MEASURE ENERGY CONSUMPTION

Abstract Level:

Units: Define energy consumption units, such as joules, kilowatt-hours (kWh), or calories.**Metrics**: Establish key metrics, like total energy consumed, energy per unit time, or energy efficiency.**Context**: Specify the context, like a building, device, or system, where energy consumption will be measured.

Module Level:

Components: Identify the individual modules or components contributing to energy consumption.**Sensors**: Install energy measurement sensors or meters within each module.**Data Collection**: Gather real-time data on energy usage from each module.Integration: Aggregate data from all modules to calculate

overall energy consumption.**Analysis**: Analyze data to understand which modules consume the most energy and how they interact.**Optimization**: Implement energy-saving strategies at the module level, if necessary.

Combining abstract principles and modular analysis allows for a comprehensive understanding and management of energy consumption in various contexts, from a single device to an entire infrastructure.Context: Specify the context, like a building, device, or system, where energy consumption will be measured.

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from xgboost import XGBRegressor
import xgboost as xgb
from sklearn.metrics import mean
_squared_error

color_pal = sns.color_palette()
plt.style.use('fivethirtyeight')
```

```
/opt/conda/lib/python3.10/site-p
ackages/scipy/__init__.py:146: U
serWarning: A NumPy version >=1.
16.5 and <1.23.0 is required for
this version of SciPy (detected
version 1.23.5
```

```
  warnings.warn(f"A NumPy versio
n >={np_minversion} and <{np_max
version}"
```

In [2]:

```
df = pd.read_csv('/kaggle/input/
hourly-energy-consumption/PJME_h
ourly.csv',parse_dates=['Datetim
e'], index_col='Datetime')
df.head()
```

Out[2]:

In [3]:

```
df.tail()
```

Out[3]:

	PJME_MW
Datetime	
2018-01-01 20:00:00	44284.0
2018-01-01 21:00:00	43751.0
2018-01-01 22:00:00	42402.0
2018-01-01 23:00:00	40164.0
2018-01-02 00:00:00	38608.0

In [4]:

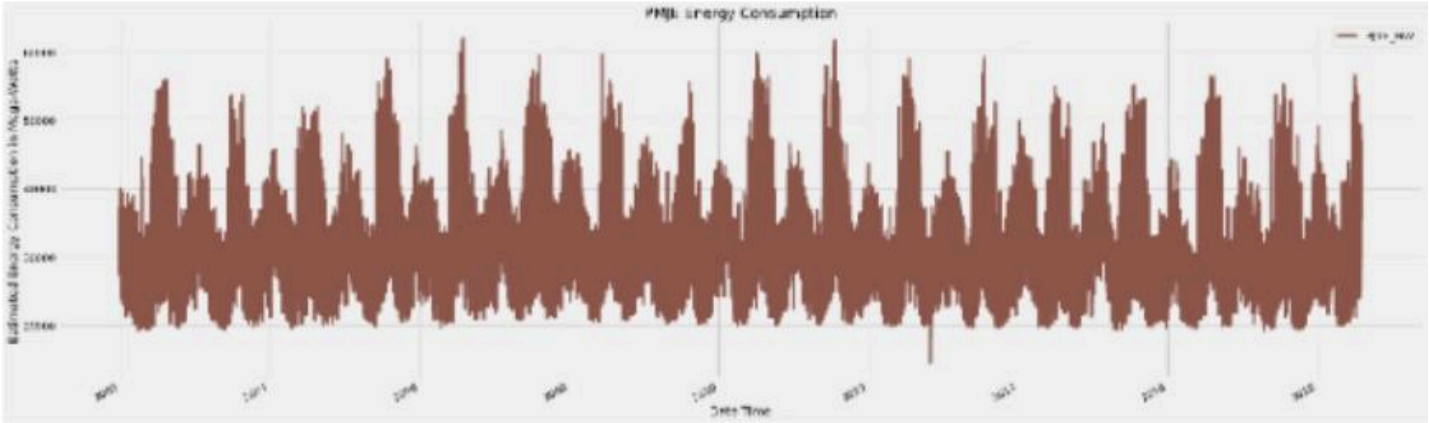
```
ax = df.plot(
    figsize=(30,10),
    color=color_pal[5],
    title='PMJE Energy Consumpti
```

In [4]:

```
ax = df.plot(
    figsize=(30,10),
    color=color_pal[5],
    title='PMJE Energy Consumpti
on',
)

ax.set_xlabel('Date-Time')
ax.set_ylabel('Estimated Energy
Consumption in Mega-Watts')

plt.show()
```

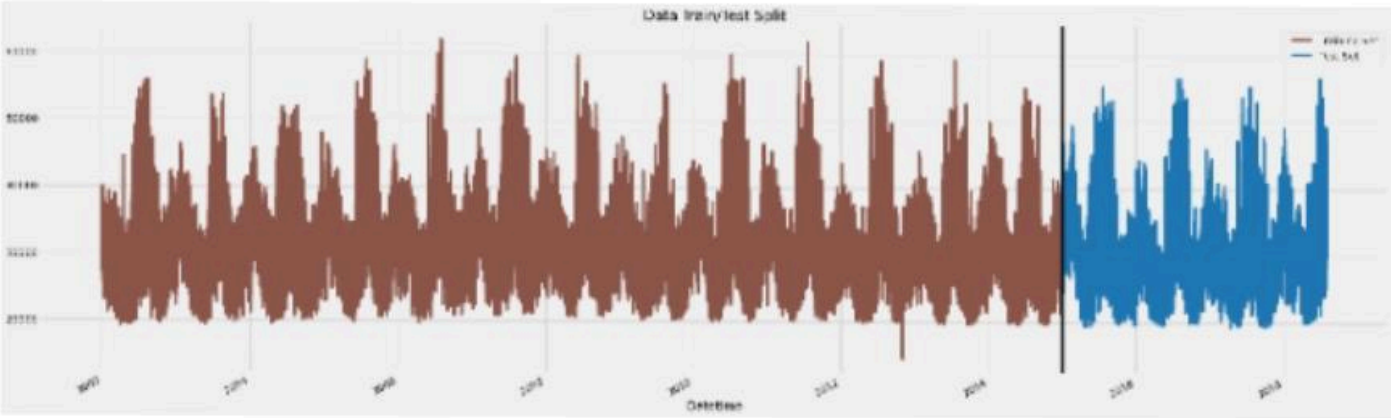


In [5]:

```
train = df.loc[df.index < '2015-
01-01']
test = df.loc[df.index >= '2015-
01-01']
```

In [6]:

```
fig, ax = plt.subplots(figsize=
(30, 10))
train.plot(ax=ax, label='Trainin
g Set', title='Data Train/Test S
plit', color=color_pal[5])
test.plot(ax=ax, label='Test Se
t', color=color_pal[0])
ax.axvline('2015', color='blac
k', ls='-')
ax.legend(['Training Set', 'Test
Set'])
plt.show()
```



In [7]:

```
df_2010 = df.loc['2010-01-01':'2
010-12-31']

fig = px.scatter(df_2010, x = df
_2010.index, y = df_2010.PJME_M
W, color=df_2010.PJME_MW)

fig.update_traces(marker=dict(si
ze=0))

fig.update_layout(
    xaxis_title='Month',
    yaxis_title='Energy Consumpt
```

```
    xaxis_title='Month',
    yaxis_title='Energy Consumpt
ion',
    title='Monthly Energy Consum
ption in 2010 (Bubble Chart)',
    xaxis=dict(
        tickvals=df.index,
        ticktext=df.index.strfti
me('%b'),
        tickangle=45
    )
)
fig.show()
```