In [ ]:

```

```

In [11]:

```
!pip install -q datascience
!pip install -q pandas-profiling
```

In [13]:

```
!pip install -q --upgrade pandas-profiling
```

In [ ]:

```
import pandas as pd
from pandas_profiling import ProfileReport
```

In [14]:

```
import numpy as np

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

In [15]:

```
import scipy as sp

pd.set_option('display.max_columns', None)
pd.set_option('display.max_colwidth', None)
pd.set_option('display.max_rows', None)
pd.set_option('mode.chained_assignment', None)
pd.set_option('display.float_format', lambda x: '%.5f' % x)
```

In [16]:

```
from scipy.stats import randint as sp_randint

from matplotlib.pylab import rcParams
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import r2_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score,train_test_split,GridSe
from sklearn.model_selection import RandomizedSearchCV
from sklearn.feature_selection import RFE
import statsmodels.api as sm

import warnings
warnings.filterwarnings("ignore")
```

In [17]:

```
avocado_df = pd.read_csv('avocado.csv')
```

In [18]:
```
avocado_df.head
```

Out[18]:
```
<bound method NDFrame.head of        Unnamed: 0        Date  AveragePri
ce   Total Volume           4046  \
0         0.00000  27-12-2015       1.33000    64236.62000     1036.740
00
1         1.00000  20-12-2015       1.35000    54876.98000      674.280
00
2         2.00000  13-12-2015       0.93000   118220.22000      794.700
00
3         3.00000  06-12-2015       1.08000    78992.15000     1132.000
00
4         4.00000  29-11-2015       1.28000    51039.60000      941.480
00
5         5.00000  22-11-2015       1.26000    55979.78000     1184.270
00
6         6.00000  15-11-2015       0.99000    83453.76000     1368.920
00
7         7.00000  08-11-2015       0.98000   109428.33000      703.750
00
8         8.00000  01-11-2015       1.02000    99811.42000     1022.150
```

In [21]:
```
!pip install pandas-profiling --upgrade
```

```
Requirement already satisfied: pandas-profiling in c:\users\lenovo\anacon
da3\lib\site-packages (3.2.0)
Requirement already satisfied: joblib~=1.1.0 in c:\users\lenovo\anaconda3
\lib\site-packages (from pandas-profiling) (1.1.1)
Requirement already satisfied: scipy>=1.4.1 in c:\users\lenovo\anaconda3
\lib\site-packages (from pandas-profiling) (1.10.1)
Requirement already satisfied: pandas!=1.0.0,!=1.0.1,!=1.0.2,!=1.1.0,>=0.
25.3 in c:\users\lenovo\anaconda3\lib\site-packages (from pandas-profilin
g) (1.5.3)
Requirement already satisfied: matplotlib>=3.2.0 in c:\users\lenovo\anaco
nda3\lib\site-packages (from pandas-profiling) (3.7.1)
Requirement already satisfied: pydantic>=1.8.1 in c:\users\lenovo\anacond
a3\lib\site-packages (from pandas-profiling) (2.3.0)
Requirement already satisfied: PyYAML>=5.0.0 in c:\users\lenovo\anaconda3
\lib\site-packages (from pandas-profiling) (6.0)
Requirement already satisfied: jinja2>=2.11.1 in c:\users\lenovo\anaconda
3\lib\site-packages (from pandas-profiling) (3.1.2)
Requirement already satisfied: markupsafe~=2.1.1 in c:\users\lenovo\anaco
nda3\lib\site-packages (from pandas-profiling) (2.1.1)
Requirement already satisfied: visions[type_image_path]==0.7.4 in c:\user
s\lenovo\anaconda3\lib\site-packages (from pandas-profiling) (0.7.4)
Requirement already satisfied: numpy>=1.16.0 in c:\users\lenovo\anaconda3
\lib\site-packages (from pandas-profiling) (1.24.3)
Requirement already satisfied: htmlmin>=0.1.12 in c:\users\lenovo\anacond
a3\lib\site-packages (from pandas-profiling) (0.1.12)
Requirement already satisfied: missingno>=0.4.2 in c:\users\lenovo\anacon
da3\lib\site-packages (from pandas-profiling) (0.5.2)
Requirement already satisfied: phik>=0.11.1 in c:\users\lenovo\anaconda3
\lib\site-packages (from pandas-profiling) (0.12.3)
Requirement already satisfied: tangled-up-in-unicode==0.2.0 in c:\users\l
enovo\anaconda3\lib\site-packages (from pandas-profiling) (0.2.0)
Requirement already satisfied: requests>=2.24.0 in c:\users\lenovo\anacon
da3\lib\site-packages (from pandas-profiling) (2.29.0)
Requirement already satisfied: tqdm>=4.48.2 in c:\users\lenovo\anaconda3
\lib\site-packages (from pandas-profiling) (4.65.0)
Requirement already satisfied: seaborn>=0.10.1 in c:\users\lenovo\anacond
a3\lib\site-packages (from pandas-profiling) (0.12.2)
Requirement already satisfied: multimethod>=1.4 in c:\users\lenovo\anacon
da3\lib\site-packages (from pandas-profiling) (1.9.1)
Requirement already satisfied: attrs>=19.3.0 in c:\users\lenovo\anaconda3
\lib\site-packages (from visions[type_image_path]==0.7.4->pandas-profilin
g) (22.1.0)
Requirement already satisfied: networkx>=2.4 in c:\users\lenovo\anaconda3
\lib\site-packages (from visions[type_image_path]==0.7.4->pandas-profilin
g) (2.8.4)
Requirement already satisfied: imagehash in c:\users\lenovo\anaconda3\lib
\site-packages (from visions[type_image_path]==0.7.4->pandas-profiling)
(4.3.1)
Requirement already satisfied: Pillow in c:\users\lenovo\anaconda3\lib\si
te-packages (from visions[type_image_path]==0.7.4->pandas-profiling) (9.
4.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\lenovo\anacon
da3\lib\site-packages (from matplotlib>=3.2.0->pandas-profiling) (1.0.5)
Requirement already satisfied: cycler>=0.10 in c:\users\lenovo\anaconda3
\lib\site-packages (from matplotlib>=3.2.0->pandas-profiling) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\lenovo\anaco
nda3\lib\site-packages (from matplotlib>=3.2.0->pandas-profiling) (4.25.
```

```
0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\lenovo\anaco
nda3\lib\site-packages (from matplotlib>=3.2.0->pandas-profiling) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\lenovo\anacond
a3\lib\site-packages (from matplotlib>=3.2.0->pandas-profiling) (23.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\lenovo\anacon
da3\lib\site-packages (from matplotlib>=3.2.0->pandas-profiling) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\lenovo\an
aconda3\lib\site-packages (from matplotlib>=3.2.0->pandas-profiling) (2.
8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\lenovo\anaconda3
\lib\site-packages (from pandas!=1.0.0,!=1.0.1,!=1.0.2,!=1.1.0,>=0.25.3->
pandas-profiling) (2022.7)
Requirement already satisfied: annotated-types>=0.4.0 in c:\users\lenovo
\anaconda3\lib\site-packages (from pydantic>=1.8.1->pandas-profiling) (0.
5.0)
Requirement already satisfied: pydantic-core==2.6.3 in c:\users\lenovo\an
aconda3\lib\site-packages (from pydantic>=1.8.1->pandas-profiling) (2.6.
3)
Requirement already satisfied: typing-extensions>=4.6.1 in c:\users\lenov
o\anaconda3\lib\site-packages (from pydantic>=1.8.1->pandas-profiling)
(4.6.3)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\lenov
o\anaconda3\lib\site-packages (from requests>=2.24.0->pandas-profiling)
(2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\lenovo\anaconda3
\lib\site-packages (from requests>=2.24.0->pandas-profiling) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\lenovo\a
naconda3\lib\site-packages (from requests>=2.24.0->pandas-profiling) (1.2
6.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\lenovo\anac
onda3\lib\site-packages (from requests>=2.24.0->pandas-profiling) (2023.
5.7)
Requirement already satisfied: colorama in c:\users\lenovo\anaconda3\lib
\site-packages (from tqdm>=4.48.2->pandas-profiling) (0.4.6)
Requirement already satisfied: six>=1.5 in c:\users\lenovo\anaconda3\lib
\site-packages (from python-dateutil>=2.7->matplotlib>=3.2.0->pandas-prof
iling) (1.16.0)
Requirement already satisfied: PyWavelets in c:\users\lenovo\anaconda3\li
b\site-packages (from imagehash->visions[type_image_path]==0.7.4->pandas-
profiling) (1.4.1)
```

In [ ]:

```
profile = avocado_df.profile_report(title="Avocado before Data Preprocessir
profile.to_file(output_file="Avocado_profiling_before_preprocessing.html")
```

In [23]:
```
avocado_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16468 entries, 0 to 16467
Data columns (total 14 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Unnamed: 0    1517 non-null   float64
 1   Date          1517 non-null   object 
 2   AveragePrice  1517 non-null   float64
 3   Total Volume  1517 non-null   float64
 4   4046          1517 non-null   float64
 5   4225          1517 non-null   float64
 6   4770          1517 non-null   float64
 7   Total Bags    1517 non-null   float64
 8   Small Bags    1517 non-null   float64
 9   Large Bags    1517 non-null   float64
 10  XLarge Bags   1517 non-null   float64
 11  type          1517 non-null   object 
 12  year          1517 non-null   float64
 13  region        1517 non-null   object 
dtypes: float64(11), object(3)
memory usage: 1.8+ MB
```

In [24]:
```
avocado_df.describe()
```

Out[24]:

	Unnamed: 0	AveragePrice	Total Volume	4046	4225	
count	1517.00000	1517.00000	1517.00000	1517.00000	1517.00000	1517.0
mean	26.99539	1.07499	1601879.06784	646438.65411	611437.50259	50405.4
std	14.84829	0.18889	4433142.82075	1947613.56974	1672906.16466	137781.2
min	0.00000	0.49000	38750.74000	467.72000	1783.77000	0.0
25%	14.00000	0.98000	147469.99000	20400.34000	41476.06000	911.2
50%	29.00000	1.08000	402791.86000	81751.17000	118664.89000	7688.1
75%	39.00000	1.19000	981975.08000	377578.48000	485150.34000	29167.3
max	51.00000	1.68000	44655461.51000	18933038.04000	18956479.74000	1381516.1

In [25]:
```
d = avocado_df.copy()
d.head
```

Out[25]:
```
<bound method NDFrame.head of        Unnamed: 0        Date  AveragePri
ce   Total Volume           4046  \
0         0.00000  27-12-2015       1.33000    64236.62000     1036.740
00
1         1.00000  20-12-2015       1.35000    54876.98000      674.280
00
2         2.00000  13-12-2015       0.93000   118220.22000      794.700
00
3         3.00000  06-12-2015       1.08000    78992.15000     1132.000
00
4         4.00000  29-11-2015       1.28000    51039.60000      941.480
00
5         5.00000  22-11-2015       1.26000    55979.78000     1184.270
00
6         6.00000  15-11-2015       0.99000    83453.76000     1368.920
00
7         7.00000  08-11-2015       0.98000   109428.33000      703.750
00
8         8.00000  01-11-2015       1.02000    99811.42000     1022.150
```

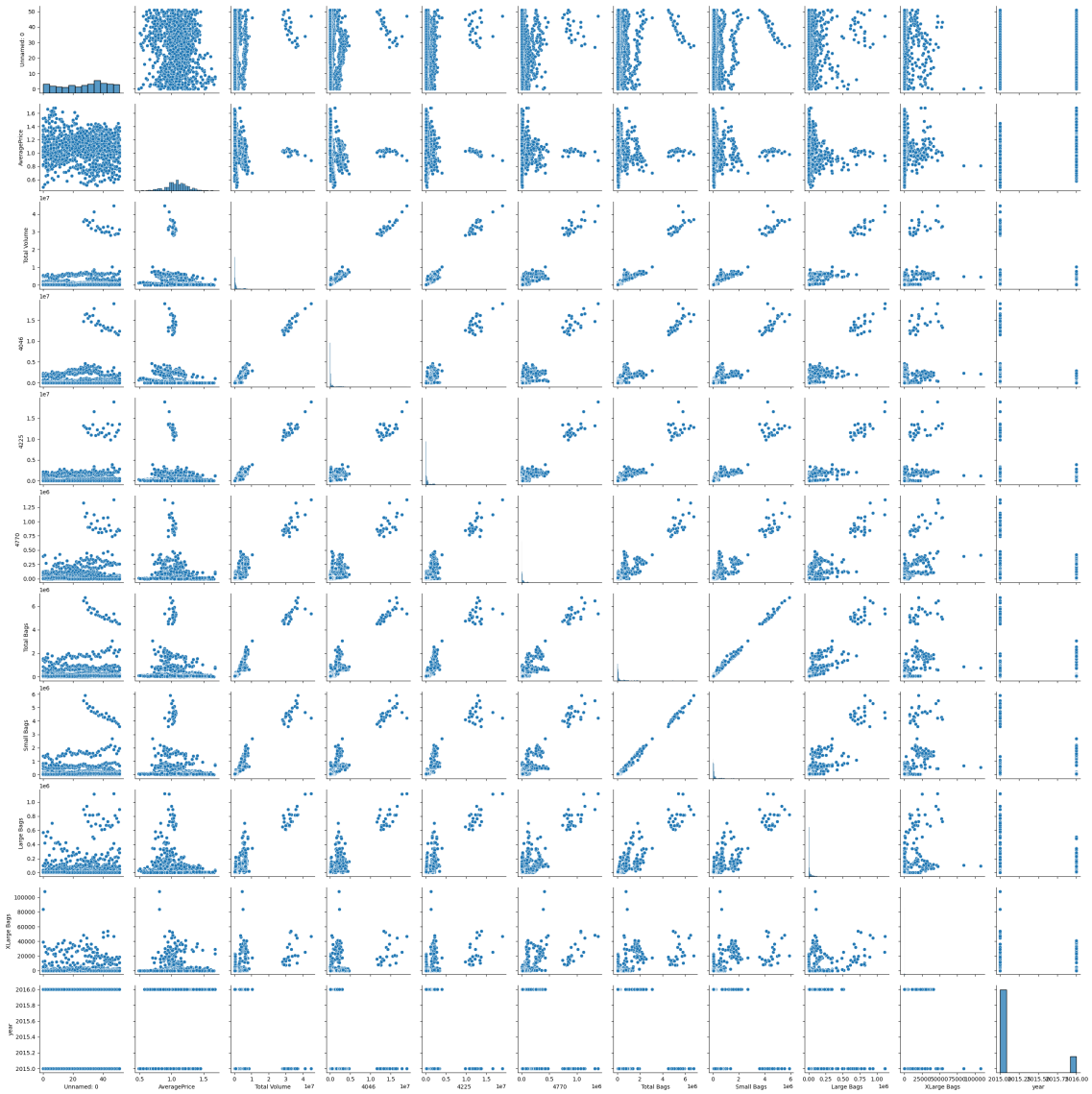
In [26]:
```
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16468 entries, 0 to 16467
Data columns (total 14 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Unnamed: 0    1517 non-null   float64
 1   Date          1517 non-null   object
 2   AveragePrice  1517 non-null   float64
 3   Total Volume  1517 non-null   float64
 4   4046          1517 non-null   float64
 5   4225          1517 non-null   float64
 6   4770          1517 non-null   float64
 7   Total Bags    1517 non-null   float64
 8   Small Bags    1517 non-null   float64
 9   Large Bags    1517 non-null   float64
 10  XLarge Bags   1517 non-null   float64
 11  type          1517 non-null   object
 12  year          1517 non-null   float64
 13  region        1517 non-null   object
dtypes: float64(11), object(3)
memory usage: 1.8+ MB
```

In [27]:
```
def datetime_to_int(dt):
    return int(dt.strftime("%Y%m%d"))

#d['Date']=d['Date'].apply(datetime_to_int)
d['Date']=pd.to_datetime(d['Date'])
```

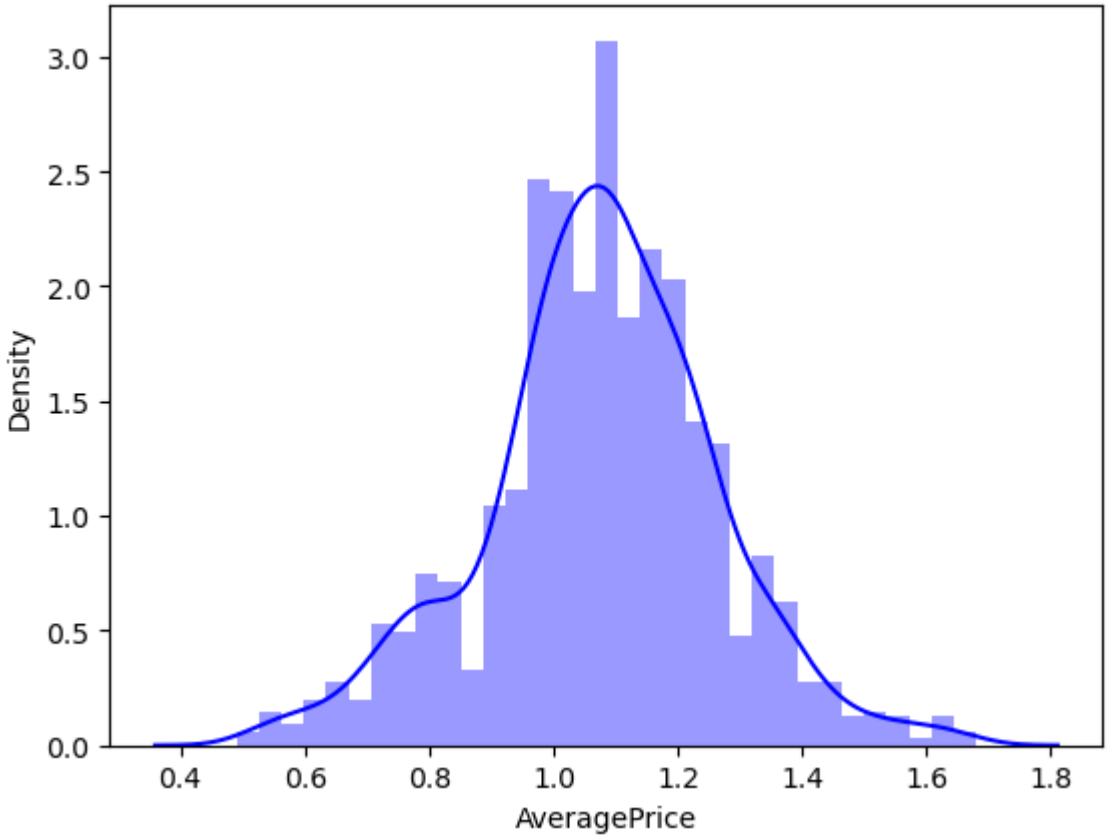
In [28]:
```
sns.pairplot(avocado_df)
```

Out[28]: <seaborn.axisgrid.PairGrid at 0x191b301d8d0>



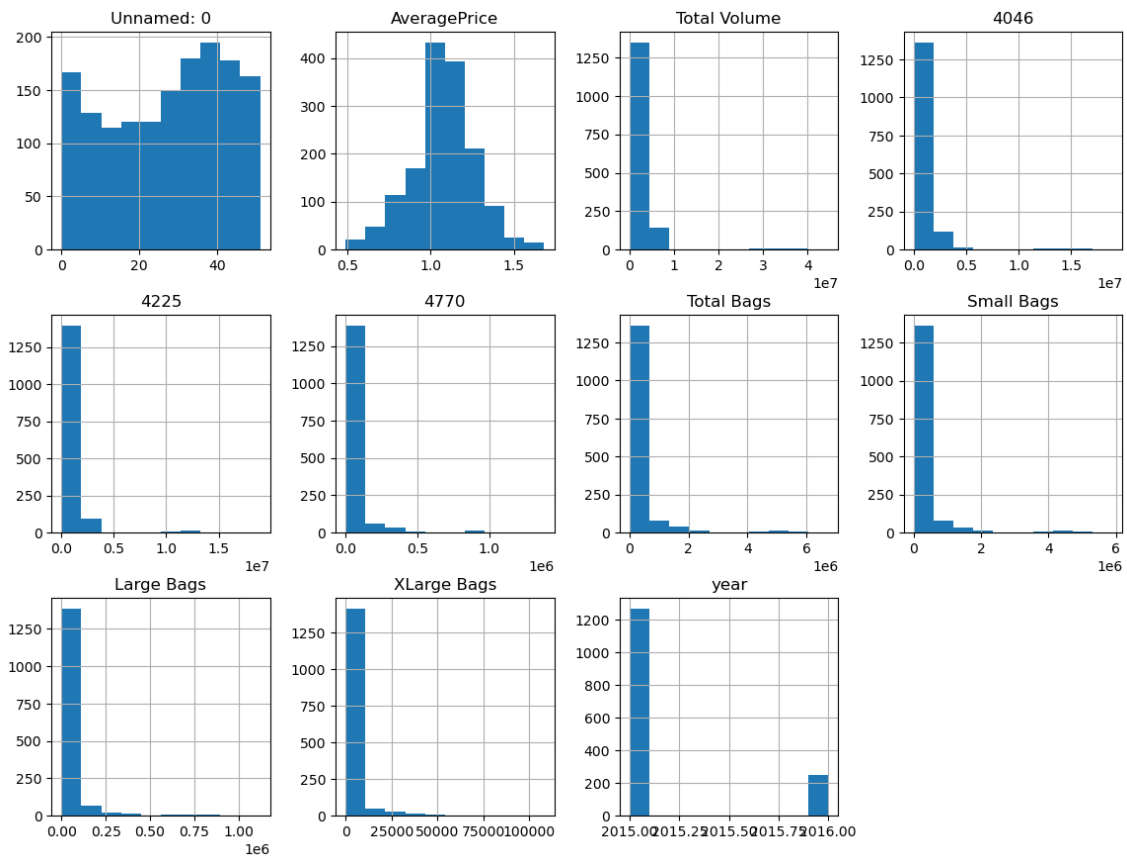
In [29]:
```
sns.distplot(d['AveragePrice'],color='b',hist=True)
```

Out[29]: <Axes: xlabel='AveragePrice', ylabel='Density'>



In [30]:
```
avocado_df.hist(figsize=(14,14),grid=True,layout=(4,4))
```

Out[30]:
```
array([[<Axes: title={'center': 'Unnamed: 0'}>,
        <Axes: title={'center': 'AveragePrice'}>,
        <Axes: title={'center': 'Total Volume'}>,
        <Axes: title={'center': '4046'}>],
       [<Axes: title={'center': '4225'}>,
        <Axes: title={'center': '4770'}>,
        <Axes: title={'center': 'Total Bags'}>,
        <Axes: title={'center': 'Small Bags'}>],
       [<Axes: title={'center': 'Large Bags'}>,
        <Axes: title={'center': 'XLarge Bags'}>,
        <Axes: title={'center': 'year'}>, <Axes: >],
       [<Axes: >, <Axes: >, <Axes: >, <Axes: >]], dtype=object)
```



In [31]:
```
d.skew()
```

Out[31]:
```
Unnamed: 0     -0.23482
AveragePrice   -0.10944
Total Volume    6.20014
4046            6.05183
4225            6.39493
4770            5.40516
Total Bags      5.36638
Small Bags      5.35518
Large Bags      5.14450
XLarge Bags     5.77508
year            1.82833
dtype: float64
```

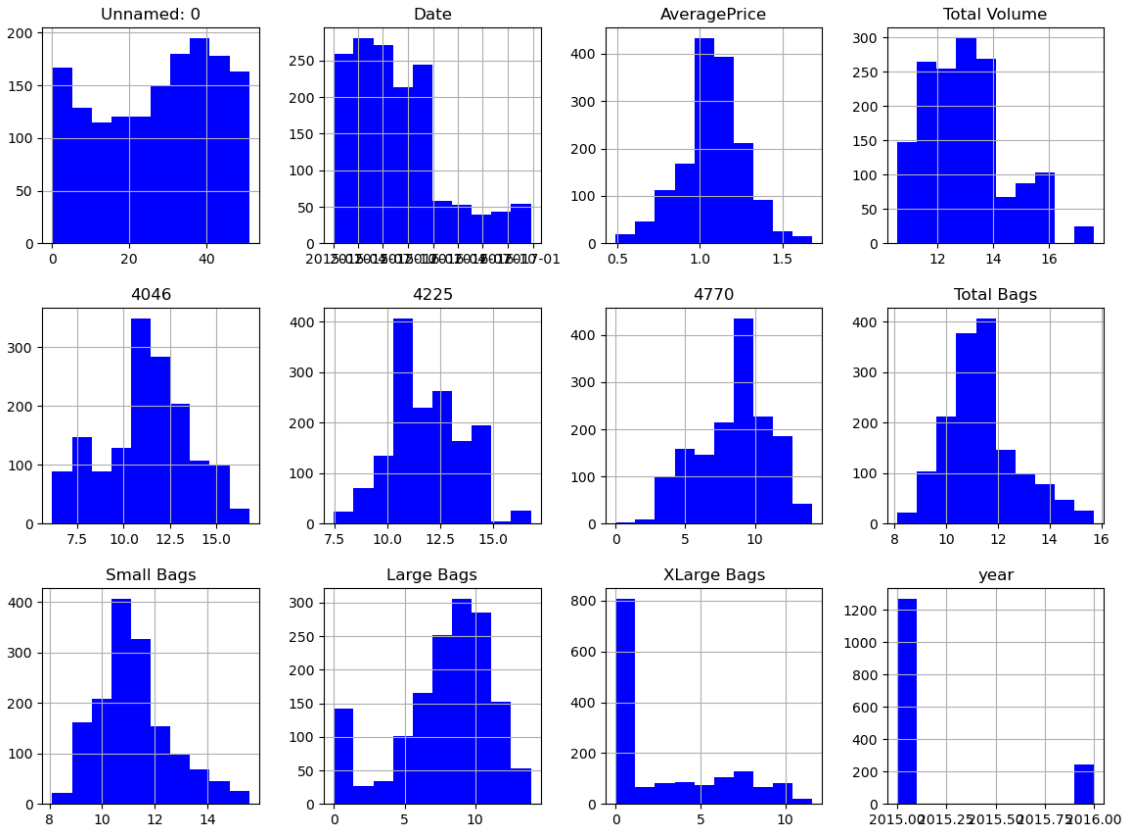
In [32]:
```
skew=('Total Volume','4046','4225','4770','Total Bags','Small Bags','Large
for col in skew :
    if d.skew().loc[col]>0.55:
        d[col]=np.log1p(d[col])
```

In [33]:
```
d.skew()
```

Out[33]:
```
Unnamed: 0     -0.23482
AveragePrice   -0.10944
Total Volume    0.66747
4046           -0.16027
4225            0.18444
4770           -0.35551
Total Bags      0.69550
Small Bags      0.71384
Large Bags     -0.91277
XLarge Bags     0.78391
year            1.82833
dtype: float64
```

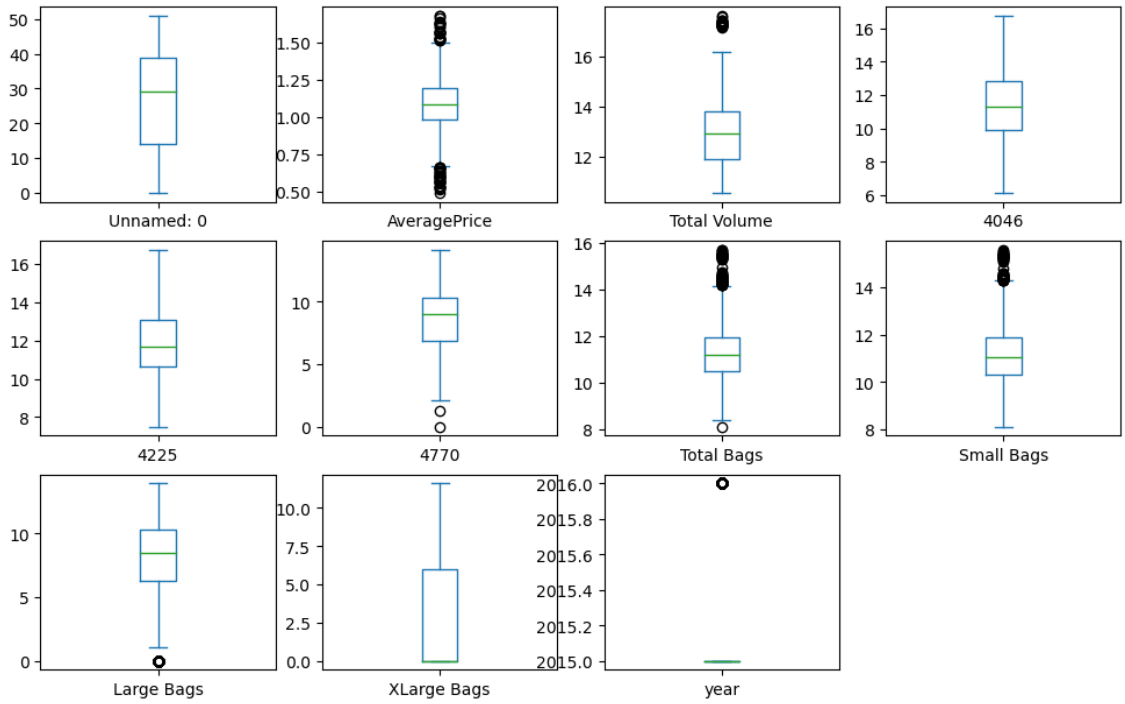
In [34]:
```
d.hist(figsize=(14,14),grid=True,layout=(4,4),color='b')
```

Out[34]:
```
array([[<Axes: title={'center': 'Unnamed: 0'}>,
        <Axes: title={'center': 'Date'}>,
        <Axes: title={'center': 'AveragePrice'}>,
        <Axes: title={'center': 'Total Volume'}>],
       [<Axes: title={'center': '4046'}>,
        <Axes: title={'center': '4225'}>,
        <Axes: title={'center': '4770'}>,
        <Axes: title={'center': 'Total Bags'}>],
       [<Axes: title={'center': 'Small Bags'}>,
        <Axes: title={'center': 'Large Bags'}>,
        <Axes: title={'center': 'XLarge Bags'}>,
        <Axes: title={'center': 'year'}>],
       [<Axes: >, <Axes: >, <Axes: >, <Axes: >]], dtype=object)
```



In [35]:
```
d.plot(kind='box',subplots=True,layout=(4,4),figsize=(12,10))
```

Out[35]:
```
Unnamed: 0         Axes(0.125,0.712609;0.168478x0.167391)
AveragePrice     Axes(0.327174,0.712609;0.168478x0.167391)
Total Volume     Axes(0.529348,0.712609;0.168478x0.167391)
4046             Axes(0.731522,0.712609;0.168478x0.167391)
4225                Axes(0.125,0.511739;0.168478x0.167391)
4770             Axes(0.327174,0.511739;0.168478x0.167391)
Total Bags       Axes(0.529348,0.511739;0.168478x0.167391)
Small Bags       Axes(0.731522,0.511739;0.168478x0.167391)
Large Bags           Axes(0.125,0.31087;0.168478x0.167391)
XLarge Bags       Axes(0.327174,0.31087;0.168478x0.167391)
year              Axes(0.529348,0.31087;0.168478x0.167391)
dtype: object
```



In [36]:
```
df=d.copy()
df.drop(['Date'],axis=1,inplace=True)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16468 entries, 0 to 16467
Data columns (total 13 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Unnamed: 0    1517 non-null   float64
 1   AveragePrice  1517 non-null   float64
 2   Total Volume  1517 non-null   float64
 3   4046          1517 non-null   float64
 4   4225          1517 non-null   float64
 5   4770          1517 non-null   float64
 6   Total Bags    1517 non-null   float64
 7   Small Bags    1517 non-null   float64
 8   Large Bags    1517 non-null   float64
 9   XLarge Bags   1517 non-null   float64
 10  type          1517 non-null   object 
 11  year          1517 non-null   float64
 12  region        1517 non-null   object 
dtypes: float64(11), object(2)
memory usage: 1.6+ MB
```

In [37]:
```
from scipy.stats import zscore
z =np.abs(zscore(d['AveragePrice']))
print(z)
print(np.where(z<3))
dn=d[(z<3)]
print('Shape of New Dataframe dn:',dn.shape)
```

```
0       NaN
1       NaN
2       NaN
3       NaN
4       NaN
5       NaN
6       NaN
7       NaN
8       NaN
9       NaN
10      NaN
11      NaN
12      NaN
13      NaN
14      NaN
15      NaN
16      NaN
17      NaN
18      NaN
```

In [39]:
```
z =np.abs(zscore(dn['4225']))
print(z)
print(np.where(z<3))
dn1=dn[(z<3)]
print('Shape of New Dataframe dn1:',dn1.shape)
```

```
[]
(array([], dtype=int64),)
Shape of New Dataframe dn1: (0, 14)
```

In [40]:
```
z =np.abs(zscore(dn1['Total Bags']))
print(z)
print(np.where(z<3))
dn2=dn1[(z<3)]
print('Shape of New Dataframe dn2:',dn2.shape)
```

```
[]
(array([], dtype=int64),)
Shape of New Dataframe dn2: (0, 14)
```

In [41]:
```
z =np.abs(zscore(dn2['Small Bags']))
print(z)
print(np.where(z<3))
dn3=dn2[(z<3)]
print('Shape of New Dataframe dn3:',dn3.shape)
```

```
[]
(array([], dtype=int64),)
Shape of New Dataframe dn3: (0, 14)
```

In [42]:
```
z =np.abs(zscore(dn3['XLarge Bags']))
print(z)
print(np.where(z<3))
dn4=dn3[(z<3)]
print('Shape of New Dataframe dn4:',dn4.shape)
```

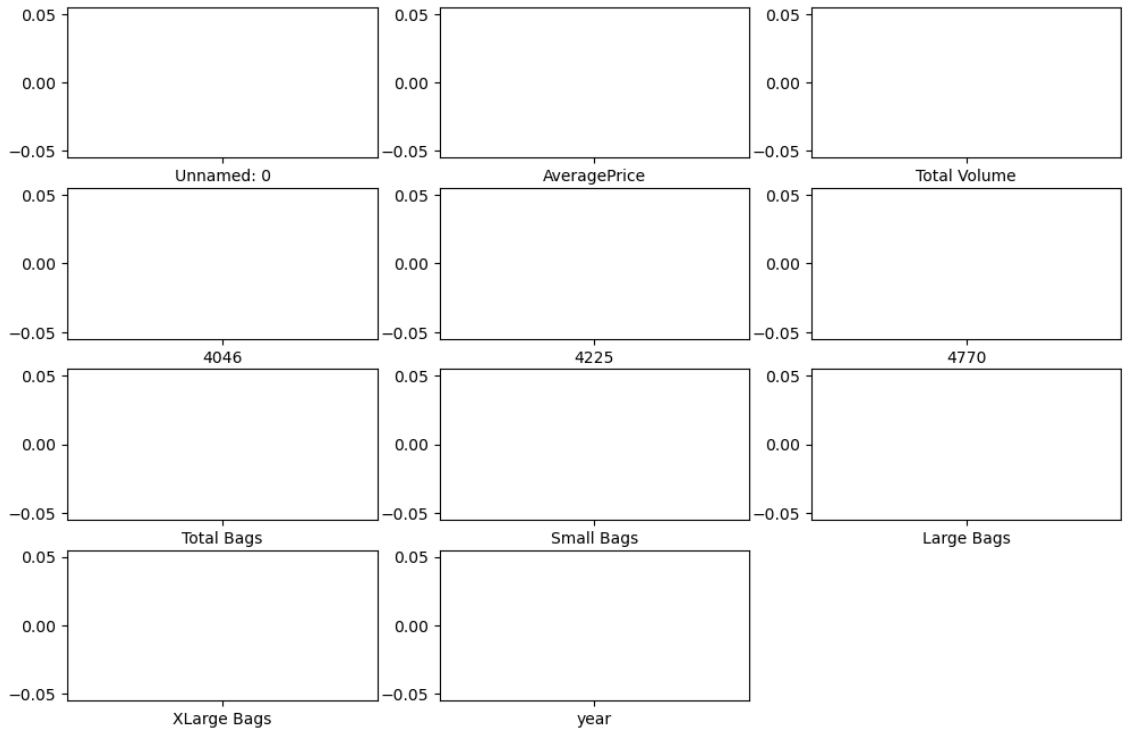
```
[]
(array([], dtype=int64),)
Shape of New Dataframe dn4: (0, 14)
```

In [43]:
```
dn4.head
```

Out[43]:
```
<bound method NDFrame.head of Empty DataFrame
Columns: [Unnamed: 0, Date, AveragePrice, Total Volume, 4046, 4225, 4770,
Total Bags, Small Bags, Large Bags, XLarge Bags, type, year, region]
Index: []>
```

In [44]:
```
dn4.plot(kind='box',subplots=True,layout=(5,3),figsize=(12,10),color='m')
```

Out[44]:
```
Unnamed: 0         Axes(0.125,0.747241;0.227941x0.132759)
AveragePrice    Axes(0.398529,0.747241;0.227941x0.132759)
Total Volume    Axes(0.672059,0.747241;0.227941x0.132759)
4046               Axes(0.125,0.587931;0.227941x0.132759)
4225            Axes(0.398529,0.587931;0.227941x0.132759)
4770            Axes(0.672059,0.587931;0.227941x0.132759)
Total Bags         Axes(0.125,0.428621;0.227941x0.132759)
Small Bags      Axes(0.398529,0.428621;0.227941x0.132759)
Large Bags      Axes(0.672059,0.428621;0.227941x0.132759)
XLarge Bags         Axes(0.125,0.26931;0.227941x0.132759)
year             Axes(0.398529,0.26931;0.227941x0.132759)
dtype: object
```



In [45]:
```
dn4.head
```

Out[45]:
```
<bound method NDFrame.head of Empty DataFrame
Columns: [Unnamed: 0, Date, AveragePrice, Total Volume, 4046, 4225, 4770,
Total Bags, Small Bags, Large Bags, XLarge Bags, type, year, region]
Index: []>
```

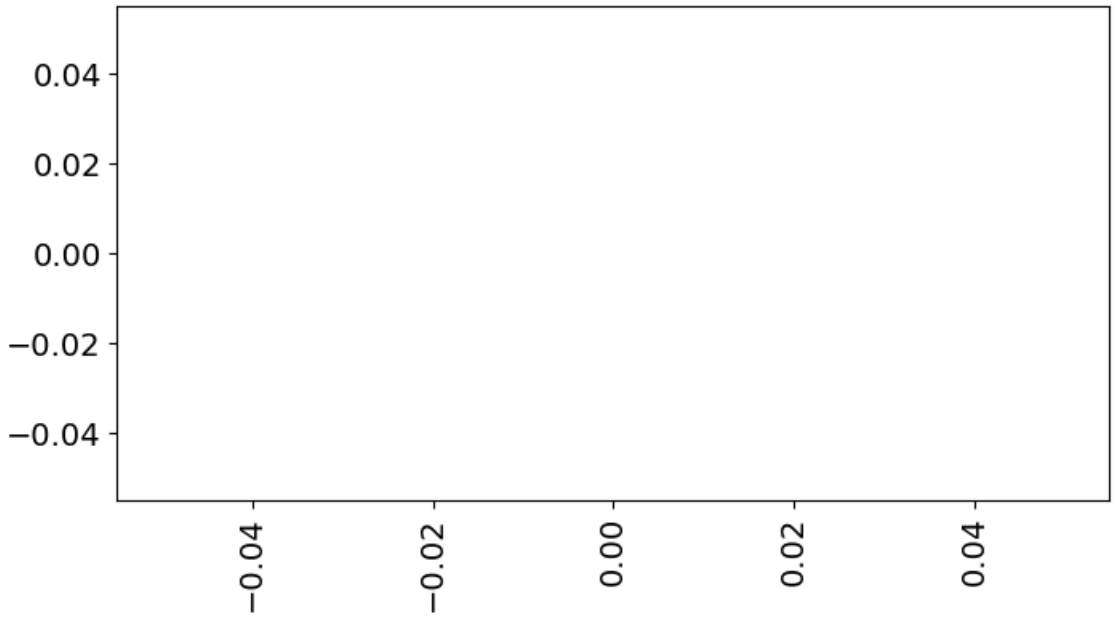
In [46]:
```
dn4.shape
```

Out[46]:
```
(0, 14)
```

In [48]:

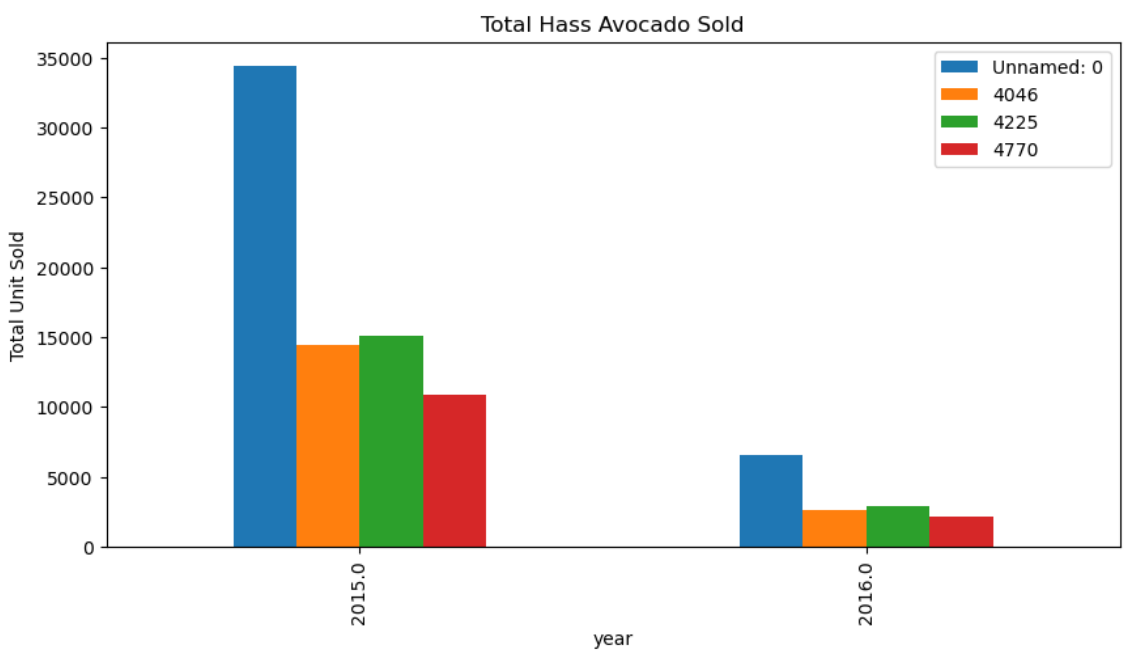
```
f,ax = plt.subplots(1,figsize=(6,3))
dn4.groupby(['year'])['AveragePrice'].mean().plot(kind='bar', figsize=(8,4)
plt.title('Hass Avocado - Average Price / Year')
print('Average Price(in $):',avocado_df.groupby(['year'])['AveragePrice'].m
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[48], line 2
      1 f,ax = plt.subplots(1,figsize=(6,3))
----> 2 dn4.groupby(['year'])['AveragePrice'].mean().plot(kind='bar', figsize=(8,4), fontsize=14, color='green')
      3 plt.title('Hass Avocado - Average Price / Year')
      4 print('Average Price(in $):',avocado_df.groupby(['year'])['AveragePrice'].mean())

File ~\anaconda3\Lib\site-packages\pandas\plotting\_core.py:1000, in PlotAccessor.__call__(self, *args, **kwargs)
    997             label_name = label_kw or data.columns
    998             data.columns = label_name
-> 1000 return plot_backend.plot(data, kind=kind, **kwargs)

File ~\anaconda3\Lib\site-packages\pandas\plotting\_matplotlib\__init__.py:71, in plot(data, kind, **kwargs)
     69         kwargs["ax"] = getattr(ax, "left_ax", ax)
     70 plot_obj = PLOT_CLASSES[kind](data, **kwargs)
---> 71 plot_obj.generate()
     72 plot_obj.draw()
     73 return plot_obj.result

File ~\anaconda3\Lib\site-packages\pandas\plotting\_matplotlib\core.py:459, in MPLPlot.generate(self)
    457 for ax in self.axes:
    458     self._post_plot_logic_common(ax, self.data)
--> 459     self._post_plot_logic(ax, self.data)

File ~\anaconda3\Lib\site-packages\pandas\plotting\_matplotlib\core.py:1739, in BarPlot._post_plot_logic(self, ax, data)
   1736 else:
   1737     str_index = [pprint_thing(key) for key in range(data.shape[0])]
-> 1739 s_edge = self.ax_pos[0] - 0.25 + self.lim_offset
   1740 e_edge = self.ax_pos[-1] + 0.25 + self.bar_width + self.lim_offset
   1742 self._decorate_ticks(ax, self._get_index_name(), str_index, s_edge, e_edge)

IndexError: index 0 is out of bounds for axis 0 with size 0
```


0.04
0.02
0.00
−0.02
−0.04
−0.04
−0.02
0.00
0.02
0.04

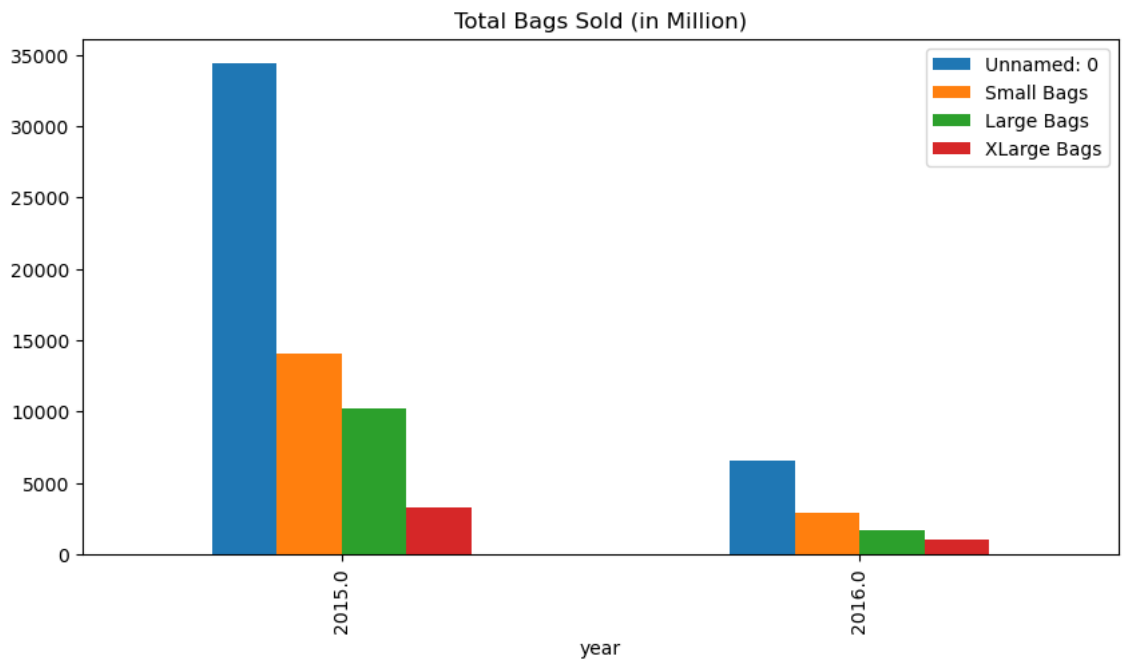
In [49]:
```
d2=d.copy()
d2.drop(['Date','AveragePrice','Total Volume','region','Total Bags','Small
d2.groupby(['year']).sum().plot(kind='bar',figsize=(10,5),legend=True)
plt.title ('Total Hass Avocado Sold')
plt.ylabel('Total Unit Sold')
print('Total Unit Sold 4046:',(avocado_df.groupby(['year'])['4046'].sum())
print('\n')
print('Total Unit Sold 4225:',(avocado_df.groupby(['year'])['4225'].sum())
print('\n')
print('Total Unit Sold 4770:',(avocado_df.groupby(['year'])['4770'].sum()
```

```
Total Unit Sold 4046: year
2015.00000   865.07418
2016.00000   115.57326
Name: 4046, dtype: float64


Total Unit Sold 4225: year
2015.00000   783.32014
2016.00000   144.23055
Name: 4225, dtype: float64


Total Unit Sold 4770: year
2015.00000   61.76072
2016.00000   14.70441
Name: 4770, dtype: float64
```



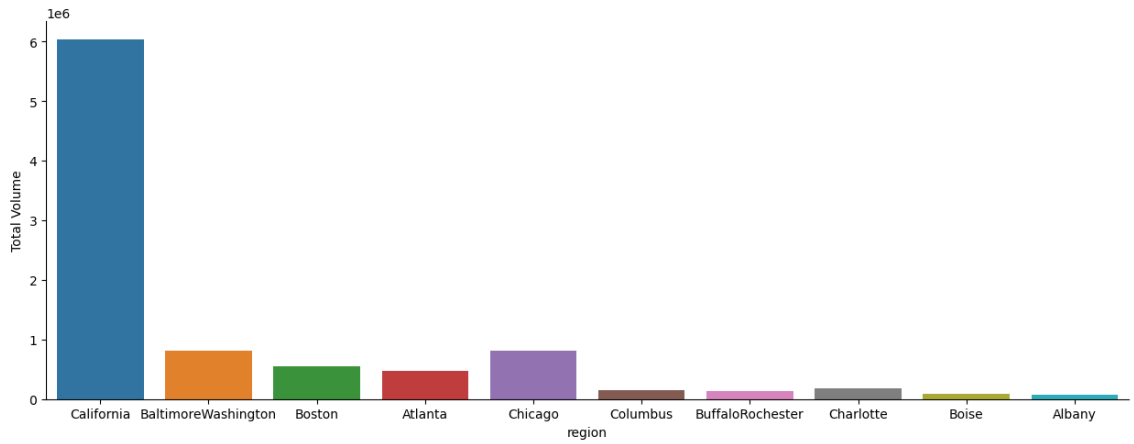
In [51]:
```
d3=d.copy()
d3.drop(['Date','AveragePrice','Total Volume','4046','4225','4770','region'
d3.groupby(['year']).sum().plot(kind='bar',figsize=(10,5),legend=True)
plt.title ('Total Bags Sold (in Million)')
print('Total Small Bags sold (in Million):',(avocado_df.groupby(['year'])['
print('\n')
print('Total Large Bags sold (in Million):',(avocado_df.groupby(['year'])['
```

```
Total Small Bags sold (in Million): year
2015.00000   277.37695
2016.00000   100.01259
Name: Small Bags, dtype: float64


Total Large Bags sold (in Million): year
2015.00000   55.39169
2016.00000    9.29630
Name: Large Bags, dtype: float64
```

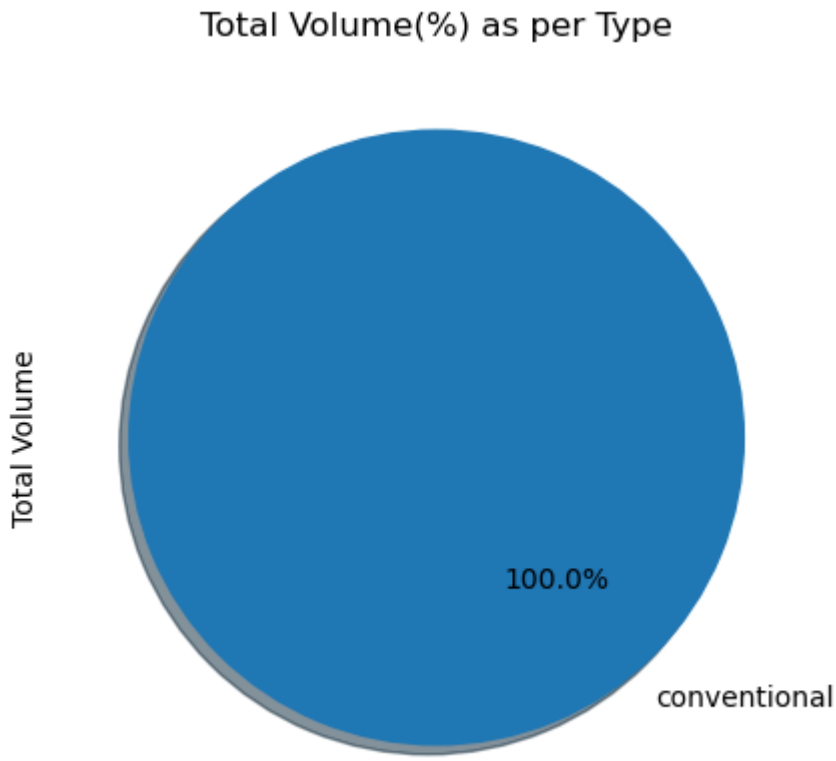


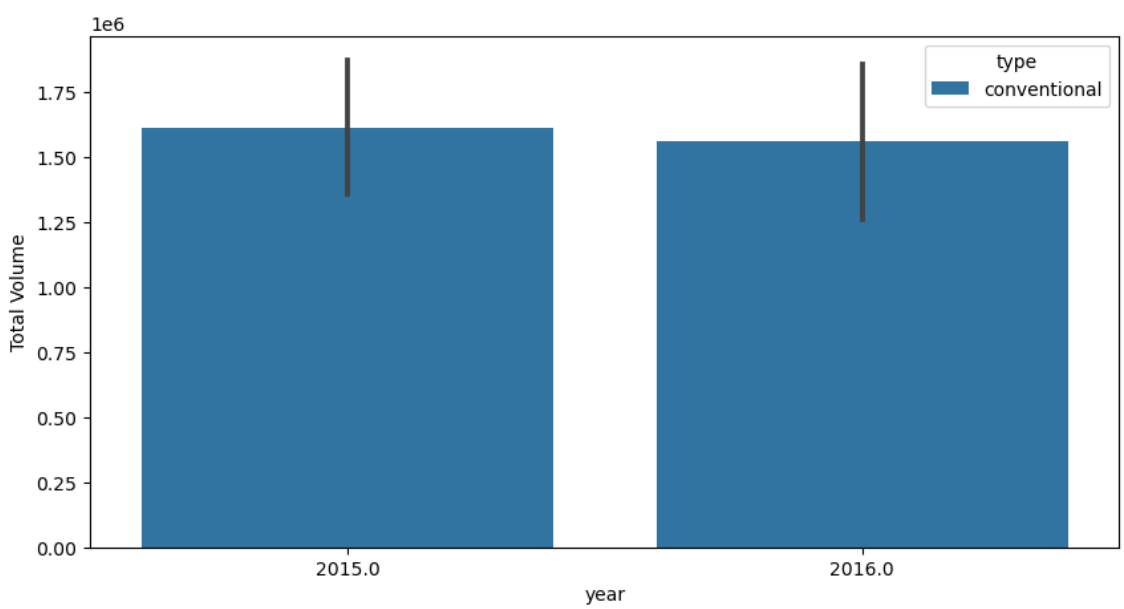
In [53]:
```
#sns.pairplot(d3, palette="viridis")
t_r=avocado_df.groupby(['region'])['Total Volume'].sum().head(10).sort_valu
sns.catplot(data=avocado_df, x='region',  y='Total Volume',kind='bar',ci=No
print('Top Selling Region (in Million)',(avocado_df.groupby(['region'])['To
```

```
Top Selling Region (in Million) region
California            458.68102
BaltimoreWashington    52.49687
Boston                 34.31443
Atlanta                25.25241
Chicago                18.67329
Columbus                6.85074
BuffaloRochester        6.74140
Charlotte               5.46462
Boise                   5.26800
Albany                  5.11144
Name: Total Volume, dtype: float64
```



In [54]:
```
avocado_df.groupby(['type'])['Total Volume'].sum().plot(kind='pie',subplots
autopct='%1.1f%%', shadow=True, startangle=130)
plt.title('Total Volume(%) as per Type')

plt.figure(figsize=(10,5))
sns.barplot(x='year',y='Total Volume',hue='type', data=avocado_df)
plt.show()

print('Based on Type:',(avocado_df.groupby(['type']).sum())/1000000)
```





```
Based on Type:                Unnamed: 0  AveragePrice  Total Volume
4046      4225  \
type
conventional     0.04095       0.00163    2430.05055 980.64744 927.55069

                 4770  Total Bags  Small Bags  Large Bags  XLarge Bags
year
type
conventional 76.46514   445.38728   377.38954    64.68799      3.30975 3.
05700
```

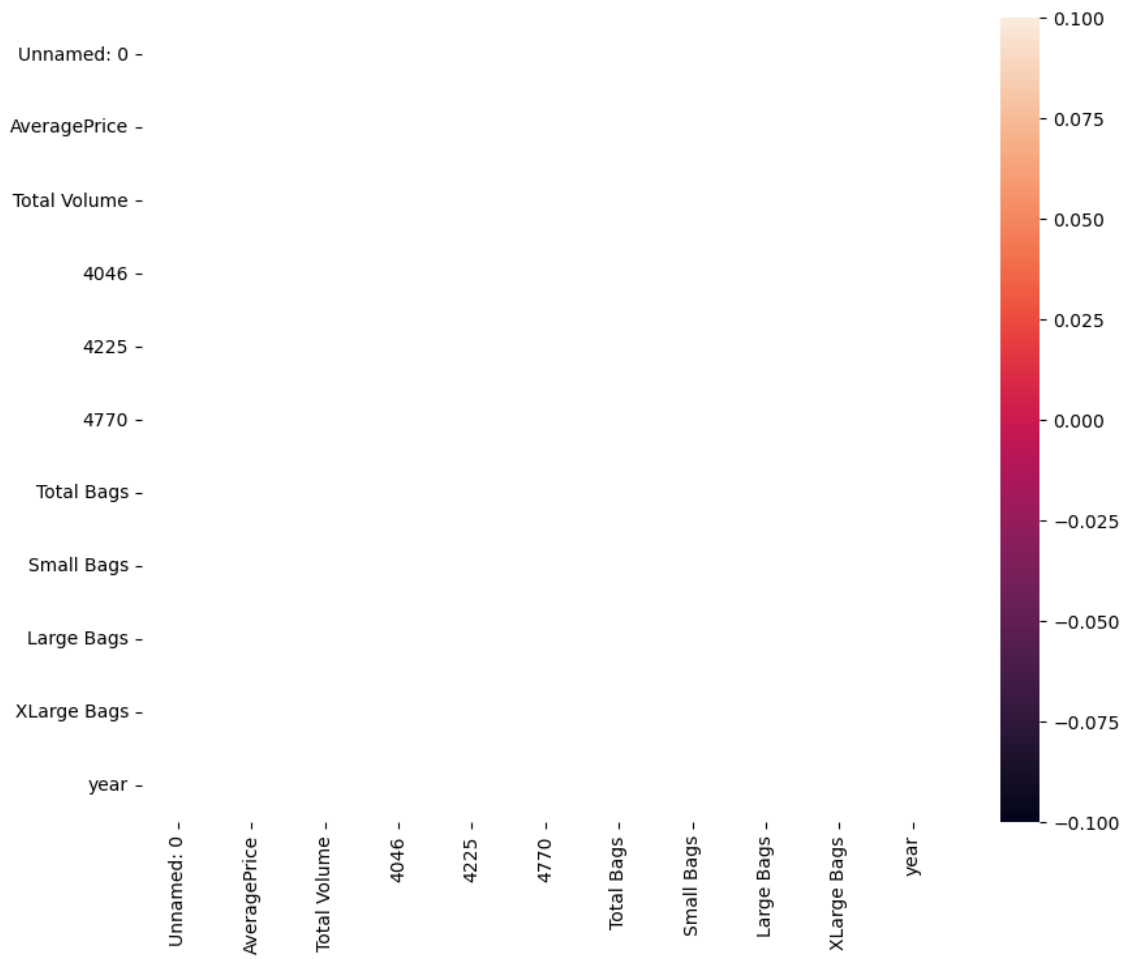
In [55]:

```
dn5=pd.get_dummies(dn4.drop(['region','Date'],axis=1),drop_first=True)
dn5.head(2)
dn6=pd.get_dummies(dn4.drop(['Date'],axis=1),drop_first=True)
dn6.head(2)
print(dn5.shape)
print(dn6.shape)
```

```
(0, 11)
(0, 11)
```

In [56]:
```
dn5.corr()
plt.figure(figsize=(10,8))
sns.heatmap( dn5.corr(), annot=True);plt.figure(figsize=(10,8))
```

Out[56]:
```
<Figure size 1000x800 with 0 Axes>
```




```
<Figure size 1000x800 with 0 Axes>
```

In [57]:
```
dnf=dn5.copy()
dnf.head
```

Out[57]:
```
<bound method NDFrame.head of Empty DataFrame
Columns: [Unnamed: 0, AveragePrice, Total Volume, 4046, 4225, 4770, Total
Bags, Small Bags, Large Bags, XLarge Bags, year]
Index: []>
```

In [63]:
```
feature_cols = ['Total Volume','4046','4225','4770','Small Bags','Large Bag
x=dn5[feature_cols]
y=dn5['AveragePrice']
```

In [64]:
```
print(x.shape)
print(y.shape)
```

```
(0, 7)
(0,)
```

In [65]:
```
x.head()
```

Out[65]:

Total Volume	4046	4225	4770	Small Bags	Large Bags	XLarge Bags

In [66]:
```
y.head()
```

Out[66]:
```
Series([], Name: AveragePrice, dtype: float64)
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: