In [93]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder,StandardScaler
from sklearn.linear_model import LinearRegression,Lasso
from sklearn.metrics import mean_squared_error,mean_absolute_error
from sklearn.ensemble import RandomForestRegressor
import warnings
warnings.filterwarnings("ignore")
```

In [94]:
```
Bigmac=pd.read_csv(r'C:\Users\18f18004\Desktop\BigmacPrice.csv')
```

In [95]:
```
Bigmac.head()
```

Out[95]:

	currency_code	name	local_price	dollar_ex	dollar_price
0	ARS	Argentina	2.50	1	2.50
1	AUD	Australia	NaN	1	2.59
2	BRL	Brazil	2.95	1	2.95
3	GBP	Britain	1.90	1	1.90
4	CAD	Canada	NaN	1	2.85

In [96]:
```
Bigmac.shape
```

Out[96]:
```
(1946, 5)
```

In [97]:
```
Bigmac.isnull().sum()
```

Out[97]:
```
currency_code    0
name             0
local_price      8
dollar_ex        0
dollar_price     0
dtype: int64
```

In [98]:
```
Bigmac=Bigmac.dropna()
```

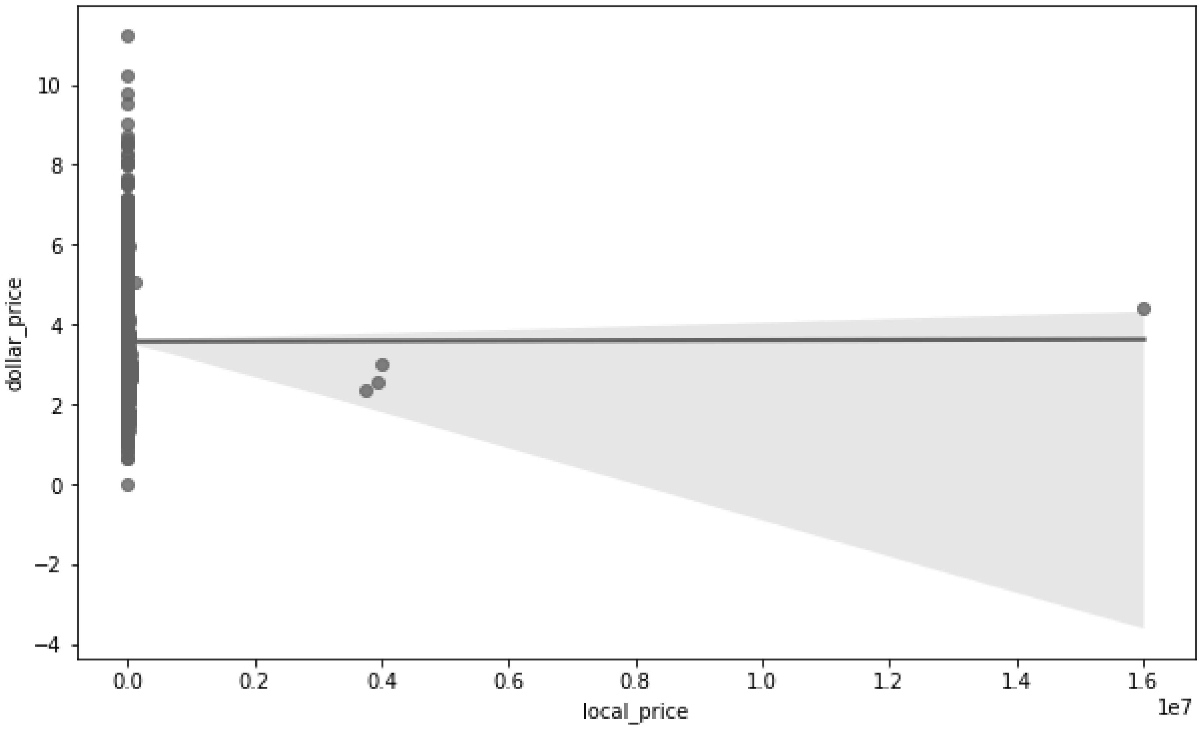
In [99]:
```
Bigmac.isnull().sum()
```

Out[99]:
```
currency_code    0
name             0
local_price      0
dollar_ex        0
dollar_price     0
dtype: int64
```

```
In [100]: Bigmac.shape

Out[100]: (1938, 5)
```

```
In [101]: Bigmac.dtypes

Out[101]: currency_code     object
          name              object
          local_price      float64
          dollar_ex          int64
          dollar_price     float64
          dtype: object
```

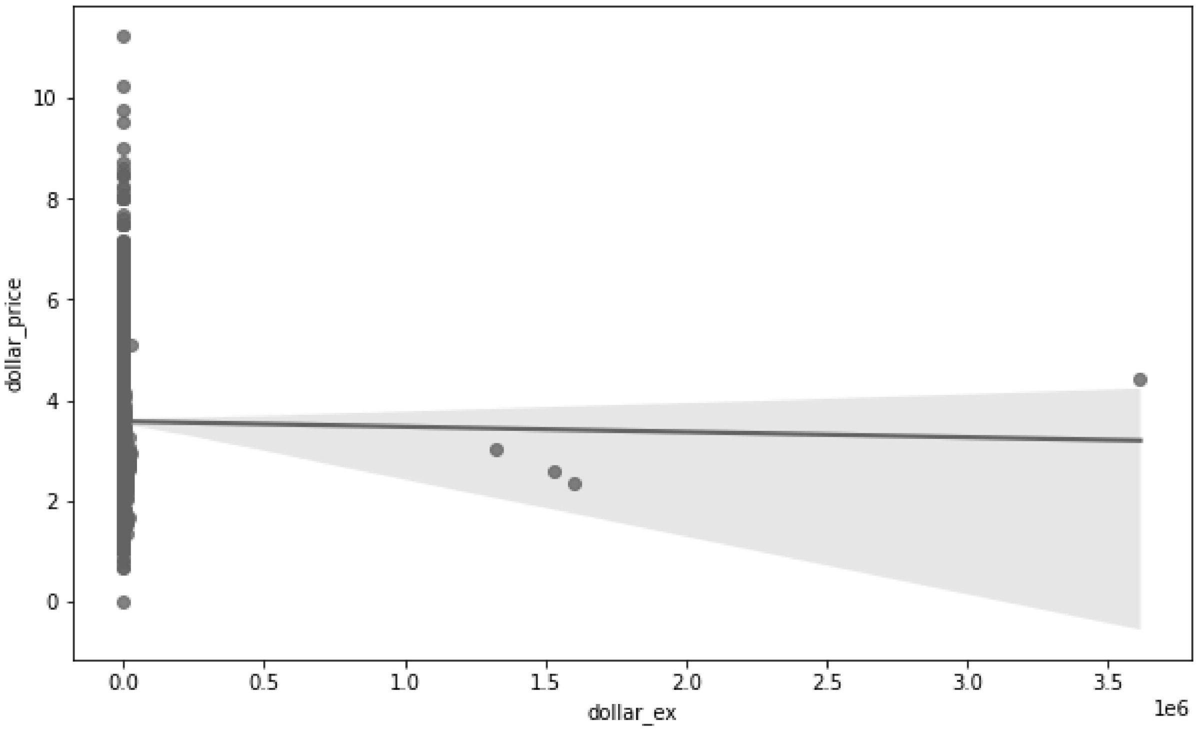
```
In [102]: plt.figure(figsize=(10,6))
          sns.regplot(x="local_price", y="dollar_price", data=Bigmac)

Out[102]: <AxesSubplot:xlabel='local_price', ylabel='dollar_price'>
```



```
In [103]: from scipy import stats
          pearson_coef, p_value = stats.pearsonr(Bigmac['dollar_ex'],Bigmac['dollar_price']
          print("The Pearson Correlation Coefficient is", pearson_coef, " with a P-value of

          The Pearson Correlation Coefficient is -0.007493347114254968  with a P-value of
          P = 0.7416498319793564
```

```
In [104]: plt.figure(figsize=(10,6))
          sns.regplot(x="dollar_ex", y="dollar_price", data=Bigmac)
```

```
Out[104]: <AxesSubplot:xlabel='dollar_ex', ylabel='dollar_price'>
```

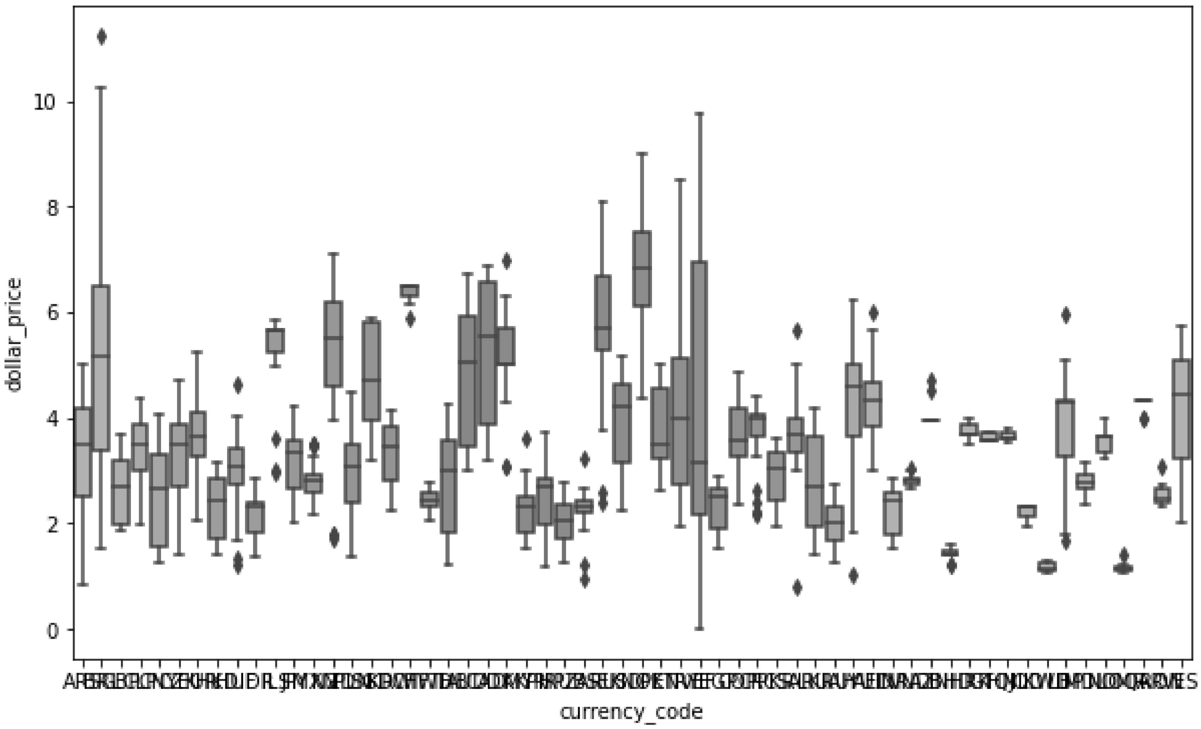


In [105]:

```
plt.figure(figsize=(10,6))
sns.boxplot(x="currency_code", y="dollar_price", data=Bigmac)
```

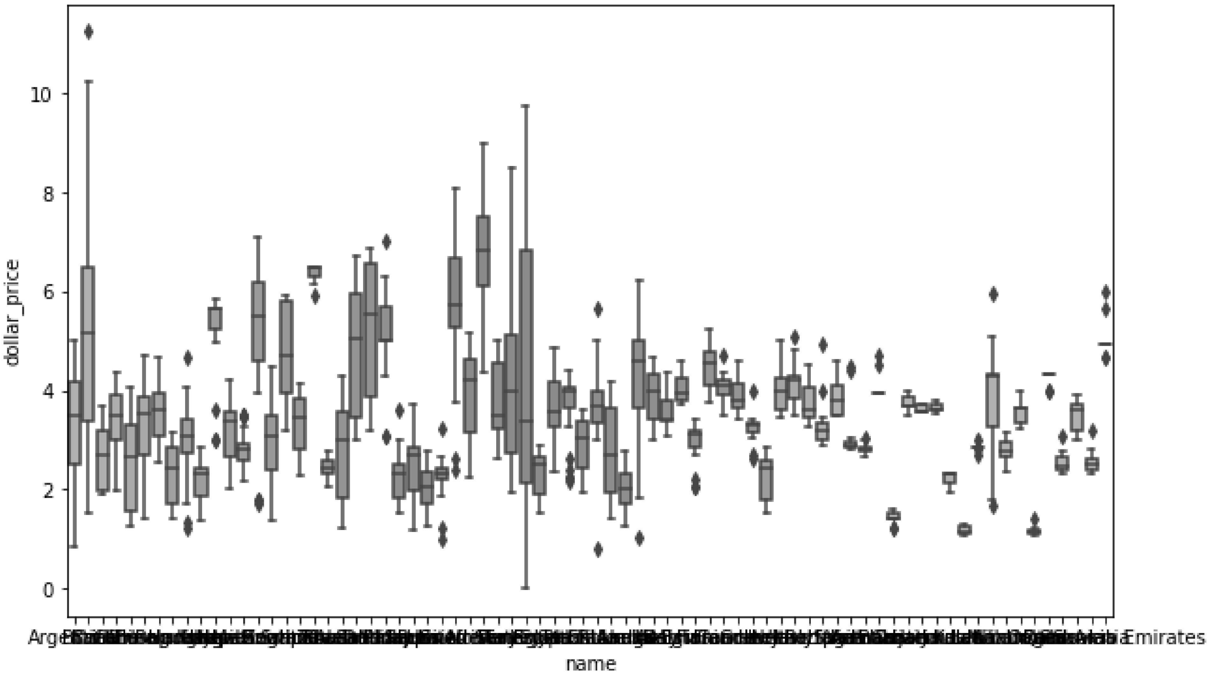
Out[105]:

```
<AxesSubplot:xlabel='currency_code', ylabel='dollar_price'>
```



```
In [106]: plt.figure(figsize=(10,6))
          sns.boxplot(x="name", y="dollar_price", data=Bigmac)
```

```
Out[106]: <AxesSubplot:xlabel='name', ylabel='dollar_price'>
```



```
In [ ]:
```

```
In [ ]:
```

Type *Markdown* and LaTeX: α^2

```
In [107]: Bigmac.shape
```

```
Out[107]: (1938, 5)
```

In [108]:
```
Bigmac.describe()
```

Out[108]:

	local_price	dollar_ex	dollar_price
count	1.938000e+03	1.938000e+03	1938.000000
mean	1.588132e+04	4.741720e+03	3.573354
std	3.948165e+05	1.008304e+05	1.416882
min	0.000000e+00	1.000000e+00	0.000000
25%	4.450000e+00	1.000000e+00	2.580000
50%	1.500000e+01	5.000000e+00	3.400000
75%	8.850000e+01	3.200000e+01	4.250000
max	1.602000e+07	3.613989e+06	11.250000

In [109]:
```
Bigmac.describe(include=['object'])
```

Out[109]:

	currency_code	name
count	1938	1938
unique	58	74
top	EUR	Argentina
freq	351	37

In [110]:
```
labelencoder = LabelEncoder()
Bigmac.currency_code= labelencoder.fit_transform(Bigmac.currency_code)
Bigmac.name = labelencoder.fit_transform(Bigmac.name)
```

In [111]:
```
Bigmac.head(10)
```

Out[111]:

	currency_code	name	local_price	dollar_ex	dollar_price
0	1	0	2.50	1	2.50
2	5	6	2.95	1	2.95
3	16	7	1.90	1	1.90
5	8	9	1260.00	514	2.45
6	9	10	9.90	8	1.24
7	12	14	54.37	39	1.39
9	15	18	2.56	1	2.56
10	18	25	10.20	7	1.46
11	21	26	339.00	279	1.22
12	22	28	14500.00	7945	1.83

In [112]:
```
import scipy.stats as stats
Bigmac = stats.zscore(Bigmac)
Bigmac = stats.zscore(Bigmac)
```

In [113]:
```
Bigmac
```

Out[113]:

	currency_code	name	local_price	dollar_ex	dollar_price
0	-1.526485	-1.653458	-0.040229	-0.047029	-0.757742
2	-1.281433	-1.381295	-0.040227	-0.047029	-0.440061
3	-0.607541	-1.335934	-0.040230	-0.047029	-1.181317
5	-1.097644	-1.245213	-0.037043	-0.041940	-0.793040
6	-1.036381	-1.199853	-0.040210	-0.046959	-1.647248
...	...	...	...	...	...
1941	-1.587748	1.476416	-0.040189	-0.047009	1.713109
1942	1.597927	1.521777	-0.040222	-0.047029	1.113045
1943	1.659190	1.567137	-0.039589	-0.046632	1.868419
1944	1.781716	1.612498	-0.040210	-0.046989	-1.110721
1945	1.842979	1.657858	0.134575	0.185263	-0.440061

1938 rows × 5 columns

In [114]:
```
x_train=Bigmac.iloc[:,0:4]
y_train=Bigmac.iloc[:,4]
x_test=Bigmac.iloc[:,0:4]
y_test=Bigmac.iloc[:,4]
```

In [115]:

```
x_train
```

Out[115]:

	currency_code	name	local_price	dollar_ex
0	-1.526485	-1.653458	-0.040229	-0.047029
2	-1.281433	-1.381295	-0.040227	-0.047029
3	-0.607541	-1.335934	-0.040230	-0.047029
5	-1.097644	-1.245213	-0.037043	-0.041940
6	-1.036381	-1.199853	-0.040210	-0.046959
...	...	...	...	...
1941	-1.587748	1.476416	-0.040189	-0.047009
1942	1.597927	1.521777	-0.040222	-0.047029
1943	1.659190	1.567137	-0.039589	-0.046632
1944	1.781716	1.612498	-0.040210	-0.046989
1945	1.842979	1.657858	0.134575	0.185263

1938 rows × 4 columns

In [116]:

```
rg = LinearRegression()
mdl=rg.fit(x_train,y_train)
```

In [117]:

```
y_pred1 = rg.predict(x_test)
```

In [118]:

```
print('The R-square for Multiple Linear regression is: ',
rg.score(x_train,y_train))
```

```
The R-square for Multiple Linear regression is:  0.04722870369435617
```

In [119]:

```
mse1 = mean_squared_error(y_test, y_pred1)
print('The mean square error for Multiple Linear Regression: ', mse1)
```

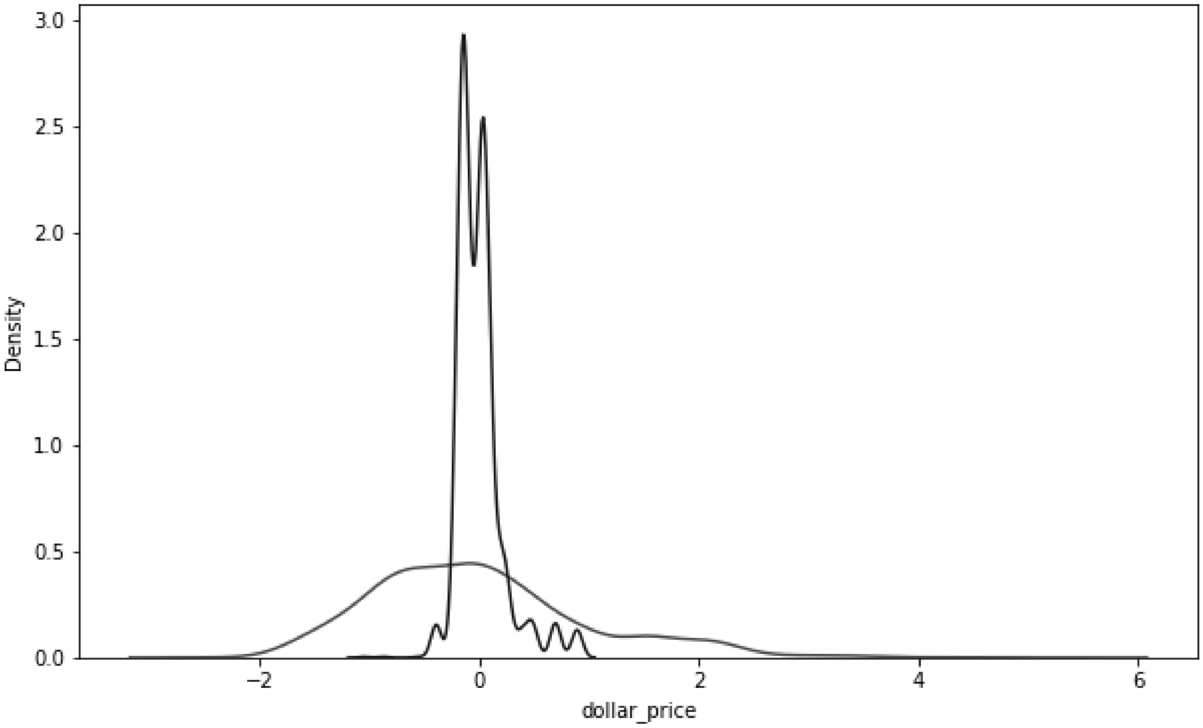
```
The mean square error for Multiple Linear Regression:  0.952771296305644
```

In [120]:

```
mae1= mean_absolute_error(y_test, y_pred1)
print('The mean absolute error for Multiple Linear Regression: ', mae1)
```

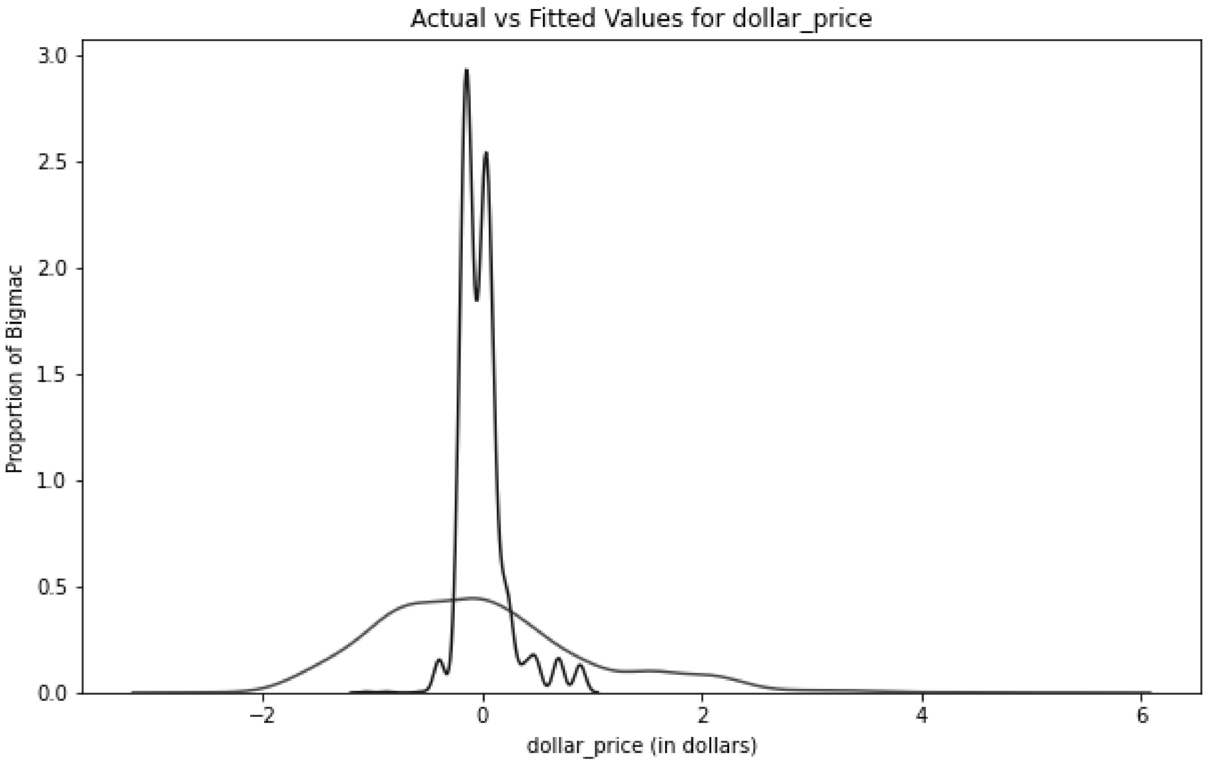
```
The mean absolute error for Multiple Linear Regression:  0.7441704806317442
```

```
In [121]: plt.figure(figsize=(10,6))
          ax1 = sns.distplot(y_test, hist=False, color="r", label="Actual Value")
          sns.distplot(y_pred1, hist=False, color="b", label="Fitted Values" , ax=ax1)

Out[121]: <AxesSubplot:xlabel='dollar_price', ylabel='Density'>
```



In [123]:

```
plt.figure(figsize=(10,6))
ax1 = sns.distplot(y_test, hist=False, color="r", label="Actual Value")
sns.distplot(y_pred1, hist=False, color="b", label="Fitted Values" , ax=ax1)
plt.title('Actual vs Fitted Values for dollar_price')
plt.xlabel('dollar_price (in dollars)')
plt.ylabel('Proportion of Bigmac')
plt.show()
plt.close()
```



In [ ]:

```
In [ ]:
```

```
In [124]: rf = RandomForestRegressor()
          model=rf.fit(x_train,y_train)
```

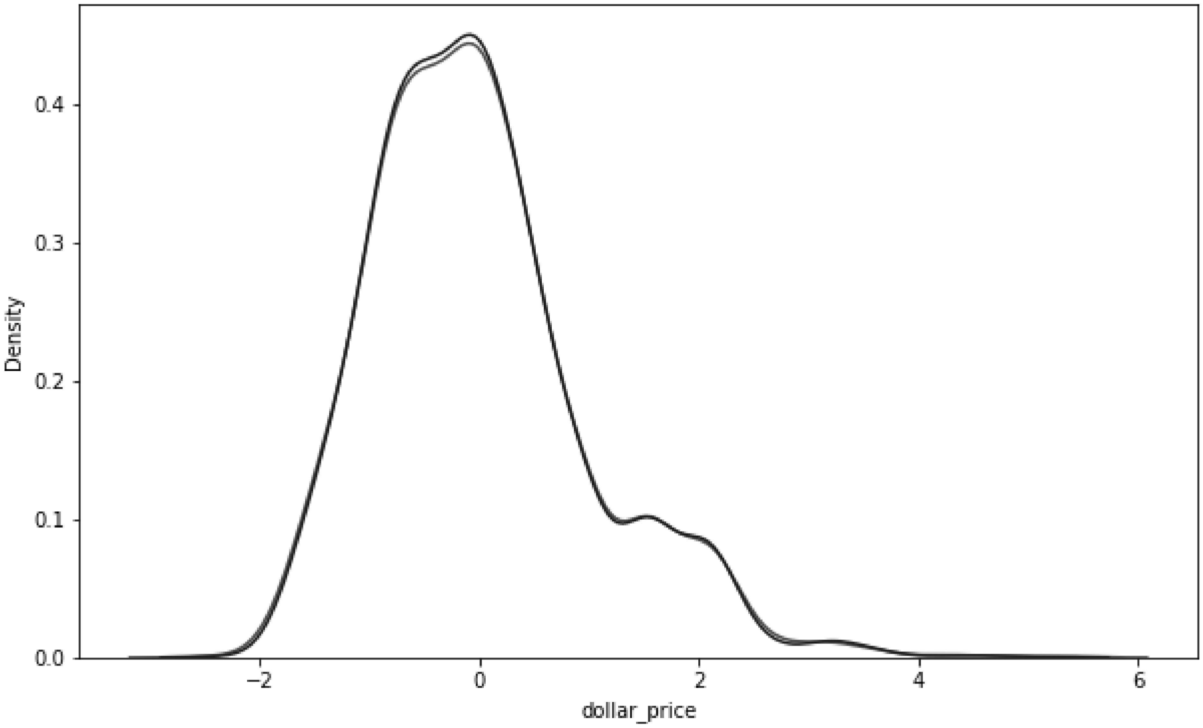
```
In [125]: y_pred2 = rf.predict(x_test)
```

```
In [126]: print('The R-square for Random Forest is: ', rf.score(x_train,y_train))

          The R-square for Random Forest is:  0.9971163449675606
```

```
In [127]: mse2 = mean_squared_error(y_test, y_pred2)
          print('The mean square error of price and predicted value is: ', mse2)

          The mean square error of price and predicted value is:  0.0028836550324394064
```

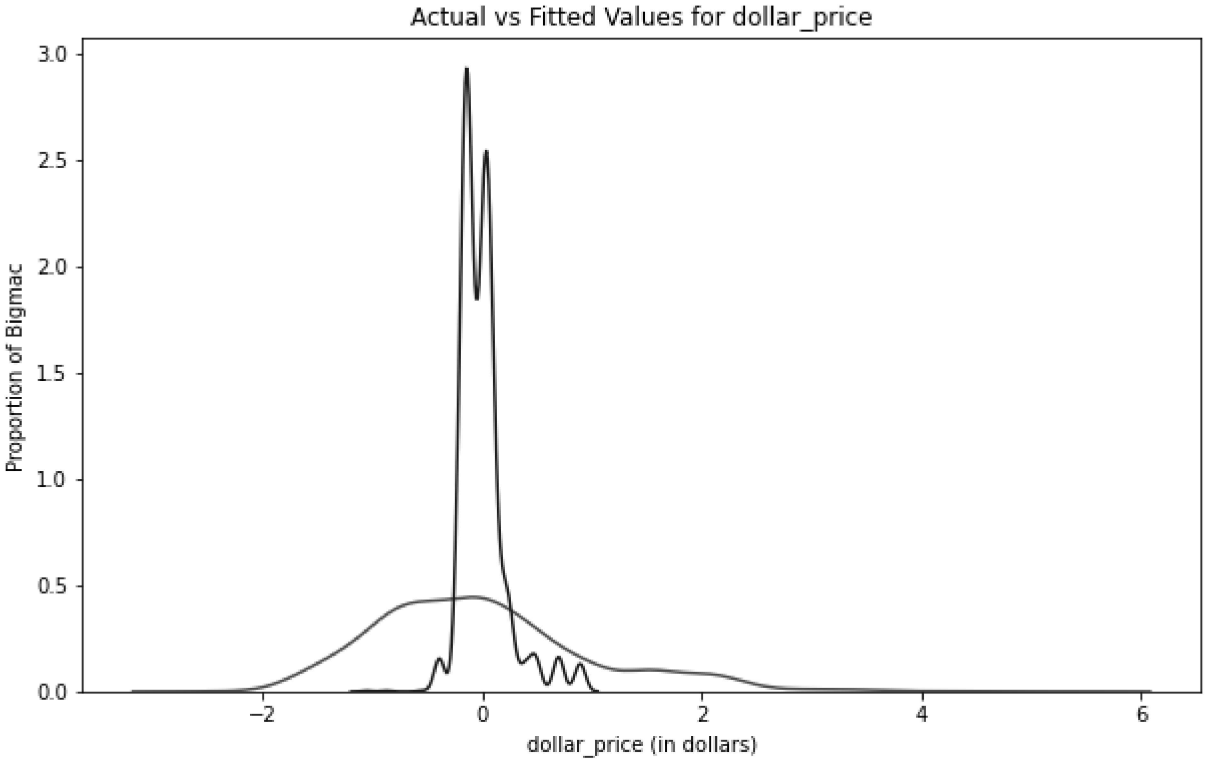
```
In [128]: mae2= mean_absolute_error(y_test, y_pred2)
          print('The mean absolute error of price and predicted value is: ', mae2)

          The mean absolute error of price and predicted value is:  0.020697227814428117
```

```
In [129]: plt.figure(figsize=(10,6))
          ax1 = sns.distplot(y_test, hist=False, color="r", label="Actual Value")
          sns.distplot(y_pred2, hist=False, color="b", label="Fitted Values" , ax=ax1)

Out[129]: <AxesSubplot:xlabel='dollar_price', ylabel='Density'>
```



In [132]:

```
plt.figure(figsize=(10,6))
ax1 = sns.distplot(y_test, hist=False, color="r", label="Actual Value")
sns.distplot(y_pred1, hist=False, color="b", label="Fitted Values" , ax=ax1)
plt.title('Actual vs Fitted Values for dollar_price')
plt.xlabel('dollar_price (in dollars)')
plt.ylabel('Proportion of Bigmac')
plt.show()
plt.close()
```



In [133]:

```
LassoModel=Lasso()
lm=LassoModel.fit(x_train,y_train)
```

In [134]:

```
y_pred3 = lm.predict(x_test)
```

In [135]:

```
print('The R-square for LASSO is: ', lm.score(x_train,y_train))
```

```
The R-square for LASSO is:  0.0
```

In [136]:

```
mae3= mean_absolute_error(y_test, y_pred3)
print('The mean absolute error of price and predicted value is: ', mae3)
```

```
The mean absolute error of price and predicted value is:  0.7653887100012792
```

In [137]:

```
mse3 = mean_squared_error(y_test, y_pred3)
print('The mean square error of price and predicted value is: ', mse3)
```

```
The mean square error of price and predicted value is:  1.0000000000000002
```

In [138]:

```
scores = [('MLR', mae1),
 ('Random Forest', mae2),
 ('LASSO', mae3) ]
```

In [139]:

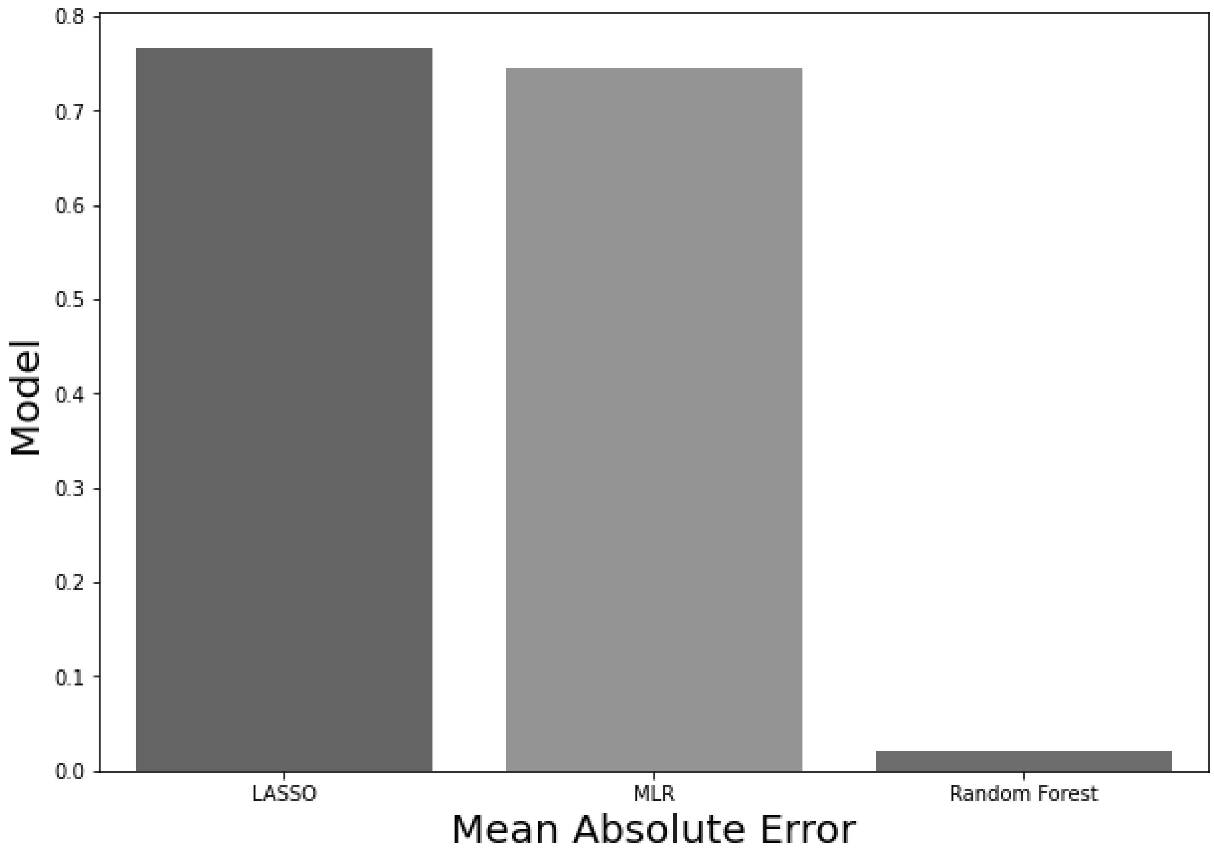
```
mae = pd.DataFrame(data = scores, columns=['Model', 'MAE Score'])
mae
```

Out[139]:

	Model	MAE Score
0	MLR	0.744170
1	Random Forest	0.020697
2	LASSO	0.765389

In [140]:

```
mae.sort_values(by=(['MAE Score']), ascending=False, inplace=True)
f, axe = plt.subplots(1,1, figsize=(10,7))
sns.barplot(x = mae['Model'], y=mae['MAE Score'], ax = axe)
axe.set_xlabel('Mean Absolute Error', size=20)
axe.set_ylabel('Model', size=20)
plt.show()
```



In [ ]: