**Title:** Household Power Consumption Analysis

➢ Measurements of electric power consumption in one household with a one-minute sampling rate over a period of almost 4 years.

**Objective:**

The objective of this lab assignment is to explore and analyze a dataset containing measurements of electric power consumption in a household over a period of almost 4 years. You will perform various data visualization tasks to gain insights into electrical quantities, sub#metering values, and overall trends.

**Code:**

**1. Load the dataset**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

# 1. Load the data

url = "/kaggle/input/ml-household-power-consumption/household\_power\_consumption.csv"

data = pd.read\_csv(url, sep=';', na\_values='?')

# Combine Date and Time

data['Datetime'] = pd.to\_datetime(data['Date'] + ' ' +

data['Time'], dayfirst=True)

data.drop(columns=['Date', 'Time'], inplace=True)

data.set\_index('Datetime', inplace=True)

# Convert numeric columns

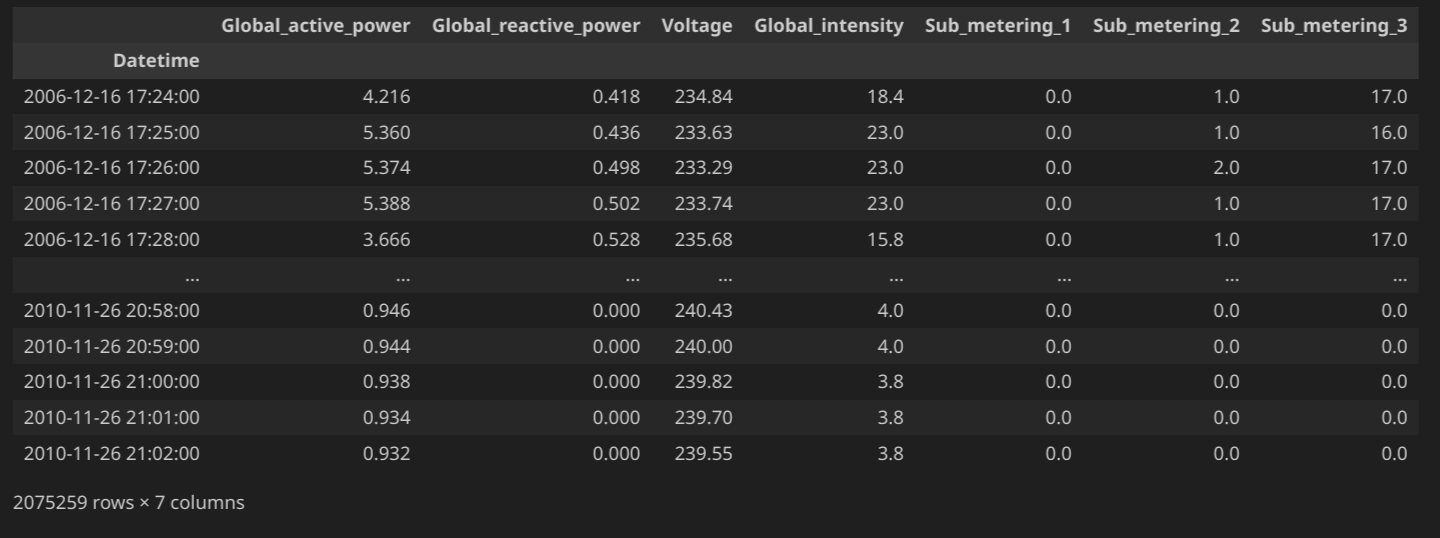
cols\_to\_numeric = ['Global\_active\_power',

'Global\_reactive\_power', 'Voltage',

'Global\_intensity', 'Sub\_metering\_1',

'Sub\_metering\_2', 'Sub\_metering\_3']

data[cols\_to\_numeric].apply(pd.to\_numeric, errors='coerce')



**2. Detect outlier, missing value from the data.**

# 2. Detect Missing Values and Outliers

print("Missing Values:\n", data.isnull().sum())

# Drop rows with missing values

data.dropna(inplace=True)

# Detect outliers in Global\_active\_power

Q1 = data['Global\_active\_power'].quantile(0.25)

Q3 = data['Global\_active\_power'].quantile(0.75)

IQR = Q3 - Q1

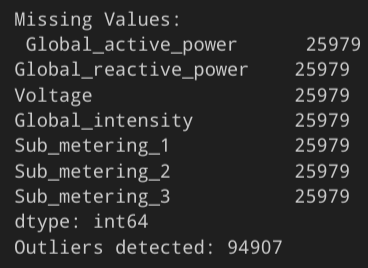
outliers = data[(data['Global\_active\_power'] < (Q1 - 1.5 \*

IQR)) |

(data['Global\_active\_power'] > (Q3 + 1.5 \*

IQR))]

print("Outliers detected:", len(outliers))



**3. Subset the data from the given dates (December 2006 and November 2009)**

# 3. Subset Data: December 2006 and November 2009

dec\_2006 = data.loc['2006-12']

nov\_2009 = data.loc['2009-11']

subset\_data = pd.concat([dec\_2006, nov\_2009])

**4. Create a histogram**

# 4. Histogram

plt.figure(figsize=(8,5))

subset\_data['Global\_active\_power'].hist(bins=50,

color='skyblue')

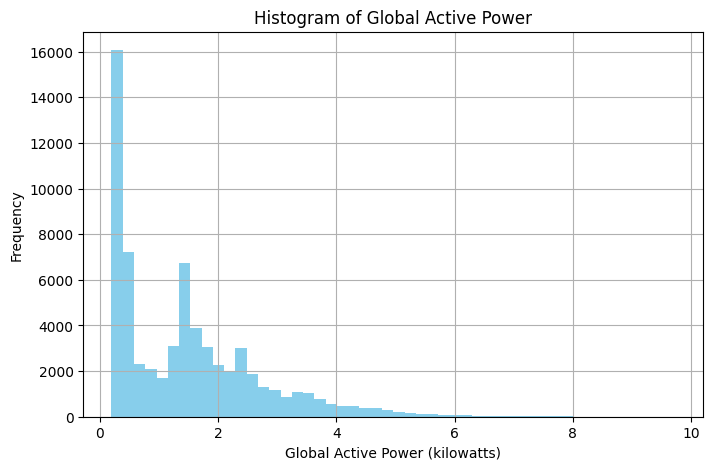
plt.title('Histogram of Global Active Power')

plt.xlabel('Global Active Power (kilowatts)')

plt.ylabel('Frequency')

plt.grid(True)

plt.show()



**5. Create a Time series**

# 5. Time Series Plot

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

subset\_data['Global\_active\_power'].plot()

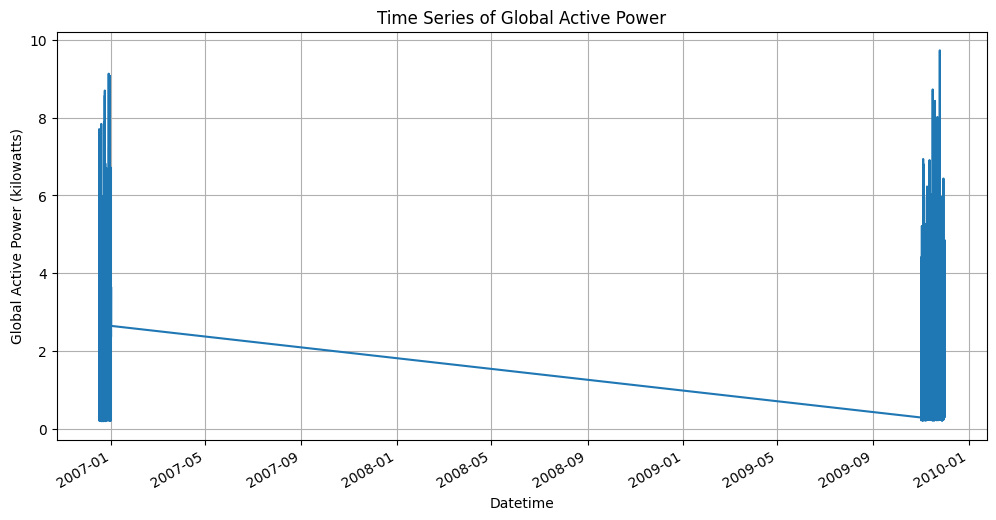
plt.title('Time Series of Global Active Power')

plt.ylabel('Global Active Power (kilowatts)')

plt.xlabel('Datetime')

plt.grid(True)

plt.show()



**6. Create a plot for sub metering**

# 6. Sub-Metering Plot

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

subset\_data['Sub\_metering\_1'].plot(label='Sub\_metering\_1')

subset\_data['Sub\_metering\_2'].plot(label='Sub\_metering\_2')

subset\_data['Sub\_metering\_3'].plot(label='Sub\_metering\_3')

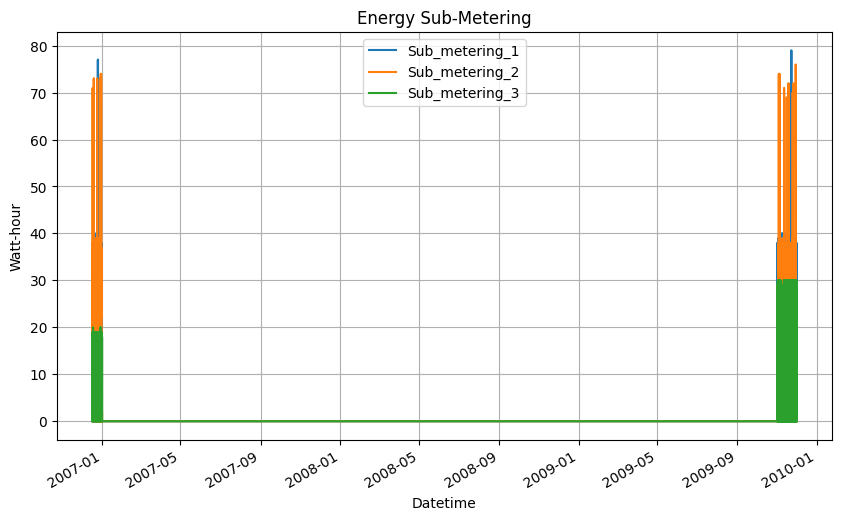
plt.legend()

plt.title('Energy Sub-Metering')

plt.ylabel('Watt-hour')

plt.grid(True)

plt.show()



**7. Create multiple plots, such as, Scatterplot, Histogram, Bar Chart, Pie Chart, Count plot, Boxplot, Heatmap, Distplot, Jointplot**

# Scatterplot

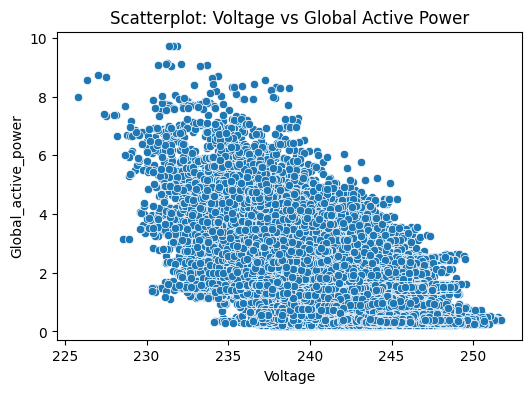
plt.figure(figsize=(6,4))

sns.scatterplot(x='Voltage', y='Global\_active\_power',

data=subset\_data)

plt.title("Scatterplot: Voltage vs Global Active Power")

plt.show()



# Bar Chart (Average Power per Month)

monthly\_avg = subset\_data.resample('ME').mean()

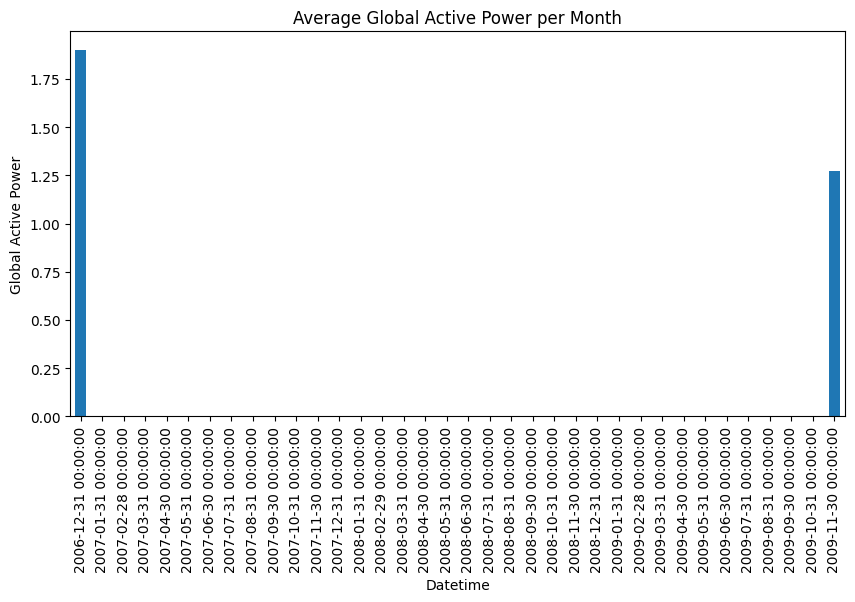
monthly\_avg['Global\_active\_power'].plot(kind='bar',

figsize=(10,5))

plt.title("Average Global Active Power per Month")

plt.ylabel("Global Active Power")

plt.show()



# Pie Chart (Total Energy by Sub Metering)

sub\_totals = subset\_data[['Sub\_metering\_1',

'Sub\_metering\_2', 'Sub\_metering\_3']].sum()

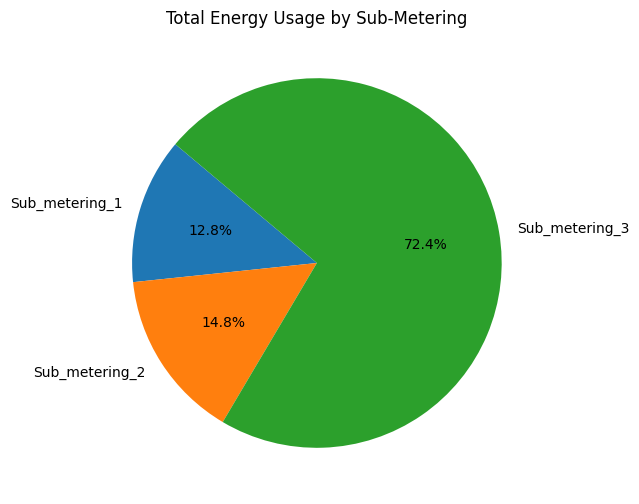
plt.figure(figsize=(6,6))

plt.pie(sub\_totals, labels=sub\_totals.index,

autopct='%1.1f%%', startangle=140)

plt.title("Total Energy Usage by Sub-Metering")

plt.show()



# Countplot (Days of Week)

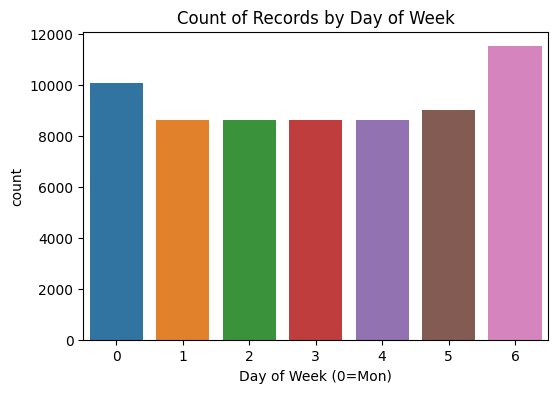
plt.figure(figsize=(6,4))

sns.countplot(x=subset\_data.index.dayofweek)

plt.title("Count of Records by Day of Week")

plt.xlabel("Day of Week (0=Mon)")

plt.show()



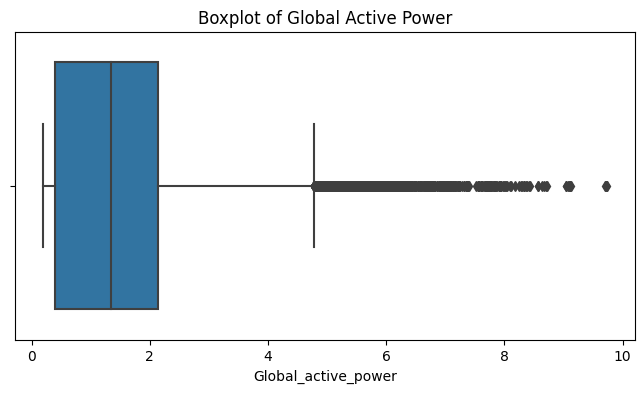
# Boxplot

plt.figure(figsize=(8,4))

sns.boxplot(x=subset\_data['Global\_active\_power'])

plt.title("Boxplot of Global Active Power")

plt.show()



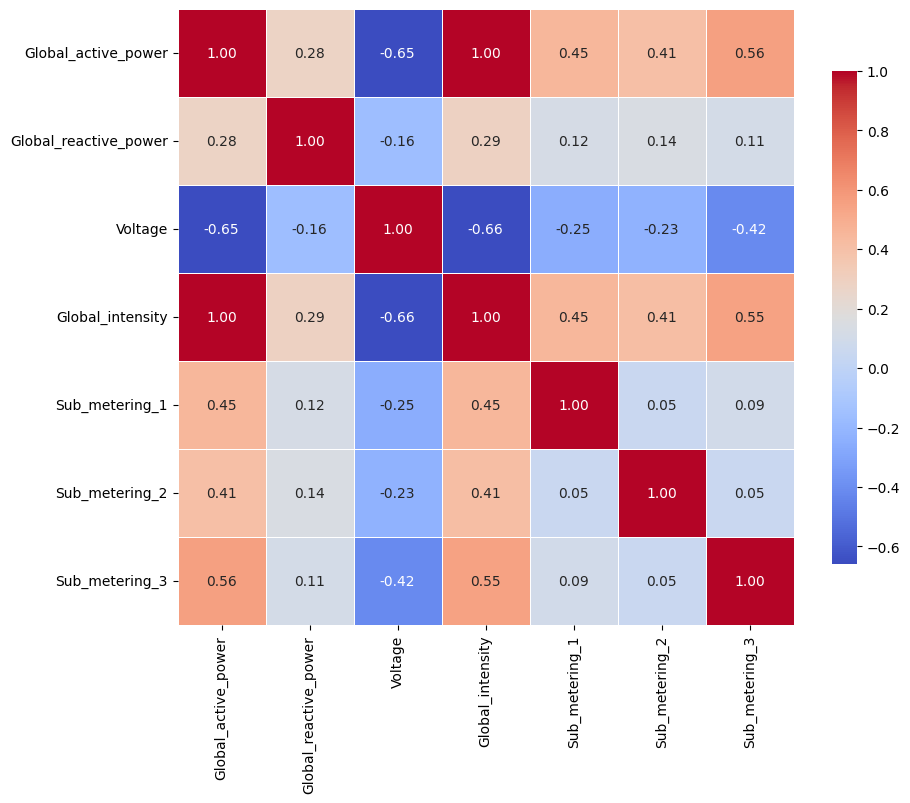
# Heatmap

plt.figure(figsize=(10, 8))

sns.heatmap(subset\_data.corr(), annot=True,

cmap='coolwarm', fmt=".2f", linewidths=0.5, square=True,

cbar\_kws={"shrink": 0.8}, annot\_kws={"size": 10})



# Correlation Heatmap

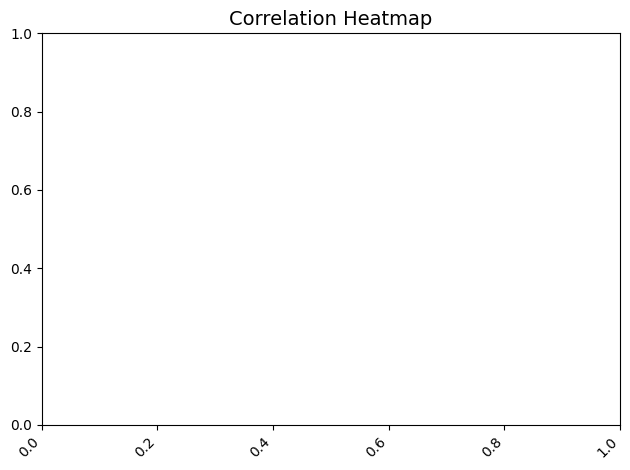
plt.title("Correlation Heatmap", fontsize=14)

plt.xticks(rotation=45, ha='right', fontsize=10)

plt.yticks(rotation=0, fontsize=10)

plt.tight\_layout()

plt.show()



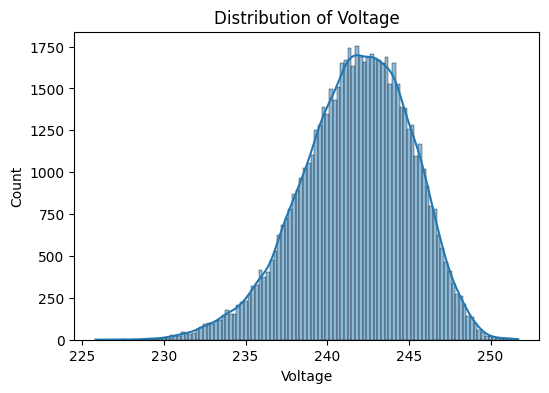
# Distplot (using histplot with KDE)

plt.figure(figsize=(6,4))

sns.histplot(subset\_data['Voltage'], kde=True)

plt.title("Distribution of Voltage")

plt.show()



# Jointplot

sns.jointplot(x='Global\_active\_power',

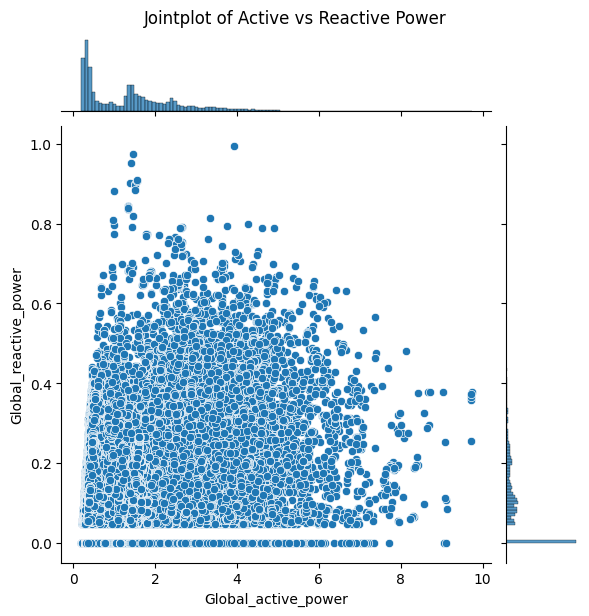
y='Global\_reactive\_power', data=subset\_data,

kind='scatter')

plt.suptitle("Jointplot of Active vs Reactive Power",

y=1.02)

plt.show()



* **Which visualization techniques shows optimum visualization.**

The visualization techniques that show optimum visualization—based on clarity, insight, and usefulness for the given dataset—are:

**Time Series Plot**

Why: It effectively shows how power consumption changes over time, highlighting trends, peaks, and fluctuations.

**Sub-Metering Plot (Line Plot)**

Why: It clearly compares energy usage across different appliances or zones over time, making it easy to identify consumption patterns.

**Correlation Heatmap**

Why: It quickly reveals relationships (positive or negative) between variables, helping identify which variables influence each other.

These three techniques are considered most optimum for visualizing the dataset because they provide the most actionable insights in an interpretable manner.