Contents

1- Introduction

The issue of household power consumption has become a global concern due to the continuous growth of power resources and demand, resulting in an increase in energy costs (Chiavarviglio, Mellia, & Neri, 2009).

The primary aim of this assignment is to investigate the electric power consumption dataset of a specific household situated in Sceaux, which is located approximately 7km from Paris, France. The objective is to develop a reliable model for analysing and predicting the user's power consumption data. The main focus of the analysis is to determine whether there is a significant relationship between the household's global active power and electric equipment usage by constructing a linear regression model that can provide accurate predictions of power consumption based on electric equipment usage.

The linear regression model will be used to predict the power consumption, and the quality of the regression model must satisfy the assumptions. These assumptions include no multicollinearity, linearity, homoscedasticity, normality, and independence. Failure to meet these assumptions may result in unreliable linear regression analysis, requiring alternative models or methods to be employed.

The analysis of the dataset will include the use of descriptive statistics, data visualisation, correlation analysis, multiple linear regression analysis, analysis of results, and evidence of assumptions` satisfaction or unsatisfaction. Lastly, the findings will be presented with a conclusion about the regression analysis and model`s reliability.

From data exploration and regression analysis of the individual household electrical power consumption dataset, this study seeks to contribute to the understanding of household power consumption and economy. The findings could improve households to optimize their electrical power consumption and reduce energy costs, benefiting the environment and society.

1- Data Exploration

The dataset of Individual Household Electrical Power Consumption for this study is from the Machine Learning Repository – UC.

Data information:

- Measurements of electric power consumption in one household with a one-minute sampling rate over period of almost 4 years, between December 2006 and November 2010. The active energy consumed every minute (in watt hour) is represented by

global_active_power*1000/60 – submetering 1 - submetering 2 - submetering 3, electrical equipment not measured.

- The dataset contains some missing values the measurements (nearly 1,25% of the rows). All calendar timestamps are present in the dataset but for some timestamps, the measurement values are missing: a missing value is represented by the absence of value between two consecutive semi-colon attribute separators. For instance, the dataset shows missing values on April 28, 2007.
- Variables information:
    1. Date: date in format dd/mm/yyyy
    2. Time: time in format hh:mm:ss
    3. Global_active_power: household global minute-averaged active power (in kilowatt). Global active power is the power consumed by appliances other than the appliances mapped to Sub Meters. Global active power is the real power consumption i.e., the power consumed by electrical appliances other than the sub metered appliances. It is basically called watt full power.
    4. Global_reactive_power: household global minute-averaged reactive power (in kilowatt). Global reactive power is the power which bounces back and forth without any usage or leakage. It is the imaginary power consumption. It is basically called wattless power.
    5. Voltage: minute-averaged voltage (in volt).
    6. Global_intensity: household global minute-averaged current intensity (in ampere). Intensity is magnitude of the power consumed. Also called as strength of current.
    7. Sub_metering_1: energy sub-metering No. 1 (in watt-hour of active energy). It corresponds to the kitchen, containing mainly a dishwasher, an oven, and a microwave (hot plates are not electric but gas powered).
    8. Sub_metering_2: energy sub-metering No. 2 (in watt-hour of active energy). It corresponds to the laundry room, containing a washing-machine, a tumble-drier, a refrigerator and a light.
    9. Sub_metering_3: energy sub-metering No. 3 (in watt-hour of active energy). It corresponds to an electric water-heater and an air-conditioner.

2.1- Approach

The Python programming language was used to import a dataset with 2075259 rows and 9 columns into a Jupyter notebook. The dataset was checked for duplicates, and it was found that the data contained the strings ‘nan’ and ‘?’. To address this issue, both were converted to NumPy’s NaN and removed, resulting in a dataset with 2049280 rows and 9 columns. The variables were initially imported as objects but were converted to ‘float64’. The ’Date’ and ‘Time’ columns were combined into a ‘data_time’ column using datetime format (yyyy-mm-dd), and new columns for ‘Year’, ‘Month’, and ‘Weekday’ were added. The ‘Date’ and ‘Time’ columns were then dropped to facilitate further exploration of the dataframe. The focus of the study will be on ‘Global_active_power’ by electric equipment categorized into ‘Sub_metering_1’, ‘Sub_metering_2’ and ‘Sub_metering_3’.

The information shown in Table 1, the summary statistics reveal that the dataset contains 2,049,280 reading across 9 columns. The average value for ‘Sub_metering_3’ is higher than that of Submetering 1 and 2, with value of 6.56 watt-hour. Additionally, the average value for ‘Global_active_power’ is 1.09 kilowatts. Notably, the minimum reading for all submetering variable is zero. Conversely, ‘Sub_metering_1’ has the highest maximum readings of 88.00 watt-hour, surpassing the maximum readings for Submetering 2 and 3.

Table 1: Summary statistic of the household electric power consumption dataset.

	count	mean	std	min	25%	50%	75%	max
Year	2049280.0	2008.424761	1.124388	2006.000	2007.000	2008.000	2009.000	2010.000
Month	2049280.0	6.497968	3.446016	1.000	4.000	7.000	10.000	12.000
Global_active_power	2049280.0	1.091615	1.057294	0.076	0.308	0.602	1.528	11.122
Global_reactive_power	2049280.0	0.123714	0.112722	0.000	0.048	0.100	0.194	1.390
Voltage	2049280.0	240.839858	3.239987	223.200	238.990	241.010	242.890	254.150
Global_Intensity	2049280.0	4.627759	4.444396	0.200	1.400	2.600	6.400	48.400
Sub_metering_1	2049280.0	1.121923	6.153031	0.000	0.000	0.000	0.000	88.000
Sub_metering_2	2049280.0	1.298520	5.822026	0.000	0.000	0.000	1.000	80.000
Sub_metering_3	2049280.0	6.458447	8.437154	0.000	0.000	1.000	17.000	31.000

2.2- Visualisation

Exploring the dataset by visualisation, from histogram Figure 1 was observed that the frequency of ‘Global_active_power’ presented distribution skewed to the right similarity with ‘Global_intensity’, meanwhile ‘Voltage’ presented a bell-shaped distribution, and Submetering categories uneven distribution, which may indicate each bar

does not contain enough data points to accurately show the distribution of the data, no satisfying normality assumption.

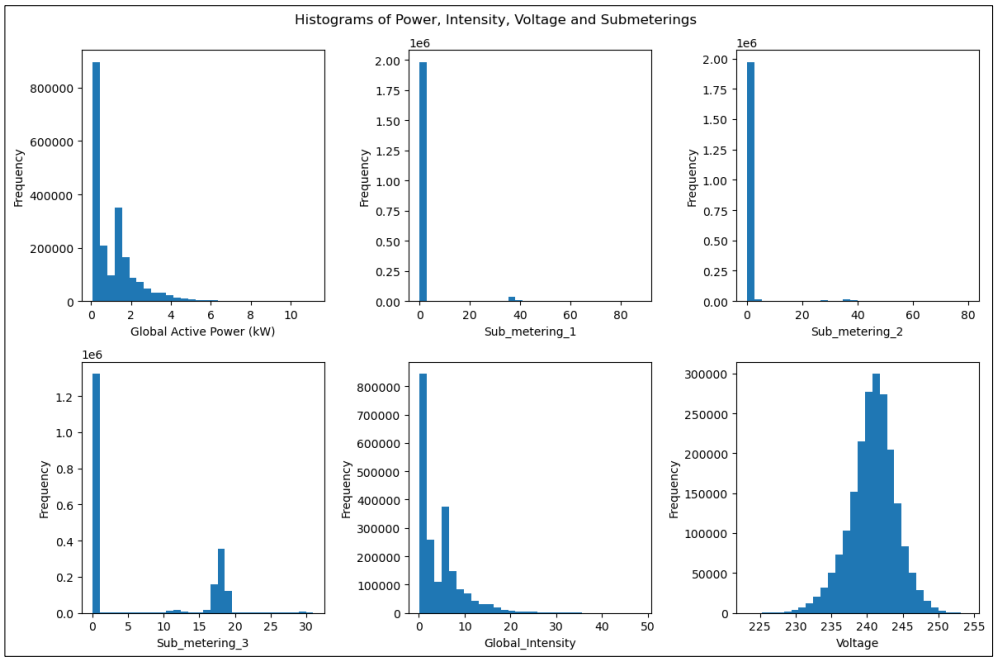


Figure 1. Variables histograms measured by frequency.

Further investigation by time consumption, it is evident from the 3 lines graph displaying the 'Global_active_power' by years, month, and weekdays (Figure 2), that there is a clear pattern. The readings begin in 2006 with a peak, drop, and subsequently remain relatively constant until 2010. When analysing the readings on monthly basis, it is evident that consumption is higher during winter months and lower during summer months.

This may imply that households require more electric power during winter to keep their homes warm. Furthermore, analysing the line graph by weekdays shows that the readings are higher on weekends. This observation suggests that the household tends to spend more time at home on weekends and leave their homes for working during weekdays.

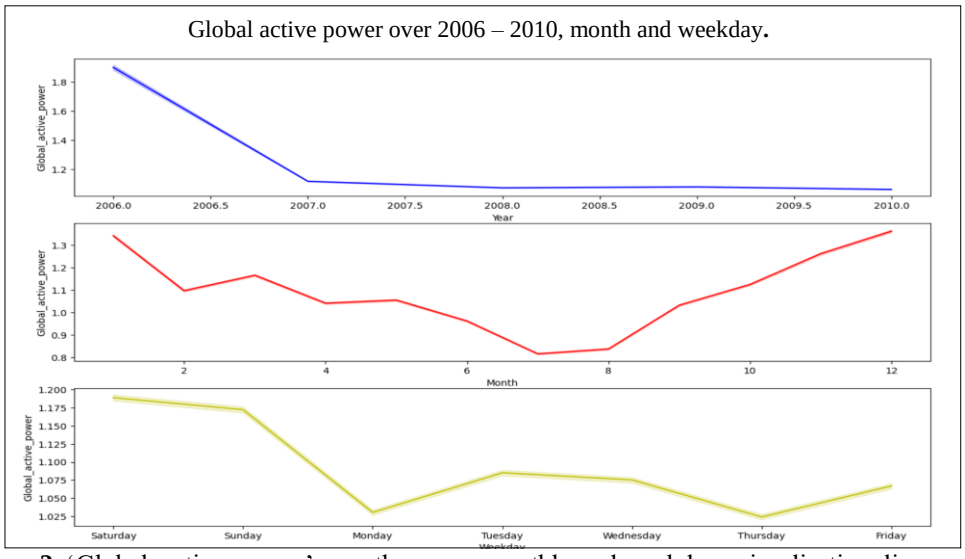


Figure 2. 'Global_active_power' over the years, monthly and weekdays visualisation, line graph.

Sub_metering_1, Sub_metering_2, Sub_metering_3 box plot of data distribution, as shown in Figure 3.

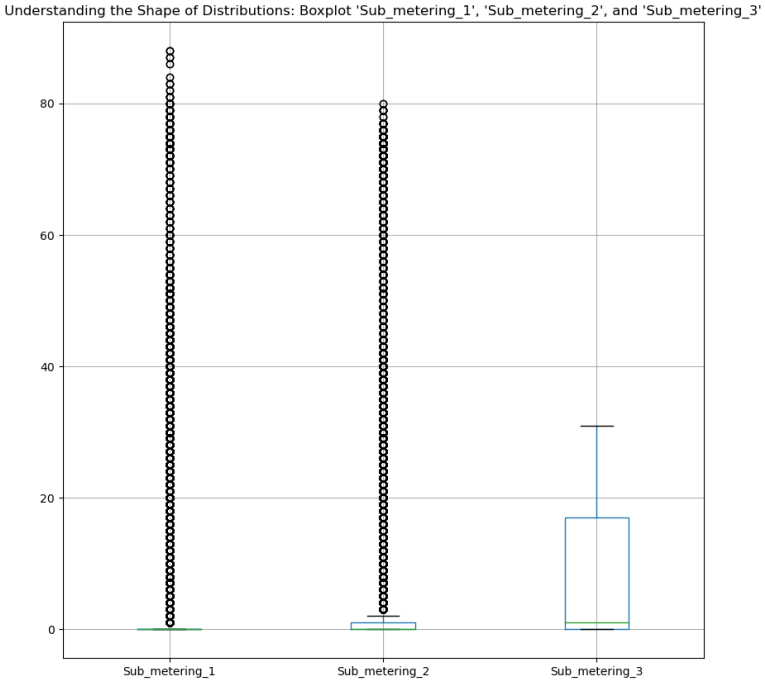


Figure 3. Box plot for electric equipment power consumption (Sub_metering_1, Sub_metering_2, Sub_metering_3).

Table 2: Summary statistic of electric equipment power consumption (Sub_metering_1, Sub_metering_2, Sub_metering_3).

	Type	Max value	Min value	Mean value	Median value
0	Sub_metering_1	88.0	0.0	1.121923	0.0
1	Sub_metering_2	80.0	0.0	1.298520	0.0
2	Sub_metering_3	31.0	0.0	6.458447	1.0

As illustrated in Figure 3, it is apparent that the Sub_metering_3 exhibits the highest average power consumption, followed by Sub_metering_2, and lastly, Sub_metering_1.

According to the data, power consumption tends to be higher during winter season and weekend. Additionally, the electric equipment that contributes the most to consumption is ‘Sub_metering_3’, which includes water-heaters and air conditioners, with an average consumption of 6.46 watt-hour.

2- Correlation Analysis

A correlation analysis was conducted, and a regression model discussed, taking into consideration the assumptions that needed to be satisfied, including but not limited to the absence of multicollinearity, linearity between the dependent and independent variables, homoscedasticity, normality, and independence.

3.1- Multicollinearity

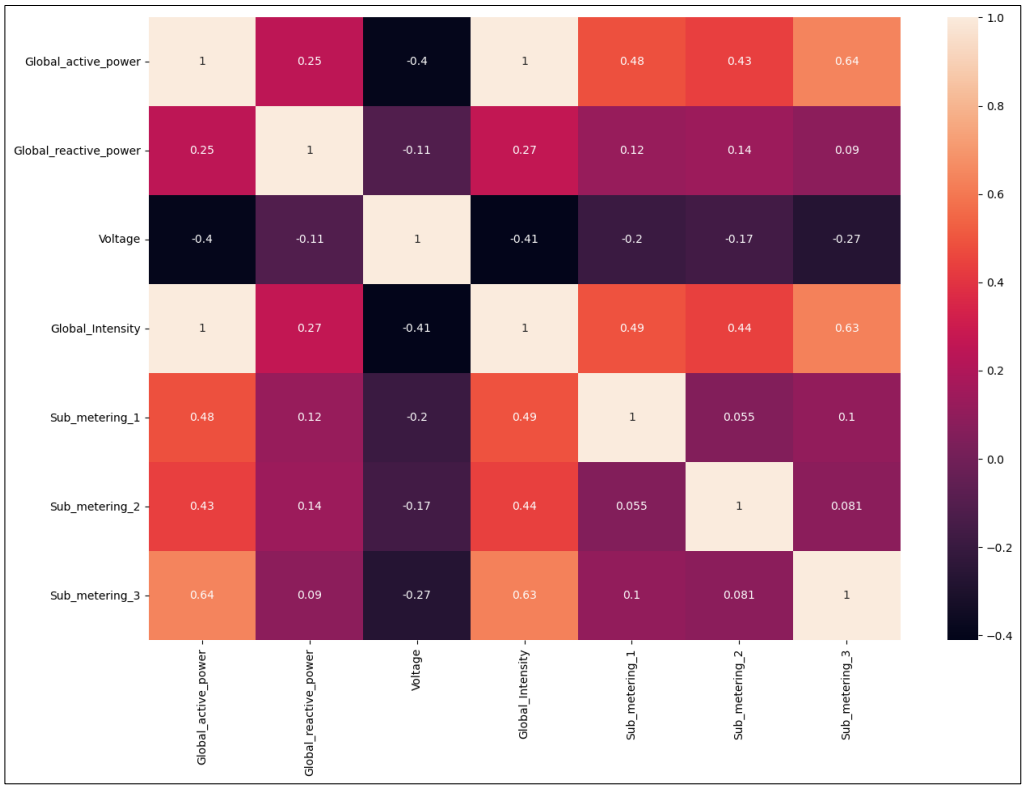


Figure 4. Heatmap correlation matrix.

The correlation coefficients presented on heatmap matrix a range from -0.4 to 1, with values closer to 1 indicating a strong positive correlation and values closer to -0.4 indicating a weak negative correlation. A correlation coefficient of 0 indicates no correlation between the variables.

The highest positive correlation coefficient observed on the heatmap is between 'Global_active_power' and 'Global_intensity', which a coefficient of 1. This indicates that an increase in current intensity results in a corresponding increase in active power consumption. This is an expected result since active power represents the rate at which energy is consumed, while current intensity measures the amount of current flowing through the circuit (Amin & Nasserzadeh, 2014). Additionally, the heatmap matrix indicates that there is no significant correlation between the independent variables, indicating that there is no multicollinearity among them.

Furthermore, by scatter plotting the matrix can be analysed, as shown in Figure 5.

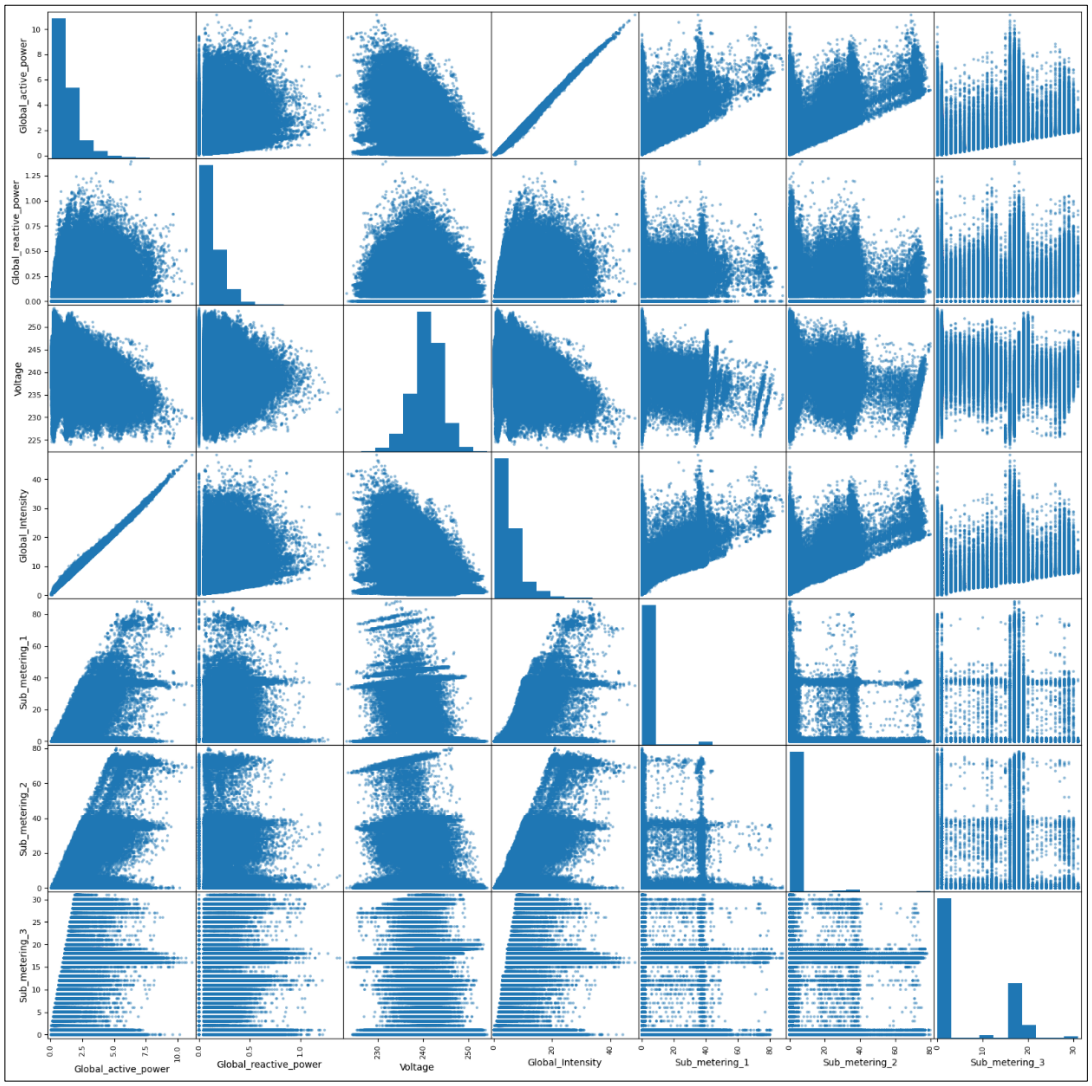


Figure 5. Pairwise scatterplots of all combinations.

From the scatterplots is visible the perfect positive linear correlation between 'Global_active_power' and 'Global_intensity". Moreover, a moderate positive linear correlation between the 'Sub_metering_1', Sub_metering_2' and 'Sub_metering_3'.

From the correlation between independent variables presented only moderate positive and negative correlation, no high correlation between them.

3.2- Linearity and Homoscedasticity

From the scatterplots figure 5, could be observed the linearity between the variables, with positive linear and negative linear. Further analyses were performed applying linear regression between the dependent variable 'Global_active_power' and independent variables 'Global_intensity', 'Sub_metering_1', 'Sub_metering_2', 'Sub_metering_3', Voltage and 'Global_reactive_power'.

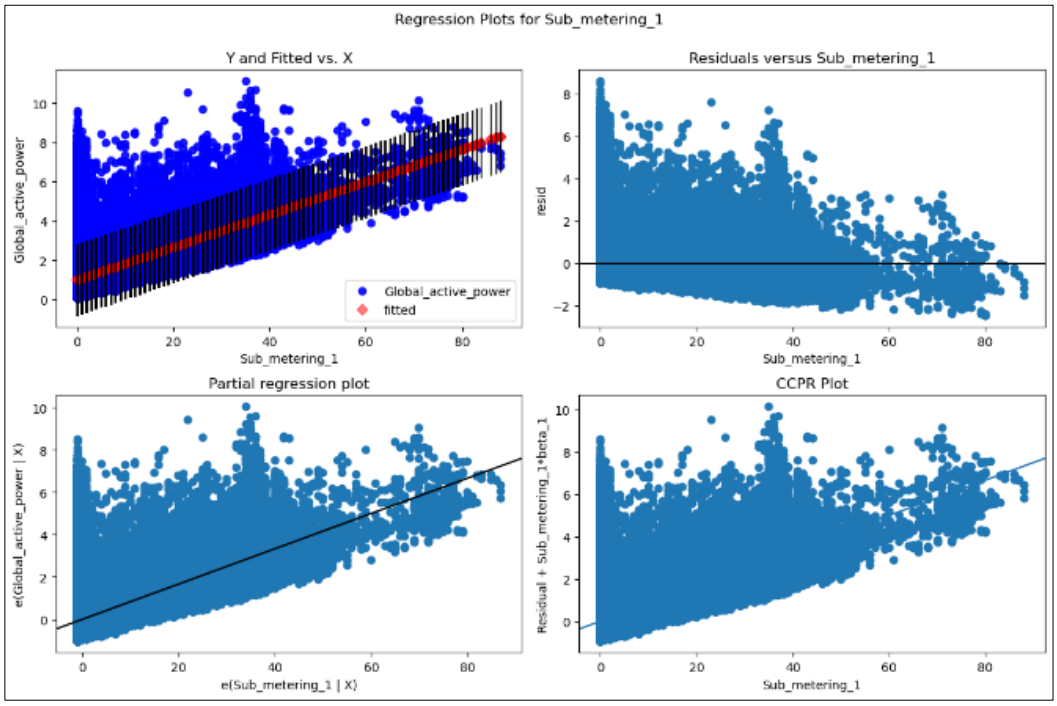


Figure 5.0. Regression plot for 'Sub_metering_1'.

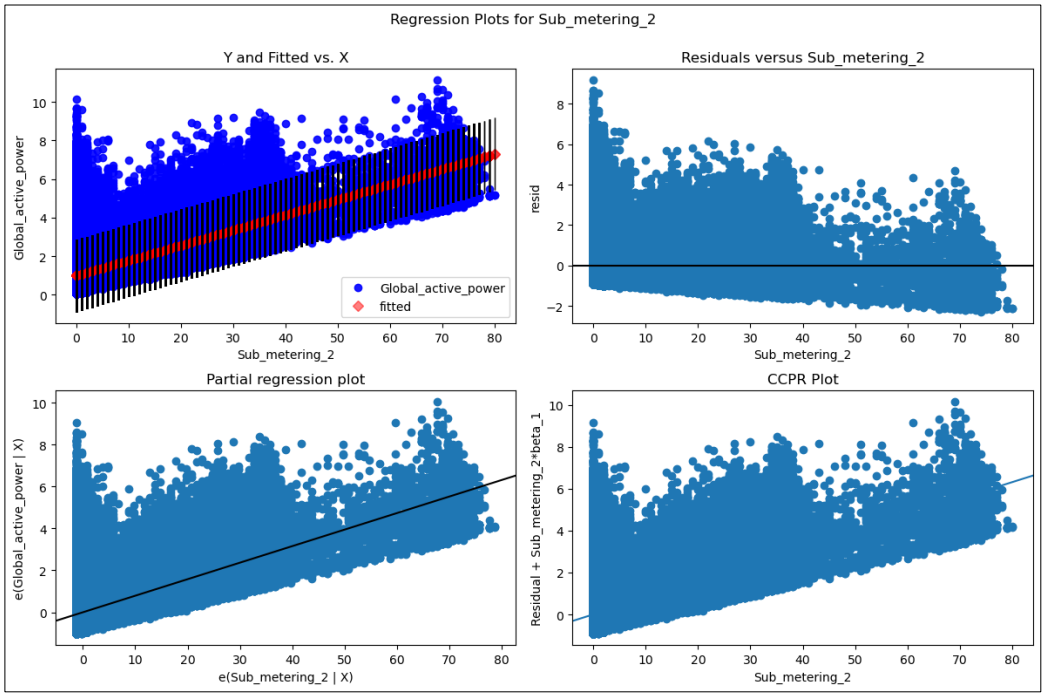


Figure 5.1. Regression plot for 'Sub_metering_2'.

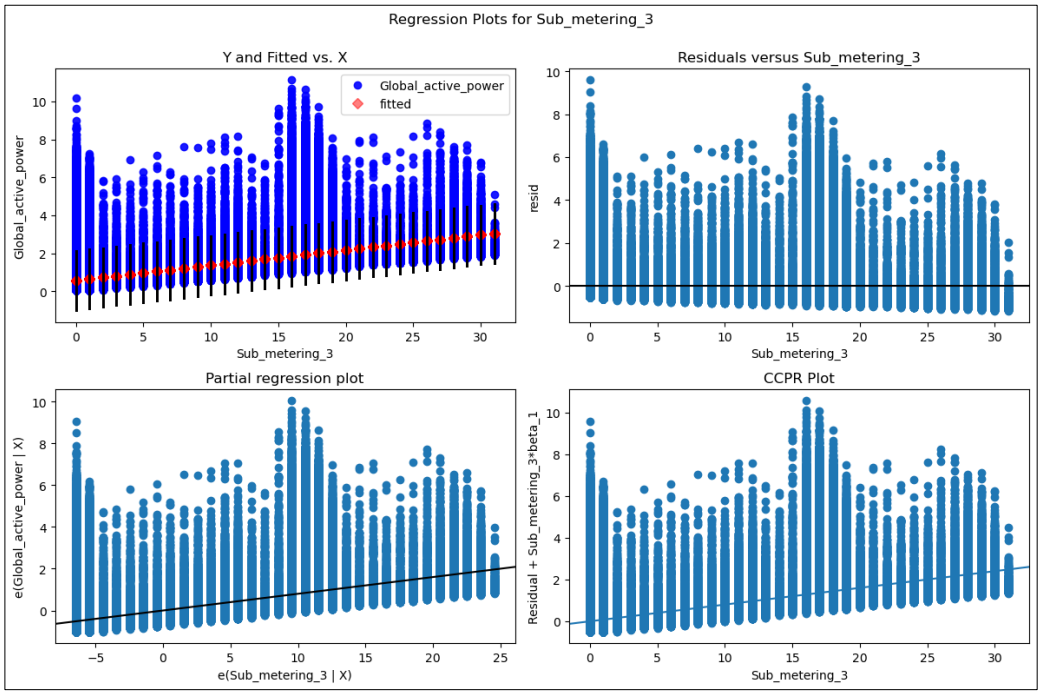


Figure 5.2. Regression plot for 'Sub_metering_2'.

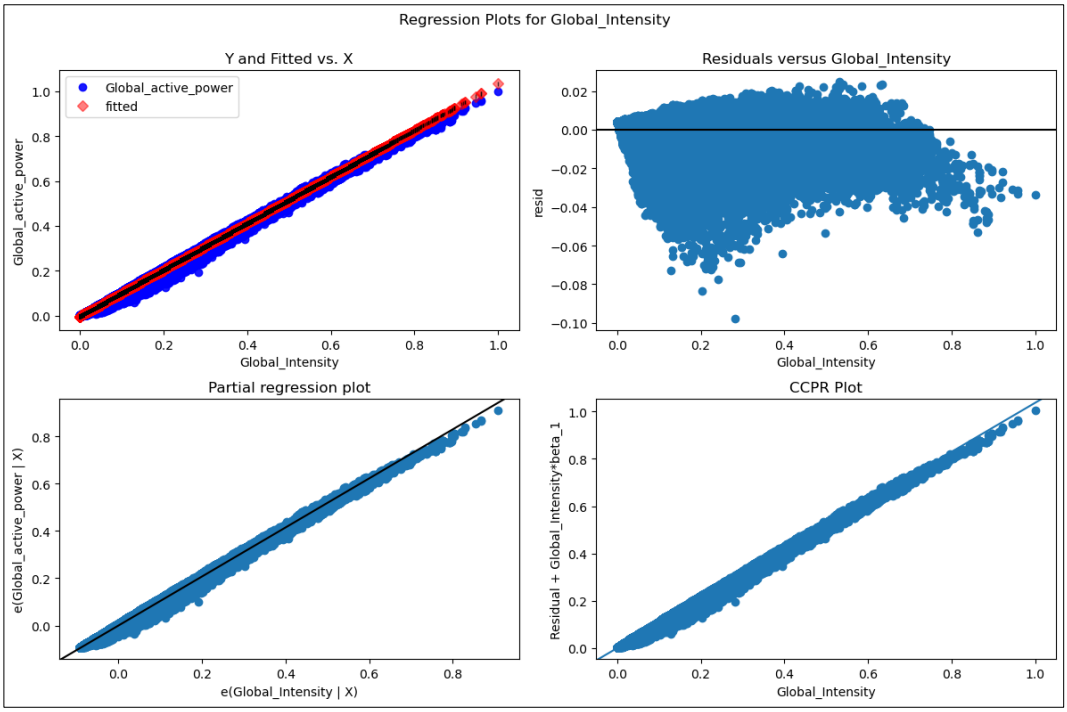


Figure 5.3. Regression plot for 'Global_intensity'.

The residuals in a regression analysis from the Figures 5.0, 5.1, 5.2, and 5.3 are typically expected to be randomly distributed around zero, indicating that the model is a good fit for the data. However, upon examining the residuals for Submetering 1, 2 and 3, the distribution is not centred around zero. Instead, most of the residuals for these submetering are clustered above zero, while some are under zero. Meanwhile, the residuals for 'Global_intensity' presented cluster more concentrated under zero than above zero. This

pattern suggests that the model may overestimate or underestimate the dependent variable for certain data points, leading to a bias towards higher values.

Upon further examination of the residuals, it is evident that for 'Sub_metering_1' and 'Sub_metering_2', the residuals with lower predicted values are mostly positive, while some are negative, and for higher predicted values, the residuals are centred around the zero line. For 'Sub_metering_3', most of the residuals are positive and constant, indicating that the model tends to overestimate the dependent variable for most data points. These observations suggest the presence of heteroscedasticity, which can make it difficult to draw accurate conclusions from the analysis.

To improve the accuracy of the model, it may be necessary to explore alternative regression techniques or data transformation methods. The coefficient of result table (Appendix A) shows that the R^2 values for Submetering 1, 3 and 3 are low in simple linear regression. However, 'Global_intensity' shows a perfect linear relationship with the dependent variable, with R^2 value of 0.998. Adding more independent variables may increase the R^2 value, as the study focuses on the Submetering categories.

Table 3: R^2, Coefficient and p-value for Submetering 1, 2 and 3.

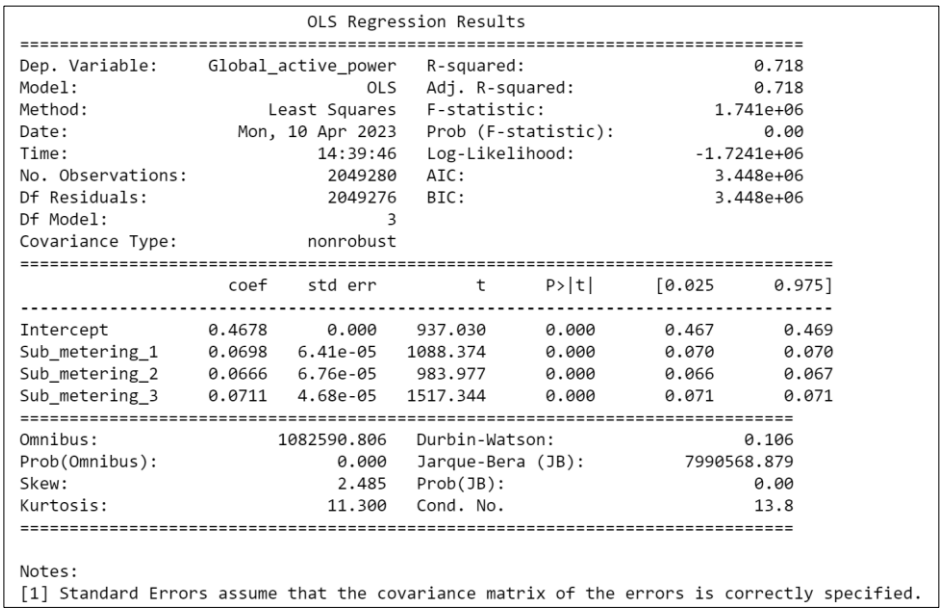
	Variable	R-squared	Coefficient	P-Value
0	Global_Intensity	0.998	1.0369	0.0
1	Sub_metering_1	0.235	0.0832	0.0
2	Sub_metering_2	0.189	0.0789	0.0
3	Sub_metering_3	0.408	0.0800	0.0

3- Model

To enhance the Model, two iterations were performed by adding more independent variables to increase the R^2 value. The first iteration, Model 1, included the independent variables 'Sub_metering_1', 'Sub_metering_2', and 'Sub_metering_3', resulting in an R^2 value of 0.718 or 71.8%, as indicated in the coefficient table results shown in Figure 6. In contrast, Model 2 included the independent variables from Model 1 along with 'Global_intensity' and 'Voltage', resulting a higher R^2 value of 0.998 or 99.8%, as shown in the coefficient table results in Figure 7.

The substantial increase in R^2 from Model 1 to Model 2 suggests that adding the additional independent variables significantly improve the fit of the model. The combined independent variables in Model 2 explain a larger proportion of the variance in the

dependent variable than the variables included in Model 1, as evidence by the higher R^2 value. These results provide evidence that the additional independent variables are valuable predictors of the dependent variable and suggest that Model 2 may be a more reliable model for predicting the outcome variable.



OLS Regression Results

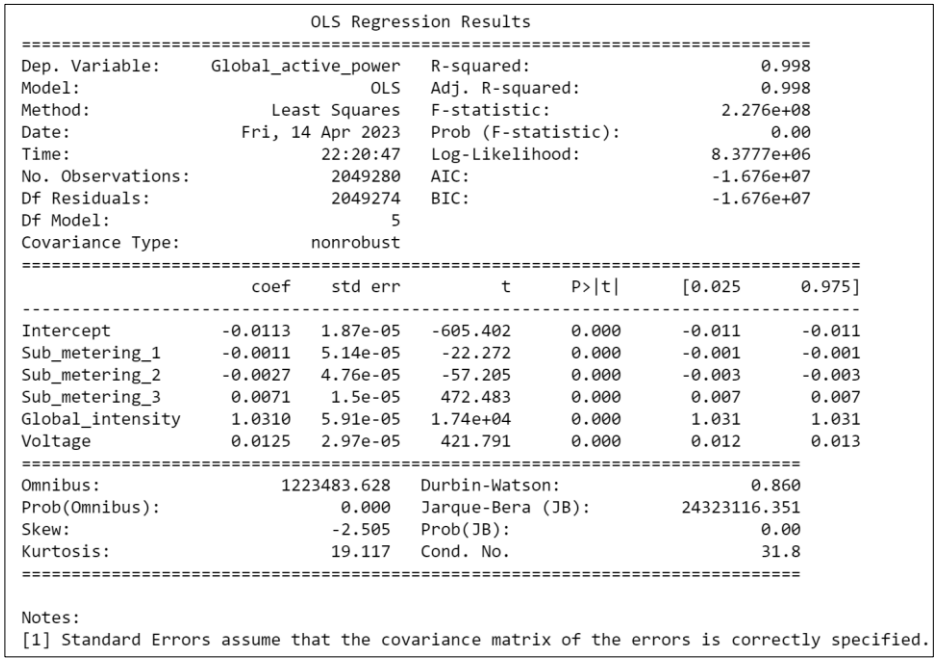
Dep. Variable:	Global_active_power	R-squared:	0.718
Model:	OLS	Adj. R-squared:	0.718
Method:	Least Squares	F-statistic:	1.741e+06
Date:	Mon, 10 Apr 2023	Prob (F-statistic):	0.00
Time:	14:39:46	Log-Likelihood:	-1.7241e+06
No. Observations:	2049280	AIC:	3.448e+06
Df Residuals:	2049276	BIC:	3.448e+06
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.4678	0.000	937.030	0.000	0.467	0.469
Sub_metering_1	0.0698	6.41e-05	1088.374	0.000	0.070	0.070
Sub_metering_2	0.0666	6.76e-05	983.977	0.000	0.066	0.067
Sub_metering_3	0.0711	4.68e-05	1517.344	0.000	0.071	0.071

Omnibus:	1082590.806	Durbin-Watson:	0.106
Prob(Omnibus):	0.000	Jarque-Bera (JB):	7990568.879
Skew:	2.485	Prob(JB):	0.00
Kurtosis:	11.300	Cond. No.	13.8

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Figure 6. Regression results for Model 1.



OLS Regression Results

Dep. Variable:	Global_active_power	R-squared:	0.998
Model:	OLS	Adj. R-squared:	0.998
Method:	Least Squares	F-statistic:	2.276e+08
Date:	Fri, 14 Apr 2023	Prob (F-statistic):	0.00
Time:	22:20:47	Log-Likelihood:	8.3777e+06
No. Observations:	2049280	AIC:	-1.676e+07
Df Residuals:	2049274	BIC:	-1.676e+07
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-0.0113	1.87e-05	-605.402	0.000	-0.011	-0.011
Sub_metering_1	-0.0011	5.14e-05	-22.272	0.000	-0.001	-0.001
Sub_metering_2	-0.0027	4.76e-05	-57.205	0.000	-0.003	-0.003
Sub_metering_3	0.0071	1.5e-05	472.483	0.000	0.007	0.007
Global_intensity	1.0310	5.91e-05	1.74e+04	0.000	1.031	1.031
Voltage	0.0125	2.97e-05	421.791	0.000	0.012	0.013

Omnibus:	1223483.628	Durbin-Watson:	0.860
Prob(Omnibus):	0.000	Jarque-Bera (JB):	24323116.351
Skew:	-2.505	Prob(JB):	0.00
Kurtosis:	19.117	Cond. No.	31.8

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Figure 7. Regression results for Model 2.

4.1- Normality

One common method to assess the normality of residuals is by using a quantile-quantile plot, known as Q-Q plot, which compares the distribution of the residuals to a normal distribution. Ideally, the points in the Q-Q plot should fall along straight line, indicating that the residuals are normally distributed. However, in some cases, the red line in the Q-Q plot may not align perfectly with the blue line, which can raise questions about the normality of residuals (Mardem, 2004).

In this analysis, Figure 8. indicates that the distribution of Residuals from Model 2 is left-skewed or negative exponential distribution, which results in a curved appearance of the Q-Q plot for left-skewed data. It is important to note that a deviation of the red line from the blue line does not necessarily indicate non-normality of residuals. Instead, it suggests that there may be some deviation from normality, but the deviation may not be significant enough to invalidate the normality assumption. Some normality shows on Q-Q plot from around -2 from Theorical Quantiles to around 5, which would match the bell-shaped from histogram of Residuals, which shows a normal distribution from -0.02 to 0.02.

To further validate the normality of residuals, other statistical tests can be conducted.

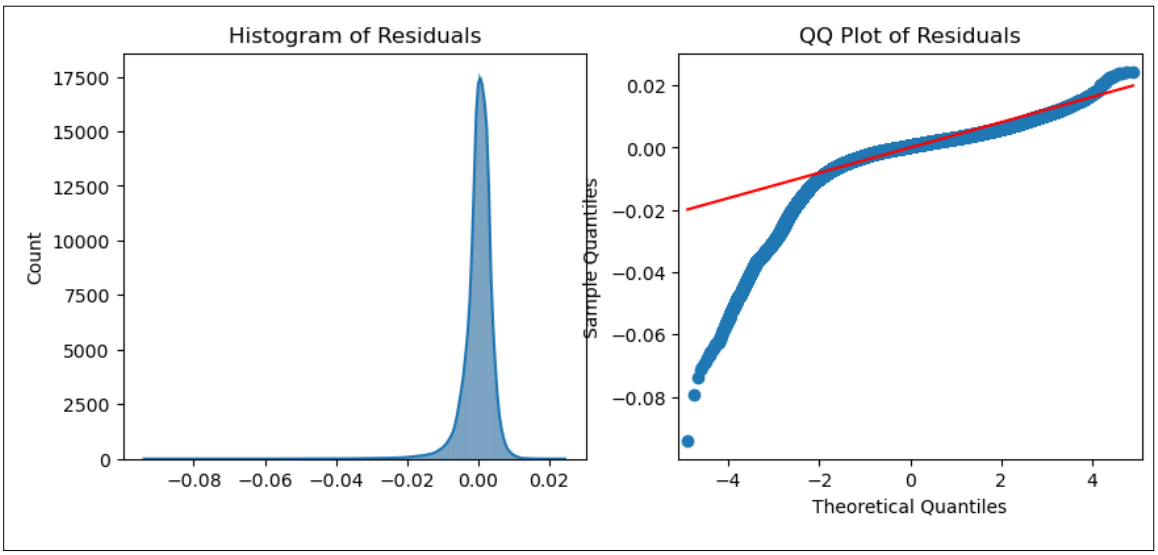


Figure 8. Residual Normal distribution, Global active power distribution and residuals Q-Q plot.

4.2- Independence

For the independence assumption to meet is require the observations to be independent of each other, usually the violation occurs in time series regression model. The violation of statistical independence indicates that the model could be improved. Meanwhile, in non-time series models, the statistical independence violation would be present if the model systematically underpredicts or overpredicts the coefficient estimates (Flatt & Jacobs, 2019).

By observing the plots on the graph from Submetering 1, 2 and 3 was noticed that the plots on the graphs were not random and presenting a positive autocorrelation. Also, a numerical analysis applying the method Durbin-Watson coefficient from the Model 2, the result is 0.8601, which to satisfied assumption the test statistic is close to 2 (between 1.5 and 2.5), therefore the test statistic is less than 2, it suggests that there may be positive autocorrelation in the residuals, which mean that the residuals are not independent and are correlated with one another over time or across observations, matching with the graphic observation. Which may need to be addressed by adjusting the model or the data, and not satisfying the independence assumption.

5- Statistical Inference

This study aimed to examine the potential relationship between household electric power consumption's Global active power and Submetering 1, 2 and 3 (electric equipment). After exploring the dataset, it was found that Global active power consumption was higher during winter and weekends. Additionally, 'Sub_metering_3' had the highest average, indicating that water heaters consume more power during the winter months.

We performed linear regression analysis and tested the model's assumptions. The multicollinearity assumption was satisfied since the correlation matrix did not show high correlation between the independent variables. However, the linearity assumption was not fully satisfied, despite scatterplots showing moderate positive linear correlation between 'Global_active_power' and Submetering 1, 2 and 3 and negative correlation with Voltage.

The homoscedasticity assumption was rejected because the scatterplot of the residuals showed unequal spread across all independent variable values. To address this issue, we suggest log transformation the dependent variable and using auto-regressive conditional heteroscedasticity models to model error variance.

The normality assumption was also not satisfied, as the histogram and Q-Q plot of the residuals showed negatively skewed data. One potential solution to this issue is to apply nonlinear transformations of the variables, exclude long-tailed variables, or remove outliers.

Furthermore, we tested the independence assumption by analysing graphs plotting from the regression analysis and using the Durbin-Watson coefficient, which indicated that the assumption was not satisfied.

All of the models showed a p-value of zero, indicating that the findings are statistically significant, as p-value is below the typical threshold of 0.05. While the displayed p-value

may appear to be zero, it is an extremely small number, often computed with several decimal places by statistical software.

Despite Model 2 having the highest R^2 value of 99.8% in the OLS Regression table, it is important to consider that violated assumptions could potentially impact the accuracy of the model and its ability to make precise predictions.

Thus, it is recommended to conduct further analysis to improve the model. Possible solutions include transforming variables, removing outliers, including lag variables or interaction terms, or adding additional variables. Implementing these solutions can help improve the model's accuracy and provide more reliable results. A high R^2 value alone is not sufficient to justify the model's accuracy.

6- Appendix

Electric Power Consumption Regression Linear Analysis Code

6.1- Importing Library and dataset

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import scipy.stats as stats

from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler

from sklearn import linear_model
```

In [2]:

```
df = pd.read_csv("household_power_consumption.csv", sep=';')
df
C:\Users\cinar\AppData\Local\Temp\ipykernel_6532\2371335474.py:1:
DtypeWarning: Columns (2,3,4,5,6,7) have mixed types. Specify dtype
option on import or set low_memory=False.
  df = pd.read_csv("household_power_consumption.csv", sep=';')
```

Out[2]:

	Date	Time	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
0	16/12/2006	17:24:00	4.216	0.418	234.840	18.400	0.000	1.000	17.0
1	16/12/2006	17:25:00	5.360	0.436	233.630	23.000	0.000	1.000	16.0
2	16/12/2006	17:26:00	5.374	0.498	233.290	23.000	0.000	2.000	17.0
3	16/12/2006	17:27:00	5.388	0.502	233.740	23.000	0.000	1.000	17.0
4	16/12/2006	17:28:00	3.666	0.528	235.680	15.800	0.000	1.000	17.0
...	...	...	...	...	...	...	...	...	...
2075254	26/11/2010	20:58:00	0.946	0.0	240.43	4.0	0.0	0.0	0.0
2075255	26/11/2010	20:59:00	0.944	0.0	240.0	4.0	0.0	0.0	0.0
2075256	26/11/2010	21:00:00	0.938	0.0	239.82	3.8	0.0	0.0	0.0
2075257	26/11/2010	21:01:00	0.934	0.0	239.7	3.8	0.0	0.0	0.0
2075258	26/11/2010	21:02:00	0.932	0.0	239.55	3.8	0.0	0.0	0.0

2075259 rows × 9 columns

6.2- Cleaning the data

Checking for duplicates, Null/ NaN values.

In [3]:

```
format (len(df[df.duplicated()]))
```

Out[3]:

```
'0'
```

In [4]:

```
df.isnull().sum()
```

Out[4]:

```
Date                        0
Time                        0
Global_active_power         0
Global_reactive_power       0
Voltage                     0
Global_intensity            0
Sub_metering_1              0
Sub_metering_2              0
Sub_metering_3          25979
dtype: int64
```

In [5]:

```
# Information about dataframe.

print(df.info())
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075259 entries, 0 to 2075258
Data columns (total 9 columns):
 #   Column                 Dtype
---  ------                 -----
 0   Date                   object
 1   Time                   object
 2   Global_active_power    object
 3   Global_reactive_power  object
 4   Voltage                object
 5   Global_intensity       object
 6   Sub_metering_1         object
 7   Sub_metering_2         object
 8   Sub_metering_3         float64
dtypes: float64(1), object(8)
memory usage: 142.5+ MB
None
```

In [6]:

```
df = df.replace(["?", ""], np.nan)

df[['Global_active_power', 'Global_reactive_power', 'Voltage',
'Global_intensity', 'Sub_metering_1', 'Sub_metering_2',
'Sub_metering_3']] = df[['Global_active_power', 'Global_reactive_power',
'Voltage', 'Global_intensity', 'Sub_metering_1', 'Sub_metering_2',
'Sub_metering_3']].apply(pd.to_numeric)

df = df.dropna()
print(df.isnull().sum())
Date                    0
Time                    0
Global_active_power     0
Global_reactive_power   0
Voltage                 0
Global_intensity        0
Sub_metering_1          0
Sub_metering_2          0
Sub_metering_3          0
dtype: int64
```

In [7]:

```
df.shape
```

Out[7]:

```
(2049280, 9)
```

In [8]:

```
print(df.dtypes)
```

```
Date                    object
Time                    object
Global_active_power     float64
Global_reactive_power   float64
Voltage                 float64
Global_intensity        float64
Sub_metering_1          float64
Sub_metering_2          float64
Sub_metering_3          float64
dtype: object
```

In [9]:

```
df
```

Out[9]:

	Date	Time	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
0	16/12/2006	17:24:00	4.216	0.418	234.84	18.4	0.0	1.0	17.0
1	16/12/2006	17:25:00	5.360	0.436	233.63	23.0	0.0	1.0	16.0
2	16/12/2006	17:26:00	5.374	0.498	233.29	23.0	0.0	2.0	17.0
3	16/12/2006	17:27:00	5.388	0.502	233.74	23.0	0.0	1.0	17.0
4	16/12/2006	17:28:00	3.666	0.528	235.68	15.8	0.0	1.0	17.0
...	...	...	...	...	...	...	...	...	...
2075254	26/11/2010	20:58:00	0.946	0.000	240.43	4.0	0.0	0.0	0.0
2075255	26/11/2010	20:59:00	0.944	0.000	240.00	4.0	0.0	0.0	0.0
2075256	26/11/2010	21:00:00	0.938	0.000	239.82	3.8	0.0	0.0	0.0
2075257	26/11/2010	21:01:00	0.934	0.000	239.70	3.8	0.0	0.0	0.0
2075258	26/11/2010	21:02:00	0.932	0.000	239.55	3.8	0.0	0.0	0.0

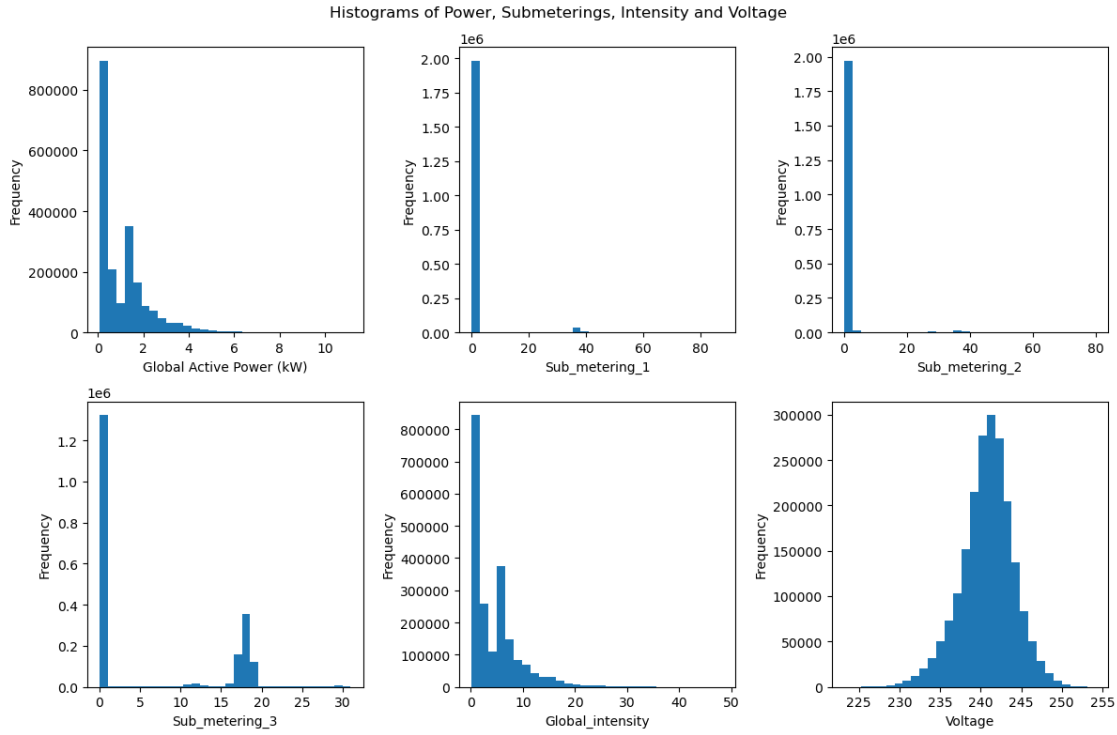
2049280 rows × 9 columns

6.3- Exploratory Data Analysis

In [10]:

```
# Investigate the data distribution by columns, using Histogram:

global_active_power = df['Global_active_power']
submetering1 = df['Sub_metering_1']
submetering2 = df['Sub_metering_2']
submetering3 = df['Sub_metering_3']
global_intensity = df['Global_intensity']
voltage = df['Voltage']

fig, axs = plt.subplots(2, 3, figsize=(12, 8))
fig.suptitle('Histograms of Power, Submeterings, Intensity and Voltage')
```

```
axs[0, 0].hist(global_active_power, bins=30)
axs[0, 0].set_xlabel('Global Active Power (kW)')
axs[0, 0].set_ylabel('Frequency')

axs[0, 1].hist(submetering1, bins=30)
axs[0, 1].set_xlabel('Sub_metering_1')
axs[0, 1].set_ylabel('Frequency')

axs[0, 2].hist(submetering2, bins=30)
axs[0, 2].set_xlabel('Sub_metering_2')
axs[0, 2].set_ylabel('Frequency')

axs[1, 0].hist(submetering3, bins=30)
axs[1, 0].set_xlabel('Sub_metering_3')
axs[1, 0].set_ylabel('Frequency')

axs[1, 1].hist(global_intensity, bins=30)
axs[1, 1].set_xlabel('Global_intensity')
axs[1, 1].set_ylabel('Frequency')

axs[1, 2].hist(voltage, bins=30)
axs[1, 2].set_xlabel('Voltage')
axs[1, 2].set_ylabel('Frequency')


plt.tight_layout()
plt.show()
```



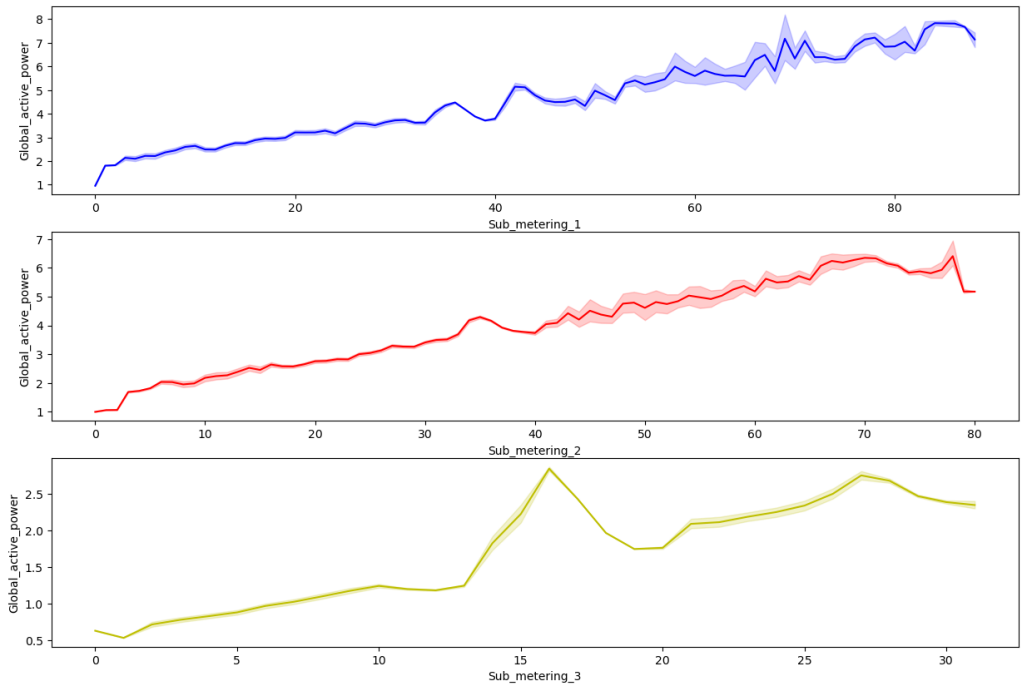
In [11]:

```
# Line chart to investigate the 'Global_active_power' consumption by
Submeterings:

fig,ax = plt.subplots(3, 1,
                      figsize=(15,10))
```

```
sns.lineplot(x=df['Sub_metering_1'], y=df['Global_active_power'], ax =
ax[0], color ='b')


sns.lineplot(x=df['Sub_metering_2'], y=df['Global_active_power'], ax =
ax[1], color ='r')


sns.lineplot(x=df['Sub_metering_3'], y=df['Global_active_power'], ax =
ax[2], color ='y')

plt.show
```

Out[11]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



6.4- Investigating the consumption by year, month and weekday.

In [12]:

```
# Converting 'Date' and 'Time' to 'date_time' column to datetime format.

df['date_time'] = pd.to_datetime(df['Date'] + ' ' + df['Time'])
```

In [13]:

```
# Creating columns for year, month and weekday.

df['Year'] = df['date_time'].apply(lambda x: x.year)
df['Month'] = df['date_time'].apply(lambda x: x.month)

df['Weekday'] = df['date_time'].dt.day_name()
```

In [14]:

```
df.sort_values(by='date_time', inplace=True)
df
```

Out[14]:

	Date	Time	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3	date_time	Year	Month	Weekday
0	16/12/2006	17:24:00	4.216	0.418	234.84	18.4	0.0	1.0	17.0	2006-12-16 17:24:00	2006	12	Saturday
1	16/12/2006	17:25:00	5.360	0.436	233.63	23.0	0.0	1.0	16.0	2006-12-16 17:25:00	2006	12	Saturday
2	16/12/2006	17:26:00	5.374	0.498	233.29	23.0	0.0	2.0	17.0	2006-12-16 17:26:00	2006	12	Saturday
3	16/12/2006	17:27:00	5.388	0.502	233.74	23.0	0.0	1.0	17.0	2006-12-16 17:27:00	2006	12	Saturday
4	16/12/2006	17:28:00	3.666	0.528	235.68	15.8	0.0	1.0	17.0	2006-12-16 17:28:00	2006	12	Saturday
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2055271	12/11/2010	23:55:00	0.690	0.062	244.16	2.8	0.0	0.0	0.0	2010-12-11 23:55:00	2010	12	Saturday
2055272	12/11/2010	23:56:00	0.688	0.060	243.82	2.8	0.0	0.0	0.0	2010-12-11 23:56:00	2010	12	Saturday
2055273	12/11/2010	23:57:00	0.688	0.062	244.20	2.8	0.0	0.0	0.0	2010-12-11 23:57:00	2010	12	Saturday
2055274	12/11/2010	23:58:00	0.688	0.062	244.21	2.8	0.0	0.0	0.0	2010-12-11 23:58:00	2010	12	Saturday
2055275	12/11/2010	23:59:00	0.688	0.064	244.65	2.8	0.0	0.0	0.0	2010-12-11 23:59:00	2010	12	Saturday

2049280 rows × 13 columns

In [15]:

```
# Dropping the column 'Global_reactive_power'.

data = df.loc[:,['date_time', 'Year', 'Month', 'Weekday',
'Global_active_power','Voltage','Global_intensity','Sub_metering_1',
'Sub_metering_2', 'Sub_metering_3']]

data.head()
```

Out[15]:

	date_time	Year	Month	Weekday	Global_active_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
0	2006-12-16 17:24:00	2006	12	Saturday	4.216	234.84	18.4	0.0	1.0	17.0
1	2006-12-16 17:25:00	2006	12	Saturday	5.360	233.63	23.0	0.0	1.0	16.0
2	2006-12-16 17:26:00	2006	12	Saturday	5.374	233.29	23.0	0.0	2.0	17.0
3	2006-12-16 17:27:00	2006	12	Saturday	5.388	233.74	23.0	0.0	1.0	17.0
4	2006-12-16 17:28:00	2006	12	Saturday	3.666	235.68	15.8	0.0	1.0	17.0

In [16]:

```
print(data.dtypes)
date_time              datetime64[ns]
Year                            int64
Month                           int64
Weekday                        object
Global_active_power           float64
Voltage                       float64
Global_intensity              float64
Sub_metering_1                float64
Sub_metering_2                float64
Sub_metering_3                float64
dtype: object
```

In [17]:

```
# Summary statistics:

data.describe().transpose()
```

Out[17]:

	count	mean	std	min	25%	50%	75%	max
Year	2049280.0	2008.424761	1.124388	2006.000	2007.000	2008.000	2009.000	2010.000
Month	2049280.0	6.497968	3.446016	1.000	4.000	7.000	10.000	12.000
Global_active_power	2049280.0	1.091615	1.057294	0.076	0.308	0.602	1.528	11.122
Voltage	2049280.0	240.839858	3.239987	223.200	238.990	241.010	242.890	254.150
Global_intensity	2049280.0	4.627759	4.444396	0.200	1.400	2.600	6.400	48.400

	count	mean	std	min	25%	50%	75%	max
Sub_metering_1	2049280.0	1.121923	6.153031	0.000	0.000	0.000	0.000	88.000
Sub_metering_2	2049280.0	1.298520	5.822026	0.000	0.000	0.000	1.000	80.000
Sub_metering_3	2049280.0	6.458447	8.437154	0.000	0.000	1.000	17.000	31.000

In [18]:

```
data.sort_values(by='date_time', inplace=True)
data
```

Out[18]:

	date_time	Year	Month	Weekday	Global_active_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
0	2006-12-16 17:24:00	2006	12	Saturday	4.216	234.84	18.4	0.0	1.0	17.0
1	2006-12-16 17:25:00	2006	12	Saturday	5.360	233.63	23.0	0.0	1.0	16.0
2	2006-12-16 17:26:00	2006	12	Saturday	5.374	233.29	23.0	0.0	2.0	17.0
3	2006-12-16 17:27:00	2006	12	Saturday	5.388	233.74	23.0	0.0	1.0	17.0
4	2006-12-16 17:28:00	2006	12	Saturday	3.666	235.68	15.8	0.0	1.0	17.0
...	...	...	...	...	...	...	...	...	...	...
2055271	2010-12-11 23:55:00	2010	12	Saturday	0.690	244.16	2.8	0.0	0.0	0.0
2055272	2010-12-11 23:56:00	2010	12	Saturday	0.688	243.82	2.8	0.0	0.0	0.0
2055273	2010-12-11 23:57:00	2010	12	Saturday	0.688	244.20	2.8	0.0	0.0	0.0
2055274	2010-12-11 23:58:00	2010	12	Saturday	0.688	244.21	2.8	0.0	0.0	0.0
2055275	2010-12-11 23:59:00	2010	12	Saturday	0.688	244.65	2.8	0.0	0.0	0.0

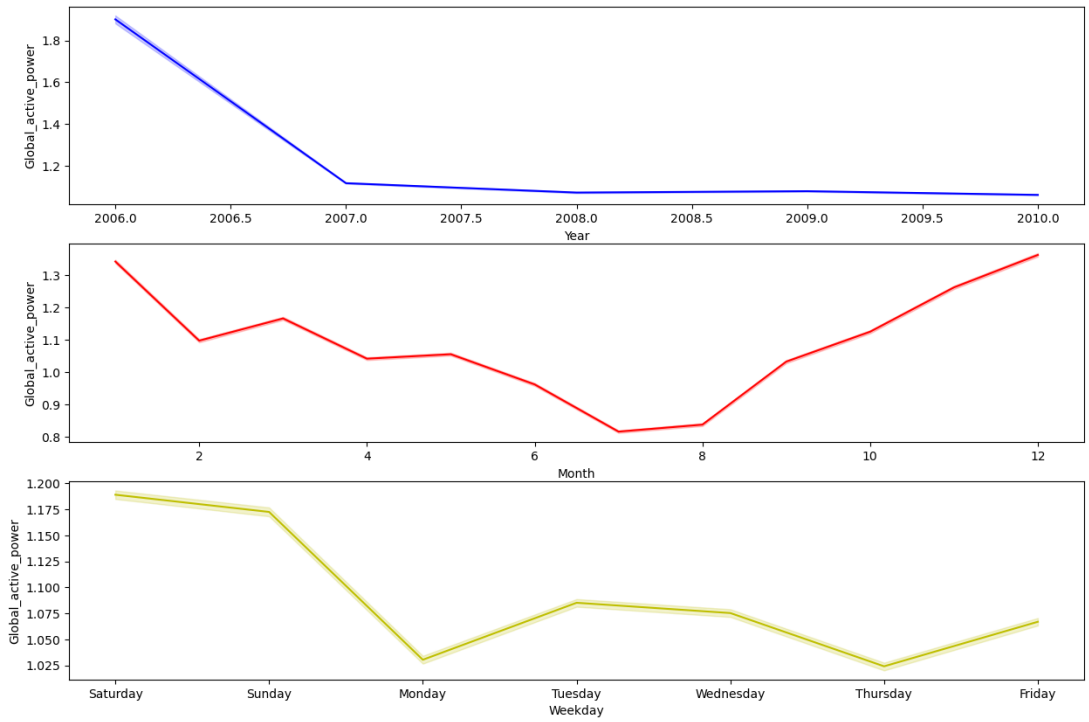
2049280 rows × 10 columns

In [19]:

```
# Visualizaton of Global active power across Yearly, Monthly and Weekly:
fig,ax = plt.subplots(3, 1,
                      figsize=(15,10))

sns.lineplot(x=data['Year'], y=data['Global_active_power'], ax = ax[0],
color ='b')


sns.lineplot(x=data['Month'], y=data['Global_active_power'], ax = ax[1],
color ='r')


sns.lineplot(x=data['Weekday'], y=data['Global_active_power'], ax =
ax[2], color ='y')

plt.show
```

Out[19]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```

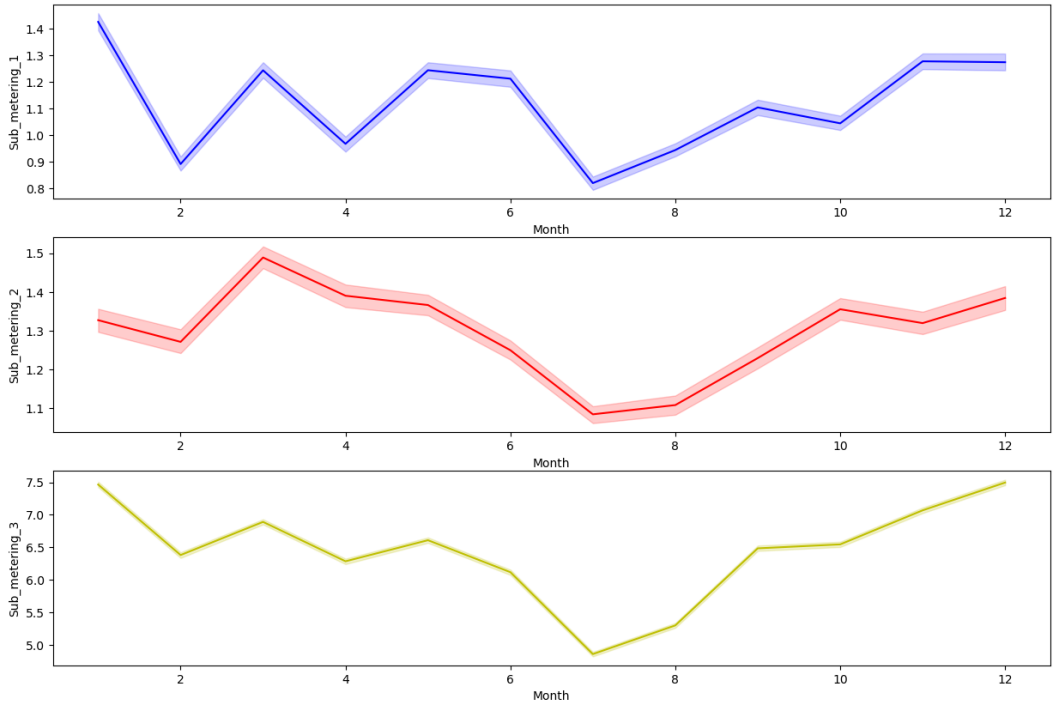


In [20]:

```
# Visualizaton of Sub power across Yearly, Monthly and Weekly:

fig,ax = plt.subplots(3, 1,
                      figsize=(15,10))

sns.lineplot(x=data['Month'], y=data['Sub_metering_1'], ax = ax[0],
color ='b')


sns.lineplot(x=data['Month'], y=data['Sub_metering_2'], ax = ax[1],
color ='r')


sns.lineplot(x=data['Month'], y=data['Sub_metering_3'], ax = ax[2],
color ='y')

plt.show
```

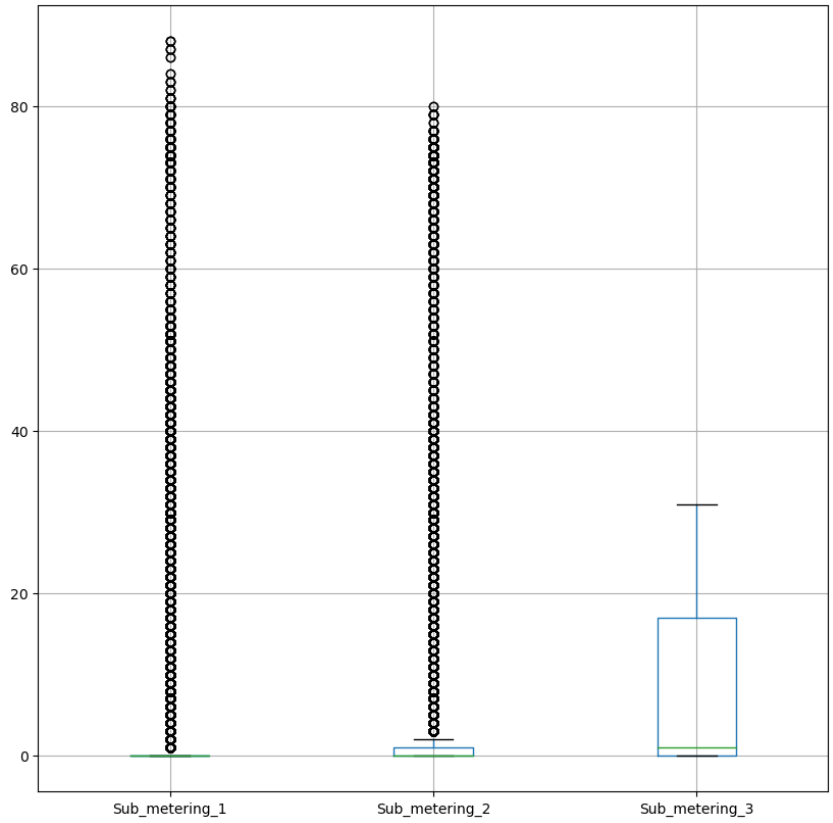
Out[20]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [21]:

```
# 'Sub_metering_1', 'Sub_metering_2' and 'Sub_metering_3' box plot of
data distribution.

ax = df.boxplot(['Sub_metering_1', 'Sub_metering_2', 'Sub_metering_3'],
figsize=(10,  10))
```



In [22]:

```
sub_1_max = df['Sub_metering_1'].max()
sub_1_min = df['Sub_metering_1'].min()
```

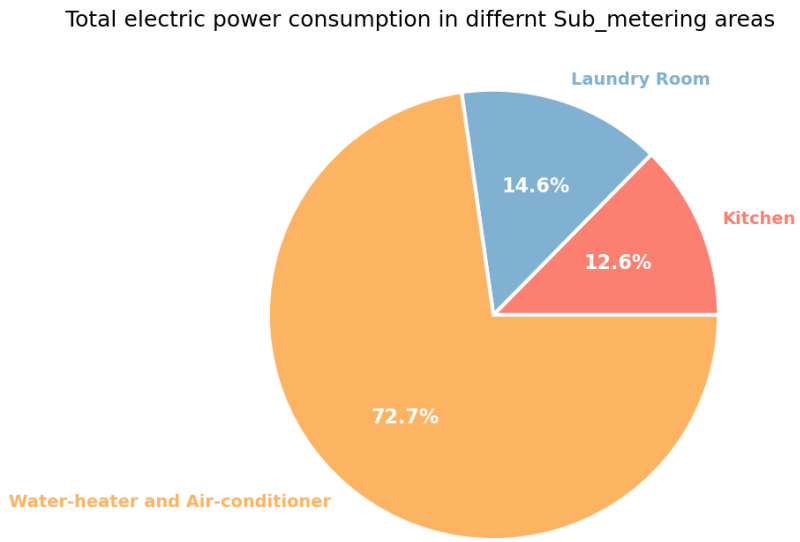
```
sub_1_mean = df['Sub_metering_1'].mean()
sub_1_median = df['Sub_metering_1'].median()

sub_2_max = df['Sub_metering_2'].max()
sub_2_min = df['Sub_metering_2'].min()
sub_2_mean = df['Sub_metering_2'].mean()
sub_2_median = df['Sub_metering_2'].median()

sub_3_max = df['Sub_metering_3'].max()
sub_3_min = df['Sub_metering_3'].min()
sub_3_mean = df['Sub_metering_3'].mean()
sub_3_median = df['Sub_metering_3'].median()

table = {'Type': ['Sub_metering_1', 'Sub_metering_2', 'Sub_metering_3'],
        'Max value': [sub_1_max, sub_2_max, sub_3_max],
        'Min value': [sub_1_min, sub_2_min, sub_3_min],
        'Mean value': [sub_1_mean, sub_2_mean, sub_3_mean],
        'Median value': [sub_1_median, sub_2_median, sub_3_median]}

_table = pd.DataFrame(table)

_table
```

Out[22]:

	Type	Max value	Min value	Mean value	Median value
0	Sub_metering_1	88.0	0.0	1.121923	0.0
1	Sub_metering_2	80.0	0.0	1.298520	0.0
2	Sub_metering_3	31.0	0.0	6.458447	1.0

In [23]:

```
# Investigating the power consumption pattern by sub-metering areas:

cls = ['Sub_metering_1', 'Sub_metering_2', 'Sub_metering_3']
pie_df = df[cls].sum(axis=0)
labels = ['Kitchen', 'Laundry Room', 'Water-heater and Air-conditioner']

plt.figure(figsize=(9,9))
colors = sns.color_palette('Set3')[3:]
patches, texts, pcts = plt.pie(
                            pie_df, labels=labels,
                            colors=colors,
                            autopct='%.1f%%',
                            wedgeprops={'linewidth': 3.0, 'edgecolor':
'white'},
                            startangle=0
)

for i, patch in enumerate(patches):
  texts[i].set_color(patch.get_facecolor())
plt.setp(pcts, color='white', fontsize=16, fontweight='bold')
plt.setp(texts, fontsize=14, fontweight='bold')
plt.title('Total electric power consumption in differnt Sub_metering
areas', fontsize=18, loc='right')
plt.tight_layout()
```



6.5- Multicollinearity investigation

In [24]:

```
# Scaling

df_drop=df.drop(["Date", "Time", "date_time", "Year", "Month",
"Weekday"], axis = 1)
df_drop.head()
```

Out[24]:

	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
0	4.216	0.418	234.84	18.4	0.0	1.0	17.0
1	5.360	0.436	233.63	23.0	0.0	1.0	16.0
2	5.374	0.498	233.29	23.0	0.0	2.0	17.0
3	5.388	0.502	233.74	23.0	0.0	1.0	17.0
4	3.666	0.528	235.68	15.8	0.0	1.0	17.0

In [25]:

```
columns = df_drop.columns

scaler = MinMaxScaler()
normalised_dataset = scaler.fit_transform(df_drop)
normalised_dataset

normalised_df = pd.DataFrame(data = normalised_dataset, columns =
columns)
normalised_df
```

Out[25]:

	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
0	0.374796	0.300719	0.376090	0.377593	0.0	0.0125	0.548387
1	0.478363	0.313669	0.336995	0.473029	0.0	0.0125	0.516129
2	0.479631	0.358273	0.326010	0.473029	0.0	0.0250	0.548387
3	0.480898	0.361151	0.340549	0.473029	0.0	0.0125	0.548387
4	0.325005	0.379856	0.403231	0.323651	0.0	0.0125	0.548387
...	...	...	...	...	...	...	...
2049275	0.055586	0.044604	0.677221	0.053942	0.0	0.0000	0.000000
2049276	0.055405	0.043165	0.666236	0.053942	0.0	0.0000	0.000000
2049277	0.055405	0.044604	0.678514	0.053942	0.0	0.0000	0.000000
2049278	0.055405	0.044604	0.678837	0.053942	0.0	0.0000	0.000000
2049279	0.055405	0.046043	0.693053	0.053942	0.0	0.0000	0.000000

2049280 rows × 7 columns

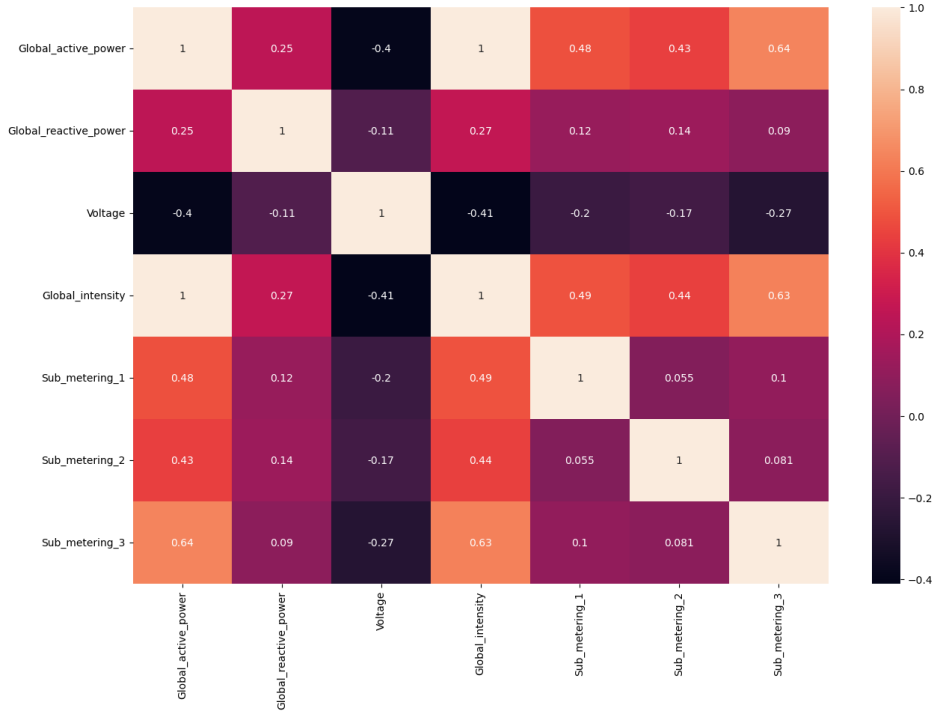
6.6- Heatmap matrix and scatter plot

In [26]:

```
plt.figure(figsize=(15,10))
sns.heatmap(normalised_df.corr(method='pearson', min_periods=1),
annot=True)
```

Out[26]:

```
<Axes: >
```

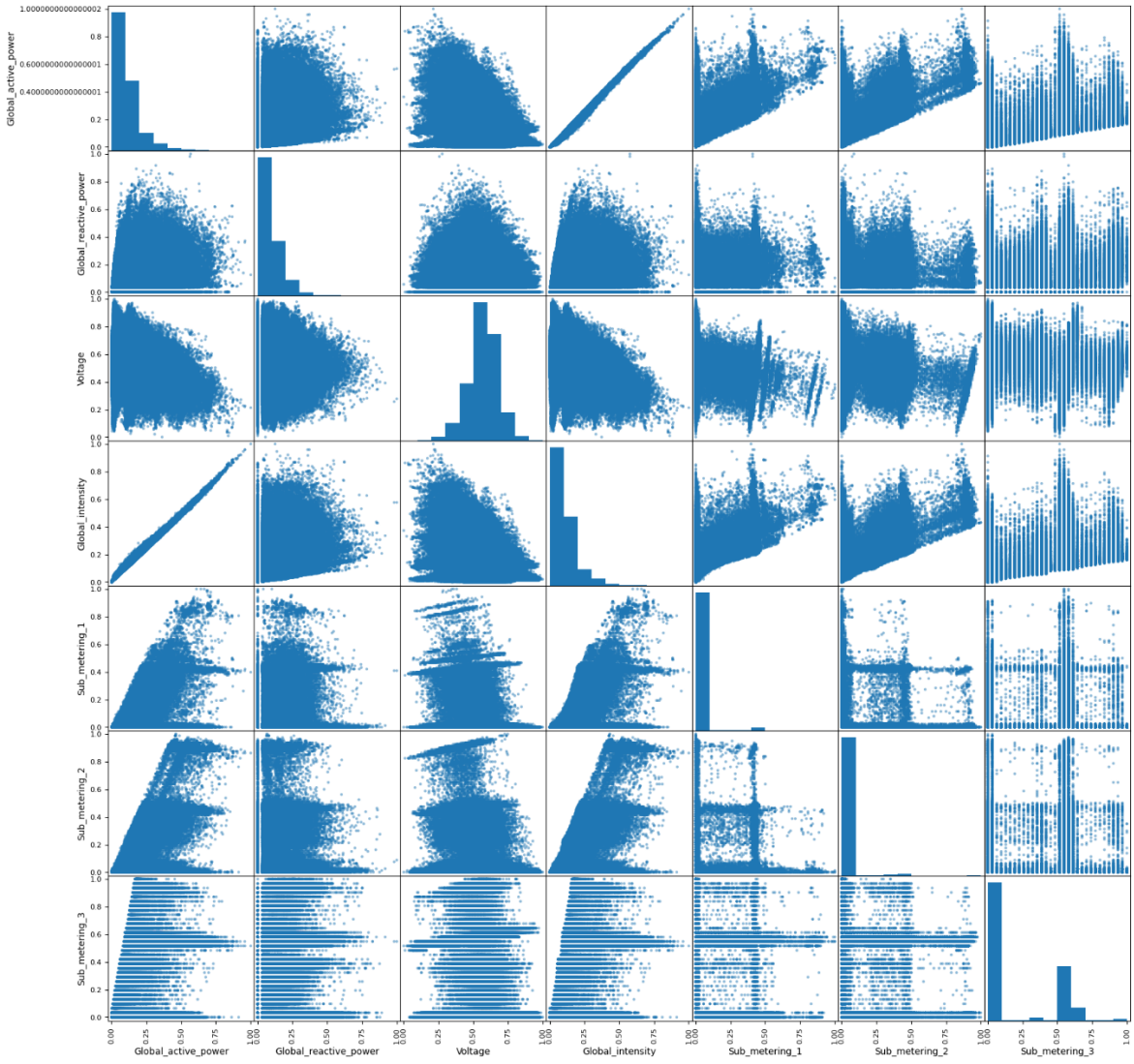


In [27]:

```
normalised_df.corr()
pd.plotting.scatter_matrix(normalised_df, figsize=[20,20])
```

Out[27]:

```
array([[<Axes: xlabel='Global_active_power',
ylabel='Global_active_power'>,
        <Axes: xlabel='Global_reactive_power',
ylabel='Global_active_power'>,
        <Axes: xlabel='Voltage', ylabel='Global_active_power'>,
        <Axes: xlabel='Global_intensity', ylabel='Global_active_power'>,
        <Axes: xlabel='Sub_metering_1', ylabel='Global_active_power'>,
        <Axes: xlabel='Sub_metering_2', ylabel='Global_active_power'>,
        <Axes: xlabel='Sub_metering_3', ylabel='Global_active_power'>],
       [<Axes: xlabel='Global_active_power',
ylabel='Global_reactive_power'>,
        <Axes: xlabel='Global_reactive_power',
ylabel='Global_reactive_power'>,
        <Axes: xlabel='Voltage', ylabel='Global_reactive_power'>,
        <Axes: xlabel='Global_intensity',
ylabel='Global_reactive_power'>,
        <Axes: xlabel='Sub_metering_1', ylabel='Global_reactive_power'>,
        <Axes: xlabel='Sub_metering_2', ylabel='Global_reactive_power'>,
        <Axes: xlabel='Sub_metering_3',
ylabel='Global_reactive_power'>],
       [<Axes: xlabel='Global_active_power', ylabel='Voltage'>,
        <Axes: xlabel='Global_reactive_power', ylabel='Voltage'>,
        <Axes: xlabel='Voltage', ylabel='Voltage'>,
        <Axes: xlabel='Global_intensity', ylabel='Voltage'>,
        <Axes: xlabel='Sub_metering_1', ylabel='Voltage'>,
        <Axes: xlabel='Sub_metering_2', ylabel='Voltage'>,
        <Axes: xlabel='Sub_metering_3', ylabel='Voltage'>],
       [<Axes: xlabel='Global_active_power', ylabel='Global_intensity'>,
        <Axes: xlabel='Global_reactive_power',
ylabel='Global_intensity'>,
        <Axes: xlabel='Voltage', ylabel='Global_intensity'>,
        <Axes: xlabel='Global_intensity', ylabel='Global_intensity'>,
        <Axes: xlabel='Sub_metering_1', ylabel='Global_intensity'>,
        <Axes: xlabel='Sub_metering_2', ylabel='Global_intensity'>,
        <Axes: xlabel='Sub_metering_3', ylabel='Global_intensity'>],
       [<Axes: xlabel='Global_active_power', ylabel='Sub_metering_1'>,
        <Axes: xlabel='Global_reactive_power', ylabel='Sub_metering_1'>,
        <Axes: xlabel='Voltage', ylabel='Sub_metering_1'>,
        <Axes: xlabel='Global_intensity', ylabel='Sub_metering_1'>,
        <Axes: xlabel='Sub_metering_1', ylabel='Sub_metering_1'>,
        <Axes: xlabel='Sub_metering_2', ylabel='Sub_metering_1'>,
        <Axes: xlabel='Sub_metering_3', ylabel='Sub_metering_1'>],
       [<Axes: xlabel='Global_active_power', ylabel='Sub_metering_2'>,
        <Axes: xlabel='Global_reactive_power', ylabel='Sub_metering_2'>,
        <Axes: xlabel='Voltage', ylabel='Sub_metering_2'>,
        <Axes: xlabel='Global_intensity', ylabel='Sub_metering_2'>,
        <Axes: xlabel='Sub_metering_1', ylabel='Sub_metering_2'>,
        <Axes: xlabel='Sub_metering_2', ylabel='Sub_metering_2'>,
        <Axes: xlabel='Sub_metering_3', ylabel='Sub_metering_2'>],
       [<Axes: xlabel='Global_active_power', ylabel='Sub_metering_3'>,
        <Axes: xlabel='Global_reactive_power', ylabel='Sub_metering_3'>,
        <Axes: xlabel='Voltage', ylabel='Sub_metering_3'>,
        <Axes: xlabel='Global_intensity', ylabel='Sub_metering_3'>,
        <Axes: xlabel='Sub_metering_1', ylabel='Sub_metering_3'>,
        <Axes: xlabel='Sub_metering_2', ylabel='Sub_metering_3'>,
        <Axes: xlabel='Sub_metering_3', ylabel='Sub_metering_3'>]],
      dtype=object)
```



6.7-Linear Regression investigation

In [28]:

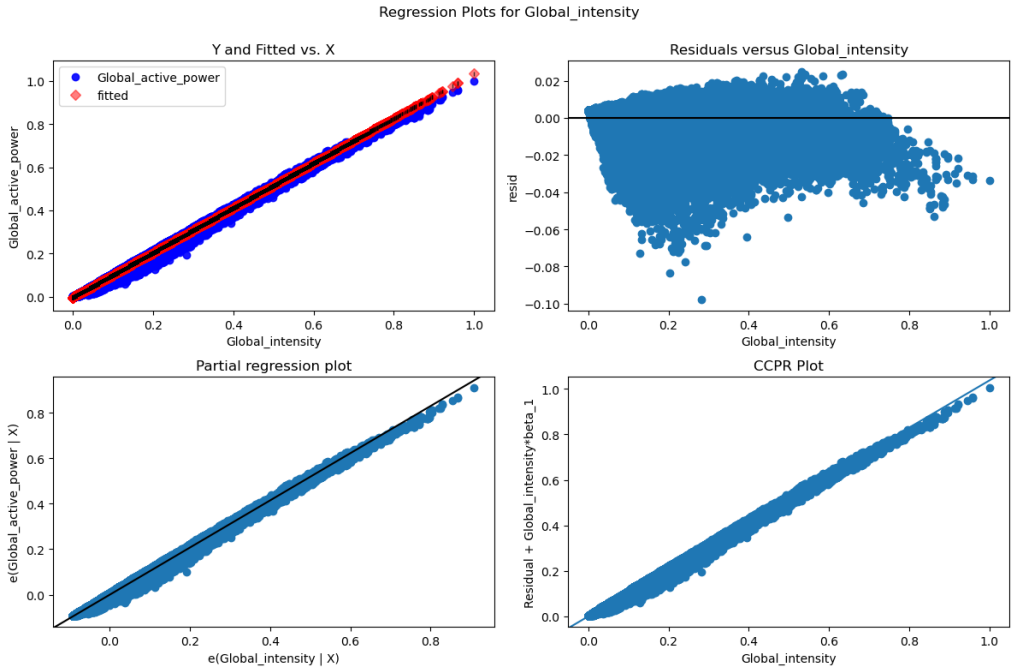
```
# Simple Linear Regression Model. Investigating by Global active power
as dependent variable.

reg = linear_model.LinearRegression()

import statsmodels.api as sm
from statsmodels.formula.api import ols

model = ols('Global_active_power ~ Global_intensity',
data=normalised_df).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
=======
Dep. Variable:     Global_active_power   R-squared:                       0.998
Model:                             OLS   Adj. R-squared:                  0.998
Method:                  Least Squares   F-statistic:                 9.204e+08
Date:                 Fri, 14 Apr 2023   Prob (F-statistic):               0.00
Time:                         22:03:39   Log-Likelihood:            8.1606e+06
No. Observations:              2049280   AIC:                        -1.632e+07
Df Residuals:                  2049278   BIC:                        -1.632e+07
Df Model:                            1
Covariance Type:             nonrobust
```

```
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0033   4.45e-06   -743.820      0.000      -0.003      -0.003
Global_intensity     1.0369   3.42e-05   3.03e+04      0.000       1.037       1.037
==============================================================================
Omnibus:                  1117754.065   Durbin-Watson:                   0.705
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         16865127.739
Skew:                          -2.301   Prob(JB):                         0.00
Kurtosis:                      16.279   Cond. No.                         10.9
==============================================================================
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
```

In [29]:

```
fig = plt.figure(figsize=(12,8))
fig = sm.graphics.plot_regress_exog(model, 'Global_intensity', fig=fig)
eval_env: 1
```

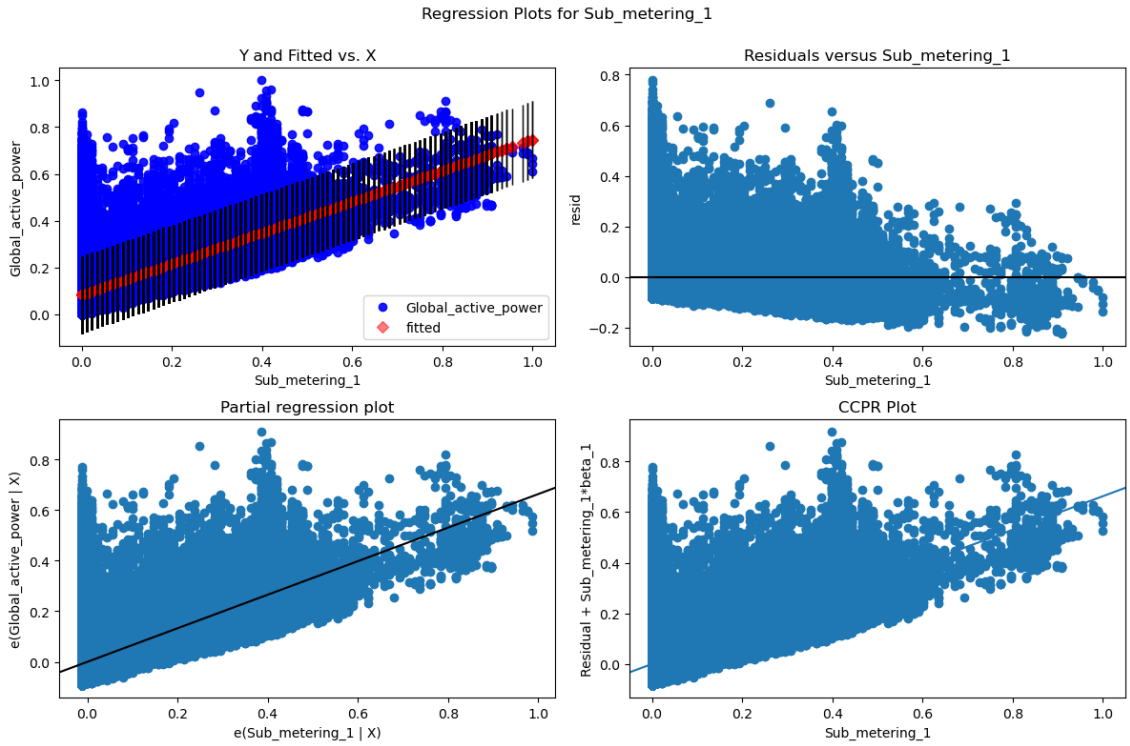


In [30]:

```
# Sub_metering_1

model = ols('Global_active_power ~ Sub_metering_1',
data=normalised_df).fit()
print(model.summary())
fig = plt.figure(figsize=(12,8))
fig = sm.graphics.plot_regress_exog(model, 'Sub_metering_1', fig=fig)
```

```
                            OLS Regression Results
```

```
==============================================================================
=======
Dep. Variable:     Global_active_power   R-squared:
0.235
Model:                             OLS   Adj. R-squared:           0.235
Method:                  Least Squares   F-statistic:           6.283e+05
Date:                 Fri, 14 Apr 2023   Prob (F-statistic):         0.00
Time:                         22:06:03   Log-Likelihood:       2.1745e+06
No. Observations:              2049280   AIC:                  -4.349e+06
Df Residuals:                  2049278   BIC:                  -4.349e+06
Df Model:                            1
Covariance Type:             nonrobust
==============================================================================
==========
                     coef    std err          t      P>|t|       [0.025
0.975]
------------------------------------------------------------------------------
----------
Intercept          0.0835   5.95e-05   1404.141      0.000        0.083
0.084
Sub_metering_1     0.6631      0.001    792.637      0.000        0.661
0.665
==============================================================================
======
Omnibus:                   642064.856   Durbin-Watson:                   0.063
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          2051263.705
Skew:                           1.614   Prob(JB):                         0.00
Kurtosis:                       6.688   Cond. No.                         14.3
==============================================================================
======
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
eval_env: 1
```



In [31]:

```
# Sub_metering_2

model = ols('Global_active_power ~ Sub_metering_2',
data=normalised_df).fit()
print(model.summary())
fig = plt.figure(figsize=(12,8))
fig = sm.graphics.plot_regress_exog(model, 'Sub_metering_2', fig=fig)
```

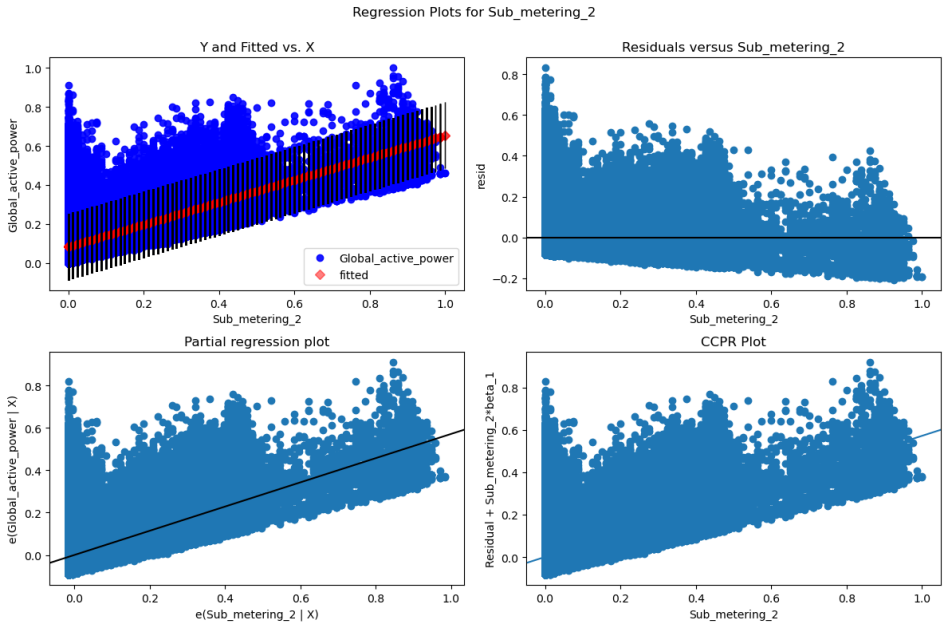
OLS Regression Results

Dep. Variable:	Global_active_power	R-squared:	0.189
Model:	OLS	Adj. R-squared:	0.189
Method:	Least Squares	F-statistic:	4.771e+05
Date:	Fri, 14 Apr 2023	Prob (F-statistic):	0.00
Time:	22:09:01	Log-Likelihood:	2.1150e+06
No. Observations:	2049280	AIC:	-4.230e+06
Df Residuals:	2049278	BIC:	-4.230e+06
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0827	6.17e-05	1339.826	0.000	0.083	0.083
Sub_metering_2	0.5716	0.001	690.730	0.000	0.570	0.573

Omnibus:	631923.799	Durbin-Watson:	0.066
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1881811.007
Skew:	1.620	Prob(JB):	0.00
Kurtosis:	6.397	Cond. No.	13.7

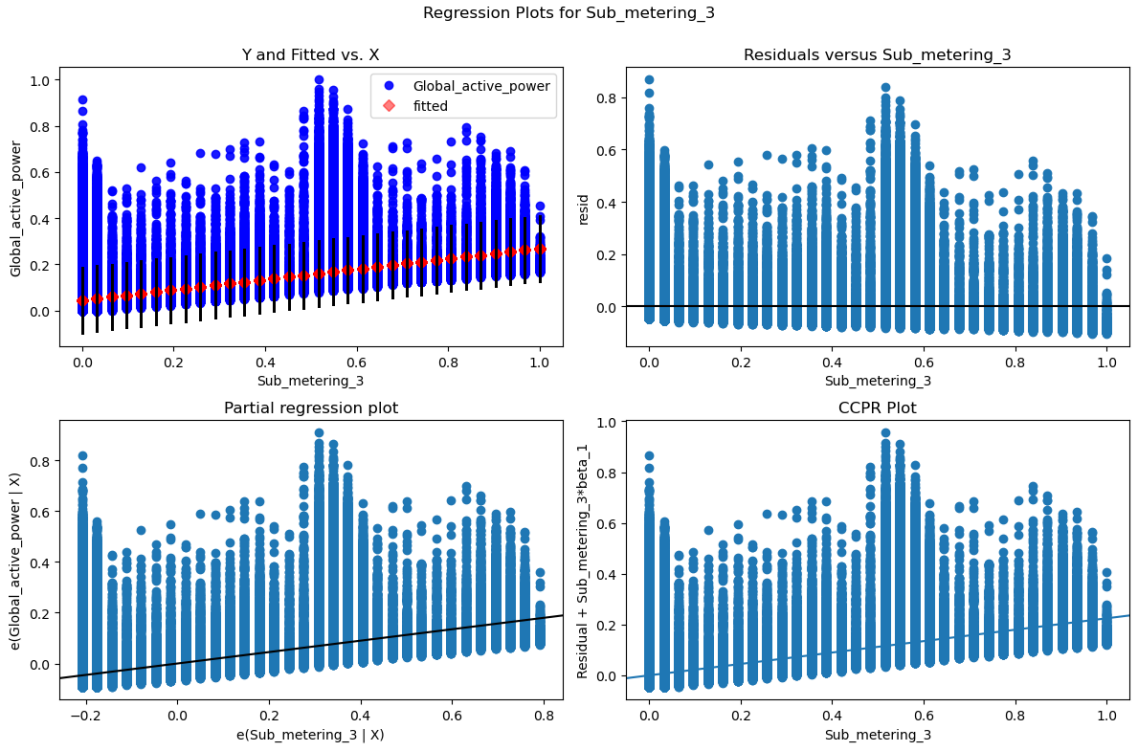
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
eval_env: 1



In [32]:

```
# Sub_metering_3:

model = ols('Global_active_power ~ Sub_metering_3',
data=normalised_df).fit()
print(model.summary())
fig = plt.figure(figsize=(12,8))
fig = sm.graphics.plot_regress_exog(model, 'Sub_metering_3', fig=fig)
```

```
                            OLS Regression Results
==============================================================================
=======
Dep. Variable:     Global_active_power   R-squared:                       0.408
Model:                             OLS   Adj. R-squared:                  0.408
Method:                  Least Squares   F-statistic:                 1.411e+06
Date:                 Fri, 14 Apr 2023   Prob (F-statistic):               0.00
Time:                         22:12:05   Log-Likelihood:             2.4373e+06
No. Observations:              2049280   AIC:                        -4.875e+06
Df Residuals:                  2049278   BIC:                        -4.875e+06
Df Model:                            1
Covariance Type:             nonrobust
==============================================================================
==========
                     coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
----------
Intercept          0.0452   6.48e-05    696.856      0.000       0.045
0.045
Sub_metering_3     0.2246      0.000   1187.812      0.000       0.224
0.225
==============================================================================
======
Omnibus:                  1155962.933   Durbin-Watson:                   0.104
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          9396574.201
Skew:                           2.665   Prob(JB):                         0.00
Kurtosis:                      12.036   Cond. No.                         3.85
==============================================================================
======
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
eval_env: 1
```

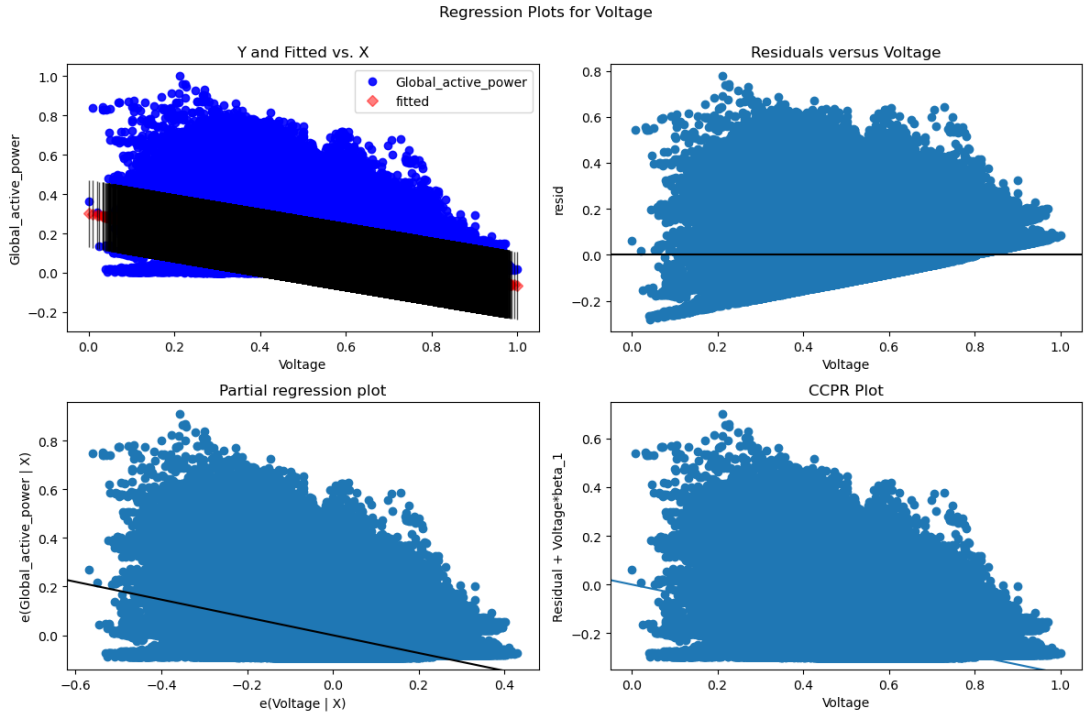


In [33]:

```
# Voltage:

model = ols('Global_active_power ~ Voltage', data=normalised_df).fit()
print(model.summary())
fig = plt.figure(figsize=(12,8))
fig = sm.graphics.plot_regress_exog(model, 'Voltage', fig=fig)
```

```
                            OLS Regression Results
==============================================================================
=======
Dep. Variable:     Global_active_power   R-squared:                       0.160
Model:                             OLS   Adj. R-squared:                  0.160
Method:                  Least Squares   F-statistic:                 3.898e+05
Date:                 Fri, 14 Apr 2023   Prob (F-statistic):               0.00
Time:                         22:14:57   Log-Likelihood:             2.0790e+06
No. Observations:              2049280   AIC:                        -4.158e+06
Df Residuals:                  2049278   BIC:                        -4.158e+06
Df Model:                            1
Covariance Type:             nonrobust
==============================================================================
======
                 coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
------
Intercept      0.3003      0.000    885.067      0.000       0.300
0.301
Voltage       -0.3655      0.001   -624.328      0.000      -0.367
-0.364
==============================================================================
======
Omnibus:                   524936.882   Durbin-Watson:                   0.062
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           466890.854
Skew:                           1.360   Prob(JB):                         0.00
```

```
Kurtosis:                       6.128   Cond. No.                         12.7
==============================================================================
======
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
eval_env: 1
```

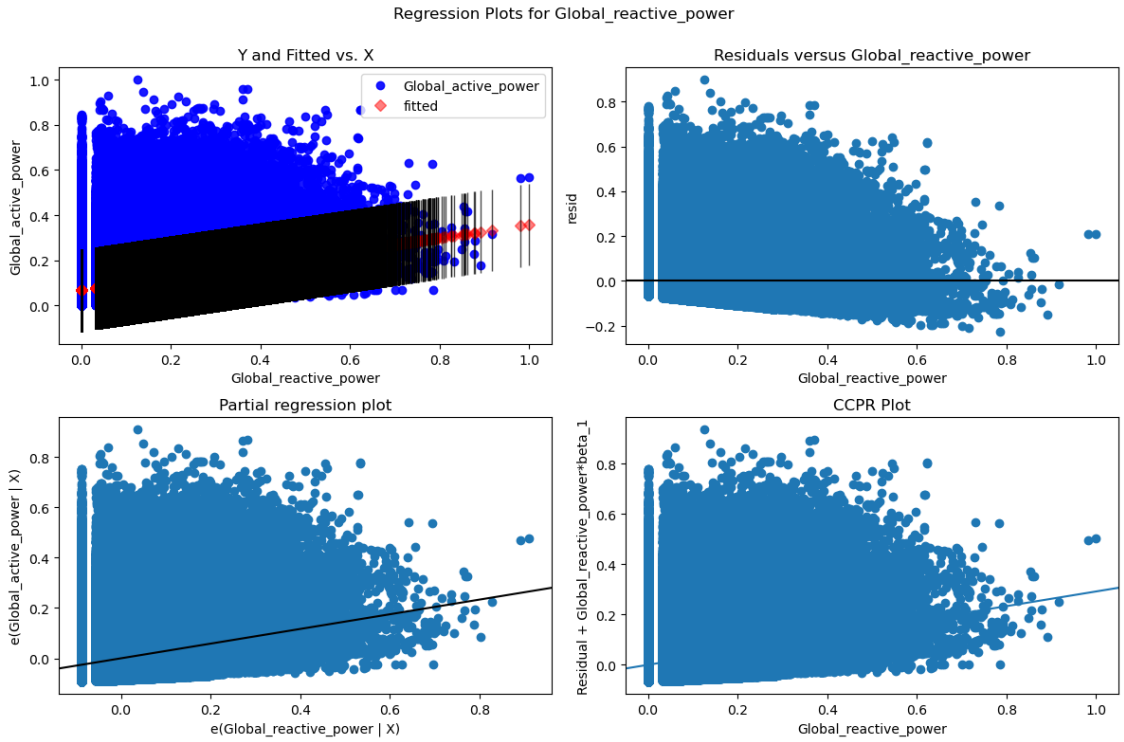


In [34]:

```
# Global_reactive_power:

model = ols('Global_active_power ~ Global_reactive_power',
data=normalised_df).fit()
print(model.summary())
fig = plt.figure(figsize=(12,8))
fig = sm.graphics.plot_regress_exog(model, 'Global_reactive_power',
fig=fig)
```

```
                            OLS Regression Results
==============================================================================
=======
Dep. Variable:     Global_active_power   R-squared:                       0.061
Model:                             OLS   Adj. R-squared:                  0.061
Method:                  Least Squares   F-statistic:                 1.332e+05
Date:                 Fri, 14 Apr 2023   Prob (F-statistic):               0.00
Time:                         22:17:46   Log-Likelihood:            1.9650e+06
No. Observations:              2049280   AIC:                        -3.930e+06
Df Residuals:                  2049278   BIC:                        -3.930e+06
Df Model:                            1
Covariance Type:             nonrobust
==============================================================================
=================
                           coef    std err          t      P>|t|
[0.025      0.975]
------------------------------------------------------------------------------
-----------------
```

```
Intercept                   0.0660    9.62e-05     686.009       0.000
0.066       0.066
Global_reactive_power       0.2916       0.001     364.921       0.000
0.290       0.293
==============================================================================
======
Omnibus:                   692019.691   Durbin-Watson:                   0.067
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           382273.439
Skew:                           1.716   Prob(JB):                         0.00
Kurtosis:                       7.015   Cond. No.                         12.4
==============================================================================
======
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
eval_env: 1
```



In [35]:

```
# Table for R-squared, coefficient and p-value:

Table = {'Variable': ['Global_intensity', 'Sub_metering_1',
'Sub_metering_2', 'Sub_metering_3'],
        'R-squared': [0.998, 0.235, 0.189, 0.408],
         'Coefficient': [1.0369, 0.0832, 0.0789, 0.0800],
         'p-Value': [0.000, 0.000, 0.000, 0.000]}
table = pd.DataFrame(Table)
print('Table 3: R-sqaured, Coefficient and P-value obtained for simple
linear regression, "Global_Intensity", "Sub_metering_1",
"Sub_metering_2" and "Sub_metering_3":')
table
```

```
Table 3: R-sqaured, Coefficient and P-value obtained for simple linear
regression, "Global_Intensity", "Sub_metering_1", "Sub_metering_2" and
"Sub_metering_3":
```

Out[35]:

	Variable	R-squared	Coefficient	p-Value
0	Global_intensity	0.998	1.0369	0.0

	Variable	R-squared	Coefficient	p-Value
1	Sub_metering_1	0.235	0.0832	0.0
2	Sub_metering_2	0.189	0.0789	0.0
3	Sub_metering_3	0.408	0.0800	0.0

6.8-Applying multiple independent variables in a model

In [36]:

```
# Model 1, included the independent variables Submetering categories:

Model_1 = ols('Global_active_power ~ Sub_metering_1 + Sub_metering_2 +
Sub_metering_3',
            data=normalised_df).fit()
print(Model_1.summary())
                            OLS Regression Results
==============================================================================
=======
Dep. Variable:     Global_active_power   R-squared:                       0.718
Model:                             OLS   Adj. R-squared:                  0.718
Method:                  Least Squares   F-statistic:                 1.741e+06
Date:                 Fri, 14 Apr 2023   Prob (F-statistic):               0.00
Time:                         22:20:45   Log-Likelihood:             3.1984e+06
No. Observations:              2049280   AIC:                        -6.397e+06
Df Residuals:                  2049276   BIC:                        -6.397e+06
Df Model:                            3
Covariance Type:             nonrobust
==============================================================================
==========
                     coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
----------
Intercept          0.0355   4.52e-05    784.796      0.000       0.035
0.036
Sub_metering_1     0.5560      0.001   1088.374      0.000       0.555
0.557
Sub_metering_2     0.4820      0.000    983.977      0.000       0.481
0.483
Sub_metering_3     0.1995      0.000   1517.344      0.000       0.199
0.200
==============================================================================
======
Omnibus:                  1082590.806   Durbin-Watson:                   0.106
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          7990568.879
Skew:                           2.485   Prob(JB):                        0.00
Kurtosis:                      11.300   Cond. No.                        14.9
==============================================================================
======
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
```

In [37]:

```
# Model 2, included the remain independent variables, Global intensity
and voltage:
```

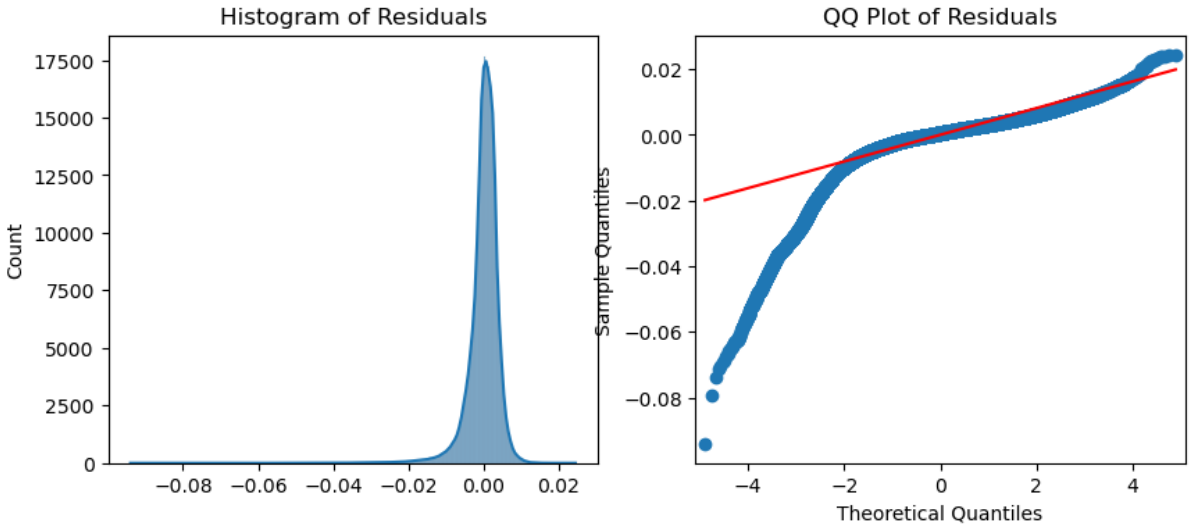
```
Model_2 = ols('Global_active_power ~ Sub_metering_1 + Sub_metering_2 +
Sub_metering_3 + Global_intensity + Voltage',
            data=normalised_df).fit()
print(Model_2.summary())
                            OLS Regression Results
==============================================================================
=======
Dep. Variable:     Global_active_power   R-squared:                       0.998
Model:                             OLS   Adj. R-squared:                  0.998
Method:                  Least Squares   F-statistic:                 2.276e+08
Date:                 Fri, 14 Apr 2023   Prob (F-statistic):               0.00
Time:                         22:20:47   Log-Likelihood:            8.3777e+06
No. Observations:              2049280   AIC:                        -1.676e+07
Df Residuals:                  2049274   BIC:                       -1.676e+07
Df Model:                            5
Covariance Type:             nonrobust
==============================================================================
============
                       coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
------------
Intercept           -0.0113   1.87e-05   -605.402      0.000      -0.011
-0.011
Sub_metering_1      -0.0011   5.14e-05    -22.272      0.000      -0.001
-0.001
Sub_metering_2      -0.0027   4.76e-05    -57.205      0.000      -0.003
-0.003
Sub_metering_3       0.0071    1.5e-05    472.483      0.000       0.007
0.007
Global_intensity     1.0310   5.91e-05   1.74e+04      0.000       1.031
1.031
Voltage              0.0125   2.97e-05    421.791      0.000       0.012
0.013
==============================================================================
======
Omnibus:                   1223483.628   Durbin-Watson:                   0.860
Prob(Omnibus):                   0.000   Jarque-Bera (JB):         24323116.351
Skew:                           -2.505   Prob(JB):                         0.00
Kurtosis:                       19.117   Cond. No.                         31.8
==============================================================================
======
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
```

6.9-Checking Residuals for Normality distribution for Global active power

In [38]:

```
# Applying histogram and Q-Q plot:

residuals = Model_2.resid
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(10, 4))

# Plot the histogram of the residuals
sns.histplot(residuals, kde=True, ax=ax1)
ax1.set_title('Histogram of Residuals')

# Plot the QQ plot of the residuals
sm.graphics.qqplot(residuals, line='s', ax=ax2)
```

```
ax2.set_title('QQ Plot of Residuals')

plt.show()
```



6.10-Checking Independence applying numerical analysis, Durbin-Watson:

In [39]:

```
from statsmodels.stats.stattools import durbin_watson
dw_test = durbin_watson(Model_2.resid)
dw_test
```

Out[39]:

```
0.8601917065688713
```

7- Reference

Amin, N & Nasserzadeh, M. (2014). Household Energy Consumption and Related Factors: Empirical Evidence from Malaysia. *Journal of Energy and Natural Resources Management*, 1(1), 1-10.

Chiaraviglio, L., Mellia, M., & Neri, F. (2009, June). Reducing power consumption in backbone networks. In *2009 IEEE international conference on communications* (pp. 1-6). IEEE.

Department of Mathematics, University of Illinois at Urbana-Champaing. (2016). Regression analysis of energy consumption data. Retrieved from https://math.illinois.edu/system/files/inline-files/Proj9AY1516-report2.pdf

Flatt, C., & Jacobs, R.L. (2019). Principle Assumptions of Regression Analysis: Testing, Techniques, and Statistical Reporting of Imperfect Data Sets. *Advances in Developing Human Resources*, *21*, 484 - 502.

Marden, J. I. (2004). Positions and QQ Plots. *Statistical Science*, *19*(4), 606–614. http://www.jstor.org/stable/4144431

University College Dublin. (nd). Quantile-quantile plots. Retrieved from https://www.ucd.ie/ecomodel/Resources/QQplots_WebVersion.html