In [105…
```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

In [106…
```python
file =pd.read_csv("avocado.csv.zip")
```

In [107…
```python
file
```

Out[107…

	Unnamed: 0	Date	AveragePrice	Total Volume	4046	4225	4770	Total Bags	Small Bags	Large Bags	XLarge Bags	type	year	region
0	0	2015-12-27	1.33	64236.62	1036.74	54454.85	48.16	8696.87	8603.62	93.25	0.0	conventional	2015	Albany
1	1	2015-12-20	1.35	54876.98	674.28	44638.81	58.33	9505.56	9408.07	97.49	0.0	conventional	2015	Albany
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
18247	10	2018-01-14	1.93	16205.22	1527.63	2981.04	727.01	10969.54	10919.54	50.00	0.0	organic	2018	WestTexNewMexico
18248	11	2018-01-07	1.62	17489.58	2894.77	2356.13	224.53	12014.15	11988.14	26.01	0.0	organic	2018	WestTexNewMexico

18249 rows × 14 columns

In [108…
```python
file.head(5)
```

Out[108…

	Unnamed: 0	Date	AveragePrice	Total Volume	4046	4225	4770	Total Bags	Small Bags	Large Bags	XLarge Bags	type	year	region
0	0	2015-12-27	1.33	64236.62	1036.74	54454.85	48.16	8696.87	8603.62	93.25	0.0	conventional	2015	Albany
1	1	2015-12-20	1.35	54876.98	674.28	44638.81	58.33	9505.56	9408.07	97.49	0.0	conventional	2015	Albany
2	2	2015-12-13	0.93	118220.22	794.70	109149.67	130.50	8145.35	8042.21	103.14	0.0	conventional	2015	Albany
3	3	2015-12-06	1.08	78992.15	1132.00	71976.41	72.58	5811.16	5677.40	133.76	0.0	conventional	2015	Albany
4	4	2015-11-29	1.28	51039.60	941.48	43838.39	75.78	6183.95	5986.26	197.69	0.0	conventional	2015	Albany

In [109…
```python
file.tail(5)
```

Out[109…

	Unnamed: 0	Date	AveragePrice	Total Volume	4046	4225	4770	Total Bags	Small Bags	Large Bags	XLarge Bags	type	year	region
18244	7	2018-02-04	1.63	17074.83	2046.96	1529.20	0.00	13498.67	13066.82	431.85	0.0	organic	2018	WestTexNewMexico
18245	8	2018-01-28	1.71	13888.04	1191.70	3431.50	0.00	9264.84	8940.04	324.80	0.0	organic	2018	WestTexNewMexico
18246	9	2018-01-21	1.87	13766.76	1191.92	2452.79	727.94	9394.11	9351.80	42.31	0.0	organic	2018	WestTexNewMexico
18247	10	2018-01-14	1.93	16205.22	1527.63	2981.04	727.01	10969.54	10919.54	50.00	0.0	organic	2018	WestTexNewMexico
18248	11	2018-01-07	1.62	17489.58	2894.77	2356.13	224.53	12014.15	11988.14	26.01	0.0	organic	2018	WestTexNewMexico

In [110…
```python
# checking data shape and size

file.shape
```

Out[110…
```
(18249, 14)
```

In [111…
```python
# checking null values

file.isnull().sum()
```

Out[111…
```
Unnamed: 0    0
Date          0
             ..
year          0
region        0
Length: 14, dtype: int64
```

In [112…
```python
file.describe()
```

Out[112…

	Unnamed: 0	AveragePrice	Total Volume	4046	4225	4770	Total Bags	Small Bags	Large Bags	XLarge Bags	year
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	18249.000000	18249.000000
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	3106.426507	2016.147899
...	...	...	...	...	...	...	...	...	...	...	...
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	132.500000	2017.000000
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	551693.650000	2018.000000

8 rows × 11 columns

In [113…
```python
# droping unnamed:0 because it is repenting the index number
```

In [114…
```python
file.drop('Unnamed: 0', axis =1 ,inplace =True)
```

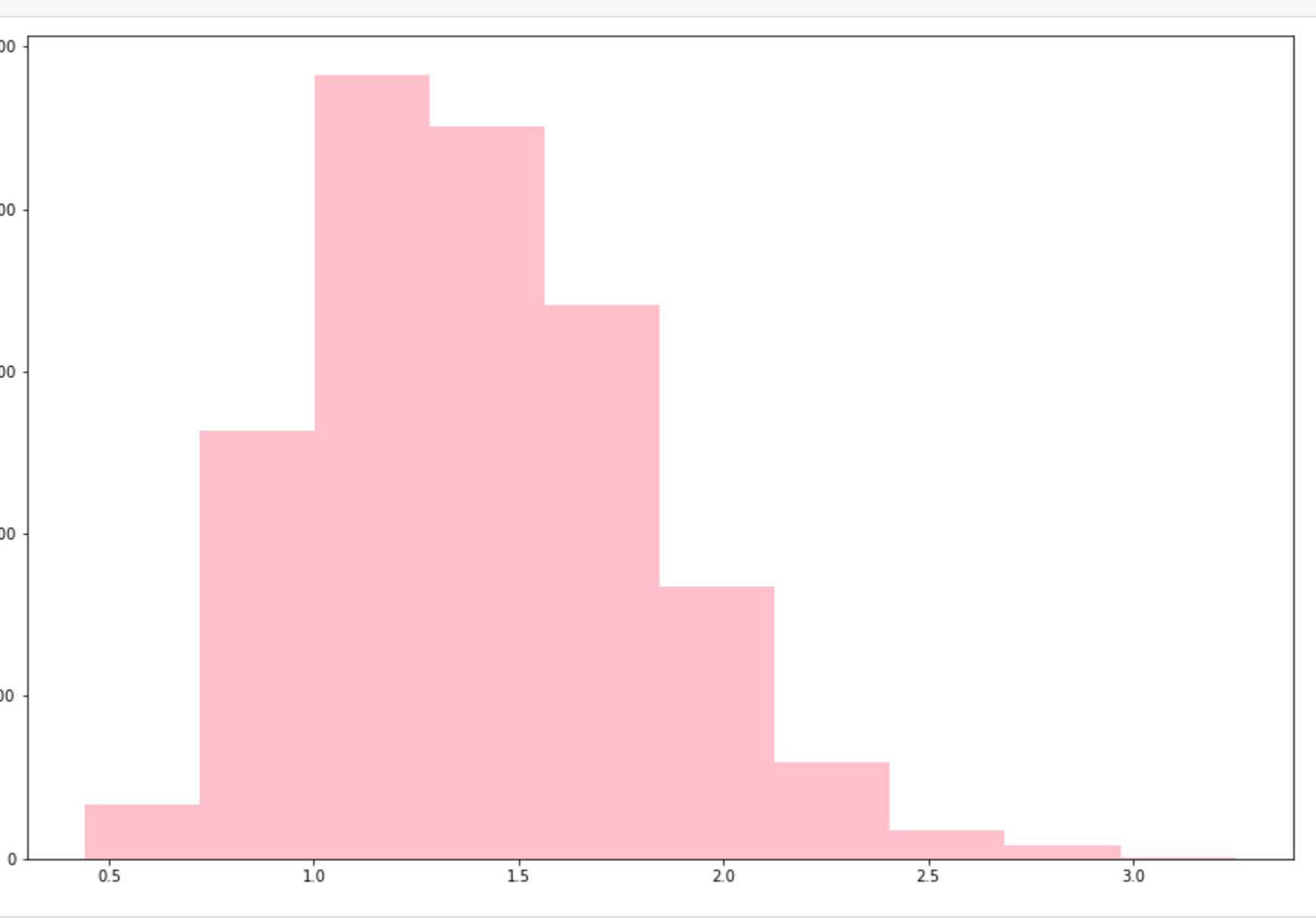
In [115…
```python
file.head(5)
```

Out[115…

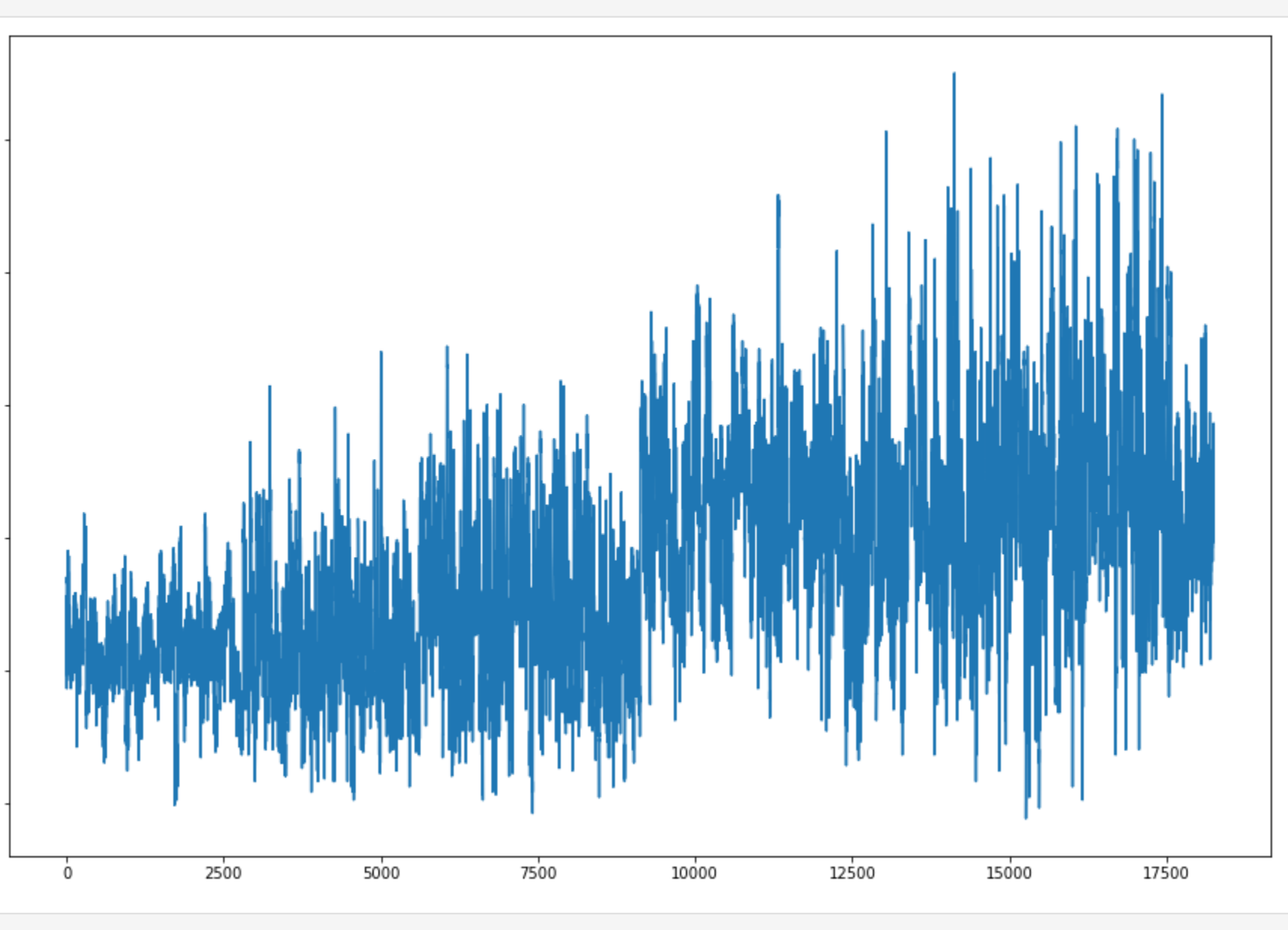
	Date	AveragePrice	Total Volume	4046	4225	4770	Total Bags	Small Bags	Large Bags	XLarge Bags	type	year	region
0	2015-12-27	1.33	64236.62	1036.74	54454.85	48.16	8696.87	8603.62	93.25	0.0	conventional	2015	Albany
1	2015-12-20	1.35	54876.98	674.28	44638.81	58.33	9505.56	9408.07	97.49	0.0	conventional	2015	Albany
2	2015-12-13	0.93	118220.22	794.70	109149.67	130.50	8145.35	8042.21	103.14	0.0	conventional	2015	Albany
3	2015-12-06	1.08	78992.15	1132.00	71976.41	72.58	5811.16	5677.40	133.76	0.0	conventional	2015	Albany
4	2015-11-29	1.28	51039.60	941.48	43838.39	75.78	6183.95	5986.26	197.69	0.0	conventional	2015	Albany

In [96]:
```python
# analysis of average price
```

In [80]:
```python
file.AveragePrice.plot(kind ='hist' , figsize =(15,10) ,color ="pink",)
plt.show()
```

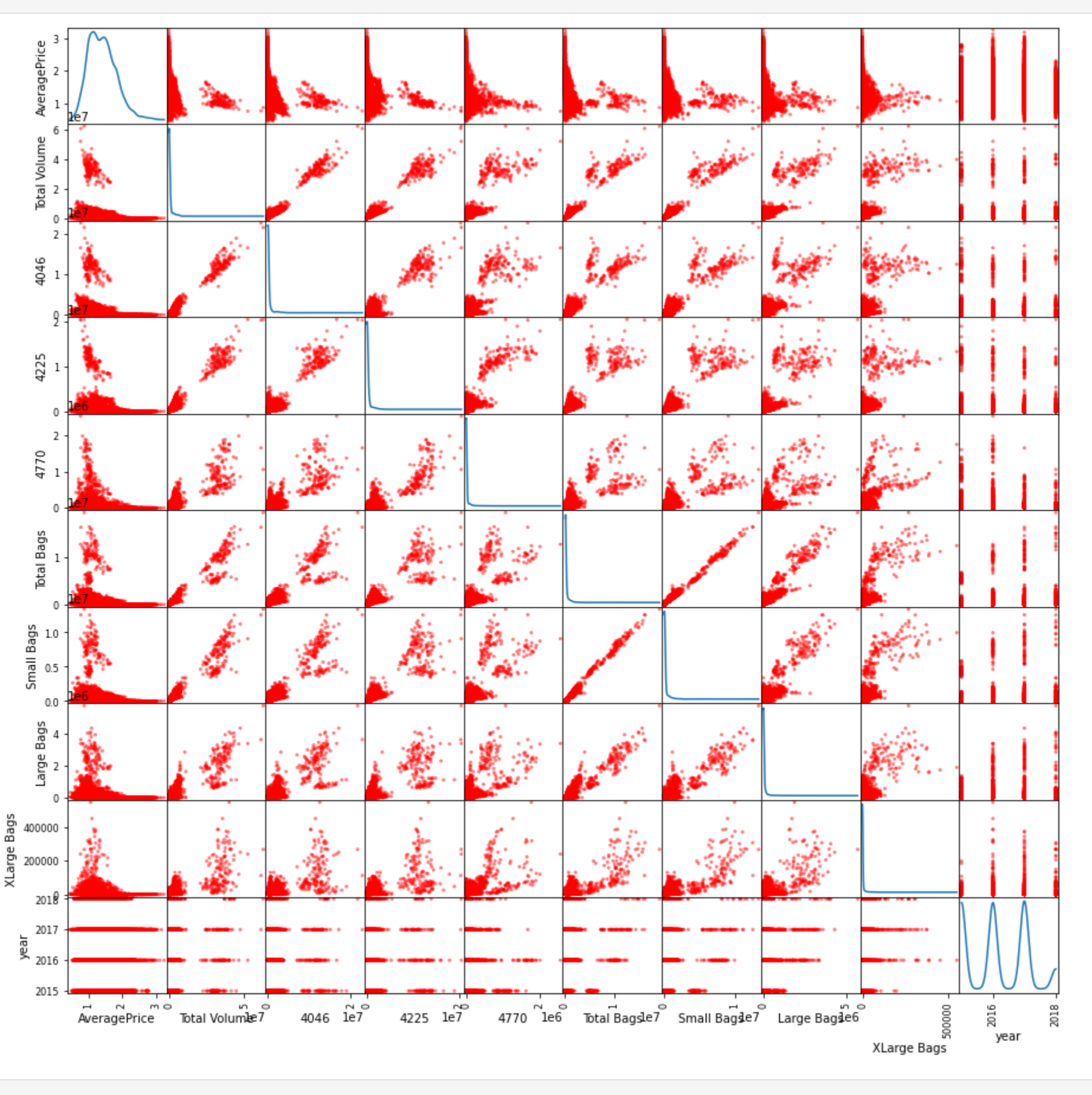


In [81]:
```python
plt.figure(figsize =(15,10))
plt.plot(file['AveragePrice'])
plt.show()
```



In [82]:
```python
from pandas.plotting import scatter_matrix
```

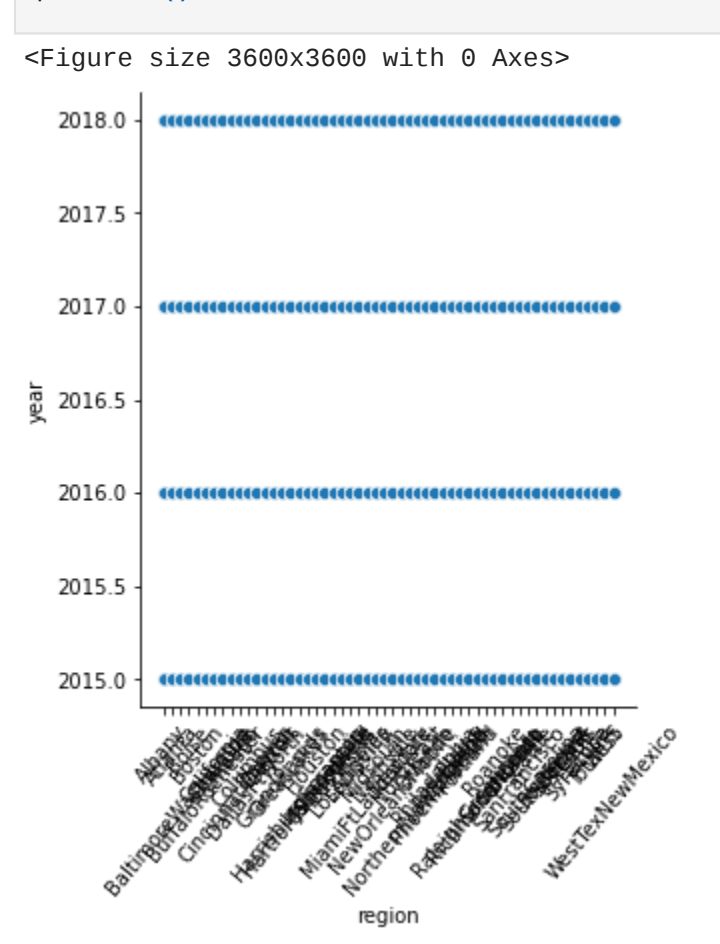
In [83]:
```python
scatter_matrix(file , figsize=(15,15) ,diagonal="kde" ,color="r")
plt.show()
```



In [84]:
```python
# indicating number of regions

plt.figure(figsize =(50,50))
sns.relplot( x ='region' , y = 'year' , data =file)
plt.xticks(rotation = 50)
plt.show()
```

```
<Figure size 3600x3600 with 0 Axes>
```



In [85]:
```python
file.head()
```

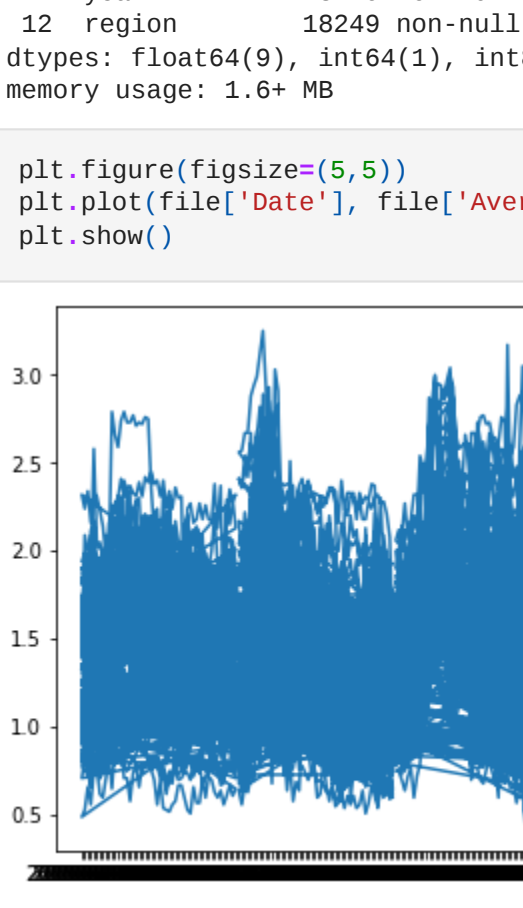
Out[85]:

	Date	AveragePrice	Total Volume	4046	4225	4770	Total Bags	Small Bags	Large Bags	XLarge Bags	type	year	region
0	2015-12-27	1.33	64236.62	1036.74	54454.85	48.16	8696.87	8603.62	93.25	0.0	conventional	2015	Albany
1	2015-12-20	1.35	54876.98	674.28	44638.81	58.33	9505.56	9408.07	97.49	0.0	conventional	2015	Albany
2	2015-12-13	0.93	118220.22	794.70	109149.67	130.50	8145.35	8042.21	103.14	0.0	conventional	2015	Albany
3	2015-12-06	1.08	78992.15	1132.00	71976.41	72.58	5811.16	5677.40	133.76	0.0	conventional	2015	Albany
4	2015-11-29	1.28	51039.60	941.48	43838.39	75.78	6183.95	5986.26	197.69	0.0	conventional	2015	Albany

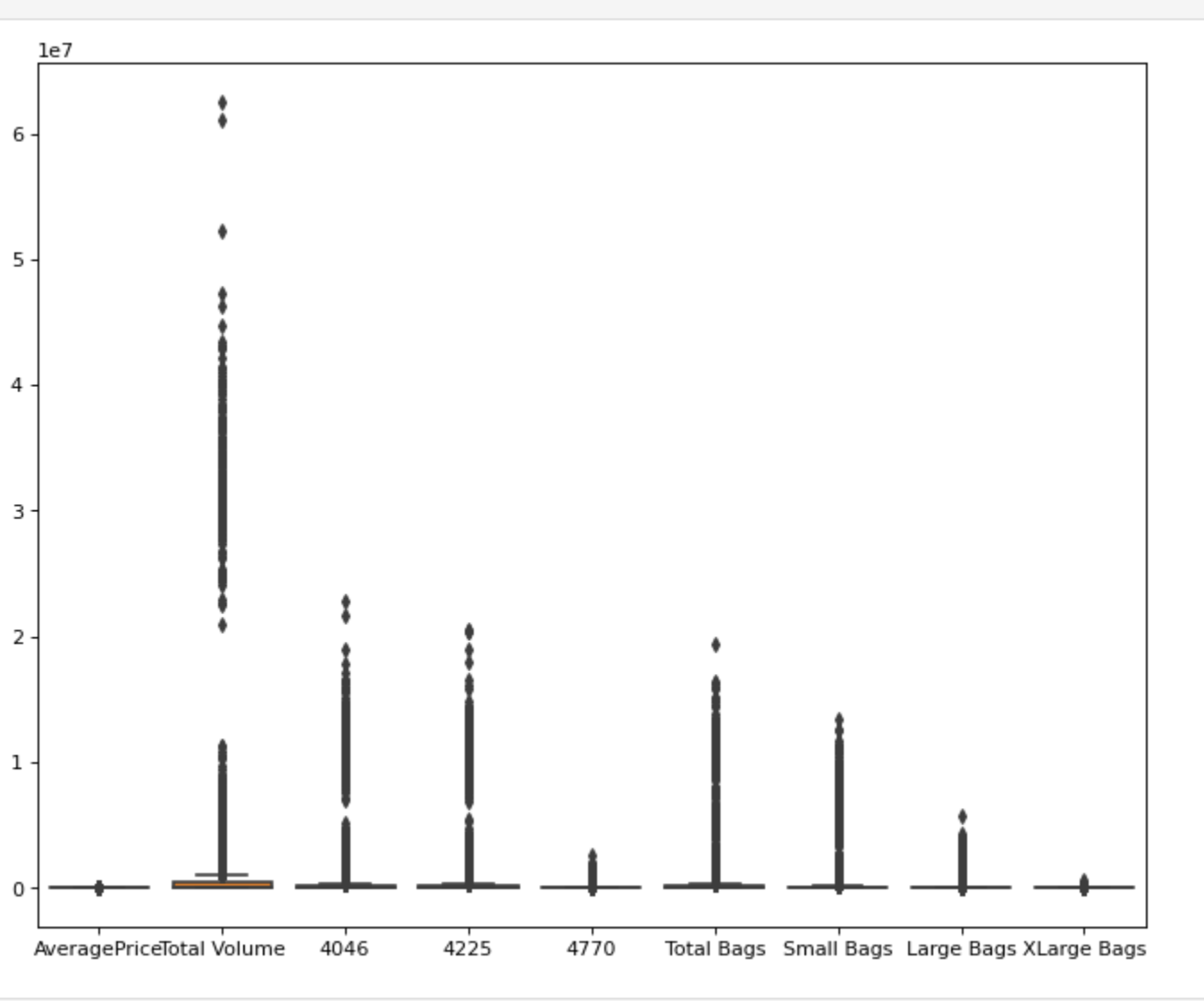
In [50]:
```python
file.info()
```

```
&lt;class 'pandas.core.frame.DataFrame'&gt;
RangeIndex: 18249 entries, 0 to 18248
Data columns (total 13 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Date          18249 non-null  object
 1   AveragePrice  18249 non-null  float64
 2   Total Volume  18249 non-null  float64
 3   4046          18249 non-null  float64
 4   4225          18249 non-null  float64
 5   4770          18249 non-null  float64
 6   Total Bags    18249 non-null  float64
 7   Small Bags    18249 non-null  float64
 8   Large Bags    18249 non-null  float64
 9   XLarge Bags   18249 non-null  float64
 10  type          18249 non-null  int8
 11  year          18249 non-null  int64
 12  region        18249 non-null  int8
dtypes: float64(9), int64(1), int8(2), object(1)
memory usage: 1.6+ MB
```

In [98]:
```python
plt.figure(figsize=(5,5))
plt.plot(file['Date'], file['AveragePrice'])
plt.show()
```



In [101…
```python
plt.figure(figsize=(10,8),dpi=80)
sns.boxplot(data = file[[
 'AveragePrice',
 'Total Volume',
 '4046',
 '4225',
 '4770',
 'Total Bags',
 'Small Bags',
 'Large Bags',
 'XLarge Bags']])
plt.show()
```



In [116…
```python
file.drop(columns=["Date"],inplace=True)
file.info()
```

```
&lt;class 'pandas.core.frame.DataFrame'&gt;
RangeIndex: 18249 entries, 0 to 18248
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   AveragePrice  18249 non-null  float64
 1   Total Volume  18249 non-null  float64
 2   4046          18249 non-null  float64
 3   4225          18249 non-null  float64
 4   4770          18249 non-null  float64
 5   Total Bags    18249 non-null  float64
 6   Small Bags    18249 non-null  float64
 7   Large Bags    18249 non-null  float64
 8   XLarge Bags   18249 non-null  float64
 9   type          18249 non-null  object
 10  year          18249 non-null  int64
 11  region        18249 non-null  object
dtypes: float64(9), int64(1), object(2)
memory usage: 1.7+ MB
```

In [117…
```python
from numpy import percentile

columns = file.columns
for i in columns:
    if isinstance(file[i][0], str) :
        continue
    else:
        #defining quartiles
        quartiles = percentile(file[i], [20,60])
        # calculate min/max
        lower_fence = quartiles[0] - (1.5*(quartiles[1]-quartiles[0]))
        upper_fence = quartiles[1] + (1.5*(quartiles[1]-quartiles[0]))
        file[i] = file[i].apply(lambda x: upper_fence if x > upper_fence else (lower_fence if x < lower_fence else x))
```

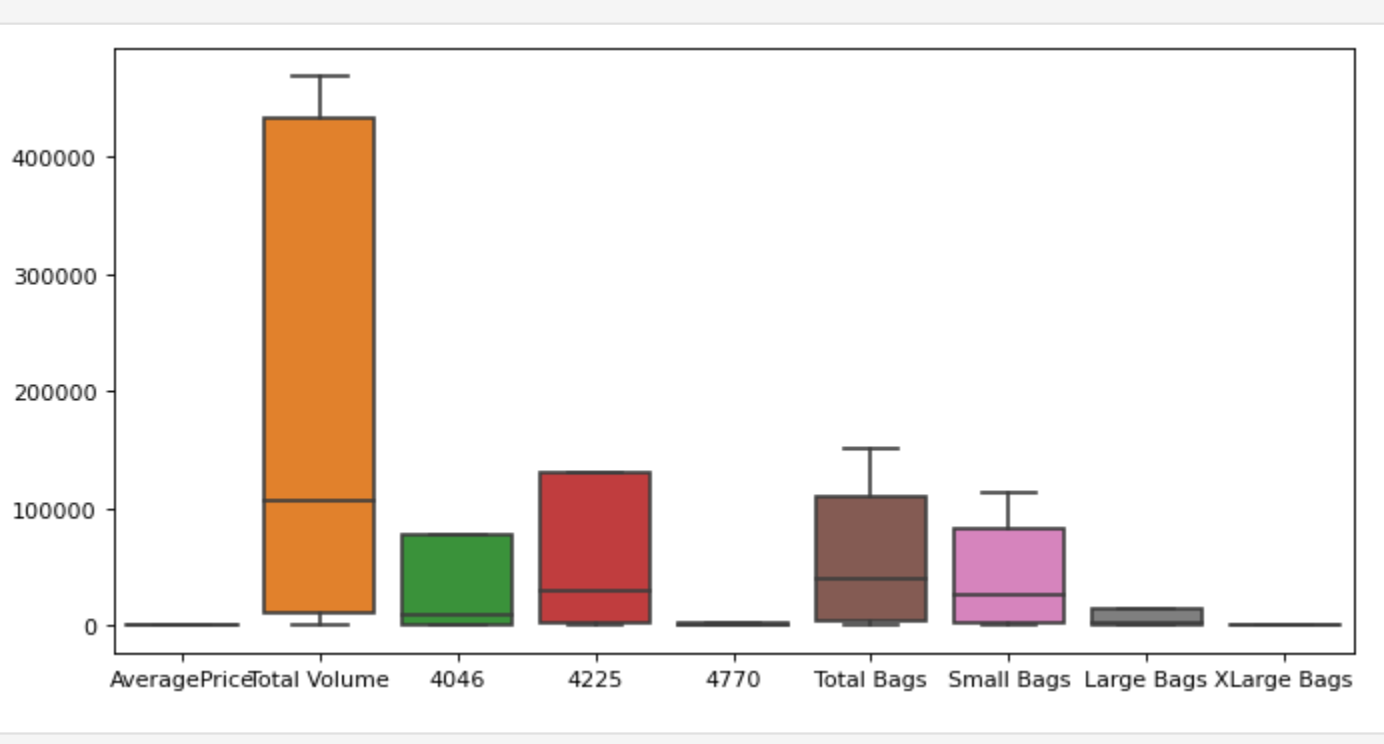
In [118…
```python
file.head()
```

Out[118…

	AveragePrice	Total Volume	4046	4225	4770	Total Bags	Small Bags	Large Bags	XLarge Bags	type	year	region
0	1.33	64236.62	1036.74	54454.85	48.16	8696.87	8603.62	93.25	0.0	conventional	2015.0	Albany
1	1.35	54876.98	674.28	44638.81	58.33	9505.56	9408.07	97.49	0.0	conventional	2015.0	Albany
2	0.93	118220.22	794.70	109149.67	130.50	8145.35	8042.21	103.14	0.0	conventional	2015.0	Albany
3	1.08	78992.15	1132.00	71976.41	72.58	5811.16	5677.40	133.76	0.0	conventional	2015.0	Albany
4	1.28	51039.60	941.48	43838.39	75.78	6183.95	5986.26	197.69	0.0	conventional	2015.0	Albany

In [120…
```python
# now we can see that data is free from outliers

plt.figure(figsize=(10,5),dpi=80)
sns.boxplot(data = file[[
 'AveragePrice',
 'Total Volume',
 '4046',
 '4225',
 '4770',
 'Total Bags',
 'Small Bags',
 'Large Bags',
 'XLarge Bags']])
plt.show()
```



In [121…
```python
file['region'] = pd.Categorical(file['region'])
dfDummies_region = pd.get_dummies(file['region'], prefix = 'region')
dfDummies_region
```

Out[121…

	region_Albany	region_Atlanta	region_BaltimoreWashington	region_Boise	region_Boston	region_BuffaloRochester	region_California	region_Charlotte	region_Chicago	region_CincinnatiDayton	...	region_SouthCarolina	region_South
0	1	0	0	0	0	0	0	0	0	0	...	0	
1	1	0	0	0	0	0	0	0	0	0	...	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
18247	0	0	0	0	0	0	0	0	0	0	...	0	
18248	0	0	0	0	0	0	0	0	0	0	...	0	

18249 rows × 54 columns

In [122…
```python
dataset = pd.concat([file, dfDummies_region], axis=1)
dataset.drop(columns="region",inplace=True)
dataset
```

Out[122…

	AveragePrice	Total Volume	4046	4225	4770	Total Bags	Small Bags	Large Bags	XLarge Bags	type	...	region_SouthCarolina	region_SouthCentral	region_Southeast	region_Spokane	region_StLouis	region_Syracuse	region_Tampa
0	1.33	64236.62	1036.74	54454.85	48.16	8696.87	8603.62	93.25	0.0	conventional	...	0	0	0	0	0	0	0
1	1.35	54876.98	674.28	44638.81	58.33	9505.56	9408.07	97.49	0.0	conventional	...	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
18247	1.93	16205.22	1527.63	2981.04	727.01	10969.54	10919.54	50.00	0.0	organic	...	0	0	0	0	0	0	0
18248	1.62	17489.58	2894.77	2356.13	224.53	12014.15	11988.14	26.01	0.0	organic	...	0	0	0	0	0	0	0

18249 rows × 65 columns

In [126…
```python
dataset
```

Out[126…

	AveragePrice	Total Volume	4046	4225	4770	Total Bags	Small Bags	Large Bags	XLarge Bags	type	...	region_SouthCarolina	region_SouthCentral	region_Southeast	region_Spokane	region_StLouis	region_Syracuse	region_Tampa
0	1.33	64236.62	1036.74	54454.85	48.16	8696.87	8603.62	93.25	0.0	conventional	...	0	0	0	0	0	0	0
1	1.35	54876.98	674.28	44638.81	58.33	9505.56	9408.07	97.49	0.0	conventional	...	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
18247	1.93	16205.22	1527.63	2981.04	727.01	10969.54	10919.54	50.00	0.0	organic	...	0	0	0	0	0	0	0
18248	1.62	17489.58	2894.77	2356.13	224.53	12014.15	11988.14	26.01	0.0	organic	...	0	0	0	0	0	0	0

18249 rows × 65 columns

In [127…
```python
from sklearn import preprocessing

label_encoder = preprocessing.LabelEncoder()
dataset['type']= label_encoder.fit_transform(dataset['type'])
dataset
```

Out[127…

	AveragePrice	Total Volume	4046	4225	4770	Total Bags	Small Bags	Large Bags	XLarge Bags	type	...	region_SouthCarolina	region_SouthCentral	region_Southeast	region_Spokane	region_StLouis	region_Syracuse	region_Tampa	regio
0	1.33	64236.62	1036.74	54454.85	48.16	8696.87	8603.62	93.25	0.0	0	...	0	0	0	0	0	0	0	
1	1.35	54876.98	674.28	44638.81	58.33	9505.56	9408.07	97.49	0.0	0	...	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
18247	1.93	16205.22	1527.63	2981.04	727.01	10969.54	10919.54	50.00	0.0	1	...	0	0	0	0	0	0	0	
18248	1.62	17489.58	2894.77	2356.13	224.53	12014.15	11988.14	26.01	0.0	1	...	0	0	0	0	0	0	0	

18249 rows × 65 columns

In [136…
```python
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
from sklearn.preprocessing import StandardScaler
import sklearn.metrics as metrics
import statsmodels.api as sm
from sklearn.model_selection import cross_val_score
```

In [129…
```python
# model fitting

X=dataset.iloc[:,1:78]
y=dataset['AveragePrice']

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=50)
y_test = np.array(y_test,dtype = float)
```

In [131…
```python
sc=StandardScaler()
X_train=sc.fit_transform(X_train)
X_test=sc.transform(X_test)
```

In [133…
```python
def regression_results(y_true, y_pred):
    explained_variance=metrics.explained_variance_score(y_true, y_pred)
    mean_absolute_error=metrics.mean_absolute_error(y_true, y_pred)
    mse=metrics.mean_squared_error(y_true, y_pred)
    r2=metrics.r2_score(y_true, y_pred)
    adjusted_r2 = 1 - (1-r2)*(len(y_true)-1)/(len(y_true)-X_test.shape[1]-1)

    print('Explained_variance: ', round(explained_variance,4))
    print('R2: ', round(r2,4))
    print('Adjusted_r2: ', round(adjusted_r2,4))
    print('MAE: ', round(mean_absolute_error,4))
    print('MSE: ', round(mse,4))
    print('RMSE: ', round(np.sqrt(mse),4))
```

In [137…
```python
# checking accuracy
def model_accuracy(model,X_train=X_train,y_train=y_train):
    accuracies = cross_val_score(estimator = model, X = X_train, y = y_train, cv = 10)
    print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
    print("Standard Deviation: {:.2f} %".format(accuracies.std()*100))
```

In [138…
```python
#  fitting multiple linear regression model
```

In [139…
```python
regressor=LinearRegression()
regressor.fit(X_train,y_train)
y_pred = regressor.predict(X_test)
regression_results(y_test,y_pred)
model_accuracy(regressor)
```

```
Explained_variance:  0.6294
R2:  0.6293
Adjusted_r2:  0.6227
MAE:  0.1799
MSE:  0.0537
RMSE:  0.2317
Accuracy: 60.14 %
Standard Deviation: 1.55 %
```

In [ ]:
```python
# fitting random forest regression model
```

In [140…
```python
from sklearn.ensemble import RandomForestRegressor
```

In [141…
```python
rand_regressor = RandomForestRegressor()
rand_regressor.fit(X_train, y_train)
y_pred_rf = rand_regressor.predict(X_test)
regression_results(y_test,y_pred_rf)
model_accuracy(rand_regressor)
```

```
Explained_variance:  0.8504
R2:  0.8503
Adjusted_r2:  0.8476
MAE:  0.1077
MSE:  0.0217
RMSE:  0.1472
Accuracy: 83.40 %
Standard Deviation: 0.81 %
```

In [ ]:
```python
# fitting decision tree regression model
```

In [142…
```python
from sklearn.tree import DecisionTreeRegressor
```

In [143…
```python
decision_tree=DecisionTreeRegressor(criterion='mse',splitter='random',random_state=10)
decision_tree.fit(X_train, y_train)
y_pred_dt = decision_tree.predict(X_test)
regression_results(y_test,y_pred_dt)
model_accuracy(decision_tree)
```

```
Explained_variance:  0.7601
R2:  0.7597
Adjusted_r2:  0.7555
MAE:  0.1264
MSE:  0.0348
RMSE:  0.1865
Accuracy: 73.57 %
Standard Deviation: 1.76 %
```

In [147…
```python
# spliting data into X and y
X = dataset.drop(['AveragePrice'],axis=1)
y =dataset['AveragePrice']
```

In [148…
```python
# spliting data into traing and testing dataset
X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    test_size=0.5,
                                                    random_state=10)
```

In [149…
```python
print("training set:",X_train.shape,' :: ','samples ::',y_train.shape[0])
print("testing set:",X_test.shape,' ::','samples ::',y_test.shape[0])
```

```
training set: (9124, 64)  ::  samples :: 9124
testing set: (9125, 64)  :: samples :: 9125
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: