset. This phase entails the creation of two essential modules, data\_analysis.py and visualize\_data.py, along with the integration of these modules into the main script main.py. The data\_analysis module is designed to analyze the dataset by extracting various time-related features and visualizing the energy consumption patterns using box plots.

On the other hand, the visualize\_data module facilitates the visualization of the dataset by plotting the training and test data points on the same graph. The main script orchestrates the execution of these modules, enabling the user to select specific datasets for analysis and visualization. The implementation adheres to best practices in modular programming, emphasizing code reusability and readability.

By encapsulating distinct functionalities into separate modules, the development process becomes more organized and manageable, allowing for streamlined data analysisand visualization tasks. The use of popular libraries such as Pandas, Matplotlib, and Seaborn ensures the efficiency and effectiveness of the data processing and visualizationprocedures. Additionally, the error handling mechanisms incorporated in the data loading process enhance the robustness and reliability of the overall application.

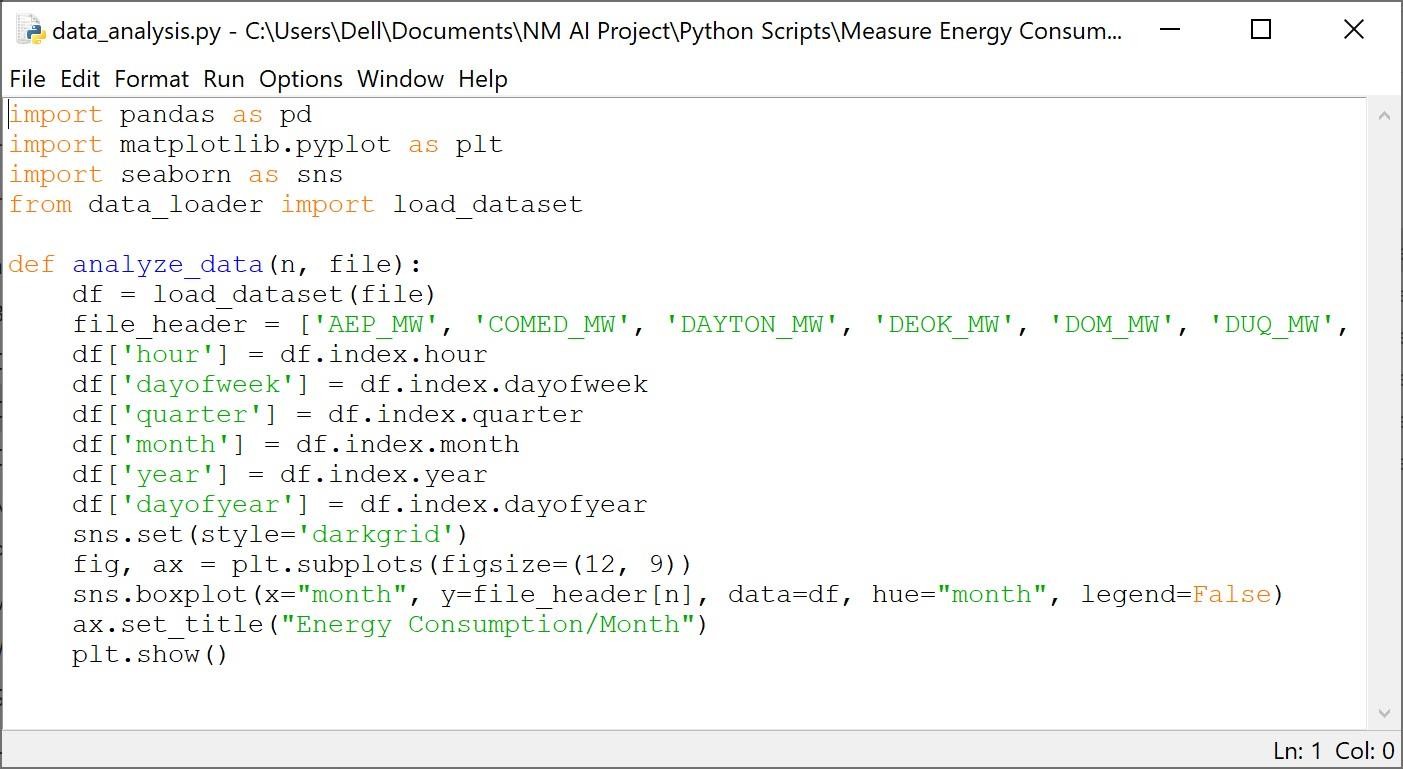
# IMPLEMENTATION STEPS:

Data Analysis and Visualization Module (data\_analysis.py)

1. Import Libraries:

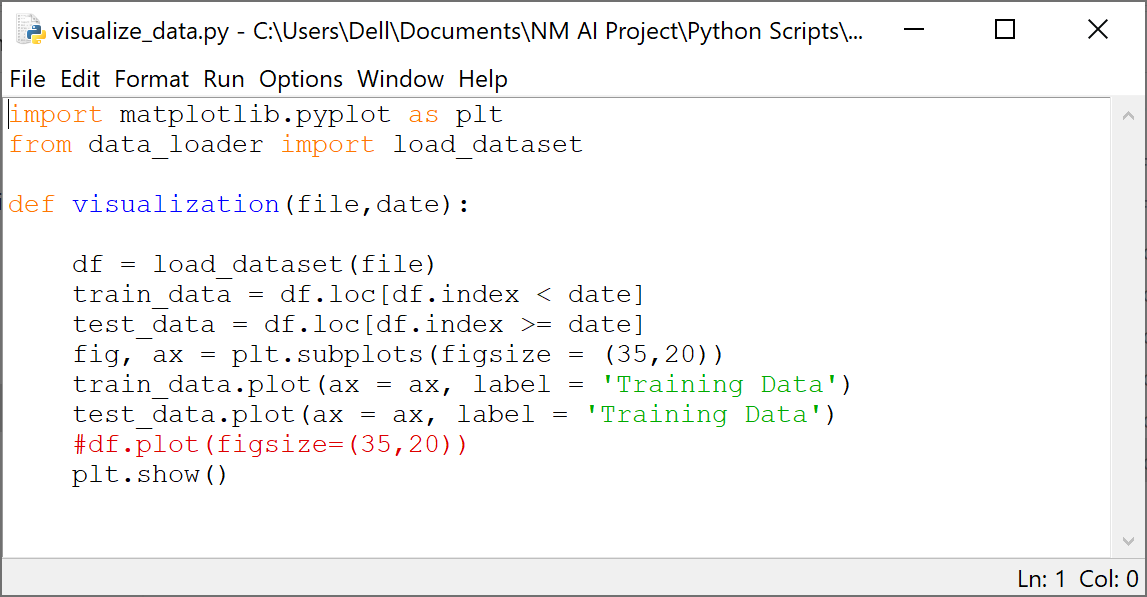
Import the necessary libraries, including pandas, matplotlib.pyplot, andseaborn, for data analysis and visualization tasks.

1. Define analyze\_data Function:
   1. Create the analyze\_data function that takes a DataFrame df and an indexn as parameters.
   2. Define the file\_header list containing the names of different energy consumption variables.
   3. Utilize the DataFrame's index to extract specific time-related features such as hour, day of the week, quarter, month, year, and day of the year.
   4. Perform data analysis and visualization using Seaborn's boxplot to understand the energy consumption pattern per month.
   5. Customize the plot with appropriate labels and title.
   6. Display the visualization using Matplotlib's plt.show().



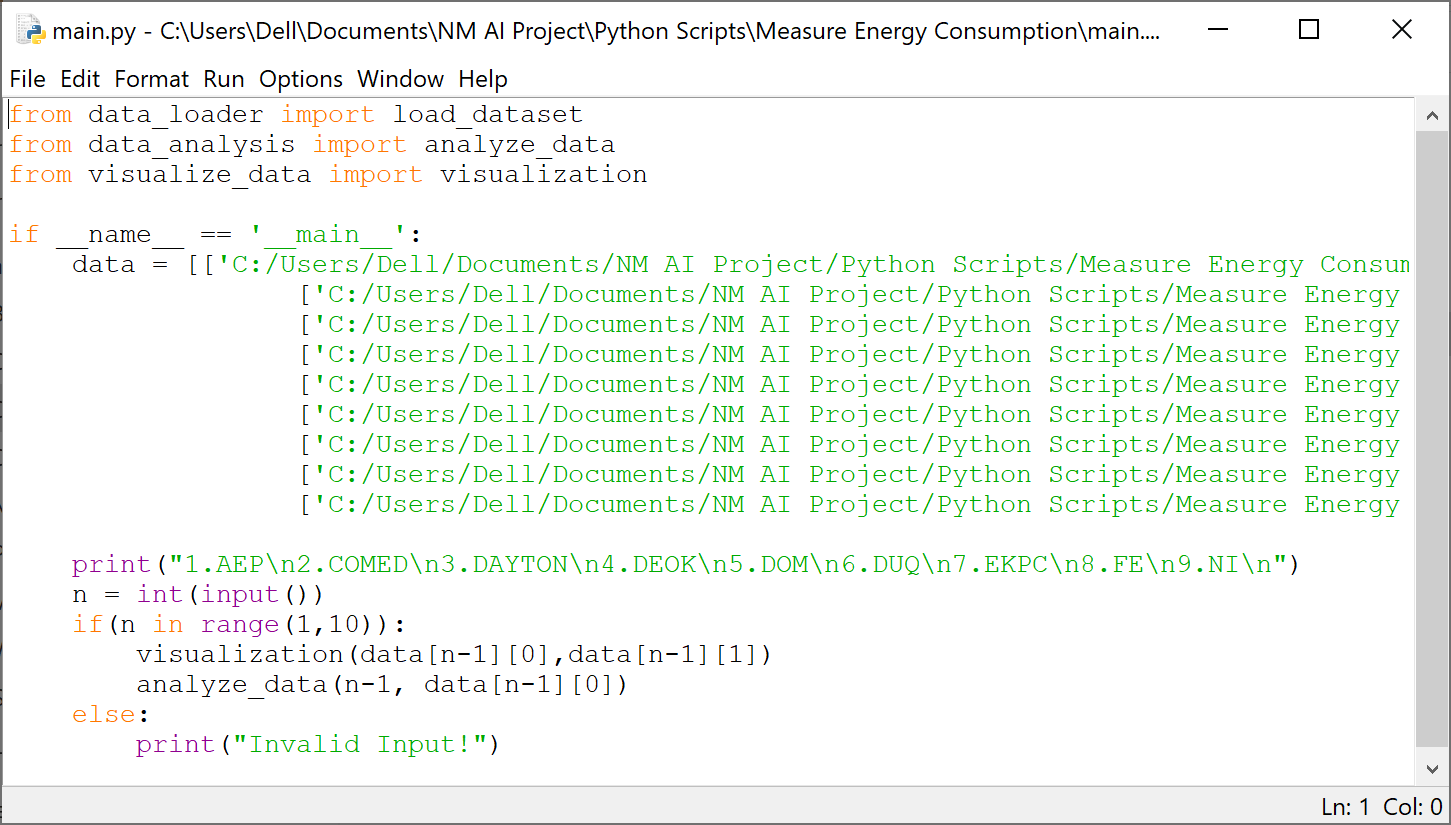
Visualization Module (visualize\_data.py)

1. Import Libraries:
   1. Import matplotlib.pyplot for plotting visualizations.
   2. Import the load\_dataset function from the data\_loader module for loading the dataset.
2. Define visualization Function:
   1. Create the visualization function that takes the parameters file and date.
   2. Use the load\_dataset function to load the dataset specified by the file parameter.
   3. Separate the data into training and test sets based on the provided date.
   4. Create a plot using Matplotlib's subplots with a specified size.
   5. Plot the training and test data on the same plot, customizing the labels accordingly.
   6. Display the plot using plt.show().



Main Script (main.py)

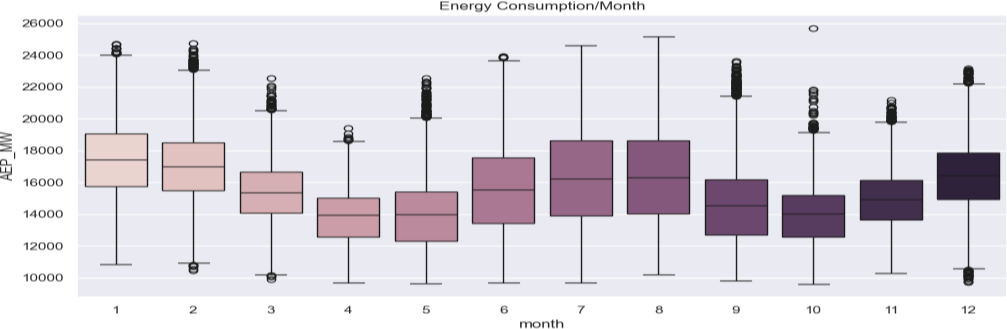
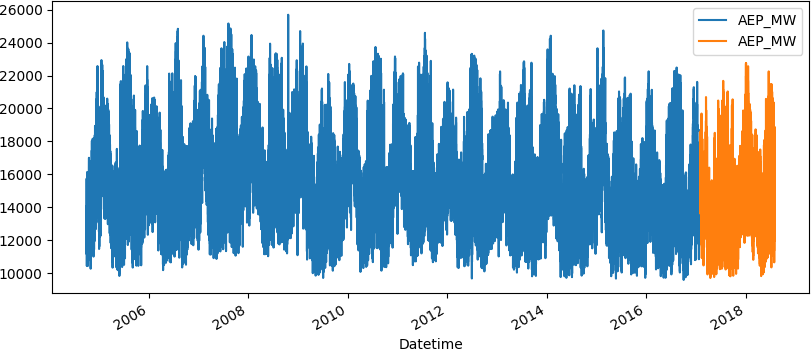
1. Import Necessary Modules:
   1. Import the load\_dataset function from the data\_loader module.
   2. Import the analyze\_data function from the data\_analysis module.
   3. Import the visualization function from the visualize\_data module.
2. Define Main Script:
   1. In the main block, create a list data containing the paths to various datasets along with specific dates.
   2. Display a menu for selecting the dataset to analyze and visualize.
   3. Based on the user's input, call the visualization and analyze\_data functions with appropriate parameters.
   4. Handle invalid user inputs by displaying an error message.



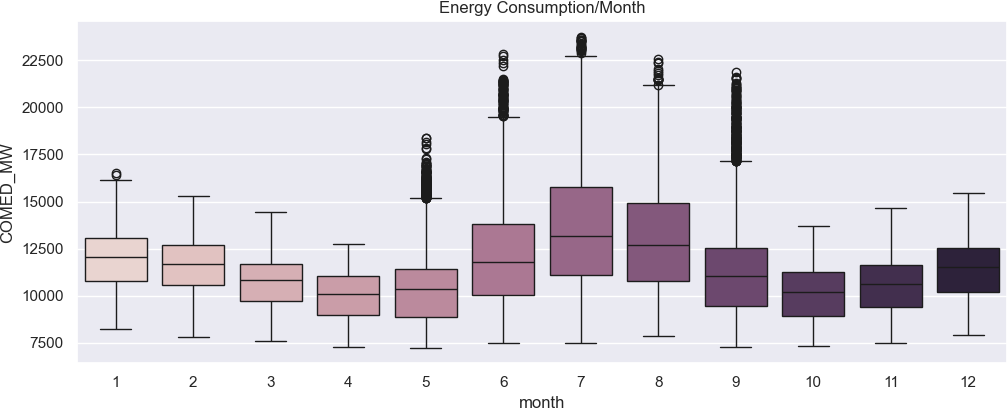
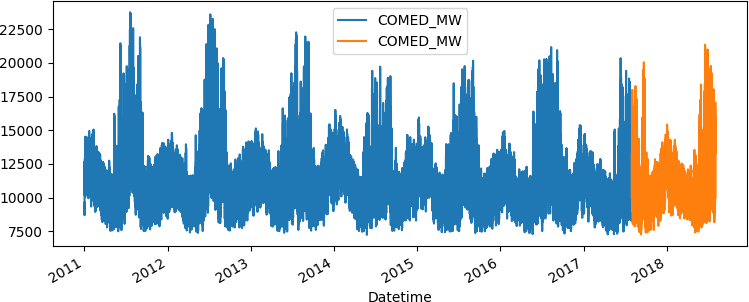
By organizing the code into these separate modules and following the outlined implementation steps, you can effectively perform data analysis and visualization on the energy consumption dataset, enabling a clear understanding of consumption patterns over time.

# RESULTS FROM VISUALIZATION:

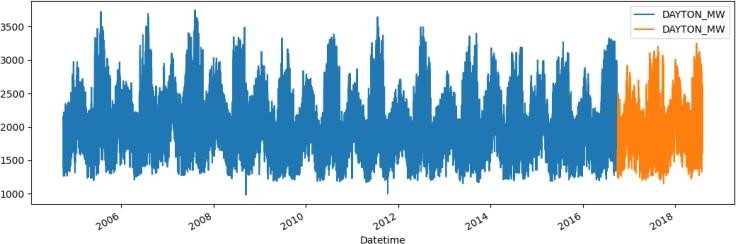
AEP:

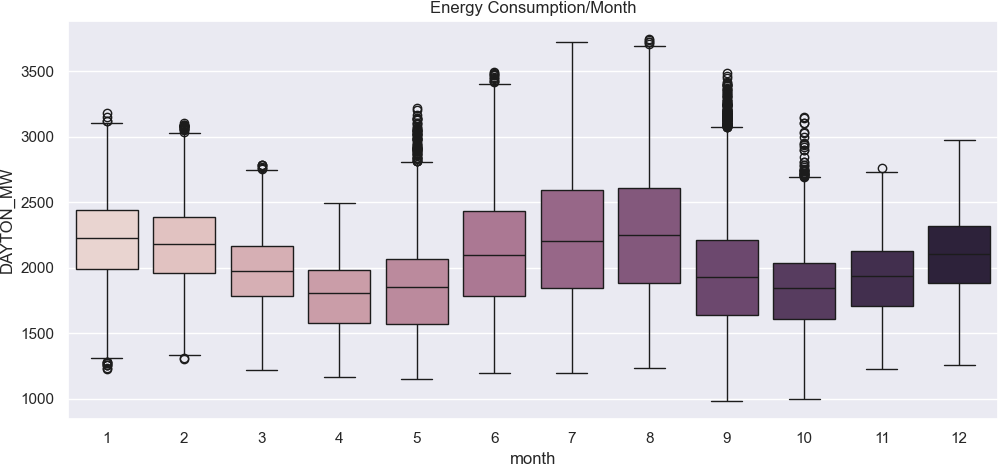


COMED:

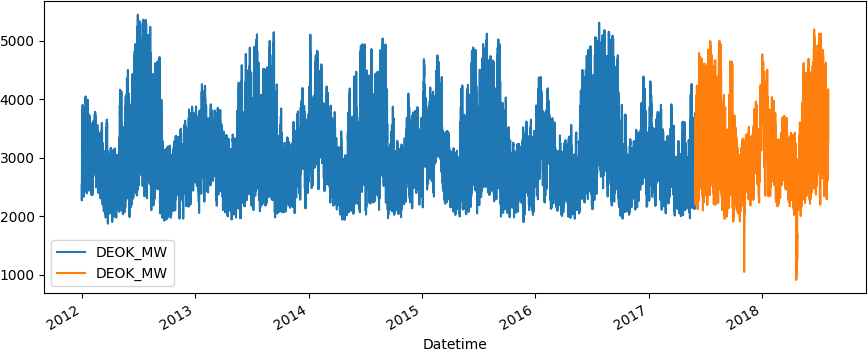


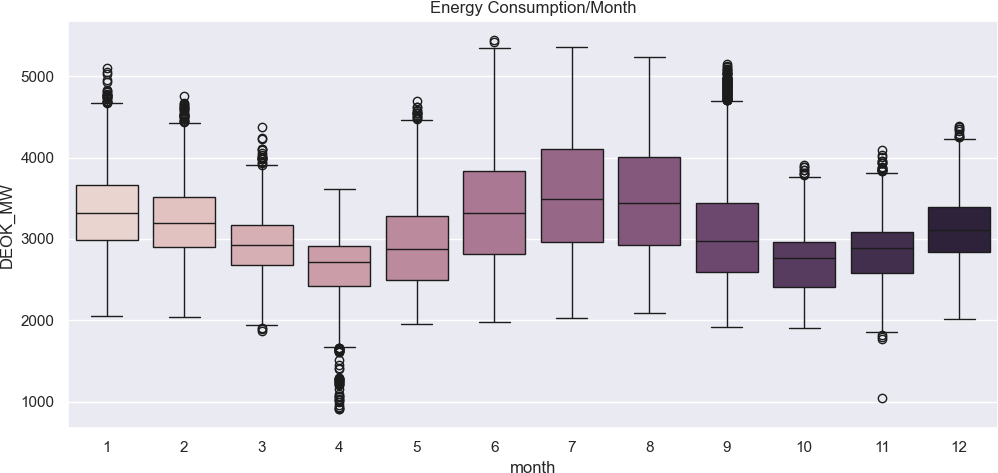
DAYTON:



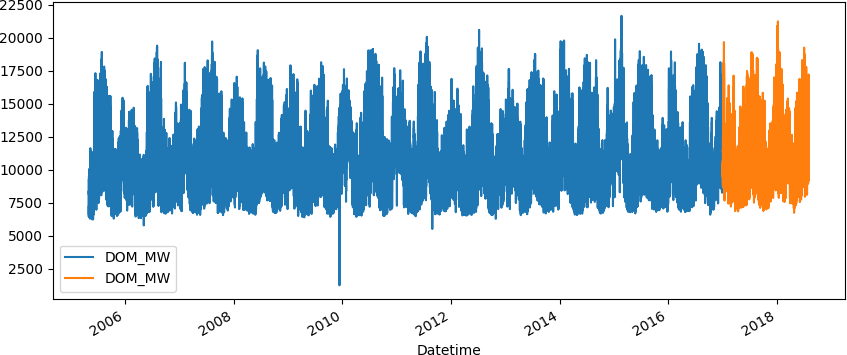


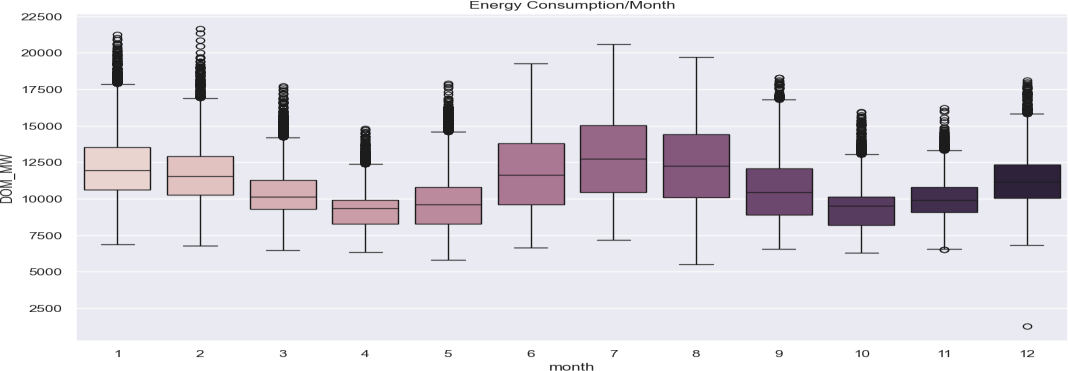
DEOK:



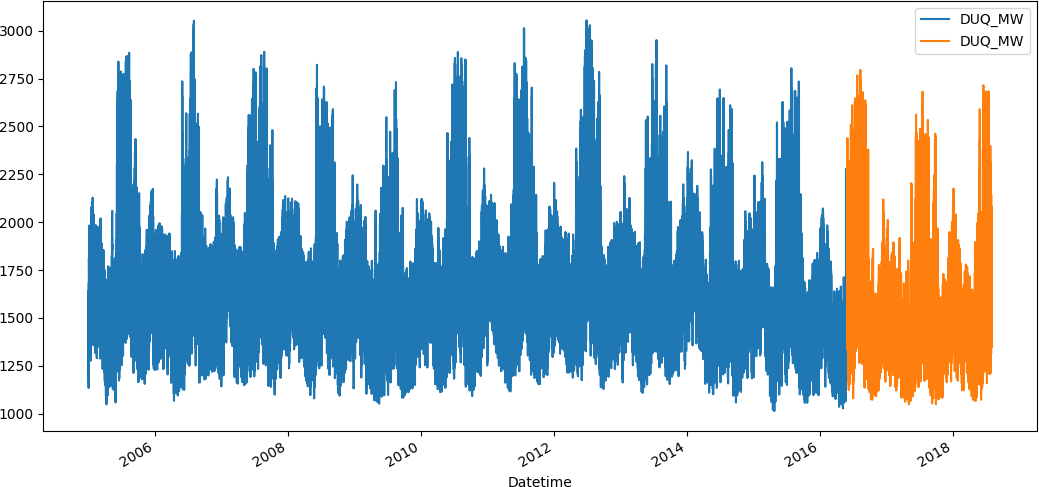


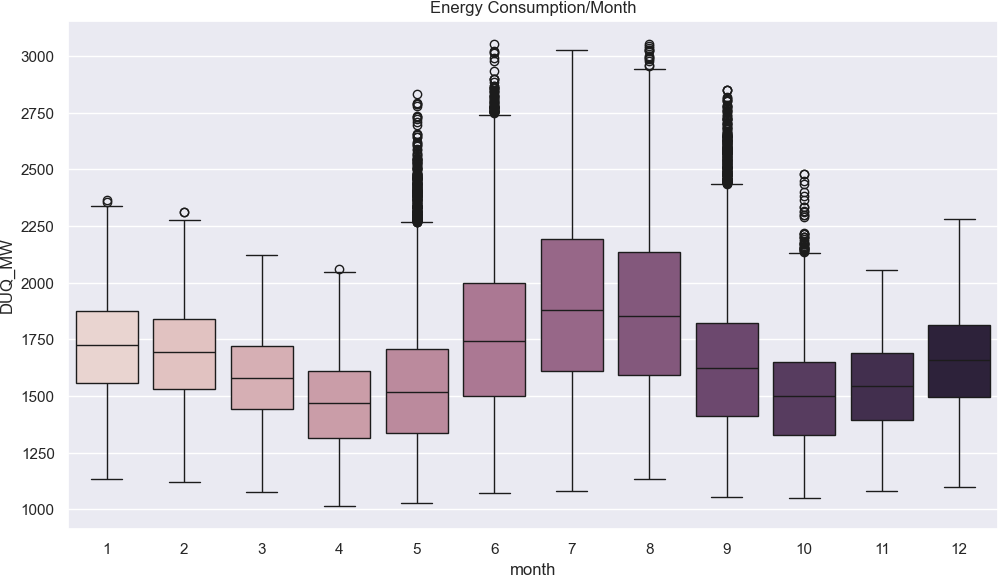
DOM:





DUQ:





# CONCLUSION:

In conclusion, the second phase of the development project successfully establishes a comprehensive framework for data analysis and visualization of energy consumption patterns. The implementation of the data\_analysis.py and visualize\_data.py modules, along with the integration of these modules into the main.py script, showcases a systematic approach to handling complex datasets. The modules' functionalities enable users to gain valuable insights into the energy consumption trends, thereby facilitating informed decision- making processes. Overall, the successful execution of the data analysis and visualization tasks in this phase marks a significant milestone in the project's progression, laying a solid foundation for the subsequent stages of advanced analytics and predictive modeling.