In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
#load first dataset
df_hist_data=pd.read_csv('ml_case_training_hist_data.csv')
df_hist_data.head()
```

Out[2]:

	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

In [3]:
```
#load second dataset
df_train_data=pd.read_csv('ml_case_training_data.csv')
df_train_data.head()
```

Out[3]:

	id	activity_new	campaign_disc_ele	channel_sales	cons_12m	cons_gas_1
0	48ada52261e7cf58715202705a0451c9	esoiiifxdlbkcsluxmfuacbdckommixw	NaN	lmkebamcaaclubfxadlmueccxoimlema	309275	
1	24011ae4ebbe3035111d65fa7c15bc57	NaN	NaN	foosdfpfkusacimwkcsosbicdxkicaua	0	54
2	d29c2c54acc38ff3c0614d0a653813dd	NaN	NaN	NaN	4660	
3	764c75f661154dac3a6c254cd082ea7d	NaN	NaN	foosdfpfkusacimwkcsosbicdxkicaua	544	

	id	activity_new	campaign_disc_ele	channel_sales	cons_12m	cons_gas_1
4	bba03439a292a1e166f80264c16191cb	NaN	NaN	lmkebamcaaclubfxadlmueccxoimlema	1584	

5 rows × 32 columns

Merging the two Trainning datasets

In [4]:
```
df_merged=df_hist_data.merge(df_train_data, how='left', on='id')
df_merged.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	wxemiwkumpibllw
1	038af19179925da21a25619c5a24b745	2015-02-01	0.151367	0.0	0.0	44.266931	0.0	0.0	wxemiwkumpibllw
2	038af19179925da21a25619c5a24b745	2015-03-01	0.151367	0.0	0.0	44.266931	0.0	0.0	wxemiwkumpibllw
3	038af19179925da21a25619c5a24b745	2015-04-01	0.149626	0.0	0.0	44.266931	0.0	0.0	wxemiwkumpibllw
4	038af19179925da21a25619c5a24b745	2015-05-01	0.149626	0.0	0.0	44.266931	0.0	0.0	wxemiwkumpibllw
5	038af19179925da21a25619c5a24b745	2015-06-01	0.149626	0.0	0.0	44.266930	0.0	0.0	wxemiwkumpibllw
6	038af19179925da21a25619c5a24b745	2015-07-01	0.150321	0.0	0.0	44.444710	0.0	0.0	wxemiwkumpibllw
7	038af19179925da21a25619c5a24b745	2015-08-01	0.145859	0.0	0.0	44.444710	0.0	0.0	wxemiwkumpibllw
8	038af19179925da21a25619c5a24b745	2015-09-01	0.145859	0.0	0.0	44.444710	0.0	0.0	wxemiwkumpibllw
9	038af19179925da21a25619c5a24b745	2015-10-01	0.145859	0.0	0.0	44.444710	0.0	0.0	wxemiwkumpibllw

10 rows × 39 columns

In [5]:
```
# drop irrelevent column
df_merged.drop(columns='campaign_disc_ele', inplace=True)
```

In [6]:
```
df_merged.shape
```

Out[6]:
```
(193002, 38)
```

In [7]:
```
#check missing values
df_merged.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
activity_new               114432
channel_sales               50595
cons_12m                        0
cons_gas_12m                    0
cons_last_month                 0
date_activ                      0
date_end                       21
date_first_activ           150960
date_modif_prod              1875
date_renewal                  477
forecast_base_bill_ele     150960
forecast_base_bill_year    150960
forecast_bill_12m          150960
forecast_cons              150960
forecast_cons_12m               0
forecast_cons_year              0
```

```
forecast_discount_energy      1507
forecast_meter_rent_12m          0
forecast_price_energy_p1      1507
forecast_price_energy_p2      1507
forecast_price_pow_p1         1507
has_gas                          0
imp_cons                         0
margin_gross_pow_ele           156
margin_net_pow_ele             156
nb_prod_act                      0
net_margin                     180
num_years_antig                  0
origin_up                     1042
pow_max                         36
dtype: int64
```

Replacing missing values for numerical data

In [8]:

```
df_merged['price_p1_var'].fillna(df_merged['price_p1_var'].mean(),inplace=True)
df_merged['price_p2_var'].fillna(df_merged['price_p2_var'].mean(),inplace=True)
df_merged['price_p3_var'].fillna(df_merged['price_p3_var'].mean(),inplace=True)
df_merged['price_p1_fix'].fillna(df_merged['price_p1_fix'].mean(),inplace=True)
df_merged['price_p2_fix'].fillna(df_merged['price_p2_fix'].mean(),inplace=True)
df_merged['price_p3_fix'].fillna(df_merged['price_p3_fix'].mean(),inplace=True)
df_merged['forecast_price_energy_p1'].fillna(df_merged['forecast_price_energy_p1'].mean(),inplace=True)
df_merged['forecast_price_energy_p2'].fillna(df_merged['forecast_price_energy_p2'].mean(),inplace=True)
df_merged['forecast_price_pow_p1'].fillna(df_merged['forecast_price_pow_p1'].mean(),inplace=True)
df_merged['margin_gross_pow_ele'].fillna(df_merged['margin_gross_pow_ele'].mean(),inplace=True)
df_merged['margin_net_pow_ele'].fillna(df_merged['margin_net_pow_ele'].mean(),inplace=True)
df_merged['net_margin'].fillna(df_merged['net_margin'].mean(),inplace=True)
df_merged['pow_max'].fillna(df_merged['pow_max'].mean(),inplace=True)
df_merged['forecast_discount_energy'].fillna(df_merged['forecast_discount_energy'].mean(),inplace=True)
```

Replacing missing values for categorical data

In [9]:

```
df_merged['channel_sales'].fillna(df_merged['channel_sales'].mode(),inplace=True)
df_merged['date_modif_prod'].fillna(df_merged['date_modif_prod'].mode(),inplace=True)
```

```
df_merged['date_renewal'].fillna(df_merged['date_renewal'].mode(),inplace=True)
df_merged['origin_up'].fillna(df_merged['origin_up'].mode(),inplace=True)
```

In [10]:
```
#Dropping missing values for those more than 30% missing values
df_merged=df_merged.dropna(how="any")
```

In [11]:
```
df_merged['activity_new'].value_counts().nunique()
```

Out[11]:
```
27
```

DATA EXPLORATION

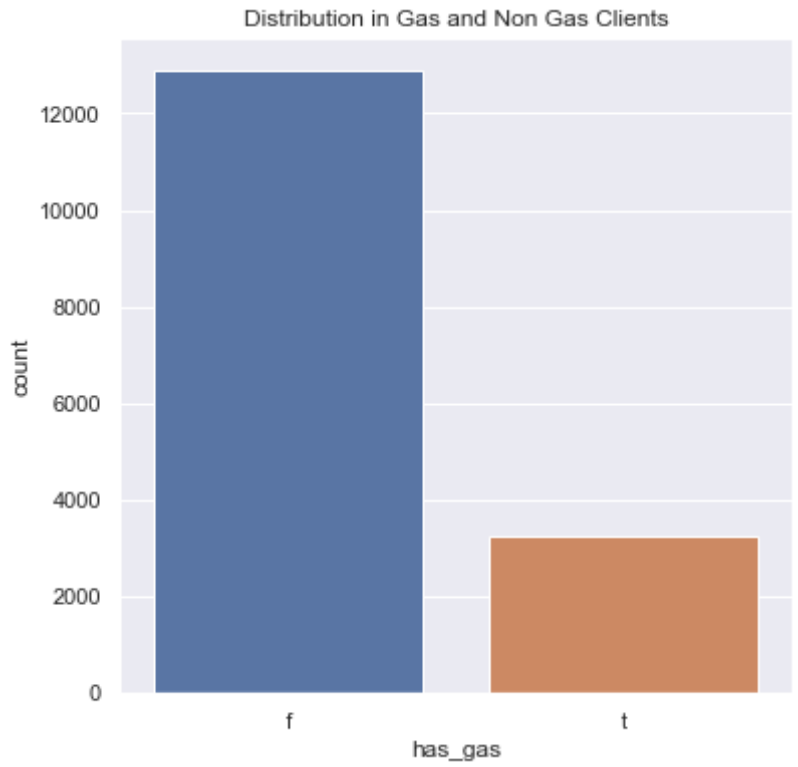
In [12]:
```
sns.set(style="darkgrid")
```

Distribution in Gas and Non Gas Clients

In [13]:
```
plt.figure(figsize=(6,6))
sns.countplot(x='has_gas',data=df_merged)
plt.title('Distribution in Gas and Non Gas Clients')


# majority of electricity clients are not gas clients too
```

Out[13]:
```
Text(0.5, 1.0, 'Distribution in Gas and Non Gas Clients')
```



In [14]:
```
## confirming data types
df_merged["date_activ"]=pd.to_datetime(df_merged["date_activ"]) #convert to datetime


dtype=df_merged['date_activ'].dtypes
print(dtype)
```

```
datetime64[ns]
```

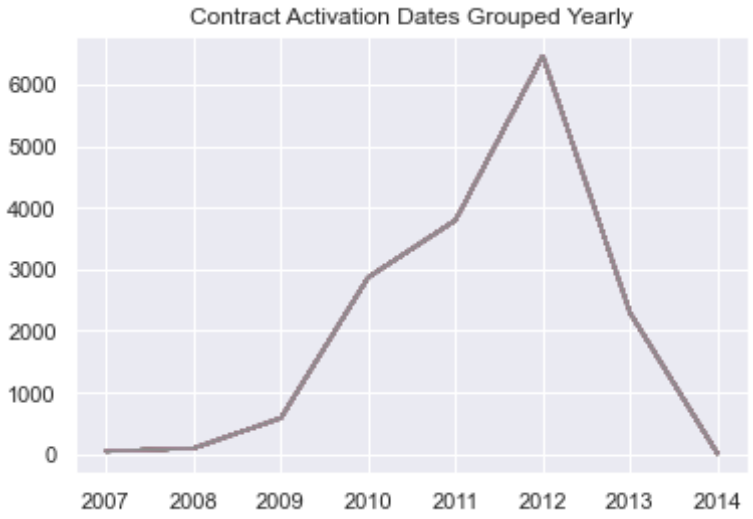
Contract Activation Dates Grouped Yearly

In [15]:
```
df_merged['year'] = pd.DatetimeIndex(df_merged['date_activ']).year
years=[year for year, df in df_merged.groupby("year")]
plt.plot(years, df_merged.groupby(["year"]).count())
```

```
plt.title('Contract Activation Dates Grouped Yearly')

#2012 had the highest number of contracts for activation
#from 2012 the numbers droped almost drastically
```

Out[15]:
```
Text(0.5, 1.0, 'Contract Activation Dates Grouped Yearly')
```

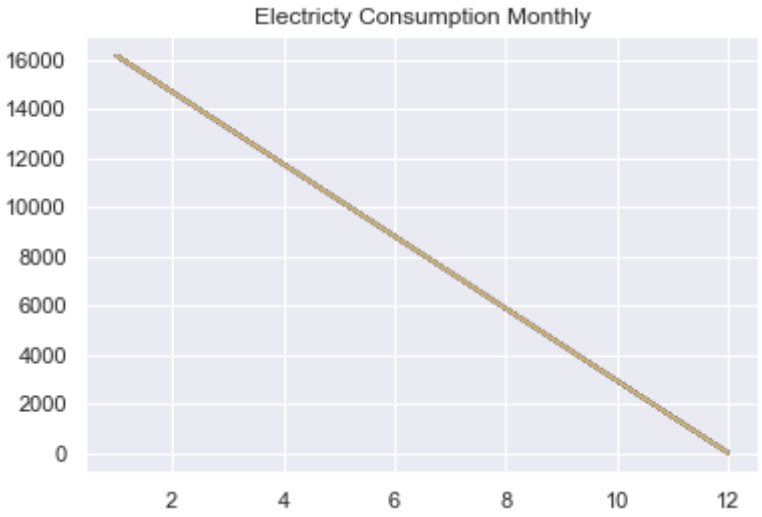


Distribution in Past 12 Months Electricty Consumption Monthly

In [16]:
```
df_merged['month_p'] = pd.DatetimeIndex(df_merged['cons_12m']).month
months_p=[month_p for month_p, df in df_merged.groupby("month_p")]
plt.plot(months_p, df_merged.groupby(["month_p"]).count())
plt.title('Electricty Consumption Monthly')

# Electricity Consumption has been decreasing from first month to the last month
```

Out[16]:
```
Text(0.5, 1.0, 'Electricty Consumption Monthly')
```



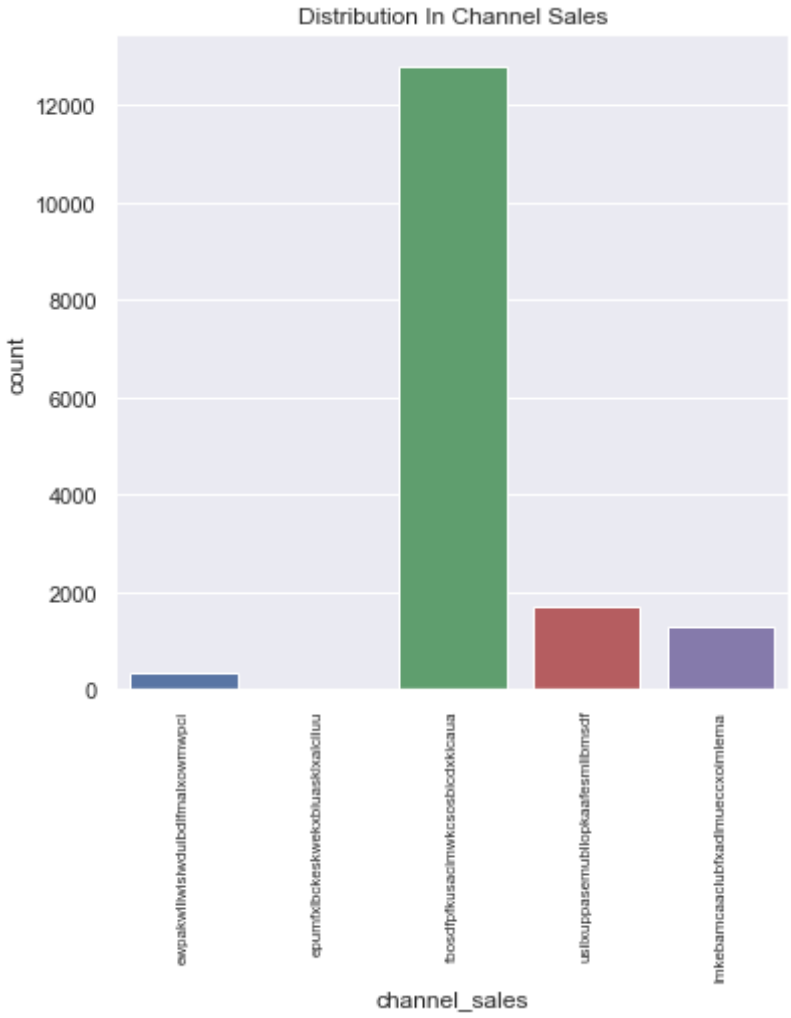
Distribution In Channel Sales

In [17]:
```
plt.figure(figsize=(6,6))

sns.countplot(x='channel_sales',data=df_merged)
plt.xticks(rotation="vertical",size=8)
plt.title('Distribution In Channel Sales')

# the third channes has the most sales
```

Out[17]:
```
Text(0.5, 1.0, 'Distribution In Channel Sales')
```



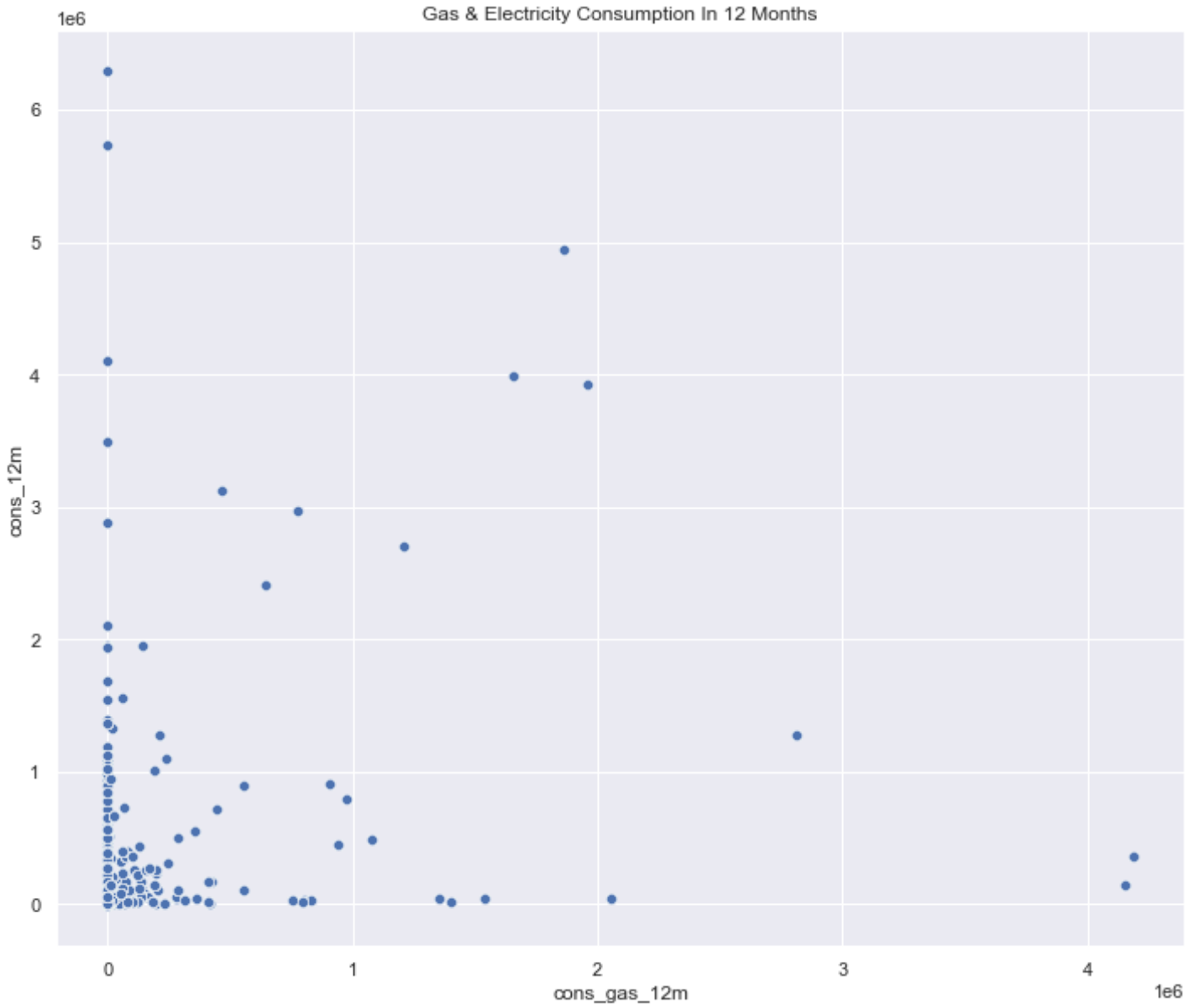
Gas & Electricity Consumption In 12 Months

In [18]:

```
plt.figure(figsize=(12,10))
axis=sns.scatterplot(x="cons_gas_12m",y="cons_12m",data=df_merged)
plt.title("Gas & Electricity Consumption In 12 Months")
```

```
# Most of gas and Electricity consumption for the last 12 months ranges below 1000000
```

Out[18]: Text(0.5, 1.0, 'Gas & Electricity Consumption In 12 Months')

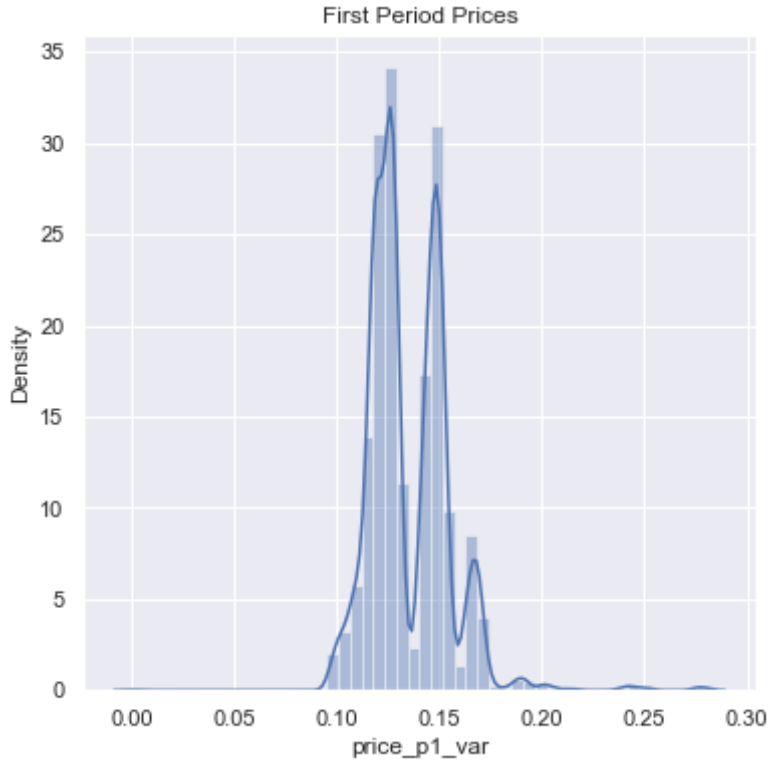


Distribution in First Period Prices

In [19]:
```
plt.figure(figsize=(6,6))
sns.distplot(df_merged['price_p1_var'])
plt.title('First Period Prices')
plt.show
```

```
C:\Users\User\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated fu
nction and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level functi
on with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[19]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



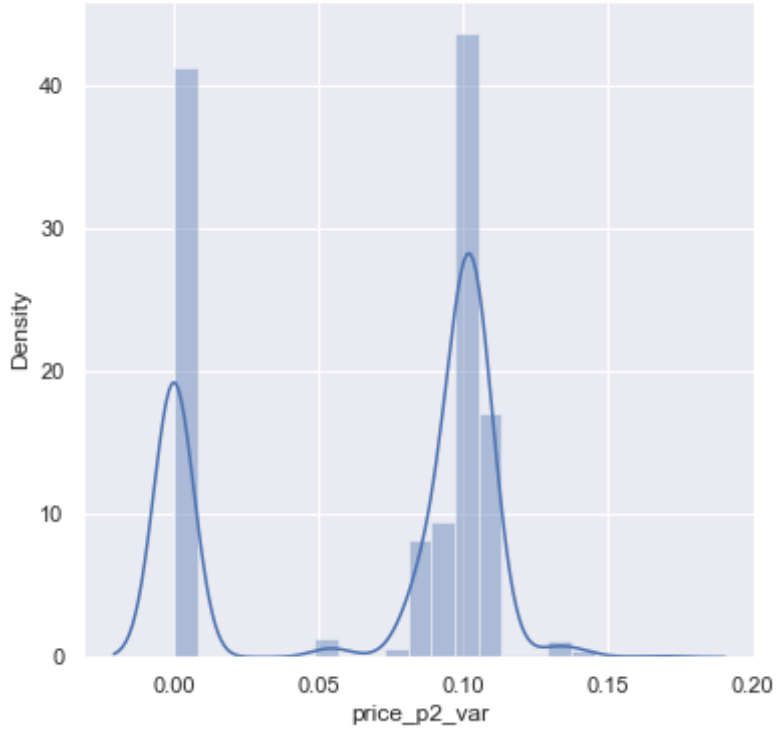
Distribution in Second Period Prices

In [20]:
```
plt.figure(figsize=(6,6))
sns.distplot(df_merged['price_p2_var'])
```

```
plt.show
```

```
C:\Users\User\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated fu
nction and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level functi
on with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[20]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



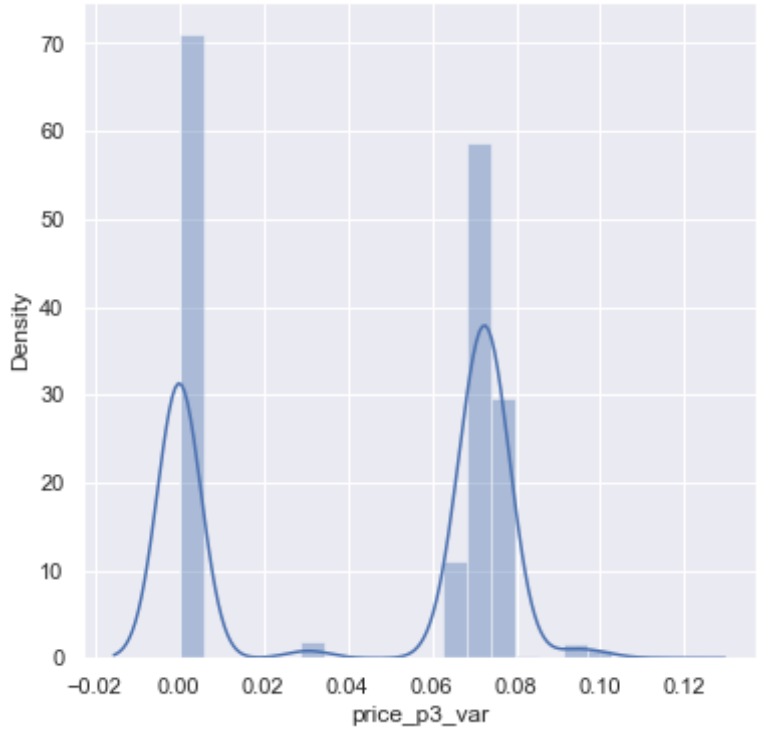
Distribution in First Period Prices

In [21]:
```
plt.figure(figsize=(6,6))
sns.distplot(df_merged['price_p3_var'])
plt.show
```

```
C:\Users\User\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated fu
```

```
nction and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level functi
on with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

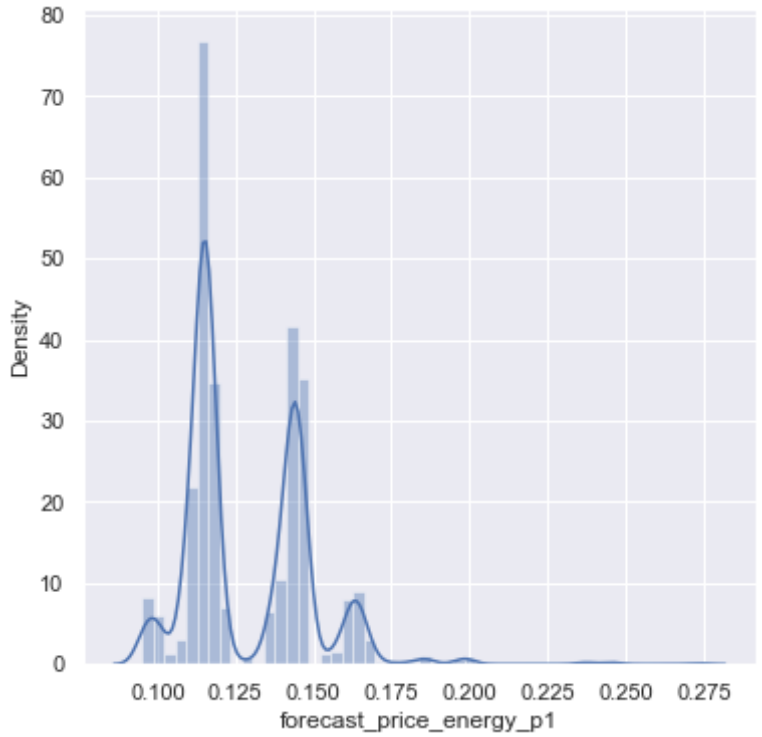
Out[21]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Distribution in Forecasted Price energy for period 1

In [22]:
```
plt.figure(figsize=(6,6))
sns.distplot(df_merged['forecast_price_energy_p1'])
plt.show()
```

```
C:\Users\User\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated fu
nction and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level functi
on with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

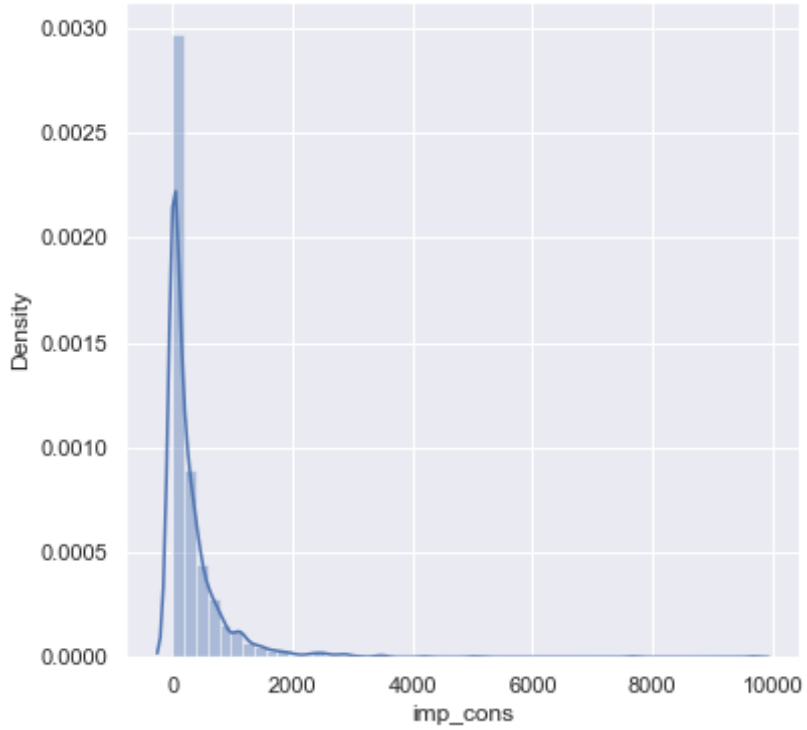


Distribution in Current Paid Consumption

In [23]:
```
plt.figure(figsize=(6,6))
sns.distplot(df_merged['imp_cons'])
plt.show
```

```
C:\Users\User\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated fu
nction and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level functi
on with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[23]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```

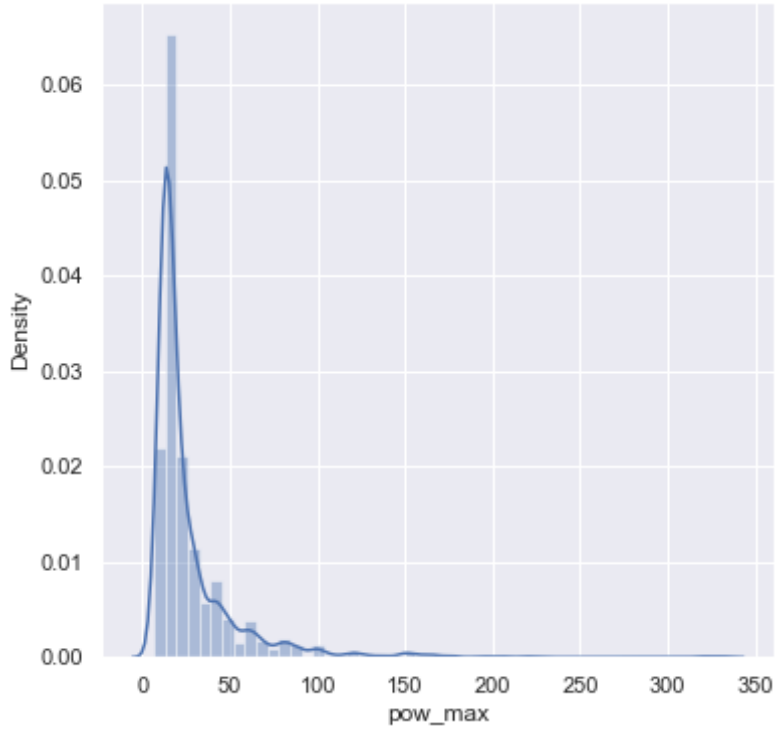


Distribution in Subscribed Power

In [24]:
```
plt.figure(figsize=(6,6))
sns.distplot(df_merged['pow_max'])
plt.show
```

```
C:\Users\User\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated fu
nction and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level functi
on with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[24]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```

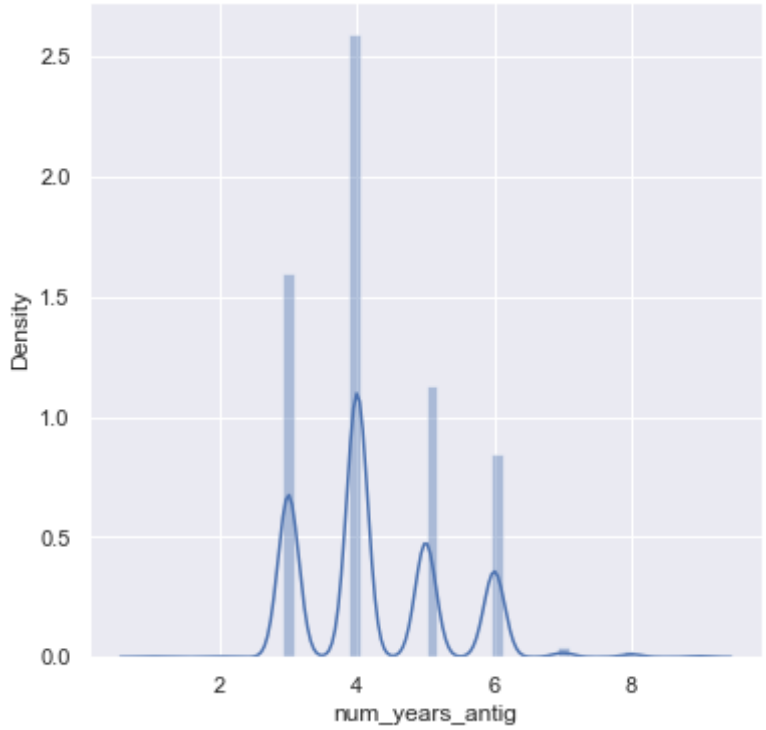


Distribution in Antiquity of Clients(Years)

In [25]:
```
plt.figure(figsize=(6,6))
sns.distplot(df_merged['num_years_antig'])
plt.show
```

```
C:\Users\User\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated fu
nction and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level functi
on with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[25]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [ ]: