Assignment 1

Part 1: Graphical Analysis

1. Scatter Diagram (10 points)

a. Choose a dataset (from UCI Machine Learning Repository).

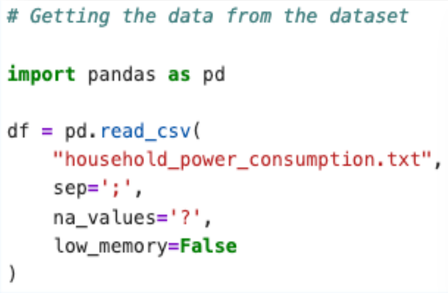
b. Create a scatter diagram to visualize the relationship between two numerical variables.

c. Describe the relationship you observe. Is it linear, non-linear, or no correlation? Provide a brief interpretation of your findings.

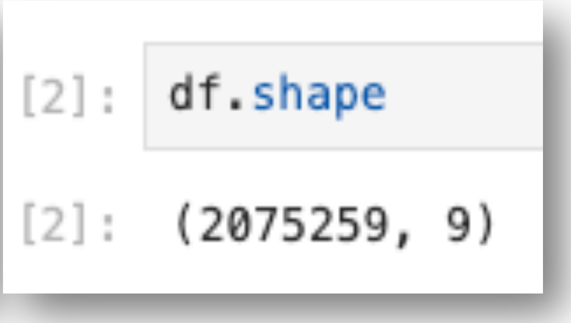
Solution

The dataset used in this assignment is Individual Household Electric Power Consumption by *Georges Hebrail* and *Alice Berard*

The data was loaded from a .txt file into Jupyter notebook using pandas library.

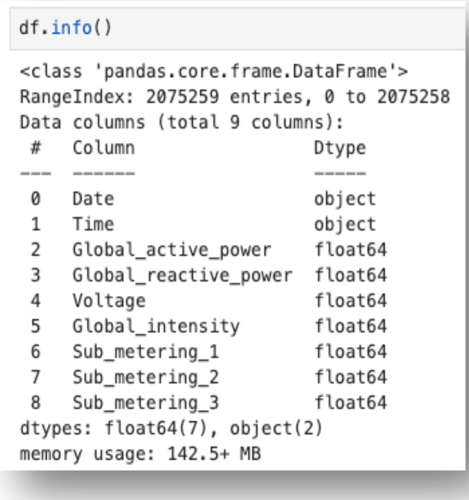


```
# Getting the data from the dataset

import pandas as pd

df = pd.read_csv(
    "household_power_consumption.txt",
    sep=';',
    na_values='?',
    low_memory=False
)
```

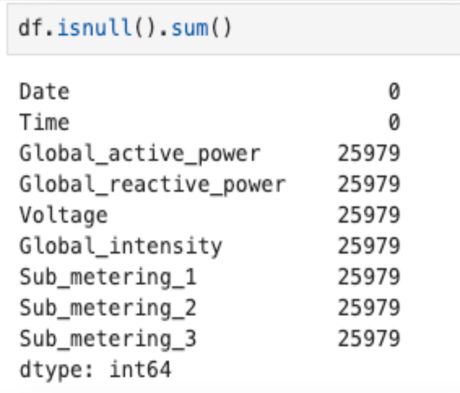


```
[2]: df.shape

[2]: (2075259, 9)
```

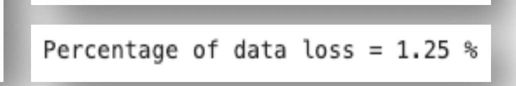
The data was examined to see the presence of null values that could interfere with the analysis. There was the presence of missing values and they were handled by removing them from the dataset.



```
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    float64
 3   Global_reactive_power  float64
 4   Voltage                float64
 5   Global_intensity       float64
 6   Sub_metering_1         float64
 7   Sub_metering_2         float64
 8   Sub_metering_3         float64
dtypes: float64(7), object(2)
memory usage: 142.5+ MB
```



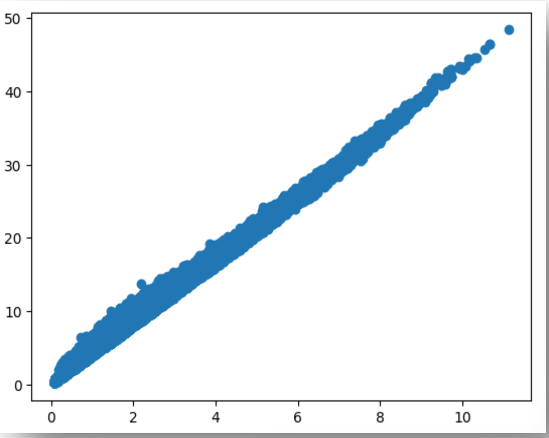
```
df.isnull().sum()

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

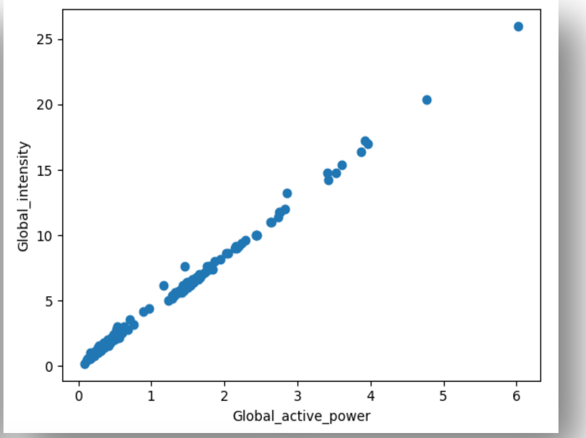


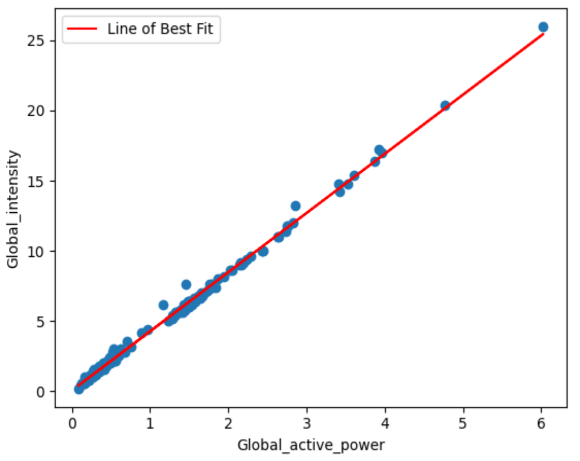
```
Percentage of data loss = 1.25 %
```

Scatter diagram was created between two numeric variables named Global_active_power and Global_intensity.

Figure below shows the scatter plot on population data, sample data of sample size 200 and deviation of points from the best fit line.







Interpretation:

From the above scatter plot of Global_active_power vs Global_intensity, we can conclude that the variables have strong positive linear correlation. Global_active_power while may not be the only factor that can predict Global_intensity values but it is a strong predictor.

2. Q-Q Plot (10 points)

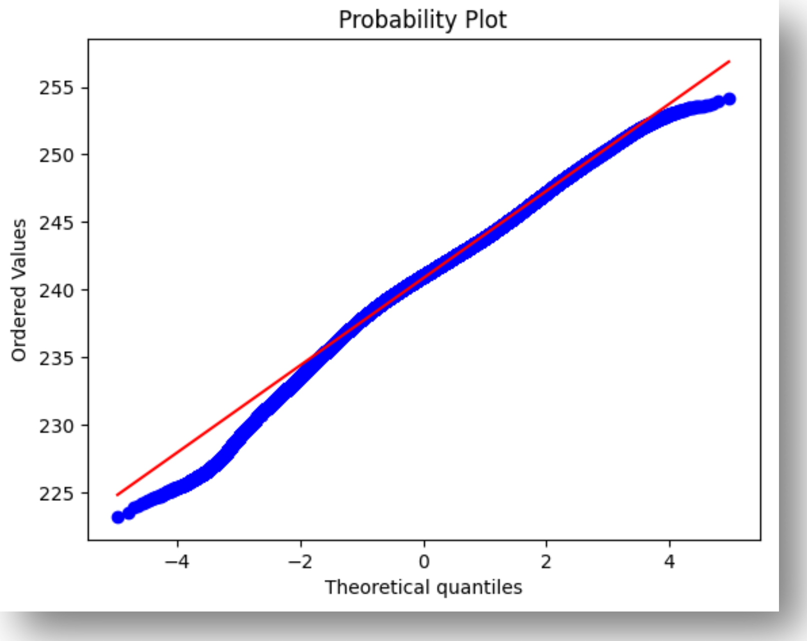
a. Using the same dataset, create a Q-Q plot to assess the normality of one of the numerical variables.

b. Explain what a Q-Q plot is and how to interpret it. Does the data appear to follow a normal distribution? Justify your answer.

Solution

Voltage was chosen as the test variable for the normality test using the Q-Q plot

The following plot shows the plot of normal line of the normal distribution against that of our test data's distribution.



Does the above data show normal distribution?

The data is approximately normally distributed in the center, but shows systematic deviations in the tails, indicating mild non-normality, likely due to skewness.

Justification:

Middle section aligns well with the red line:

- Around theoretical quantiles -1 to +3 Points closely follow the line
- Bulk of the data is roughly normal

What is Q-Q plot ?

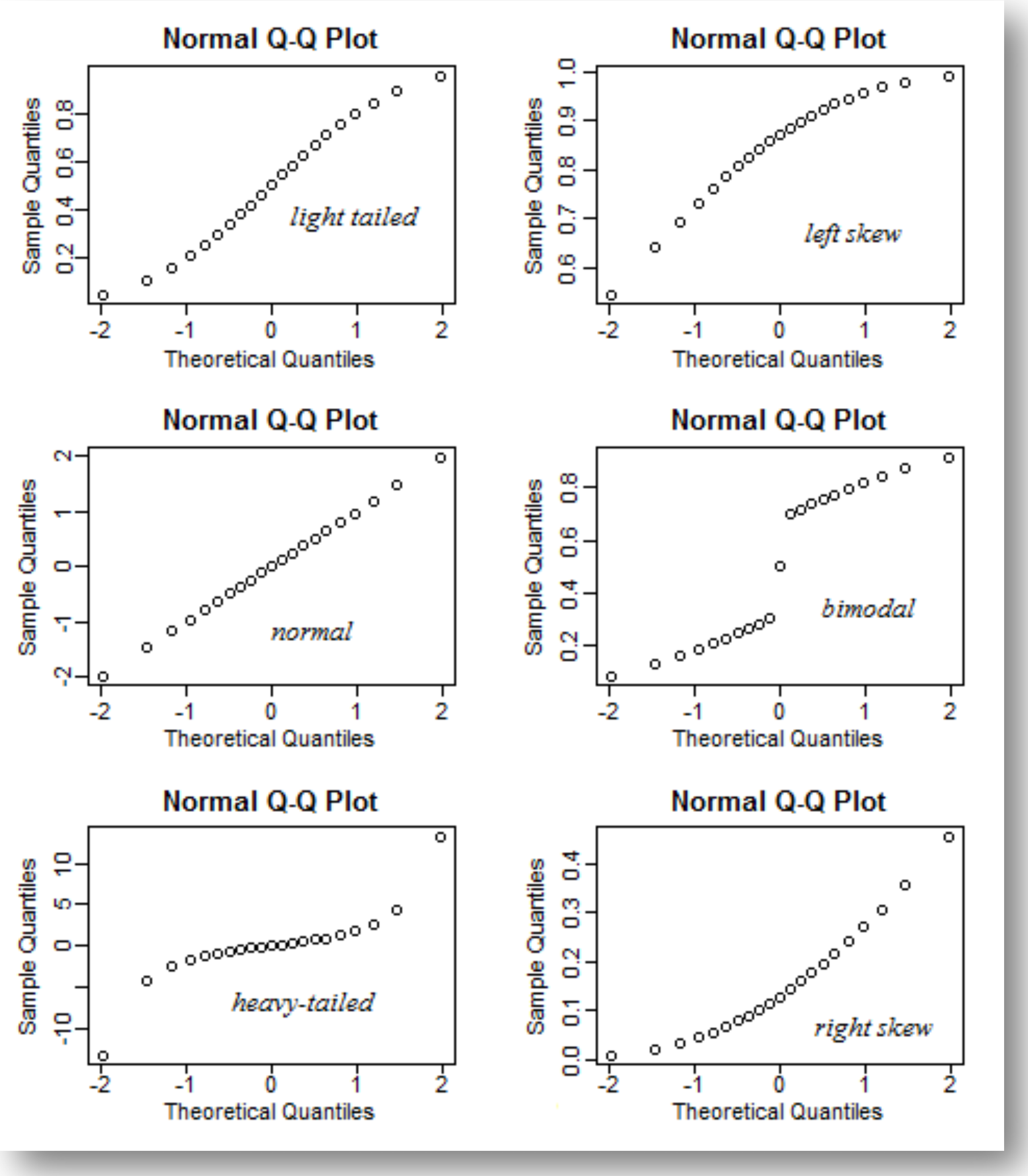
A Q-Q plot (Quantile-Quantile plot) is a plot of quantiles of test dataset against the quantiles of a theoretical distribution to visually check if the test data fits that distribution, with points forming a straight line indicating a good match, while deviations indicating skewness.

How to Interpret It

1. Perfect Fit
   - If the data perfectly matches the theoretical distribution, all points fall exactly on a straight, diagonal reference line (often $y = x$).
2. Normal Distribution (Common Use)
   - Linear Pattern: Points generally lying on a straight line suggest approximate normality.
3. Skewness
   - Right Skew (Positive Skew): An upward curve or S-shape at the end (upper tail).
   - Left Skew (Negative Skew): A downward curve or S-shape at the beginning (lower tail).
4. Kurtosis (Tail Behavior)
   - Heavy Tails (Leptokurtic): Points deviate upwards at both ends, indicating more extreme values than a normal distribution.
   - Light Tails (Platykurtic): Points deviate downwards at both ends, indicating fewer extreme values than a normal distribution.

Below is a diagram that shows what QQ-plots look like (for particular choices of distribution) on average:

Source: stackexchange



3. Box Plots (10 points)

a. Generate box plots for the numerical variable across different categories of a categorical variable in your dataset.

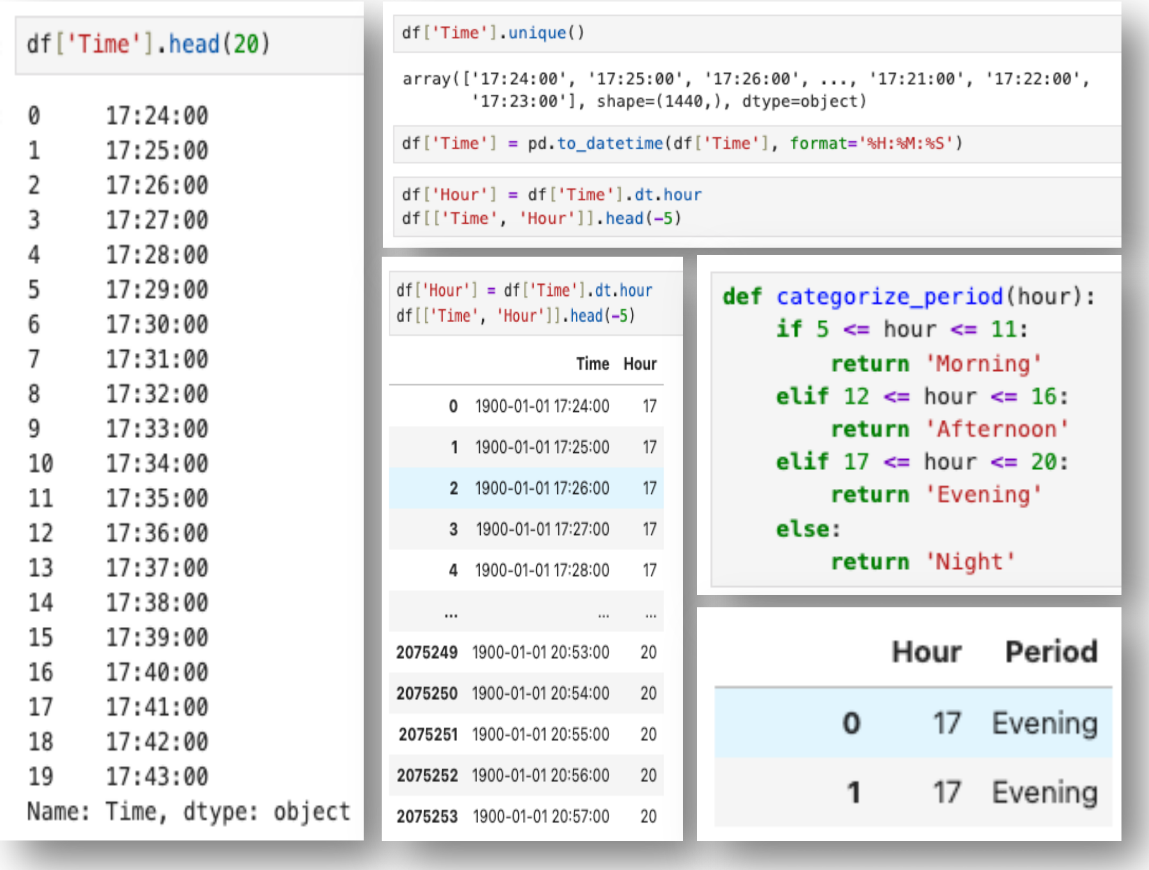
b. Analyze the box plots and discuss any outliers, the spread of the data.

Solution

The numeric variable for this test was chosen to be Global_active_power. It as a continuous variable. The categorical variable chosen was Time.

Since we are working with roughly 2 million Time data, where the unique timestamps are 1440, we need a way to categorize them into smaller quantity.

For scaling down this huge 1440 unique value, the values were first changed to just the hour value i.e. the timestamp 17:24:00 will now be just 17. This will give us 24 unique values. After that the time was changed to period like Morning, Afternoon, Evening, Night. Resulting in 4 categorical variables.



```
df['Time'].head(20)
```

```
0     17:24:00
1     17:25:00
2     17:26:00
3     17:27:00
4     17:28:00
5     17:29:00
6     17:30:00
7     17:31:00
8     17:32:00
9     17:33:00
10    17:34:00
11    17:35:00
12    17:36:00
13    17:37:00
14    17:38:00
15    17:39:00
16    17:40:00
17    17:41:00
18    17:42:00
19    17:43:00
Name: Time, dtype: object
```

```
df['Time'].unique()
```

```
array(['17:24:00', '17:25:00', '17:26:00', ..., '17:21:00', '17:22:00',
       '17:23:00'], shape=(1440,), dtype=object)
```

```
df['Time'] = pd.to_datetime(df['Time'], format='%H:%M:%S')
```

```
df['Hour'] = df['Time'].dt.hour
df[['Time', 'Hour']].head(-5)
```

```
df['Hour'] = df['Time'].dt.hour
df[['Time', 'Hour']].head(-5)
```

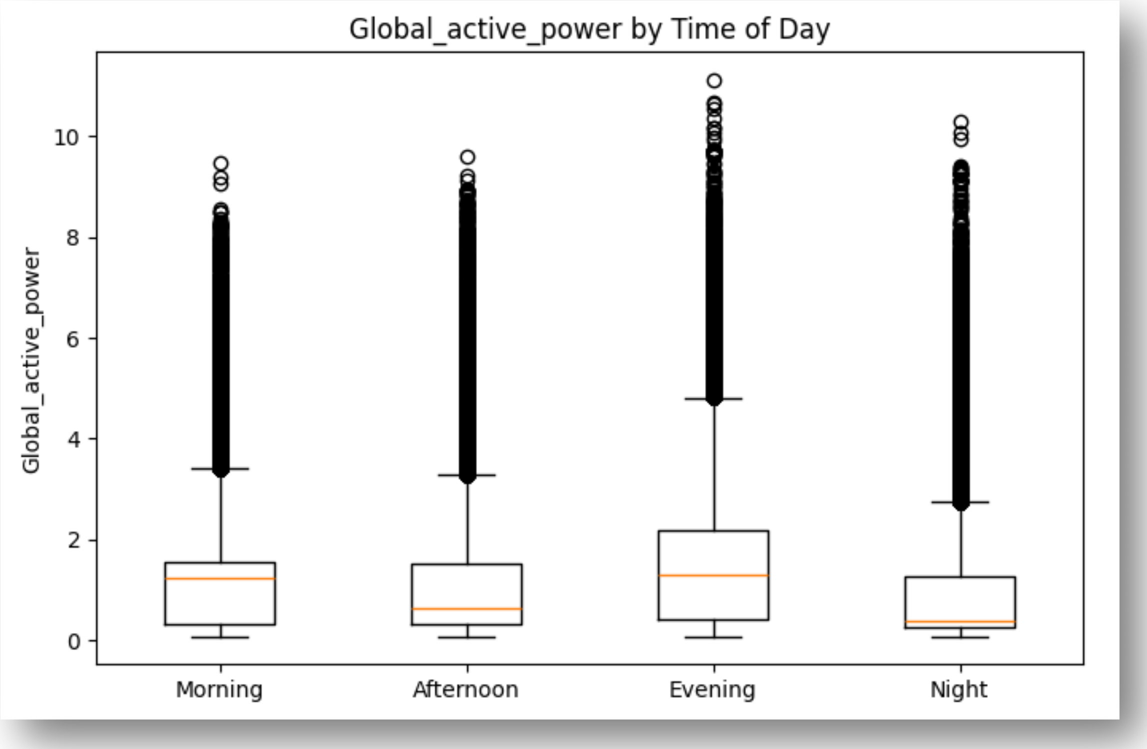
	Time	Hour
0	1900-01-01 17:24:00	17
1	1900-01-01 17:25:00	17
2	1900-01-01 17:26:00	17
3	1900-01-01 17:27:00	17
4	1900-01-01 17:28:00	17
...	...	...
2075249	1900-01-01 20:53:00	20
2075250	1900-01-01 20:54:00	20
2075251	1900-01-01 20:55:00	20
2075252	1900-01-01 20:56:00	20
2075253	1900-01-01 20:57:00	20

```
def categorize_period(hour):
    if 5 <= hour <= 11:
        return 'Morning'
    elif 12 <= hour <= 16:
        return 'Afternoon'
    elif 17 <= hour <= 20:
        return 'Evening'
    else:
        return 'Night'
```

	Hour	Period
0	17	Evening
1	17	Evening

github.com/Nayan-Chimariya/Computational-Statistics-and-Probability

Finally, after creating the categorical variable Period from the Time column, box plot was plotted against the numeric variable Global_active_power



Analysis of the plot:

1. The spread of the data
   - The spread of the data is indicated by the Interquartile Range and the length of the whiskers.
   - Evening Peak: The Evening category shows the largest spread. Both the box and the upper whisker are significantly taller than the other time periods. indicating that the power consumption is generally higher during these hours.
   - Night Lows: The Night category has the smallest spread and the lowest median. The box is compressed toward the bottom, indicating that for most of the night, power usage remains low.
   - Morning vs. Afternoon: These two are relatively similar in their range, though the Morning has a slightly higher median and a more balanced distribution within the box compared to the Afternoon.
   -

2. Presence of Outliers

   - Positive Skew: All distributions are heavily right-skewed. While the "normal" consumption stays mostly below 3 units, there are frequent instances where power usage spikes up to 8, 10, or even 11 units.
   - Extreme Events: The Evening session contains the most extreme outliers, reaching the highest point on the graph (above 11). Representing times when multiple high energy appliances are running simultaneously.
   - The Night Exception: Even though Night has the lowest average usage, it still shows a massive column of outliers. This suggests that while most people are asleep, certain households or specific nights involve high energy activity.

Part 1: Graphical Analysis

4. Normality Test (10 points)

a. Perform a normality test (e.g., Shapiro-Wilk test) on the numerical variable you analyzed.

b. Report the test statistic and p-value. Based on these results, do you reject or fail to reject the null hypothesis of normality? Explain your reasoning.

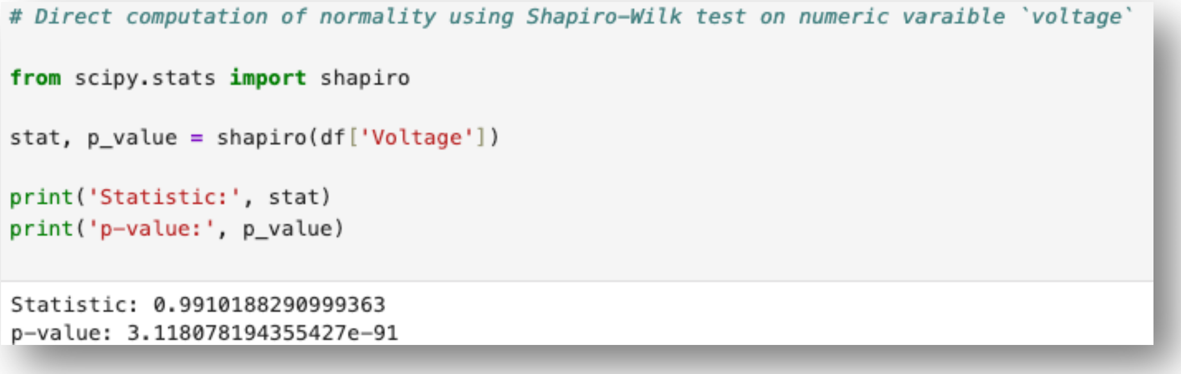
Solution

Shapiro-Wilk test was performed to test whether the distribution of numeric continuous variable Voltage is normal or not. The following hypothesis were stated.

H_0 = The data come from a normal distribution

H_1 = The data do not come from a normal distribution

Test Statistics W was computed using SciPy library and was found to be



```
# Direct computation of normality using Shapiro-Wilk test on numeric varaible `voltage`

from scipy.stats import shapiro

stat, p_value = shapiro(df['Voltage'])

print('Statistic:', stat)
print('p-value:', p_value)
```

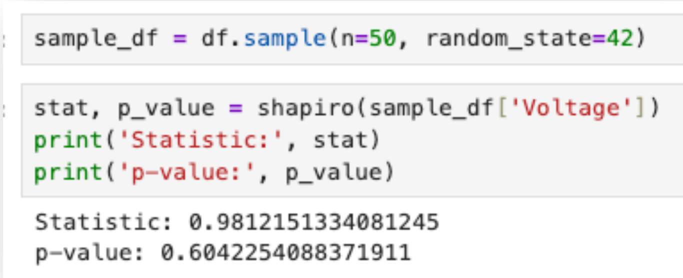
```
Statistic: 0.9910188290999363
p-value: 3.118078194355427e-91
```

As we can see the p-value is too low for a conclusion. This is because the test statistics was computed on the population data of size roughly 2 million.

The sample size is large for the normality test The Shapiro–Wilk test:

- Was designed for small to medium samples
- Uses approximations for large *N*
- Becomes unreliable for very large datasets

Then, a sample size of n=50 was taken to conclude the test of hypothesis



```
sample_df = df.sample(n=50, random_state=42)
```

```
stat, p_value = shapiro(sample_df['Voltage'])
print('Statistic:', stat)
print('p-value:', p_value)
```

```
Statistic: 0.9812151334081245
p-value: 0.6042254088371911
```

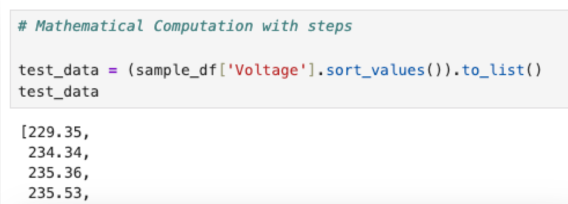
Conclusion

A Shapiro–Wilk test was conducted to assess the normality of the Voltage variable using a random sample of 50 observations.

The test produced a statistic of 0.981 and a p-value of 0.604. Since the test statistics value is close to 1 and p-value exceeds the 0.05 significance level, the null hypothesis of normality cannot be rejected, indicating that the Voltage data is approximately normally distributed.

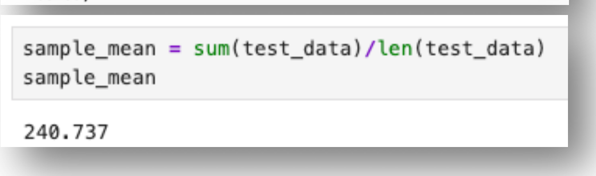
Manual Computation without using SciPy

The list of test data was sorted in ascending order and the sample mean was computed.



```
# Mathematical Computation with steps

test_data = (sample_df['Voltage'].sort_values()).to_list()
test_data
```

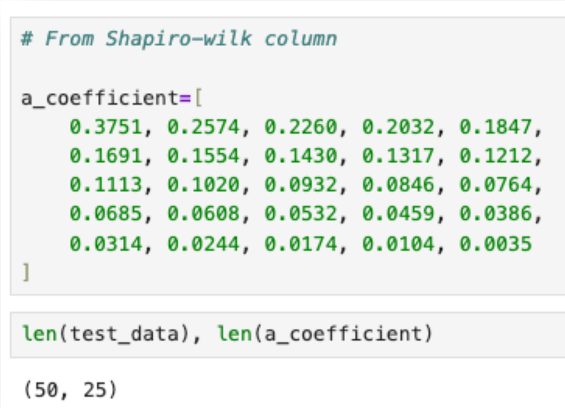
```
[229.35,
 234.34,
 235.36,
 235.53,
```



```
sample_mean = sum(test_data)/len(test_data)
sample_mean
```

```
240.737
```

A list of coefficients of a was created. The values were taken from this table

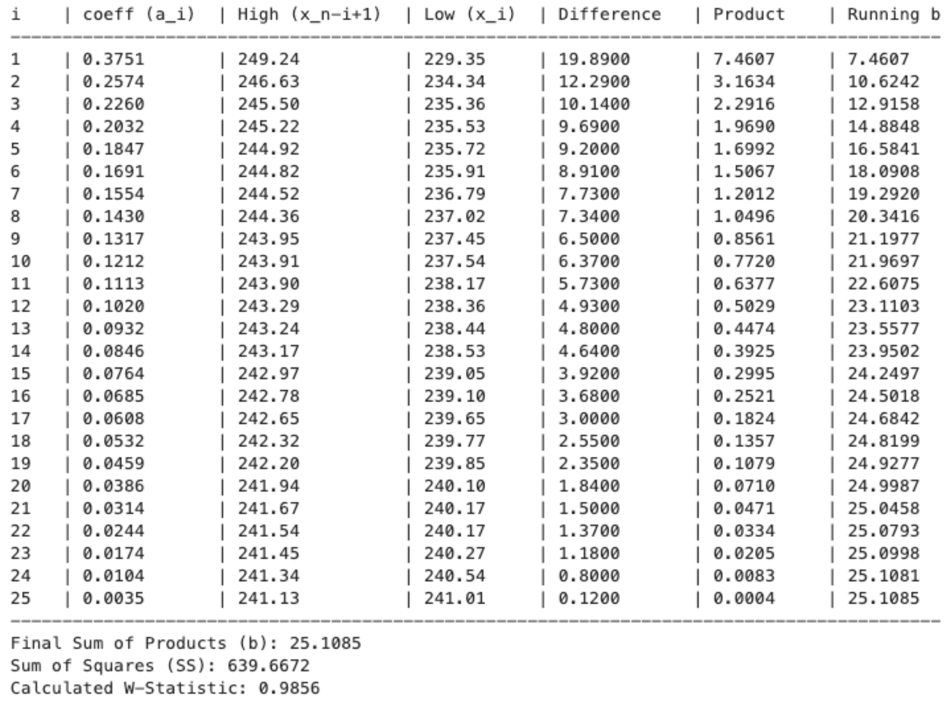


```
# From Shapiro-wilk column

a_coefficient=[
    0.3751, 0.2574, 0.2260, 0.2032, 0.1847,
    0.1691, 0.1554, 0.1430, 0.1317, 0.1212,
    0.1113, 0.1020, 0.0932, 0.0846, 0.0764,
    0.0685, 0.0608, 0.0532, 0.0459, 0.0386,
    0.0314, 0.0244, 0.0174, 0.0104, 0.0035
]
```

```
len(test_data), len(a_coefficient)
```

```
(50, 25)
```

W was computed and the table showing intermediate values are presented below



```
i     | coeff (a_i)    | High (x_n-i+1)   | Low (x_i)   | Difference    | Product      | Running b
--------------------------------------------------------------------------------------------------------
1     | 0.3751         | 249.24           | 229.35      | 19.8900       | 7.4607       | 7.4607
2     | 0.2574         | 246.63           | 234.34      | 12.2900       | 3.1634       | 10.6242
3     | 0.2260         | 245.50           | 235.36      | 10.1400       | 2.2916       | 12.9158
4     | 0.2032         | 245.22           | 235.53      | 9.6900        | 1.9690       | 14.8848
5     | 0.1847         | 244.92           | 235.72      | 9.2000        | 1.6992       | 16.5841
6     | 0.1691         | 244.82           | 235.91      | 8.9100        | 1.5067       | 18.0908
7     | 0.1554         | 244.52           | 236.79      | 7.7300        | 1.2012       | 19.2920
8     | 0.1430         | 244.36           | 237.02      | 7.3400        | 1.0496       | 20.3416
9     | 0.1317         | 243.95           | 237.45      | 6.5000        | 0.8561       | 21.1977
10    | 0.1212         | 243.91           | 237.54      | 6.3700        | 0.7720       | 21.9697
11    | 0.1113         | 243.90           | 238.17      | 5.7300        | 0.6377       | 22.6075
12    | 0.1020         | 243.29           | 238.36      | 4.9300        | 0.5029       | 23.1103
13    | 0.0932         | 243.24           | 238.44      | 4.8000        | 0.4474       | 23.5577
14    | 0.0846         | 243.17           | 238.53      | 4.6400        | 0.3925       | 23.9502
15    | 0.0764         | 242.97           | 239.05      | 3.9200        | 0.2995       | 24.2497
16    | 0.0685         | 242.78           | 239.10      | 3.6800        | 0.2521       | 24.5018
17    | 0.0608         | 242.65           | 239.65      | 3.0000        | 0.1824       | 24.6842
18    | 0.0532         | 242.32           | 239.77      | 2.5500        | 0.1357       | 24.8199
19    | 0.0459         | 242.20           | 239.85      | 2.3500        | 0.1079       | 24.9277
20    | 0.0386         | 241.94           | 240.10      | 1.8400        | 0.0710       | 24.9987
21    | 0.0314         | 241.67           | 240.17      | 1.5000        | 0.0471       | 25.0458
22    | 0.0244         | 241.54           | 240.17      | 1.3700        | 0.0334       | 25.0793
23    | 0.0174         | 241.45           | 240.27      | 1.1800        | 0.0205       | 25.0998
24    | 0.0104         | 241.34           | 240.54      | 0.8000        | 0.0083       | 25.1081
25    | 0.0035         | 241.13           | 241.01      | 0.1200        | 0.0004       | 25.1085
--------------------------------------------------------------------------------------------------------
Final Sum of Products (b): 25.1085
Sum of Squares (SS): 639.6672
Calculated W-Statistic: 0.9856
```

P value was computed using this table

Formula to calculate p value

$$p = p1 + ((p2-p1)/(w2-w1))*(w-w1)$$

- Where w1 is the largest value in the row that is just less than w while W2 is the smallest value in the same row that is just greater than w
- p1 and p2 are corresponding values of w1 and w2 respectively

P value was found to be 0.8535714285714285

Findings

The shapiro() function gave us 0.9812, while the manual loop gave us 0.9856. This happened because of the coefficients (a;):

- The values like 0.3751 we used are rounded constants from standard statistical tables.
- The scipy.stats.shapiro function uses the Royston algorithm, which calculates coefficients to a much higher precision

- Because W = b^2/SS, even a tiny change in b is squared, which amplifies the error. A difference of 0.004 in W might seem small, but in a Shapiro-Wilk table, that covers a massive jump in p-value.

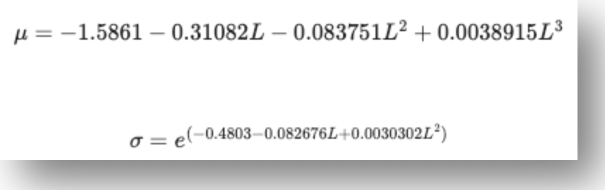
The Royston Method

For a sample size of n=50, we transform w into a y value then into a Z-score.

Calculation of y

$$y = \ln(1\text{-}w)$$

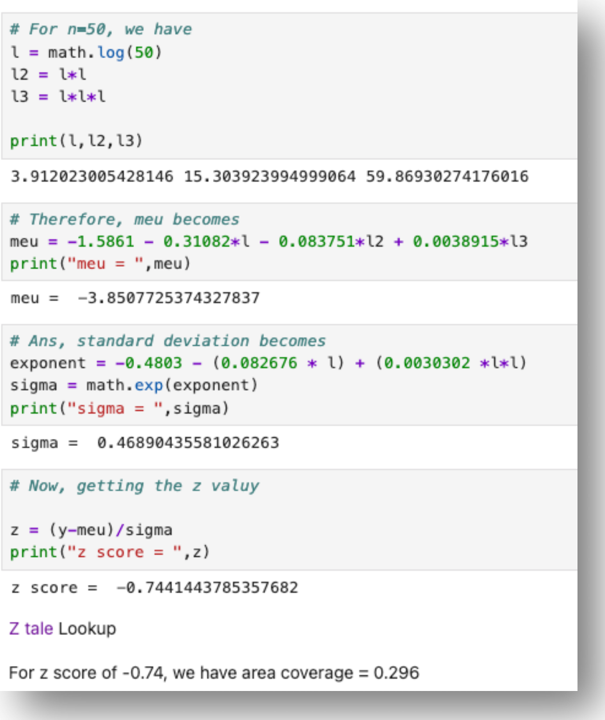
Calculate the Expected Mean (μ) and Standard Deviation (σ)



$$\mu = -1.5861 - 0.31082L - 0.083751L^2 + 0.0038915L^3$$

$$\sigma = e^{(-0.4803 - 0.082676L + 0.0030302L^2)}$$

Calculating rest of the values



```
# For n=50, we have
l = math.log(50)
l2 = l*l
l3 = l*l*l

print(l,l2,l3)
```

```
3.912023005428146 15.303923994999064 59.86930274176016
```

```
# Therefore, meu becomes
meu = -1.5861 - 0.31082*l - 0.083751*l2 + 0.0038915*l3
print("meu = ",meu)
```

```
meu =  -3.8507725374327837
```

```
# Ans, standard deviation becomes
exponent = -0.4803 - (0.082676 * l) + (0.0030302 *l*l)
sigma = math.exp(exponent)
print("sigma = ",sigma)
```

```
sigma =  0.46890435581026263
```

```
# Now, getting the z valuy

z = (y-meu)/sigma
print("z score = ",z)
```

```
z score =  -0.7441443785357682
```

Z tale Lookup

For z score of -0.74, we have area coverage = 0.296

Interpretation:

Because the p-value is significantly higher than the threshold, we fail to reject the Null Hypothesis.

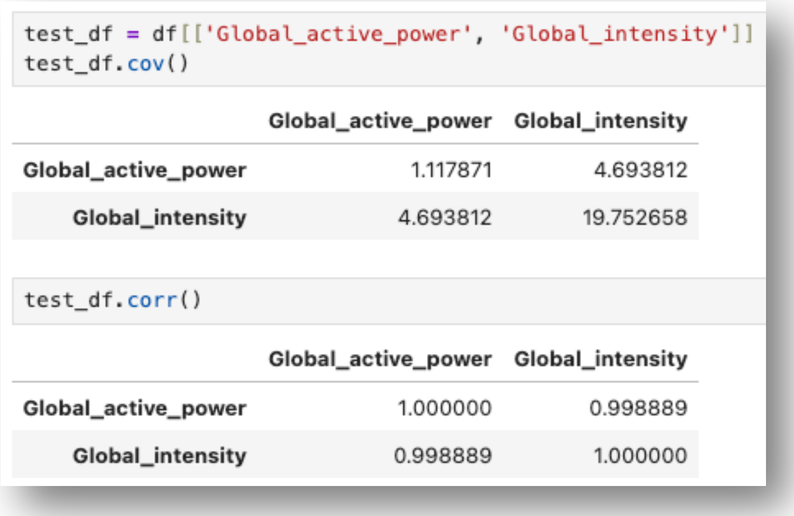
4. Correlation and Covariance (10 points)

a. Calculate the correlation coefficient and covariance between the two numerical variables you used in the scatter diagram.

b. Interpret the results. What do they tell you about the relationship between these variables?

Solution

The Correlation and covariance were computed between two numeric variables Global active power and Global intensity using the pandas library



```
test_df = df[['Global_active_power', 'Global_intensity']]
test_df.cov()
```

	Global_active_power	Global_intensity
Global_active_power	1.117871	4.693812
Global_intensity	4.693812	19.752658

```
test_df.corr()
```

	Global_active_power	Global_intensity
Global_active_power	1.000000	0.998889
Global_intensity	0.998889	1.000000

Interpretation

The covariance between the test variables `Global_active_power` and `Global_intensity` is 4.69 which is positive indicating they move together (due to same sign).

The correlation between them is 0.998 indicates that they have a very strong positive linear relationship. When Global_active_power increases, Global_intensity increases almost proportionally. One variable can predict the other very well.