```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session


# Load datasets
customer_data = pd.read_csv('/content/client_data.csv')
pricing_data = pd.read_csv('/content/price_data.csv')

# Display the first few rows of each dataset
print("Customer Data:")
print(customer_data.head())
print("\nPricing Data:")
print(pricing_data.head())
```

```
                                  id                    channel_sales  \
0  24011ae4ebbe3035111d65fa7c15bc57  foosdfpfkusacimwkcsosbicdxkicaua
1  d29c2c54acc38ff3c0614d0a653813dd                           MISSING
2  764c75f661154dac3a6c254cd082ea7d  foosdfpfkusacimwkcsosbicdxkicaua
3  bba03439a292a1e166f80264c16191cb  lmkebamcaaclubfxadlmueccxoimlema
4  149d57cf92fc41cf94415803a877cb4b                           MISSING

   cons_12m  cons_gas_12m  cons_last_month  date_activ    date_end  \
0         0         54946                0  2013-06-15  2016-06-15
1      4660             0                0  2009-08-21  2016-08-30
2       544             0                0  2010-04-16  2016-04-16
3      1584             0                0  2010-03-30  2016-03-30
4      4425             0              526  2010-01-13  2016-03-07

  date_modif_prod date_renewal  forecast_cons_12m  ...  has_gas  imp_cons  \
0      2015-11-01   2015-06-23               0.00  ...        t      0.00
1      2009-08-21   2015-08-31             189.95  ...        f      0.00
2      2010-04-16   2015-04-17              47.96  ...        f      0.00
3      2010-03-30   2015-03-31             240.04  ...        f      0.00
4      2010-01-13   2015-03-09             445.75  ...        f     52.32

   margin_gross_pow_ele  margin_net_pow_ele  nb_prod_act  net_margin  \
0                 25.44               25.44            2      678.99
1                 16.38               16.38            1       18.89
2                 28.60               28.60            1        6.60
3                 30.22               30.22            1       25.46
4                 44.91               44.91            1       47.98

  num_years_antig                        origin_up  pow_max  churn
0               3  lxidpiddsbxsbosboudacockeimpuepw   43.648      1
1               6  kamkkxfxxuwbdslkwifmmcsiusiuosws   13.800      0
2               6  kamkkxfxxuwbdslkwifmmcsiusiuosws   13.856      0
3               6  kamkkxfxxuwbdslkwifmmcsiusiuosws   13.200      0
4               6  kamkkxfxxuwbdslkwifmmcsiusiuosws   19.800      0

[5 rows x 26 columns]

Pricing Data:
                                id  price_date  price_off_peak_var  \
0  038af19179925da21a25619c5a24b745  2015-01-01            0.151367
1  038af19179925da21a25619c5a24b745  2015-02-01            0.151367
2  038af19179925da21a25619c5a24b745  2015-03-01            0.151367
3  038af19179925da21a25619c5a24b745  2015-04-01            0.149626
4  038af19179925da21a25619c5a24b745  2015-05-01            0.149626

   price_peak_var  price_mid_peak_var  price_off_peak_fix  price_peak_fix  \
0             0.0                 0.0           44.266931             0.0
1             0.0                 0.0           44.266931             0.0
2             0.0                 0.0           44.266931             0.0
3             0.0                 0.0           44.266931             0.0
4             0.0                 0.0           44.266931             0.0

   price_mid_peak_fix
0                 0.0
1                 0.0
```

```
4                0.0
```

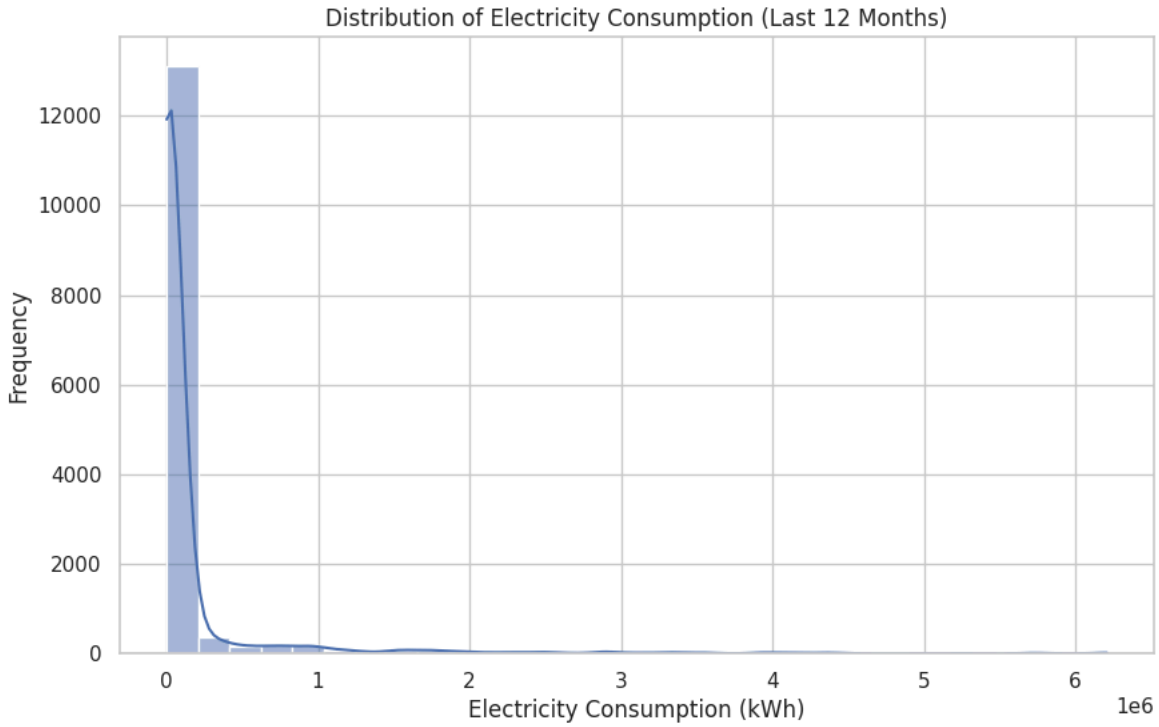
```
# Check data types
print("\nData Types of Customer Data:")
print(customer_data.dtypes)
print("\nData Types of Pricing Data:")
print(pricing_data.dtypes)
```

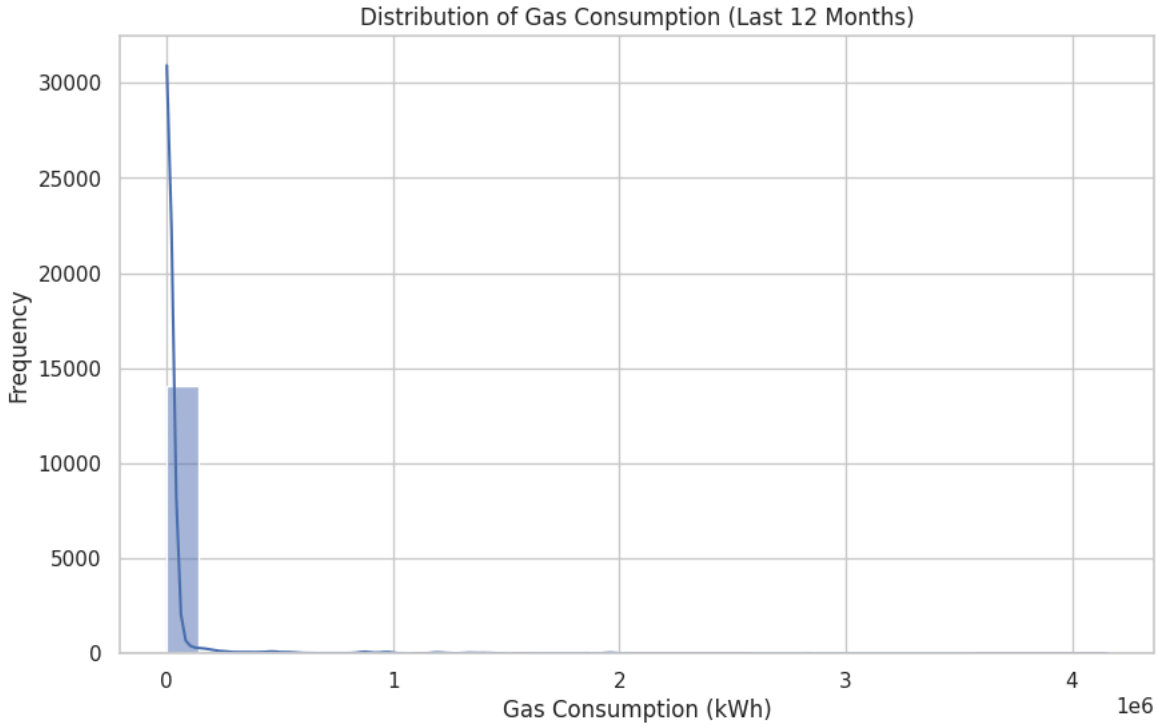
```
Data Types of Customer Data:
id                                 object
channel_sales                      object
cons_12m                            int64
cons_gas_12m                        int64
cons_last_month                     int64
date_activ                         object
date_end                           object
date_modif_prod                    object
date_renewal                       object
forecast_cons_12m                 float64
forecast_cons_year                  int64
forecast_discount_energy          float64
forecast_meter_rent_12m           float64
forecast_price_energy_off_peak    float64
forecast_price_energy_peak        float64
forecast_price_pow_off_peak       float64
has_gas                            object
imp_cons                          float64
margin_gross_pow_ele              float64
margin_net_pow_ele                float64
nb_prod_act                         int64
net_margin                        float64
num_years_antig                     int64
origin_up                          object
pow_max                           float64
churn                               int64
dtype: object

Data Types of Pricing Data:
id                     object
price_date             object
price_off_peak_var    float64
price_peak_var        float64
price_mid_peak_var    float64
price_off_peak_fix    float64
price_peak_fix        float64
price_mid_peak_fix    float64
dtype: object
```

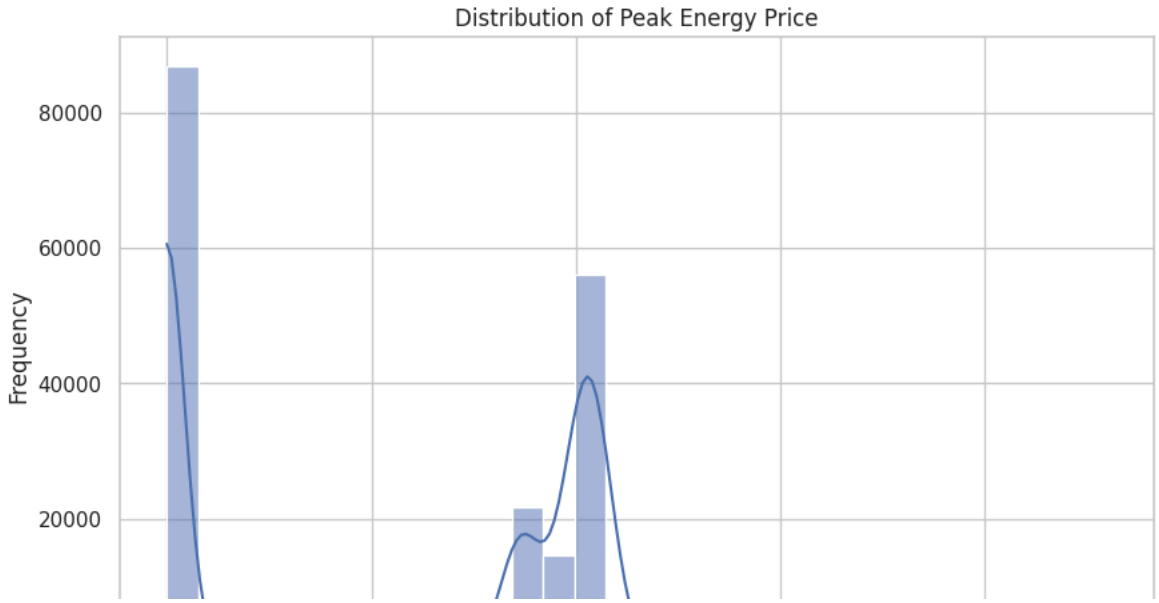
```
# Descriptive statistics for customer data
print("\nDescriptive Statistics of Customer Data:")
print(customer_data.describe(include='all'))

# Descriptive statistics for pricing data
print("\nDescriptive Statistics of Pricing Data:")
print(pricing_data.describe(include='all'))
```

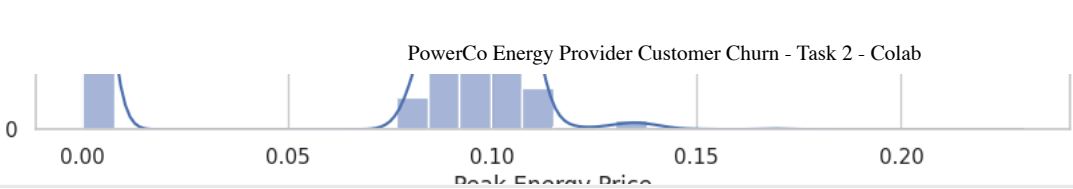
```
25%                                   NaN         NaN            0.125976
50%                                   NaN         NaN            0.146033
75%                                   NaN         NaN            0.151635
max                                   NaN         NaN            0.280700

        price_peak_var  price_mid_peak_var  price_off_peak_fix  \
count    193002.000000       193002.000000       193002.000000
unique             NaN                 NaN                 NaN
top                NaN                 NaN                 NaN
freq               NaN                 NaN                 NaN
mean          0.054630            0.030496           43.334477
std           0.049924            0.036298            5.410297
min           0.000000            0.000000            0.000000
25%           0.000000            0.000000           40.728885
50%           0.085483            0.000000           44.266930
75%           0.101673            0.072558           44.444710
max           0.229788            0.114102           59.444710

        price_peak_fix  price_mid_peak_fix
count    193002.000000       193002.000000
unique             NaN                 NaN
top                NaN                 NaN
freq               NaN                 NaN
mean         10.622875            6.409984
std          12.841895            7.773592
min           0.000000            0.000000
25%           0.000000            0.000000
50%           0.000000            0.000000
75%          24.339581           16.226389
max          36.490692           17.458221
```

```
import matplotlib.pyplot as plt
import seaborn as sns

# Set the aesthetic style of the plots
sns.set(style="whitegrid")

# Visualize distribution of electricity consumption (cons_12m) in customer data
plt.figure(figsize=(10, 6))
sns.histplot(customer_data['cons_12m'], bins=30, kde=True)
plt.title('Distribution of Electricity Consumption (Last 12 Months)')
plt.xlabel('Electricity Consumption (kWh)')
plt.ylabel('Frequency')
plt.show()

# Visualize distribution of gas consumption (cons_gas_12m) in customer data
plt.figure(figsize=(10, 6))
sns.histplot(customer_data['cons_gas_12m'], bins=30, kde=True)
plt.title('Distribution of Gas Consumption (Last 12 Months)')
plt.xlabel('Gas Consumption (kWh)')
plt.ylabel('Frequency')
plt.show()

# Visualize pricing data distribution for peak pricing
plt.figure(figsize=(10, 6))
sns.histplot(pricing_data['price_peak_var'], bins=30, kde=True)
plt.title('Distribution of Peak Energy Price')
plt.xlabel('Peak Energy Price')
plt.ylabel('Frequency')
plt.show()
```


Distribution of Electricity Consumption (Last 12 Months)
Frequency
Electricity Consumption (kWh)
1e6


Distribution of Gas Consumption (Last 12 Months)
Frequency
Gas Consumption (kWh)
1e6


Distribution of Peak Energy Price
Frequency

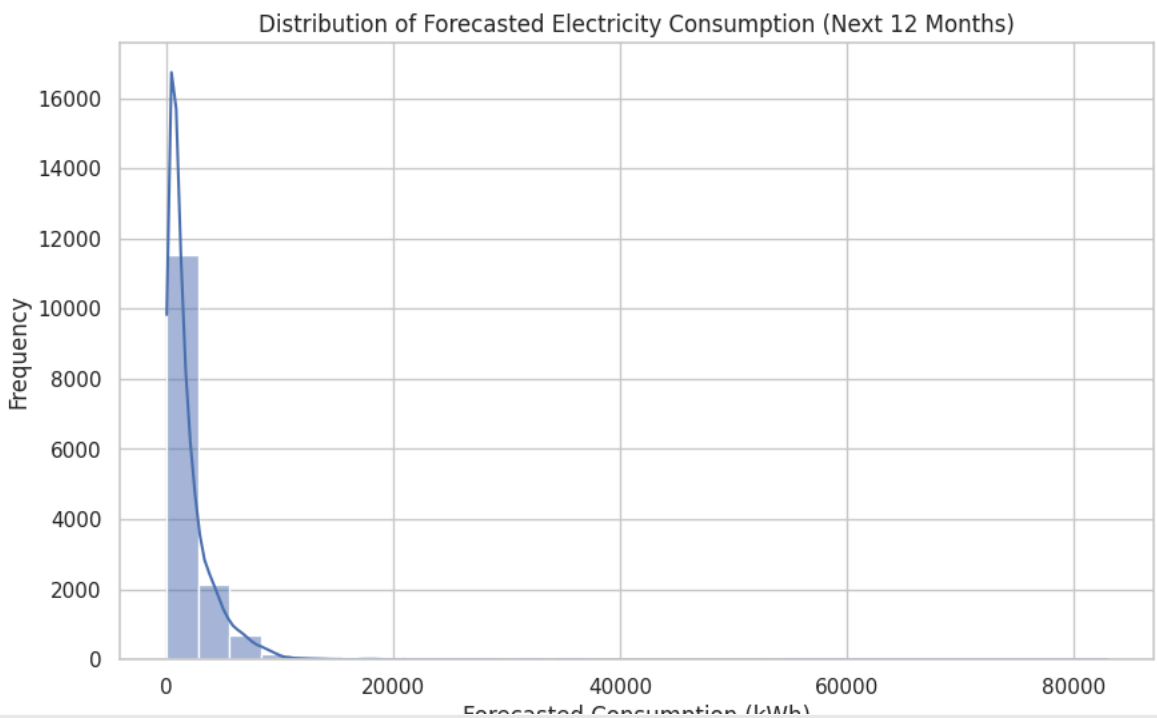


To modify the visualizations for other numerical columns or to add more plots in your exploratory data analysis (EDA), we can follow these steps:

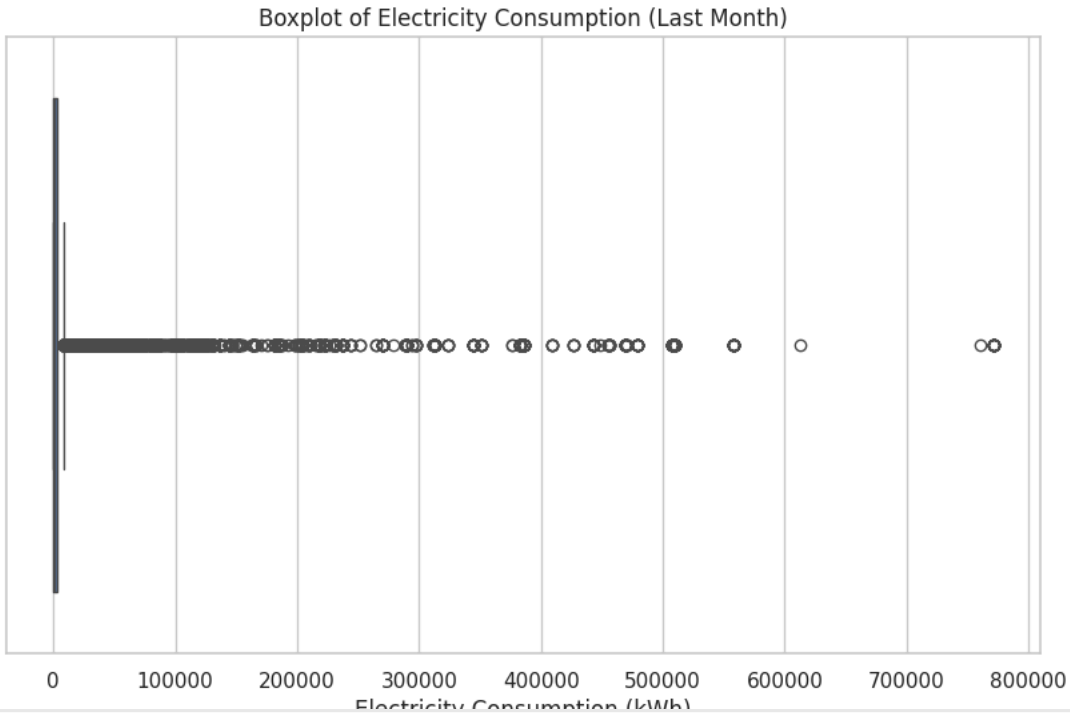
```
# List numerical columns in customer data
numerical_columns_customer = customer_data.select_dtypes(include='number').columns.tolist()
print("Numerical Columns in Customer Data:", numerical_columns_customer)

# List numerical columns in pricing data
numerical_columns_pricing = pricing_data.select_dtypes(include='number').columns.tolist()
print("Numerical Columns in Pricing Data:", numerical_columns_pricing)
```

```
Numerical Columns in Customer Data: ['cons_12m', 'cons_gas_12m', 'cons_last_month', 'forecast_cons_12m', 'forecast_cons_
Numerical Columns in Pricing Data: ['price_off_peak_var', 'price_peak_var', 'price_mid_peak_var', 'price_off_peak_fix',
```

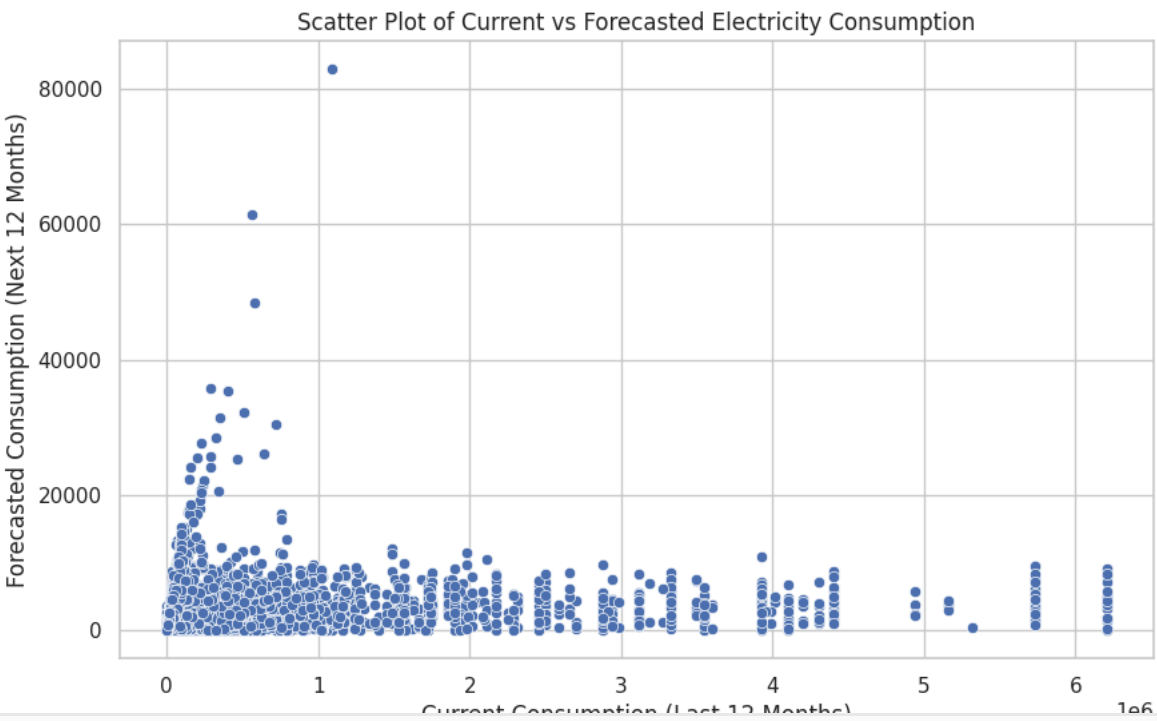
```
# Histogram for forecasted electricity consumption (next 12 months)
plt.figure(figsize=(10, 6))
sns.histplot(customer_data['forecast_cons_12m'], bins=30, kde=True)
plt.title('Distribution of Forecasted Electricity Consumption (Next 12 Months)')
plt.xlabel('Forecasted Consumption (kWh)')
plt.ylabel('Frequency')
plt.show()
```



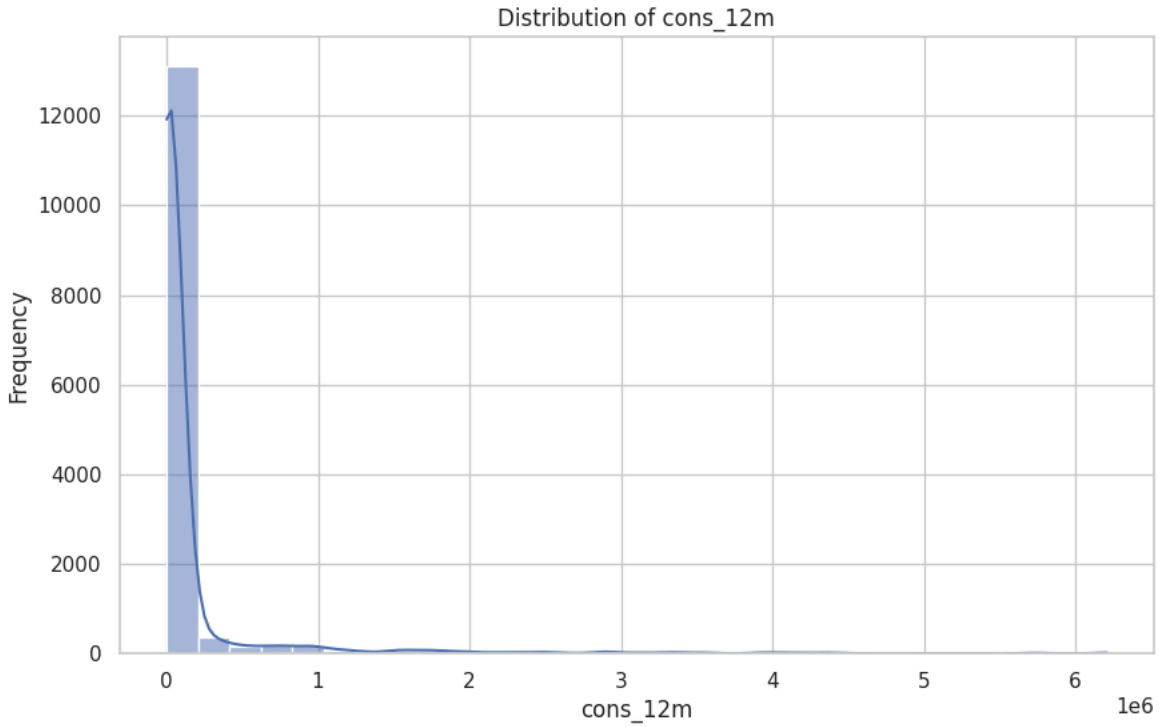

```
# Boxplot for electricity consumption (last month)
plt.figure(figsize=(10, 6))
sns.boxplot(x=customer_data['cons_last_month'])
plt.title('Boxplot of Electricity Consumption (Last Month)')
plt.xlabel('Electricity Consumption (kWh)')
plt.show()
```

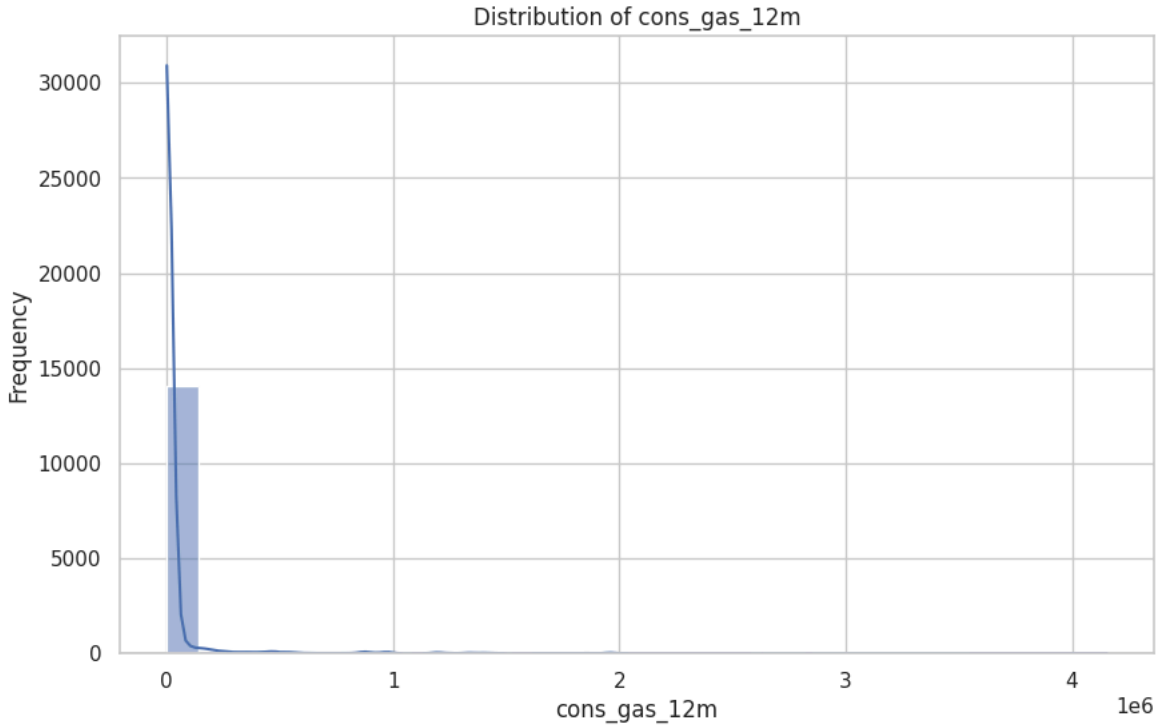



```
# Scatter plot for current vs forecasted electricity consumption
plt.figure(figsize=(10, 6))
sns.scatterplot(x=customer_data['cons_12m'], y=customer_data['forecast_cons_12m'])
plt.title('Scatter Plot of Current vs Forecasted Electricity Consumption')
plt.xlabel('Current Consumption (Last 12 Months)')
plt.ylabel('Forecasted Consumption (Next 12 Months)')
plt.show()
```

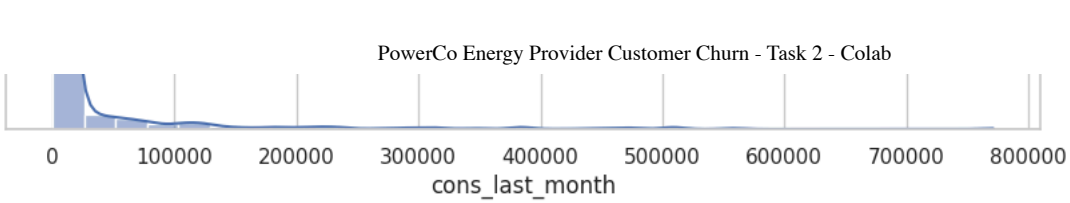


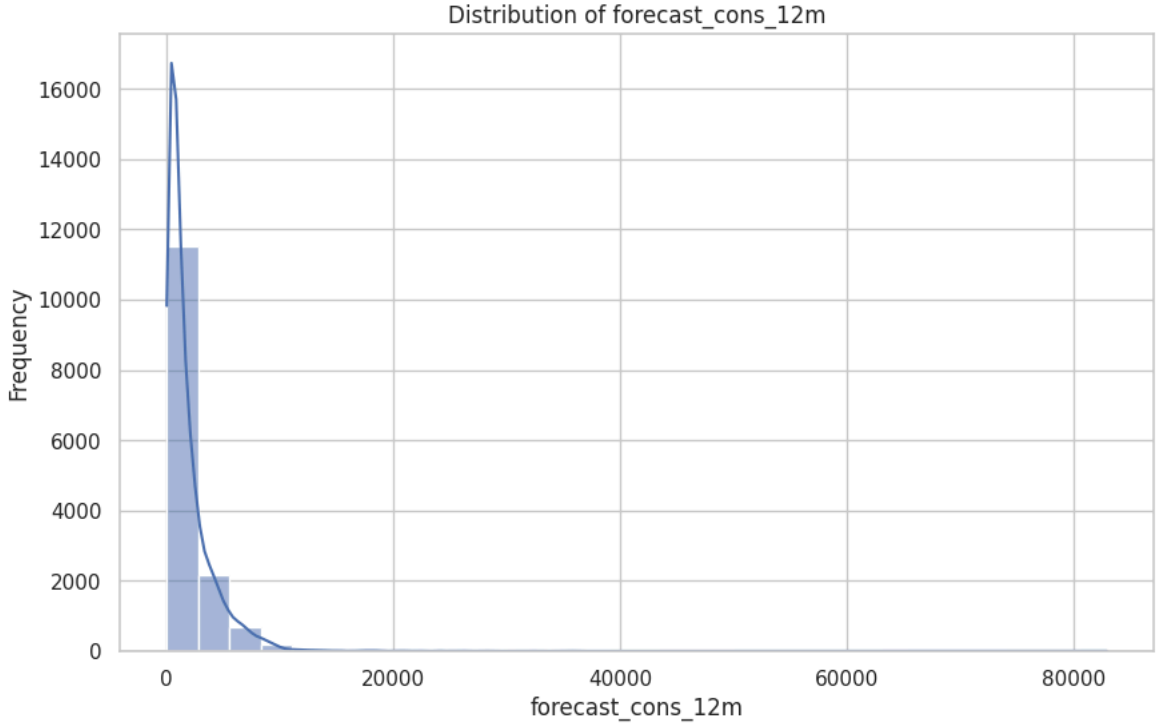

```
# Loop through numerical columns and create histograms
for column in numerical_columns_customer:
    plt.figure(figsize=(10, 6))
    sns.histplot(customer_data[column], bins=30, kde=True)
    plt.title(f'Distribution of {column}')
    plt.xlabel(column)
    plt.ylabel('Frequency')
    plt.show()
```

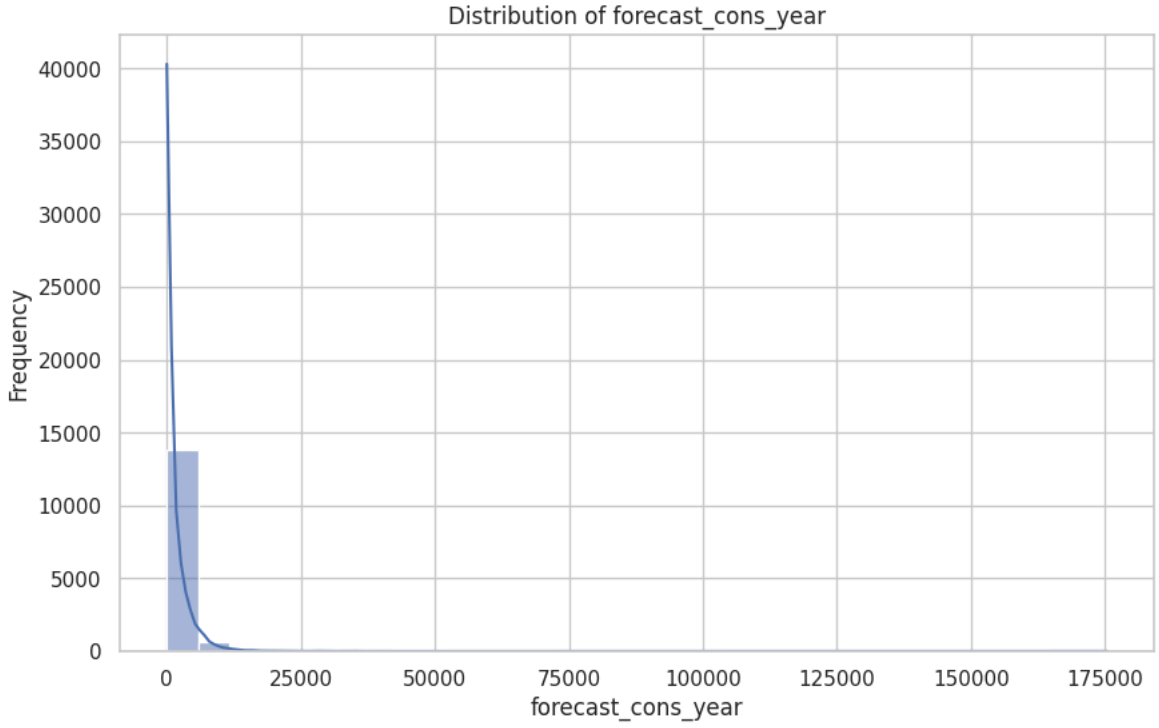

Distribution of cons_12m
Frequency
cons_12m
1e6

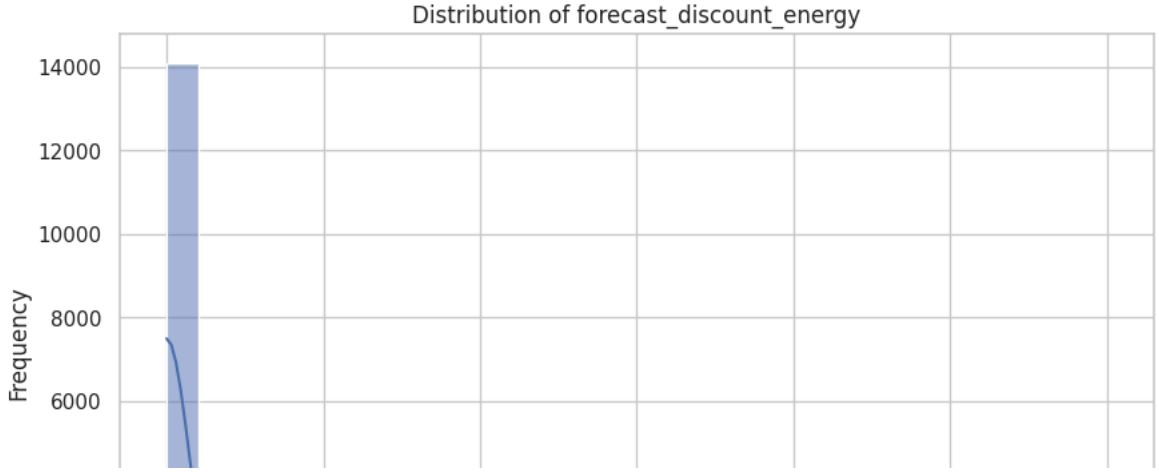

Distribution of cons_gas_12m
Frequency
cons_gas_12m
1e6

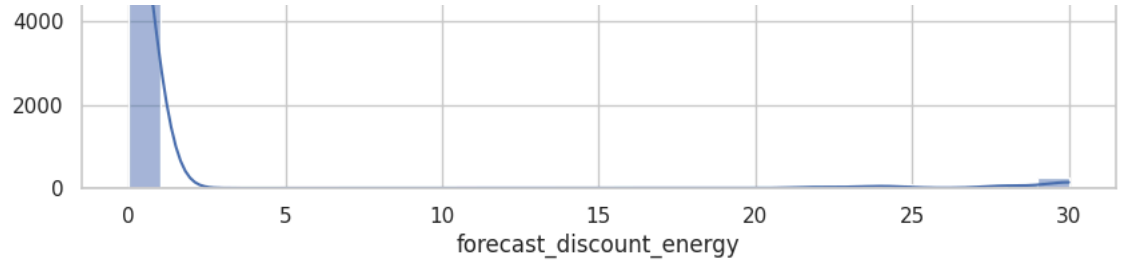

Distribution of cons_last_month
Frequency

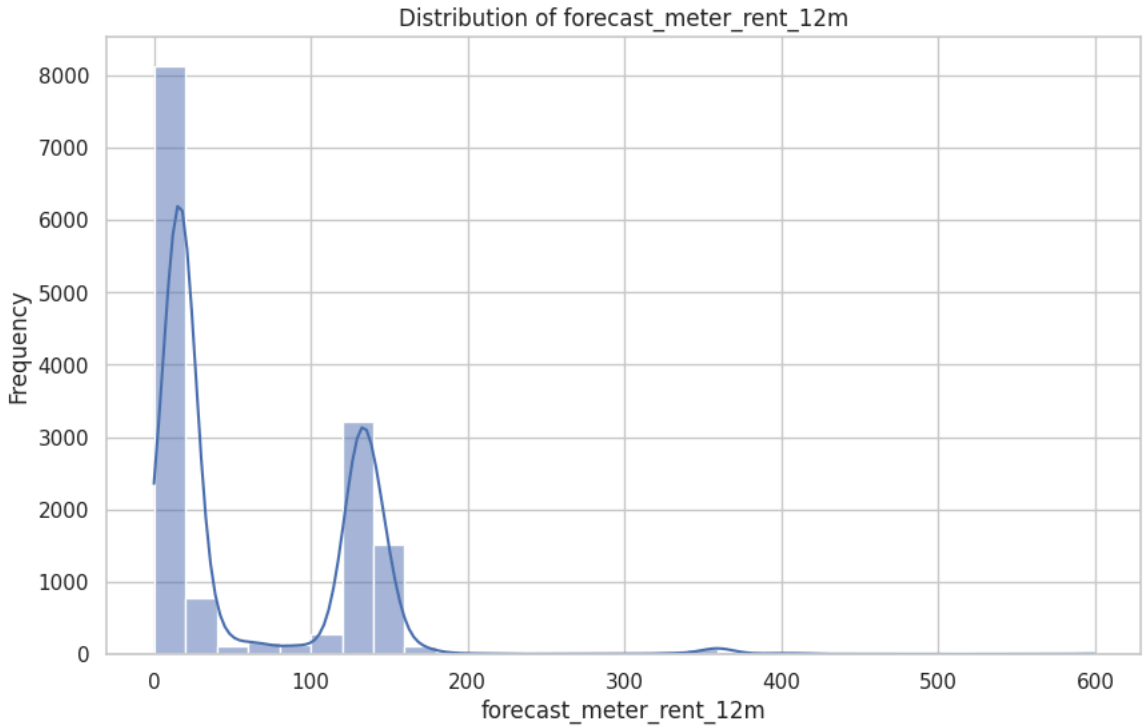

0
0
100000
200000
300000
400000
500000
600000
700000
800000
cons_last_month

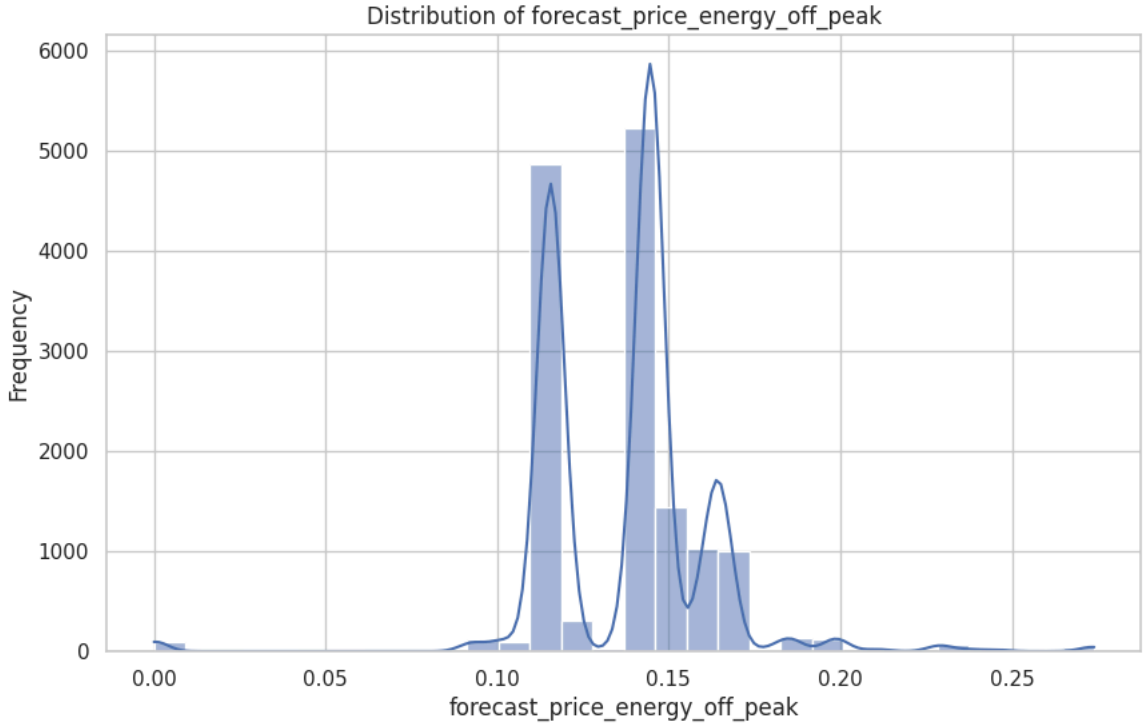

Distribution of forecast_cons_12m
Frequency
16000
14000
12000
10000
8000
6000
4000
2000
0
0
20000
40000
60000
80000
forecast_cons_12m

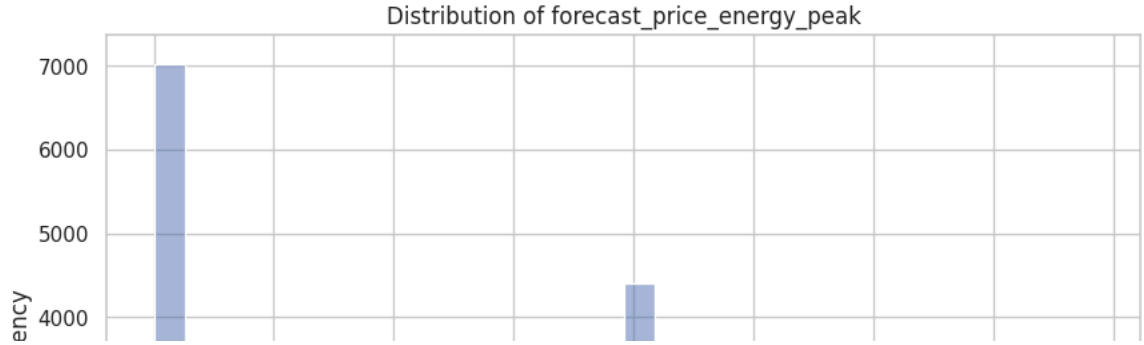

Distribution of forecast_cons_year
Frequency
40000
35000
30000
25000
20000
15000
10000
5000
0
0
25000
50000
75000
100000
125000
150000
175000
forecast_cons_year

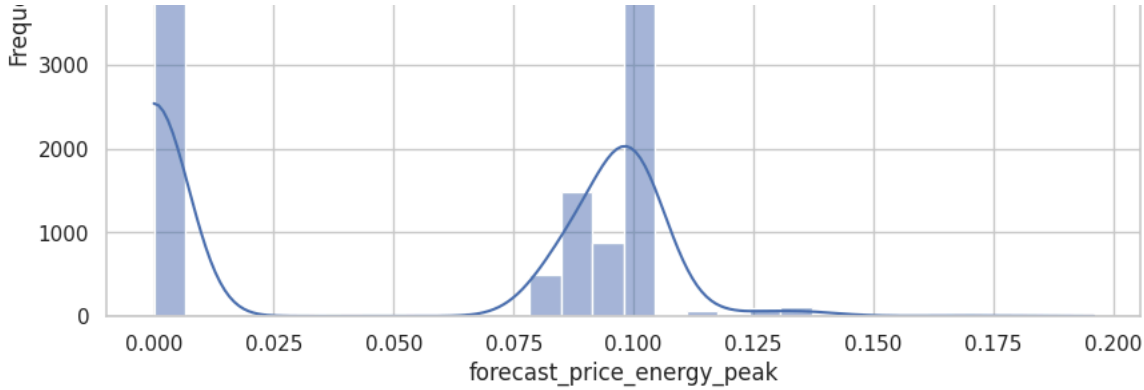

Distribution of forecast_discount_energy
Frequency
14000
12000
10000
8000
6000

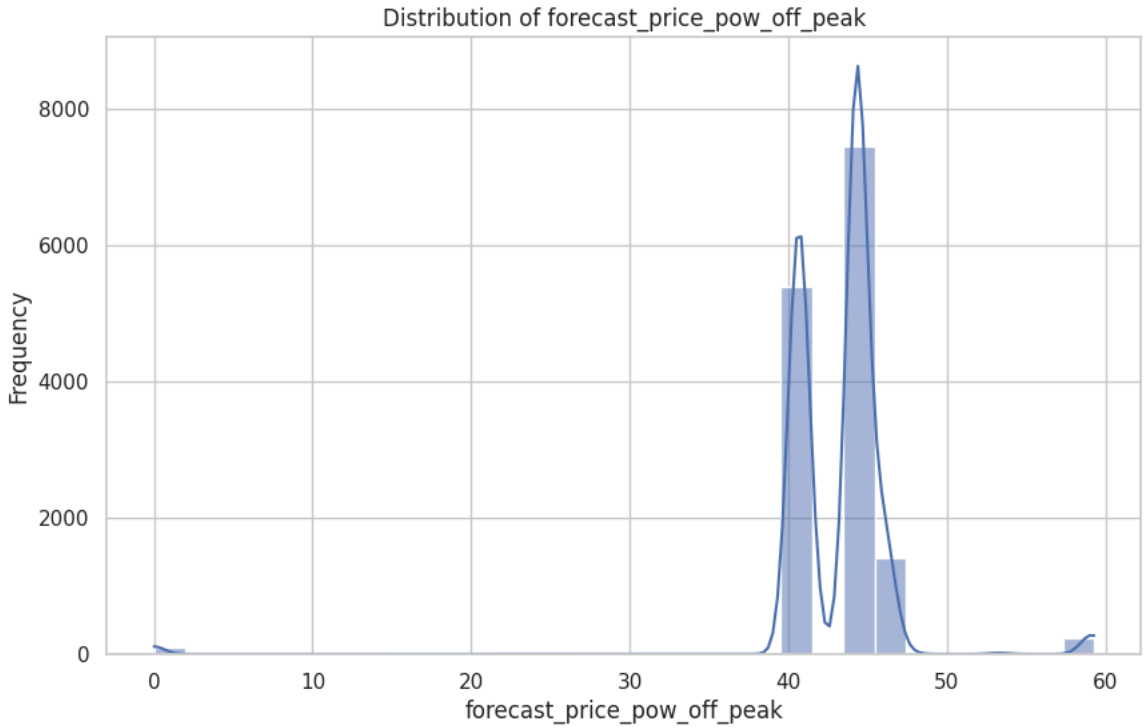

4000
2000
0
0
5
10
15
20
25
30
forecast_discount_energy

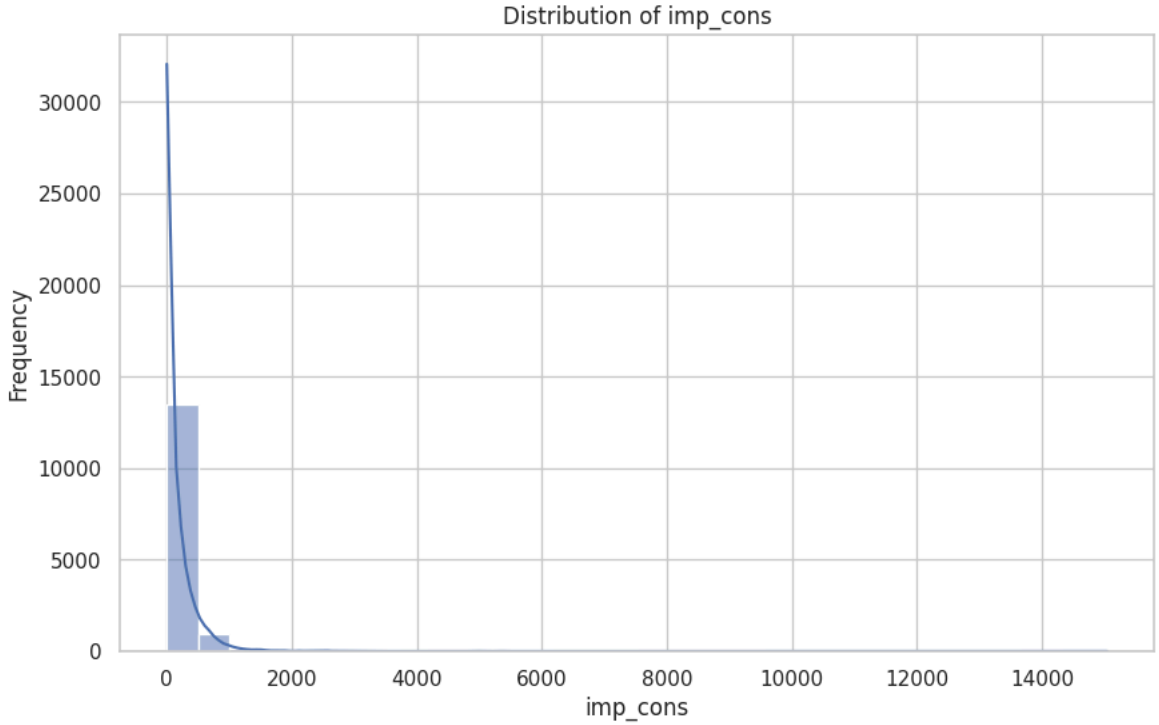

Distribution of forecast_meter_rent_12m
Frequency
8000
7000
6000
5000
4000
3000
2000
1000
0
0
100
200
300
400
500
600
forecast_meter_rent_12m

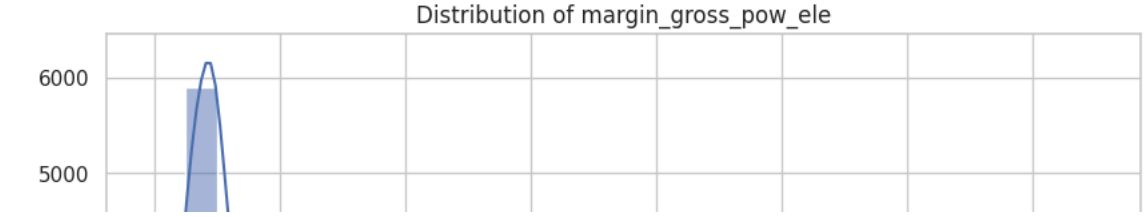

Distribution of forecast_price_energy_off_peak
Frequency
6000
5000
4000
3000
2000
1000
0
0.00
0.05
0.10
0.15
0.20
0.25
forecast_price_energy_off_peak


Distribution of forecast_price_energy_peak
7000
6000
5000
4000


Frequ
3000
2000
1000
0
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
forecast_price_energy_peak


Distribution of forecast_price_pow_off_peak
Frequency
8000
6000
4000
2000
0
0
10
20
30
40
50
60
forecast_price_pow_off_peak


Distribution of imp_cons
Frequency
30000
25000
20000
15000
10000
5000
0
0
2000
4000
6000
8000
10000
12000
14000
imp_cons


Distribution of margin_gross_pow_ele
6000
5000