PROJECT ON AVOCADO DATASET

Project By Bhagyashree Chavan

In [183]:

```
# These are the following Library using to get our results


import pandas as pd

import matplotlib

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

In [184]:

```
# to get know what is our path
import pathlib
pathlib.Path().absolute()
```

Out[184]:

```
PosixPath('/Users/bhagyashreechavan/Desktop')
```

Problem statement on Avocado dataset with focus on which part of region supply more avocado in U.S?

What is best price in which region marketers get in market?

First step is reading file and try to understand data set.

In [236]:

```
# reading file

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

Out[236]:

	Unnamed: 0	Date	AveragePrice	Total_Volume	4046	4225	4770	Total_Bags	Small_Bags	Large_Bags	Xlarge_Bags	type
0	0	12/27/15	1.33	64236.62	1036.74	54454.85	48.16	8696.87	8603.62	93.25	0.0	conventional
1	1	12/20/15	1.35	54876.98	674.28	44638.81	58.33	9505.56	9408.07	97.49	0.0	conventional
2	2	12/13/15	0.93	118220.22	794.70	109149.67	130.50	8145.35	8042.21	103.14	0.0	conventional
3	3	12/6/15	1.08	78992.15	1132.00	71976.41	72.58	5811.16	5677.40	133.76	0.0	conventional
4	4	11/29/15	1.28	51039.60	941.48	43838.39	75.78	6183.95	5986.26	197.69	0.0	conventional
...	...	...	...	...	...	...	...	...	...	...	...	...
18244	7	2/4/18	1.63	17074.83	2046.96	1529.20	0.00	13498.67	13066.82	431.85	0.0	organic
18245	8	1/28/18	1.71	13888.04	1191.70	3431.50	0.00	9264.84	8940.04	324.80	0.0	organic
18246	9	1/21/18	1.87	13766.76	1191.92	2452.79	727.94	9394.11	9351.80	42.31	0.0	organic
18247	10	1/14/18	1.93	16205.22	1527.63	2981.04	727.01	10969.54	10919.54	50.00	0.0	organic
18248	11	1/7/18	1.62	17489.58	2894.77	2356.13	224.53	12014.15	11988.14	26.01	0.0	organic

18249 rows × 14 columns

We have total 18249 rows and 14 features with index.

Now checking any missing values in data set with following codes.

In [237]:

```
# checking head of data set
df.head()
```

Out[237]:

	Unnamed: 0	Date	AveragePrice	Total_Volume	4046	4225	4770	Total_Bags	Small_Bags	Large_Bags	Xlarge_Bags	type	yea
0	0	12/27/15	1.33	64236.62	1036.74	54454.85	48.16	8696.87	8603.62	93.25	0.0	conventional	201
1	1	12/20/15	1.35	54876.98	674.28	44638.81	58.33	9505.56	9408.07	97.49	0.0	conventional	201
2	2	12/13/15	0.93	118220.22	794.70	109149.67	130.50	8145.35	8042.21	103.14	0.0	conventional	201
3	3	12/6/15	1.08	78992.15	1132.00	71976.41	72.58	5811.16	5677.40	133.76	0.0	conventional	201
4	4	11/29/15	1.28	51039.60	941.48	43838.39	75.78	6183.95	5986.26	197.69	0.0	conventional	201

In [238]:

```
# checking data types
df.dtypes
```

Out[238]:

```
Unnamed: 0        int64
Date             object
AveragePrice    float64
Total_Volume    float64
4046            float64
4225            float64
4770            float64
Total_Bags      float64
Small_Bags      float64
Large_Bags      float64
Xlarge_Bags     float64
type             object
year              int64
region           object
dtype: object
```

In [239]:

```
# now we will check what is our columns names
df.columns
```

Out[239]:

```
Index(['Unnamed: 0', 'Date', 'AveragePrice', 'Total_Volume', '4046', '4225',
       '4770', 'Total_Bags', 'Small_Bags', 'Large_Bags', 'Xlarge_Bags', 'type',
       'year', 'region'],
      dtype='object')
```

In [240]:

```
#total missing values and count of na's

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

Out[240]:

```
Unnamed: 0      0
Date            0
AveragePrice    0
Total_Volume    0
4046            0
4225            0
4770            0
Total_Bags      0
Small_Bags      0
Large_Bags      0
Xlarge_Bags     0
type            0
year            0
region          0
dtype: int64
```

So in conlusion there is no null,na or any missing values in dataset.

Now we will check what is our data describing and understanding in depth

In [241]:

```
#description of the data
df.describe
```

Out[241]:

```
<bound method NDFrame.describe of        Unnamed: 0      Date  AveragePrice  Total_Volume     4046       4225  
\
0               0  12/27/15          1.33      64236.62  1036.74   54454.85
1               1  12/20/15          1.35      54876.98   674.28   44638.81
2               2  12/13/15          0.93     118220.22   794.70  109149.67
3               3   12/6/15          1.08      78992.15  1132.00   71976.41
4               4  11/29/15          1.28      51039.60   941.48   43838.39
...           ...       ...           ...           ...      ...        ...
18244           7    2/4/18          1.63      17074.83  2046.96    1529.20
18245           8   1/28/18          1.71      13888.04  1191.70    3431.50
18246           9   1/21/18          1.87      13766.76  1191.92    2452.79
18247          10   1/14/18          1.93      16205.22  1527.63    2981.04
18248          11    1/7/18          1.62      17489.58  2894.77    2356.13

         4770  Total_Bags  Small_Bags  Large_Bags  Xlarge_Bags          type  \
0       48.16     8696.87     8603.62       93.25          0.0  conventional   
1       58.33     9505.56     9408.07       97.49          0.0  conventional   
2      130.50     8145.35     8042.21      103.14          0.0  conventional   
3       72.58     5811.16     5677.40      133.76          0.0  conventional   
4       75.78     6183.95     5986.26      197.69          0.0  conventional   
...       ...         ...         ...         ...          ...           ...   
18244    0.00    13498.67    13066.82      431.85          0.0       organic   
18245    0.00     9264.84     8940.04      324.80          0.0       organic   
18246  727.94     9394.11     9351.80       42.31          0.0       organic   
18247  727.01    10969.54    10919.54       50.00          0.0       organic   
18248  224.53    12014.15    11988.14       26.01          0.0       organic   

       year            region  
0      2015            Albany  
1      2015            Albany  
2      2015            Albany  
3      2015            Albany  
4      2015            Albany  
...     ...               ...  
18244  2018  WestTexNewMexico  
18245  2018  WestTexNewMexico  
18246  2018  WestTexNewMexico  
18247  2018  WestTexNewMexico  
18248  2018  WestTexNewMexico  

[18249 rows x 14 columns]>
```

In [242]:

```
# if we use parenthesis then we will get same data in tabulaer format with statistics
df.describe()
```

Out[242]:

	Unnamed: 0	AveragePrice	Total_Volume	4046	4225	4770	Total_Bags	Small_Bags	Large_Bags	Xlarge_Ba
count	18249.000000	18249.000000	1.824900e+04	1.824900e+04	1.824900e+04	1.824900e+04	1.824900e+04	1.824900e+04	1.824900e+04	18249.0000
mean	24.232232	1.405978	8.506440e+05	2.930084e+05	2.951546e+05	2.283974e+04	2.396392e+05	1.821947e+05	5.433809e+04	3106.4265
std	15.481045	0.402677	3.453545e+06	1.264989e+06	1.204120e+06	1.074641e+05	9.862424e+05	7.461785e+05	2.439660e+05	17692.8946
min	0.000000	0.440000	8.456000e+01	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.0000
25%	10.000000	1.100000	1.083858e+04	8.540700e+02	3.008780e+03	0.000000e+00	5.088640e+03	2.849420e+03	1.274700e+02	0.0000
50%	24.000000	1.370000	1.073768e+05	8.645300e+03	2.906102e+04	1.849900e+02	3.974383e+04	2.636282e+04	2.647710e+03	0.0000
75%	38.000000	1.660000	4.329623e+05	1.110202e+05	1.502069e+05	6.243420e+03	1.107834e+05	8.333767e+04	2.202925e+04	132.5000
max	52.000000	3.250000	6.250565e+07	2.274362e+07	2.047057e+07	2.546439e+06	1.937313e+07	1.338459e+07	5.719097e+06	551693.6500

Now here when we apply describe code then in the result we will get only those information who has data type in 'int64' or 'float'.

we won't get result for object format.

Now we will check what is trend of Average price of Avocado in US

In the region section data have contained all US and Total US values.

So that for further analysis we need to remove Total US rows so we will get correct result which we can use for analysis.

In [243]:
```
drop=df.region!='TotalUS'
df1=df[drop]
df1.head()
```

Out[243]:

	Unnamed: 0	Date	AveragePrice	Total_Volume	4046	4225	4770	Total_Bags	Small_Bags	Large_Bags	Xlarge_Bags	type	yea
0	0	12/27/15	1.33	64236.62	1036.74	54454.85	48.16	8696.87	8603.62	93.25	0.0	conventional	201
1	1	12/20/15	1.35	54876.98	674.28	44638.81	58.33	9505.56	9408.07	97.49	0.0	conventional	201
2	2	12/13/15	0.93	118220.22	794.70	109149.67	130.50	8145.35	8042.21	103.14	0.0	conventional	201
3	3	12/6/15	1.08	78992.15	1132.00	71976.41	72.58	5811.16	5677.40	133.76	0.0	conventional	201
4	4	11/29/15	1.28	51039.60	941.48	43838.39	75.78	6183.95	5986.26	197.69	0.0	conventional	201

In [417]:
```
df1.shape
# now after removing Total US values row we have 17911 rows and 13 features.
```

Out[417]:
```
(17911, 14)
```

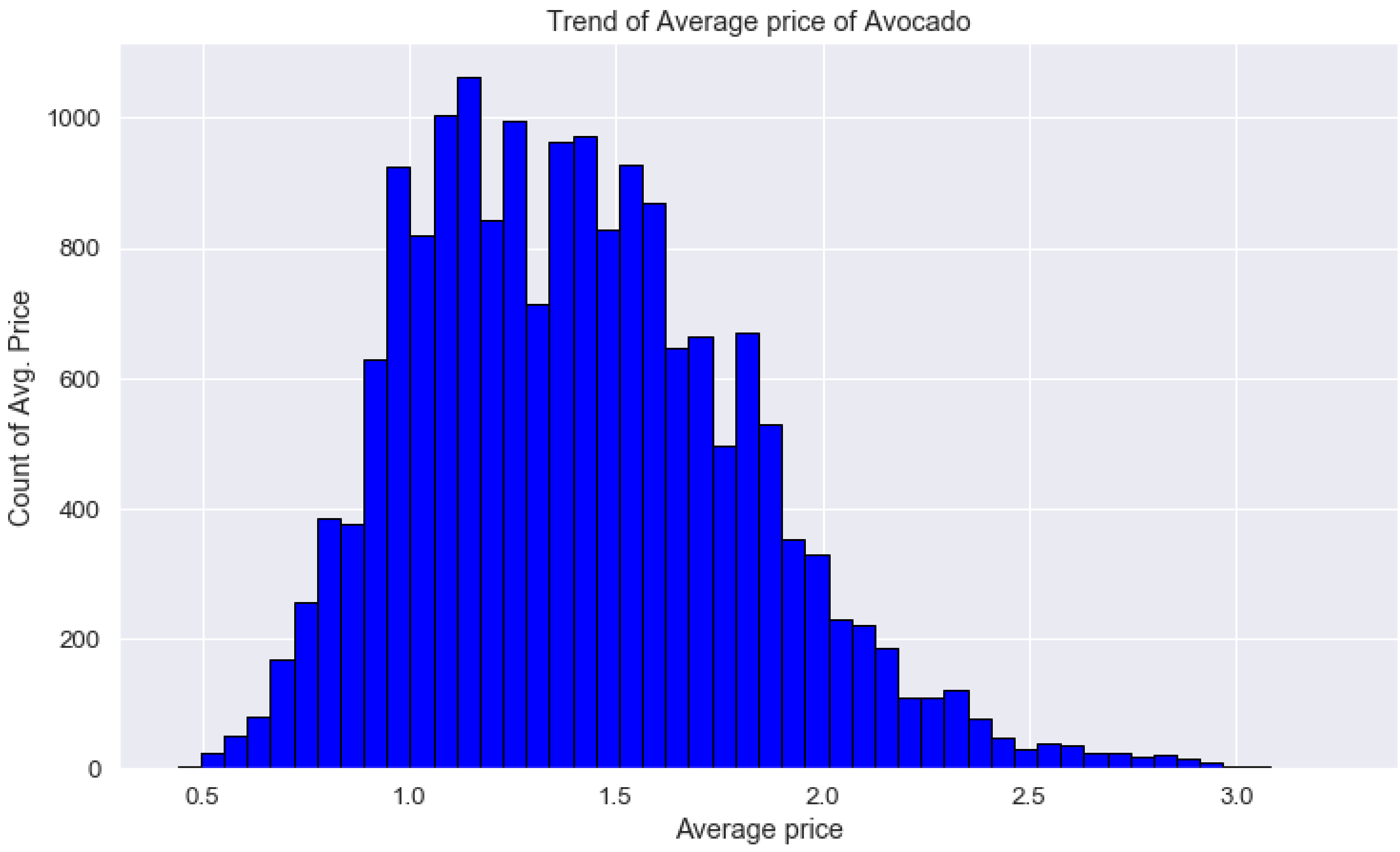
Now we will check what is trend of Average price of Avocado

For the trend I have used histogram with following Libraries.

In [265]:
```
import matplotlib.pyplot as plt
import seaborn as sns

#creating histogram
plt.figure()
plt.figure(figsize=(12,7))
plt.hist(df1['AveragePrice'],color = 'blue', edgecolor = 'black', bins= 50)
plt.title('Trend of Average price of Avocado')
plt.xlabel('Average price')
plt.ylabel('Count of Avg. Price')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```

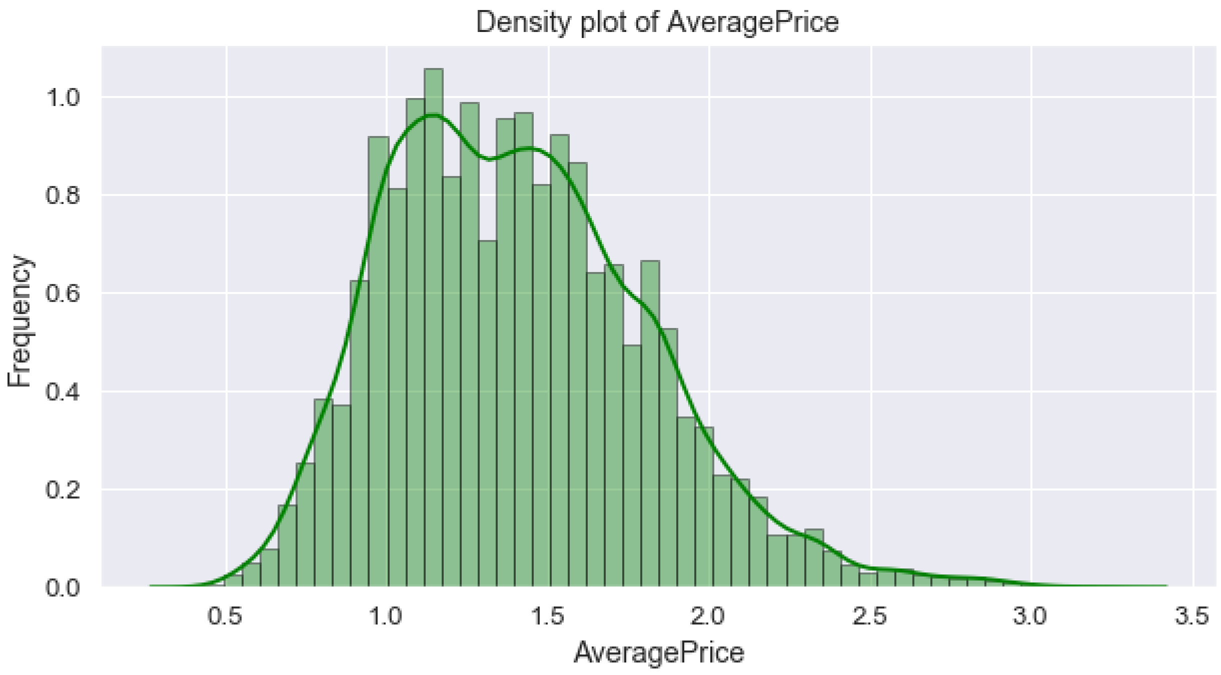


#Now here we are checking what id density of Average price with using seaborne library

In [259]:

```
plt.figure()
plt.figure(figsize=(10,5))
sns.distplot(df1['AveragePrice'],hist = True, kde= True, color = 'green',
             hist_kws ={'edgecolor':'black'},kde_kws={'linewidth':2})
plt.title('Density plot of AveragePrice')
plt.xlabel('AveragePrice')
plt.ylabel('Frequency')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



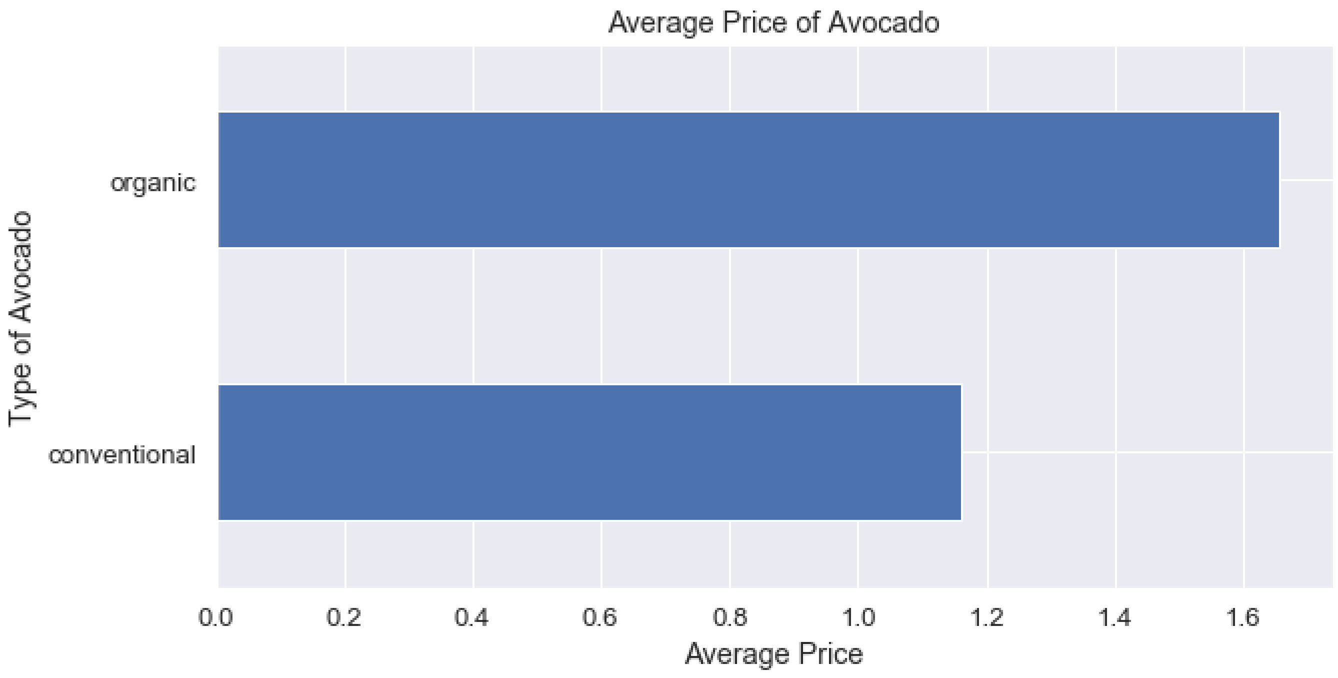
After getting result of density of average price of avocado is USD 1.3 to USD 1.8

Now we will understand what is average price for Avocado type ?

In [418]:

```
plt.figure()
plt.figure(figsize=(10,5))
df1.groupby('type').AveragePrice.mean().plot(kind='barh')
plt.title('Average Price of Avocado')
plt.xlabel('Average Price')
plt.ylabel('Type of Avocado')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```

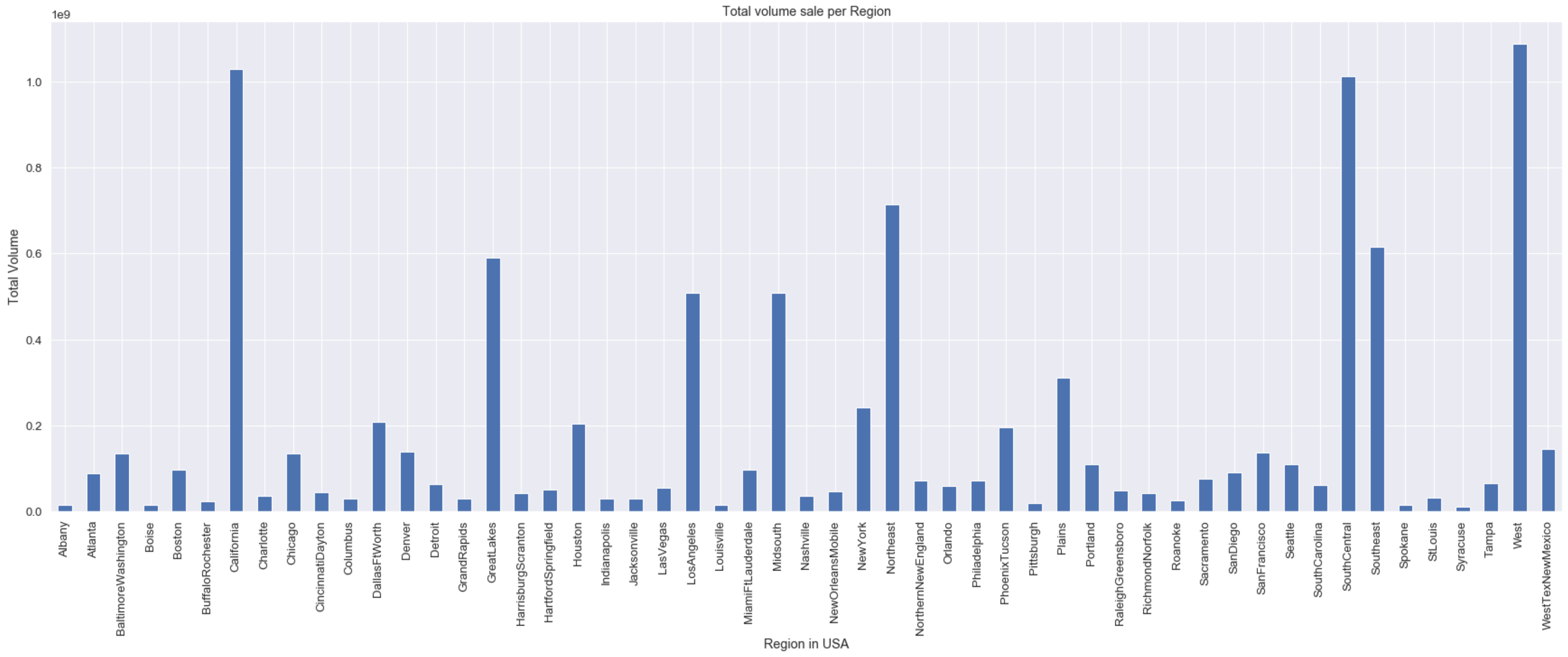


According to see results price of conventional Avocado is USD 1.2 and Organic price is USD 1.7

Now analysing Region of US with Total volume

```
In [419]: plt.figure()
          plt.figure(figsize=(30,10))
          df1.groupby('region').Total_Volume.sum().plot(kind='bar')
          plt.title('Total volume sale per Region')
          plt.xlabel('Region in USA')
          plt.ylabel('Total Volume')
          plt.show()
```

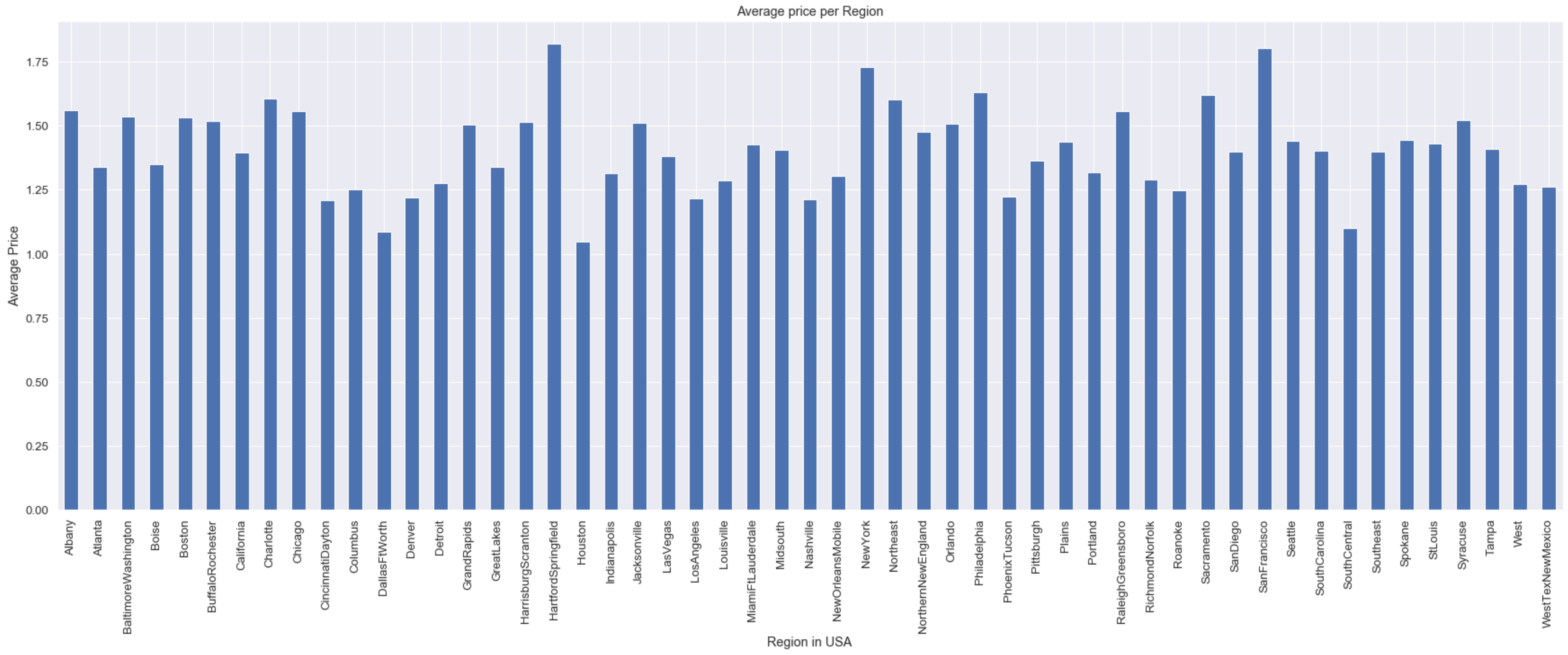
```
<Figure size 432x288 with 0 Axes>
```



Analysis of Regions with Average price of Avocado

```
In [435]: plt.figure()
          plt.figure(figsize=(30,10))
          df1.groupby('region').AveragePrice.mean().plot(kind='bar')
          plt.title('Average price per Region')
          plt.xlabel('Region in USA')
          plt.ylabel('Average Price')
          plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```

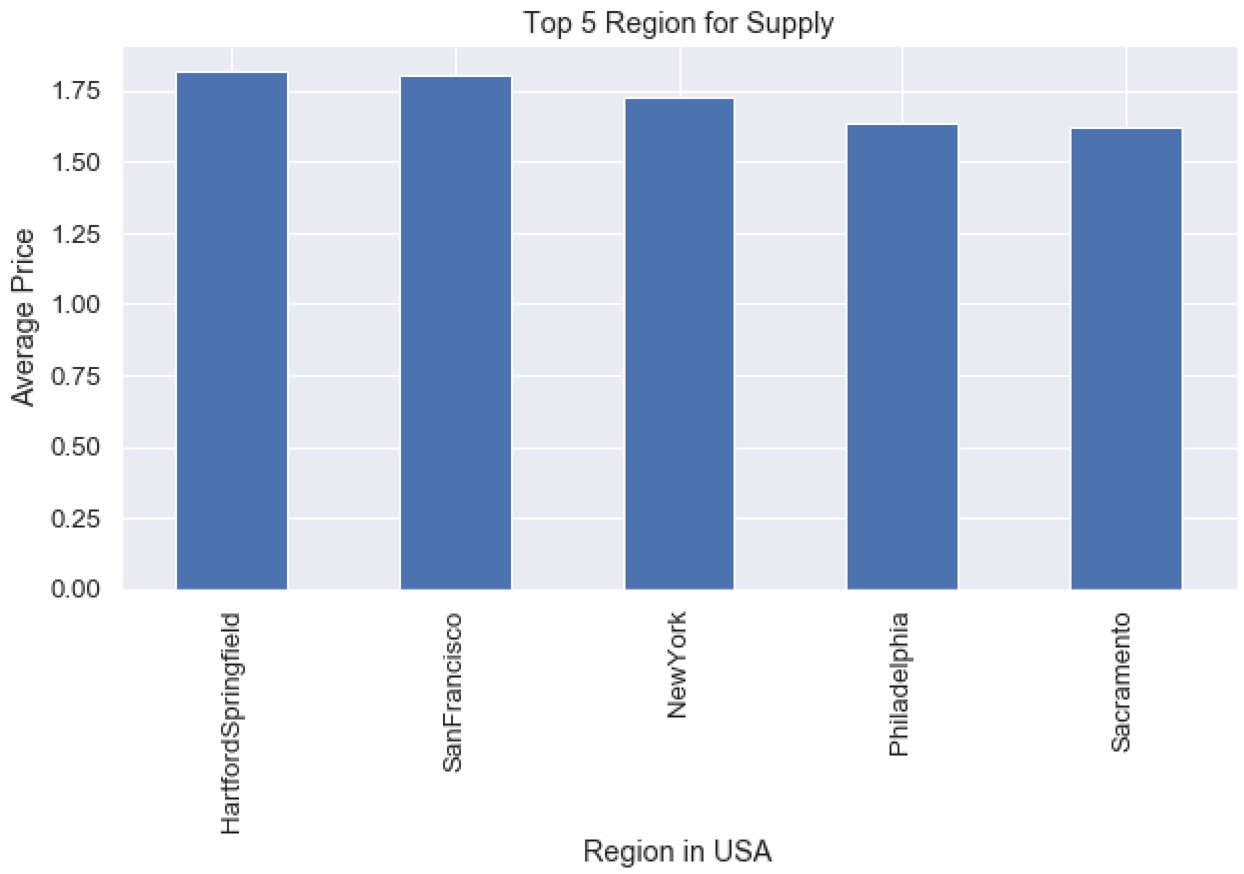


Top 5 region of best price

In [421]:

```
plt.figure()
plt.figure(figsize=(10,5))
df1.groupby("region").AveragePrice.mean().sort_values(ascending=False)[:5].plot.bar()
plt.title('Top 5 Region for Supply')
plt.xlabel('Region in USA')
plt.ylabel('Average Price')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



So that in conclusion for best 5 region where the marketers, producers can supply more in oredr to get profit Top 5 regions are following:

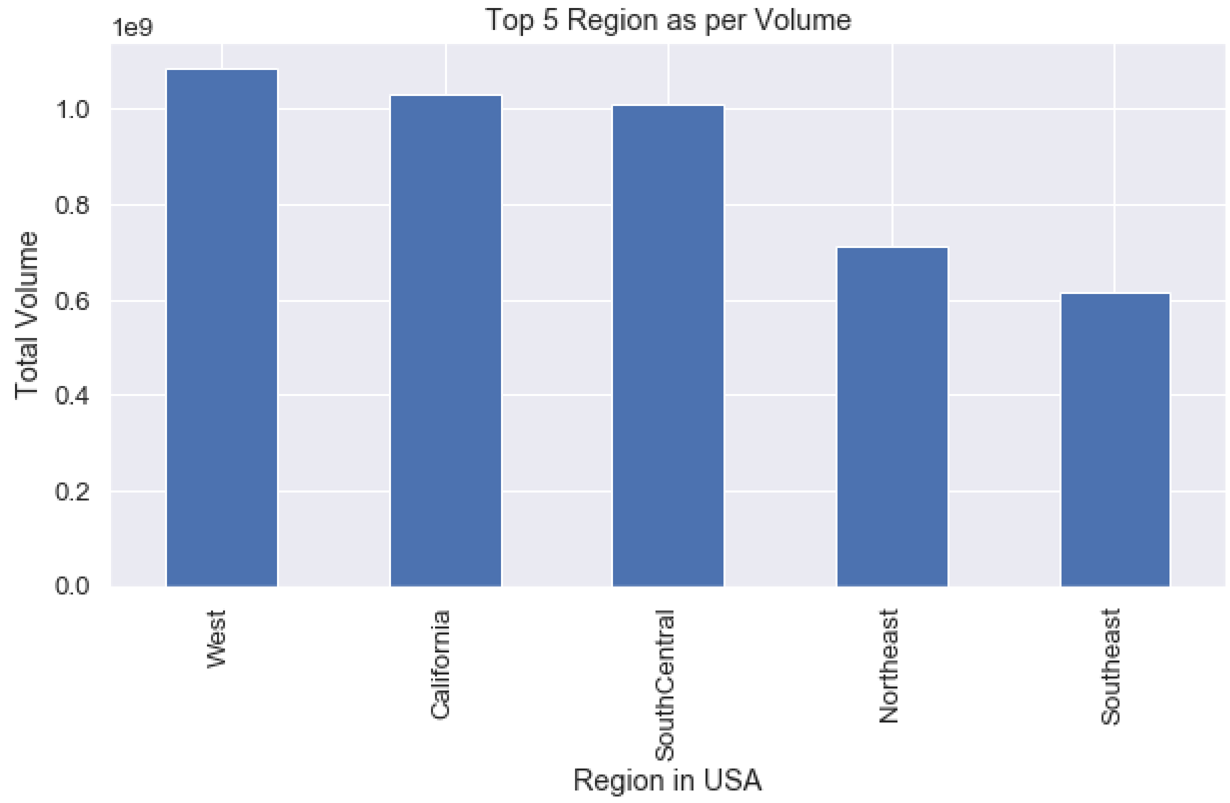
1. Hartforspringfield
2. SanFrancisco
3. New York
4. Philadelphia
5. Sacramento

Top 5 region where the Volume comsumption most

In [422]:

```
plt.figure()
plt.figure(figsize=(10,5))
df1.groupby("region").Total_Volume.sum().sort_values(ascending=False)[:5].plot.bar()
plt.title('Top 5 Region as per Volume')
plt.xlabel('Region in USA')
plt.ylabel('Total Volume')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



According to analysis of Total volume result for consumption region are different that best price region

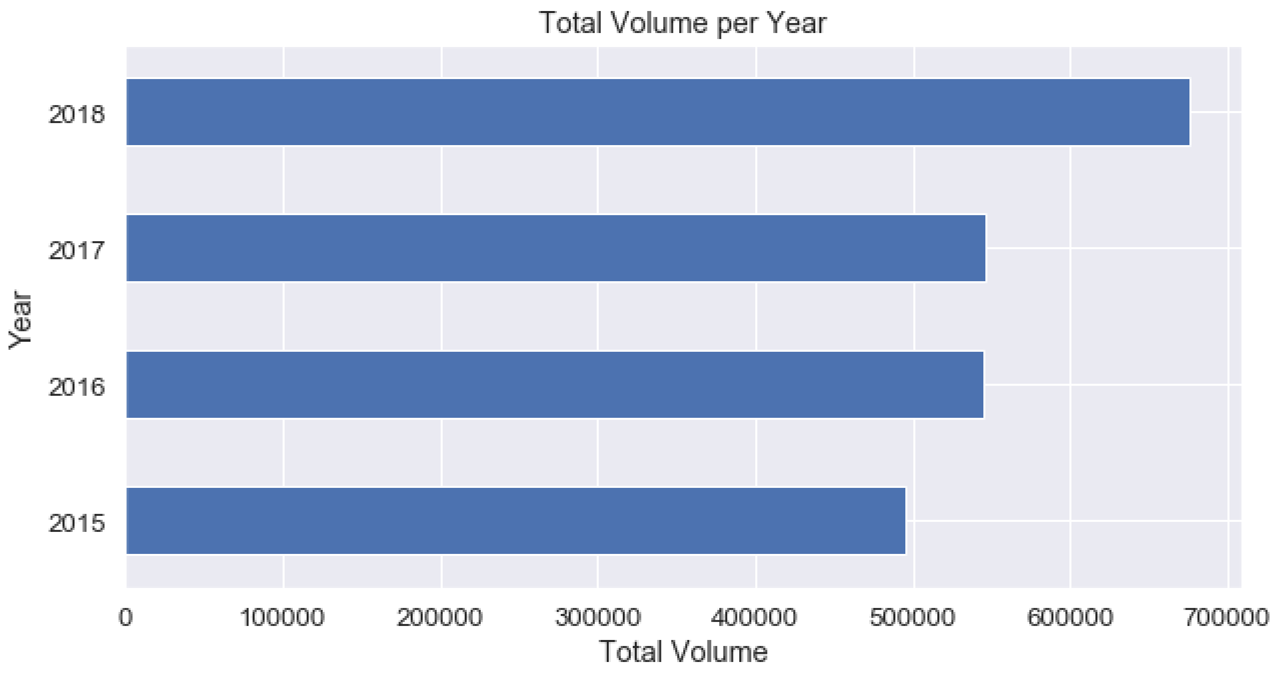
1. West
2. California
3. SouthCentral
4. Northeast
5. Southeast

Further more analysis according to year

In [423]:

```
plt.figure()
plt.figure(figsize=(10,5))
df1.groupby("year").Total_Volume.mean().plot.barh()
plt.title('Total Volume per Year')
plt.xlabel('Total Volume')
plt.ylabel('Year')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



After getting the result comparing 2015 to other year sale has been increased.

But in 2016 to 2017 volume had no changed and then major difference between 2017 to 2018.

In 2018 Total volume has been increased upto 670k.

Price according to years

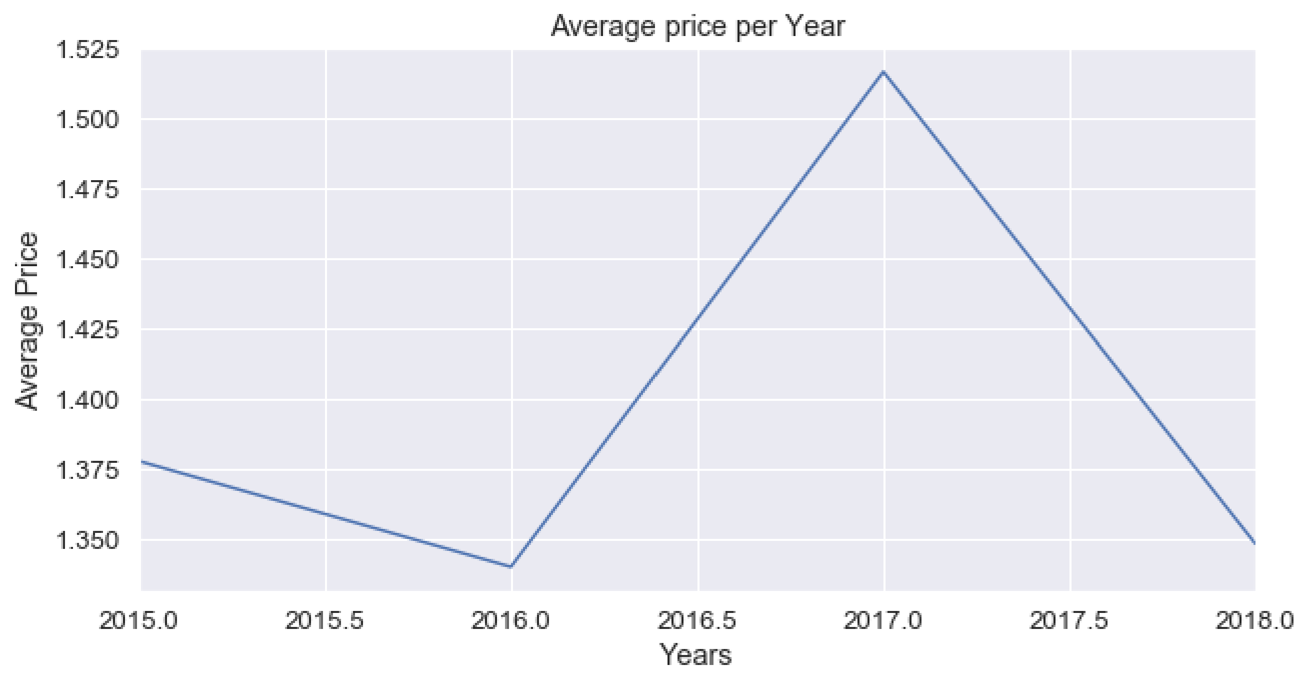
In [424]:

```
plt.figure()
plt.figure(figsize=(10,5))

df1.groupby('year').AveragePrice.mean().plot(kind='line')

plt.title('Average price per Year')
plt.xlabel('Years')
plt.ylabel('Average Price')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



In graph has shown that price has less in between 2015 to 2016.

But after starting year 2016 to 2017 price has been reached upto USD 1.5.

In year 2018 price again slow down and reached in between USD 1.375 to USD 1.350.

In [425]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as seabornInstance
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
%matplotlib inline
df1.head()
```

Out[425]:

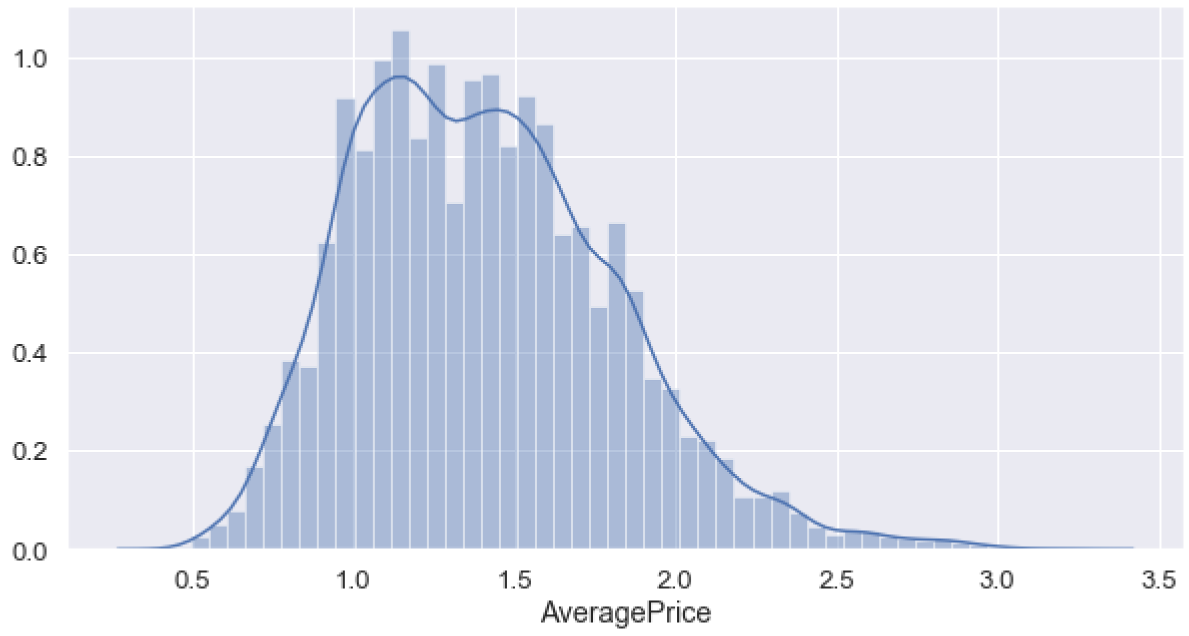
	Unnamed: 0	Date	AveragePrice	Total_Volume	4046	4225	4770	Total_Bags	Small_Bags	Large_Bags	Xlarge_Bags	type	yea
0	0	12/27/15	1.33	64236.62	1036.74	54454.85	48.16	8696.87	8603.62	93.25	0.0	conventional	201
1	1	12/20/15	1.35	54876.98	674.28	44638.81	58.33	9505.56	9408.07	97.49	0.0	conventional	201
2	2	12/13/15	0.93	118220.22	794.70	109149.67	130.50	8145.35	8042.21	103.14	0.0	conventional	201
3	3	12/6/15	1.08	78992.15	1132.00	71976.41	72.58	5811.16	5677.40	133.76	0.0	conventional	201
4	4	11/29/15	1.28	51039.60	941.48	43838.39	75.78	6183.95	5986.26	197.69	0.0	conventional	201

In [426]:

```
plt.figure(figsize=(10,5))
plt.tight_layout()
seabornInstance.distplot(df1['AveragePrice'])
```

Out[426]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x14afbfc10>
```



In [427]:

```
x = df1['Total_Volume']
y = df1['AveragePrice']

# here we have taking 20% of data to get train or to run our model
#and random state can be any number we can take but mostly people take 0 or none.
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

In [428]:

```
#Now we are apploing Linear regression model in order to get result

regres = LinearRegression()
regres.fit(X_train, y_train)
```

Out[428]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [429]:

```
#To get intercept:
print(regres.intercept_)


#To get slope:
print(regres.coef_)
```

```
-76.60036402148224
[0.03869152]
```

In [430]:

```
# Here we are assing y pred of x test
y_pred = regres.predict(X_test)
```

In [431]:

```


import numpy as np
import pandas as pd
y_test = np.array(list(y_test))
y_pred = np.array(y_pred)
df4= pd.DataFrame({'Actual': y_test.flatten(), 'Predicted': y_pred.flatten()})
df4.head()
```

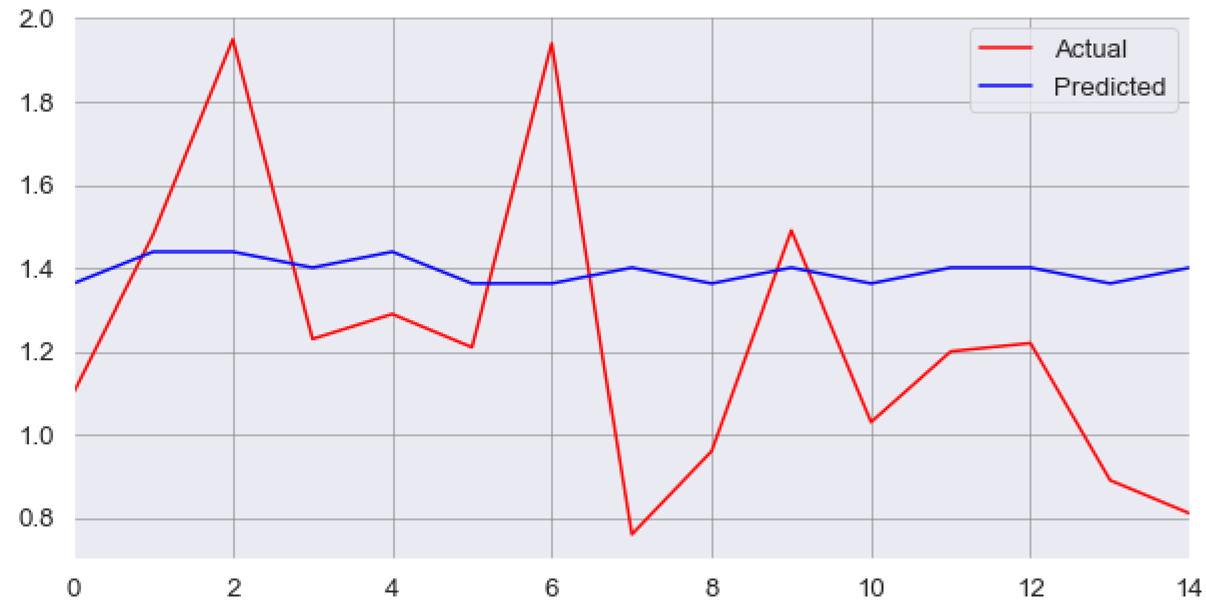
Out[431]:

	Actual	Predicted
0	2.07	1.363043
1	1.26	1.440426
2	0.58	1.401734
3	1.43	1.479117
4	1.54	1.440426

In [432]:

```
# here we are considering only 15 results

df4 = df3.head(15)
df4.plot(kind='line',figsize=(10,5),color = ('red','blue'))
plt.grid(which='major', linestyle='-', linewidth='0.5', color='gray')
plt.grid(which='minor', linestyle='dotted', linewidth='0.5', color='black')
plt.show()
```



In [433]:

```
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
Mean Absolute Error: 0.3193997805037335
Mean Squared Error: 0.157564993336592
Root Mean Squared Error: 0.3969445721213379
```

Mean square root is 0.3969 more than mean value of the percentage.

Conclusion

For supplies and marketers these are best places to get more profit:

Hartforspringfield SanFrancisco New York Philadelphia Sacramento

For Volume consumption following region are best:

West California SouthCentral Northeast Southeast

```
In [ ]:
```