In [1]:
```python
##Setting directory
import os
os.chdir('C:/Users/TOTAGOUSER4/Documents/Totago Technologies/Data Science/Projects/BCG')
```

Datasets

In [2]:
```python
import pandas as pd
cust_data = pd.read_csv('ml_case_training_data.csv')
price_data = pd.read_csv('ml_case_training_hist_data.csv')
churn_data = pd.read_csv('ml_case_training_output.csv')
```

In [3]:
```python
#Customer Dataset
cust_data.head(10)
```

Out[3]:

	id	activity_new	campaign_disc_ele	channel_sales	cons_12m
0	48ada52261e7cf58715202705a0451c9	esoiiifxdlbkcsluxmfuacbdckommixw	NaN	lmkebamcaaclubfxadlmueccxoimlema	309275
1	24011ae4ebbe3035111d65fa7c15bc57	NaN	NaN	foosdfpfkusacimwkcsosbicdxkicaua	0
2	d29c2c54acc38ff3c0614d0a653813dd	NaN	NaN	NaN	4660
3	764c75f661154dac3a6c254cd082ea7d	NaN	NaN	foosdfpfkusacimwkcsosbicdxkicaua	544
4	bba03439a292a1e166f80264c16191cb	NaN	NaN	lmkebamcaaclubfxadlmueccxoimlema	1584
5	568bb38a1afd7c0fc49c77b3789b59a3	sfisfxfcocfpcmckuekokxuseixdaoeu	NaN	foosdfpfkusacimwkcsosbicdxkicaua	121335
6	149d57cf92fc41cf94415803a877cb4b	NaN	NaN	NaN	4425
7	1aa498825382410b098937d65c4ec26d	NaN	NaN	usilxuppasemubllopkaafesmlibmsdf	8302
8	7ab4bf4878d8f7661dfc20e9b8e18011	sscfoipxikopfskekuobeuxkxmwsuucb	NaN	foosdfpfkusacimwkcsosbicdxkicaua	45097
9	01495c955be7ec5e7f3203406785aae0	NaN	NaN	foosdfpfkusacimwkcsosbicdxkicaua	29552

10 rows × 32 columns

In [4]:
```python
#Pricing Data
price_data.head(10)
```

Out[4]:

	id	price_date	price_p1_var	price_p2_var	price_p3_var	price_p1_fix	price_p2_fix	price_p3_fix
0	038af19179925da21a25619c5a24b745	2015-01-01	0.151367	0.0	0.0	44.266931	0.0	0.0
1	038af19179925da21a25619c5a24b745	2015-02-01	0.151367	0.0	0.0	44.266931	0.0	0.0
2	038af19179925da21a25619c5a24b745	2015-03-01	0.151367	0.0	0.0	44.266931	0.0	0.0
3	038af19179925da21a25619c5a24b745	2015-04-01	0.149626	0.0	0.0	44.266931	0.0	0.0
4	038af19179925da21a25619c5a24b745	2015-05-01	0.149626	0.0	0.0	44.266931	0.0	0.0
5	038af19179925da21a25619c5a24b745	2015-06-01	0.149626	0.0	0.0	44.266930	0.0	0.0
6	038af19179925da21a25619c5a24b745	2015-07-01	0.150321	0.0	0.0	44.444710	0.0	0.0
7	038af19179925da21a25619c5a24b745	2015-08-01	0.145859	0.0	0.0	44.444710	0.0	0.0
8	038af19179925da21a25619c5a24b745	2015-09-01	0.145859	0.0	0.0	44.444710	0.0	0.0
9	038af19179925da21a25619c5a24b745	2015-10-01	0.145859	0.0	0.0	44.444710	0.0	0.0

In [5]:
```python
#Churn data
churn_data.head(10)
```

Out[5]:

	id	churn
0	48ada52261e7cf58715202705a0451c9	0
1	24011ae4ebbe3035111d65fa7c15bc57	1
2	d29c2c54acc38ff3c0614d0a653813dd	0
3	764c75f661154dac3a6c254cd082ea7d	0
4	bba03439a292a1e166f80264c16191cb	0
5	568bb38a1afd7c0fc49c77b3789b59a3	0
6	149d57cf92fc41cf94415803a877cb4b	0
7	1aa498825382410b098937d65c4ec26d	1
8	7ab4bf4878d8f7661dfc20e9b8e18011	1
9	01495c955be7ec5e7f3203406785aae0	0

Data Cleaning

Dealing with missing data

In [6]:
```python
#Checking for null values in Customer Details Dataset
cust_data.isnull().sum()
```

Out[6]:
```
id                              0
activity_new                 9545
campaign_disc_ele           16096
channel_sales                4218
cons_12m                        0
cons_gas_12m                    0
cons_last_month                 0
date_activ                      0
date_end                        2
date_first_activ            12588
date_modif_prod               157
date_renewal                   40
forecast_base_bill_ele      12588
forecast_base_bill_year     12588
forecast_bill_12m           12588
forecast_cons               12588
forecast_cons_12m               0
forecast_cons_year              0
forecast_discount_energy      126
forecast_meter_rent_12m         0
forecast_price_energy_p1      126
forecast_price_energy_p2      126
forecast_price_pow_p1         126
has_gas                         0
imp_cons                        0
margin_gross_pow_ele           13
margin_net_pow_ele             13
nb_prod_act                     0
net_margin                     15
num_years_antig                 0
origin_up                      87
pow_max                         3
dtype: int64
```

Columns with null values - activity_new, campaign_disc_ele, channel_sales, date_end, date_first_activ, date_modif_prod, date_renewal, forecast_base_bill_ele, forecast_base_bill_year, forecast_bill_12m, forecast_cons, forecast_discount_energy, forecast_price_energy_p1, forecast_price_energy_p2, forecast_price_pow_p1, margin_gross_pow_ele, margin_net_pow_ele, net_margin, origin_up, pow_max

In [7]:
```python
#Checking for null values in Price Dataset
price_data.isnull().sum()
```

Out[7]:
```
id                0
price_date        0
price_p1_var   1359
price_p2_var   1359
price_p3_var   1359
price_p1_fix   1359
price_p2_fix   1359
price_p3_fix   1359
dtype: int64
```

Columns with missing data - price_p1_var, price_p2_var, price_p3_var, price_p1_fix, price_p2_fix, price_p3_fix

In [8]:
```python
#Checking for null values in Churn Dataset
churn_data.isnull().sum()
```

Out[8]:
```
id       0
churn    0
dtype: int64
```

No nulls here

In [9]:
```python
# Filling up nulls in Customer Details Dataset

#Filling with zeros
for col in ['date_end', 'date_first_activ', 'date_modif_prod', 'date_renewal', 'forecast_base_bill_ele'
, 'forecast_base_bill_year', 'forecast_bill_12m', 'forecast_cons', 'forecast_discount_energy', 'forecas
t_price_energy_p1', 'forecast_price_energy_p2', 'forecast_price_pow_p1', 'margin_gross_pow_ele', 'margi
n_net_pow_ele', 'net_margin', 'pow_max']:
    cust_data[col].fillna(0, inplace = True)

#Filling with null keyword
for col in ['channel_sales', 'origin_up']:
    cust_data[col].fillna('Null', inplace = True)

#Dropping Some columns
cust_data.drop(columns=['activity_new', 'campaign_disc_ele', ], inplace=True)

cust_data.head(10)
```

Out[9]:

	id	channel_sales	cons_12m	cons_gas_12m	cons_last_month	date_activ	date_e
0	48ada52261e7cf58715202705a0451c9	lmkebamcaaclubfxadlmueccxoimlema	309275	0	10025	2012-11-07	2016-1
1	24011ae4ebbe3035111d65fa7c15bc57	foosdfpfkusacimwkcsosbicdxkicaua	0	54946	0	2013-06-15	2016-0
2	d29c2c54acc38ff3c0614d0a653813dd	Null	4660	0	0	2009-08-21	2016-0
3	764c75f661154dac3a6c254cd082ea7d	foosdfpfkusacimwkcsosbicdxkicaua	544	0	0	2010-04-16	2016-0
4	bba03439a292a1e166f80264c16191cb	lmkebamcaaclubfxadlmueccxoimlema	1584	0	0	2010-03-30	2016-0
5	568bb38a1afd7c0fc49c77b3789b59a3	foosdfpfkusacimwkcsosbicdxkicaua	121335	0	12400	2010-04-08	2016-0
6	149d57cf92fc41cf94415803a877cb4b	Null	4425	0	526	2010-01-13	2016-0
7	1aa498825382410b098937d65c4ec26d	usilxuppasemubllopkaafesmlibmsdf	8302	0	1998	2011-12-09	2016-1
8	7ab4bf4878d8f7661dfc20e9b8e18011	foosdfpfkusacimwkcsosbicdxkicaua	45097	0	0	2011-12-02	2016-1
9	01495c955be7ec5e7f3203406785aae0	foosdfpfkusacimwkcsosbicdxkicaua	29552	0	1260	2010-04-21	2016-0

10 rows × 30 columns

In [10]:
```python
# Filling up nulls in Price Dataset

for col in ['price_p1_var', 'price_p2_var', 'price_p3_var', 'price_p1_fix', 'price_p2_fix', 'price_p3_f
ix']:
    price_data[col].fillna(0, inplace = True)

price_data.head(10)
```

Out[10]:

	id	price_date	price_p1_var	price_p2_var	price_p3_var	price_p1_fix	price_p2_fix	price_p3_fix
0	038af19179925da21a25619c5a24b745	2015-01-01	0.151367	0.0	0.0	44.266931	0.0	0.0
1	038af19179925da21a25619c5a24b745	2015-02-01	0.151367	0.0	0.0	44.266931	0.0	0.0
2	038af19179925da21a25619c5a24b745	2015-03-01	0.151367	0.0	0.0	44.266931	0.0	0.0
3	038af19179925da21a25619c5a24b745	2015-04-01	0.149626	0.0	0.0	44.266931	0.0	0.0
4	038af19179925da21a25619c5a24b745	2015-05-01	0.149626	0.0	0.0	44.266931	0.0	0.0
5	038af19179925da21a25619c5a24b745	2015-06-01	0.149626	0.0	0.0	44.266930	0.0	0.0
6	038af19179925da21a25619c5a24b745	2015-07-01	0.150321	0.0	0.0	44.444710	0.0	0.0
7	038af19179925da21a25619c5a24b745	2015-08-01	0.145859	0.0	0.0	44.444710	0.0	0.0
8	038af19179925da21a25619c5a24b745	2015-09-01	0.145859	0.0	0.0	44.444710	0.0	0.0
9	038af19179925da21a25619c5a24b745	2015-10-01	0.145859	0.0	0.0	44.444710	0.0	0.0

In [11]:
```python
#Cross checking nulls and checking variable types for Customer Details Dataset
cust_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16096 entries, 0 to 16095
Data columns (total 30 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   id                        16096 non-null  object
 1   channel_sales             16096 non-null  object
 2   cons_12m                  16096 non-null  int64
 3   cons_gas_12m              16096 non-null  int64
 4   cons_last_month           16096 non-null  int64
 5   date_activ                16096 non-null  object
 6   date_end                  16096 non-null  object
 7   date_first_activ          16096 non-null  object
 8   date_modif_prod           16096 non-null  object
 9   date_renewal              16096 non-null  object
 10  forecast_base_bill_ele    16096 non-null  float64
 11  forecast_base_bill_year   16096 non-null  float64
 12  forecast_bill_12m         16096 non-null  float64
 13  forecast_cons             16096 non-null  float64
 14  forecast_cons_12m         16096 non-null  float64
 15  forecast_cons_year        16096 non-null  int64
 16  forecast_discount_energy  16096 non-null  float64
 17  forecast_meter_rent_12m   16096 non-null  float64
 18  forecast_price_energy_p1  16096 non-null  float64
 19  forecast_price_energy_p2  16096 non-null  float64
 20  forecast_price_pow_p1     16096 non-null  float64
 21  has_gas                   16096 non-null  object
 22  imp_cons                  16096 non-null  float64
 23  margin_gross_pow_ele      16096 non-null  float64
 24  margin_net_pow_ele        16096 non-null  float64
 25  nb_prod_act               16096 non-null  int64
 26  net_margin                16096 non-null  float64
 27  num_years_antig           16096 non-null  int64
 28  origin_up                 16096 non-null  object
 29  pow_max                   16096 non-null  float64
dtypes: float64(15), int64(6), object(9)
memory usage: 3.7+ MB
```

In [12]:
```python
#Cross checking nulls and checking variable types for price Dataset
price_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 193002 entries, 0 to 193001
Data columns (total 8 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   id            193002 non-null  object
 1   price_date    193002 non-null  object
 2   price_p1_var  193002 non-null  float64
 3   price_p2_var  193002 non-null  float64
 4   price_p3_var  193002 non-null  float64
 5   price_p1_fix  193002 non-null  float64
 6   price_p2_fix  193002 non-null  float64
 7   price_p3_fix  193002 non-null  float64
dtypes: float64(6), object(2)
memory usage: 11.8+ MB
```

In [ ]:

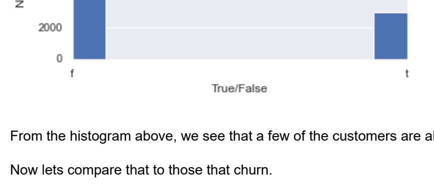
Data Analysis

In [13]:
```python
#For better visualization import matplotlib and seaborn

import matplotlib.pyplot as plt
plt.style.use('classic')
%matplotlib inline
import numpy as np
import seaborn as sns
sns.set()
```

Looking at the rate of those who are also gas clients.

In [26]:
```python
#Histogram showing number of customers that have gas

plt.hist(cust_data['has_gas'], alpha=1.0)
plt.title('Customers that are Gas Clients')
plt.ylabel('Number of Customers', fontdict = None, labelpad = None)
plt.xlabel('True/False', fontdict = None, labelpad = None)
plt.plot()
```

Out[26]: []

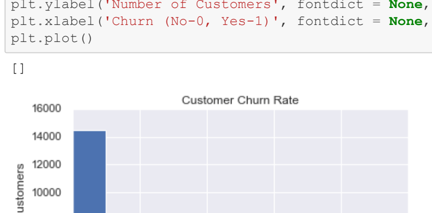


From the histogram above, we see that a few of the customers are also gas clients.

Now lets compare that to those that churn.

In [16]:
```python
plt.hist(churn_data['churn'], alpha=1.0)
plt.title('Customer Churn Rate')
plt.ylabel('Number of Customers', fontdict = None, labelpad = None)
plt.xlabel('Churn (No-0, Yes-1)', fontdict = None, labelpad = None)
plt.plot()
```

Out[16]: []



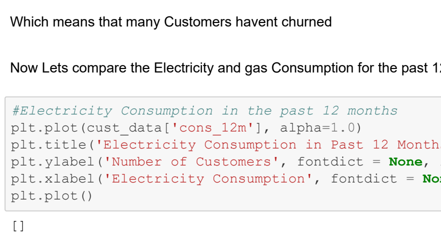
A few Customers have churned.

Which means that many Customers havent churned

Now Lets compare the Electricity and gas Consumption for the past 12 months.

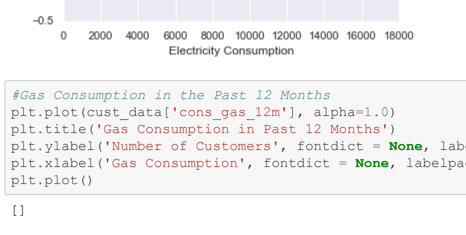
In [20]:
```python
#Electricity Consumption in the past 12 months
plt.plot(cust_data['cons_12m'], alpha=1.0)
plt.title('Electricity Consumption in Past 12 Months')
plt.ylabel('Number of Customers', fontdict = None, labelpad = None)
plt.xlabel('Electricity Consumption', fontdict = None, labelpad = None)
plt.plot()
```

Out[20]: []



In [21]:
```python
#Gas Consumption in the Past 12 Months
plt.plot(cust_data['cons_gas_12m'], alpha=1.0)
plt.title('Gas Consumption in Past 12 Months')
plt.ylabel('Number of Customers', fontdict = None, labelpad = None)
plt.xlabel('Gas Consumption', fontdict = None, labelpad = None)
plt.plot()
```

Out[21]: []



In [ ]:

In [ ]:

In [ ]:

In [ ]: