```
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")
```

```
laptopPrice=pd.read_csv('laptop_prices.csv')
laptopPrice.head()
```

	laptop_ID	Company	Product	TypeName	Inches	ScreenResolution	Cpu	Ram	Memo
0	1	Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8GB	128G SS
1	2	Apple	Macbook Air	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8GB	128G Flas Storag
2	3	HP	250 G6	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8GB	256G SS

```
laptopPrice.shape
```

```
(1303, 13)
```

```
laptopPrice.isnull().sum()
```

```
laptop_ID           0
Company             1
Product             1
TypeName            1
Inches              0
ScreenResolution    1
Cpu                 2
Ram                 0
Memory              3
Gpu                 1
OpSys               1
Weight              0
Price_euros         0
dtype: int64
```

```
laptopPrice=laptopPrice.dropna()
```

```
laptopPrice.isnull().sum()
```

```
laptop_ID           0
Company             0
Product             0
TypeName            0
Inches              0
ScreenResolution    0
Cpu                 0
Ram                 0
Memory              0
Gpu                 0
OpSys               0
Weight              0
Price_euros         0
dtype: int64
```

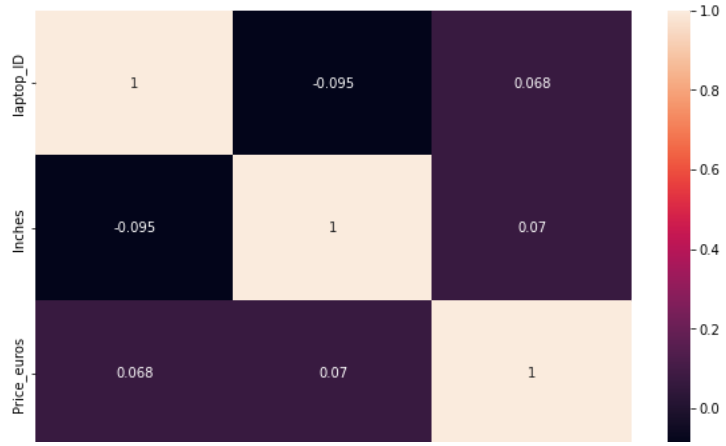
```
laptopPrice.shape
```

```
(1292, 13)
```

```
laptopPrice.dtypes
```

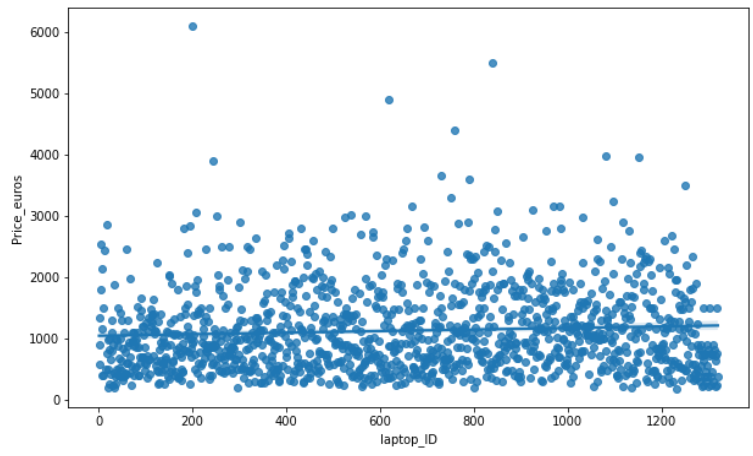
```
laptop_ID            int64
Company             object
Product             object
TypeName            object
Inches             float64
ScreenResolution    object
Cpu                 object
Ram                 object
Memory              object
Gpu                 object
OpSys               object
Weight              object
Price_euros        float64
dtype: object
```

```
plt.figure(figsize=(10,6))
corr = laptopPrice.corr()
sns.heatmap(corr,annot=True)
plt.show()
```




```
plt.figure(figsize=(10,6))
sns.regplot(x="laptop_ID", y="Price_euros", data=laptopPrice)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f080dc50100>
```

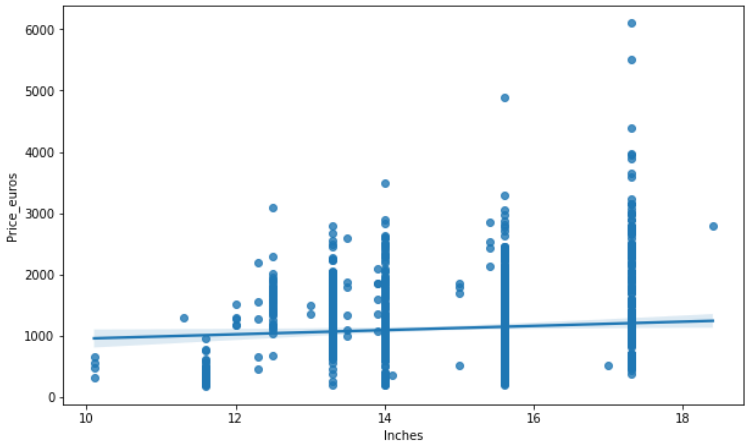



```
from scipy import stats
pearson_coef, p_value = stats.pearsonr(laptopPrice['laptop_ID'], laptopPrice['Price_euros'])
print("The Pearson Correlation Coefficient is", pearson_coef, " with a P-value of P =", p_value)
```

```
The Pearson Correlation Coefficient is 0.06841909332337447  with a P-value of P = 0.013901997508159555
```

```
plt.figure(figsize=(10,6))
sns.regplot(x="Inches", y="Price_euros", data=laptopPrice)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f080b0908e0>
```

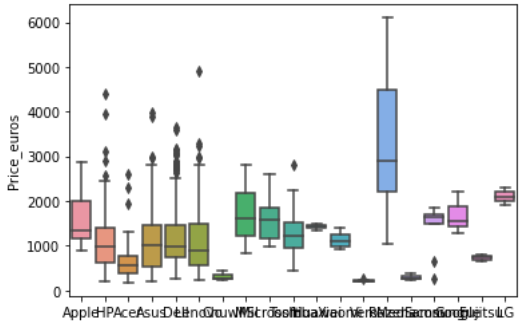



```
pearson_coef, p_value = stats.pearsonr(laptopPrice['Inches'], laptopPrice['Price_euros'])
print("The Pearson Correlation Coefficient is", pearson_coef, " with a P-value of P =", p_value)
```

```
The Pearson Correlation Coefficient is 0.07006474146467927  with a P-value of P = 0.011765731303828247
```

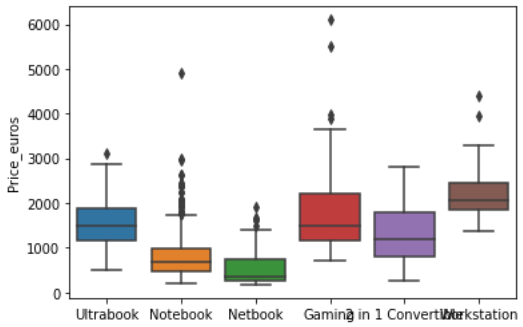
```
sns.boxplot(x="Company", y="Price_euros", data=laptopPrice)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f08093a12b0>
```



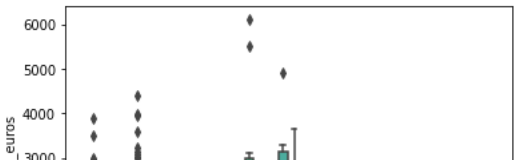

```
sns.boxplot(x="TypeName", y="Price_euros", data=laptopPrice)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0809207ca0>
```



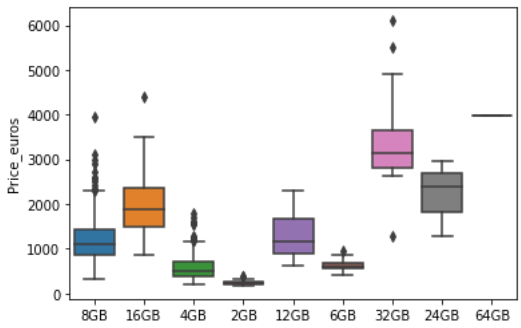

```
sns.boxplot(x="ScreenResolution", y="Price_euros", data=laptopPrice)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f08090e6640>
```



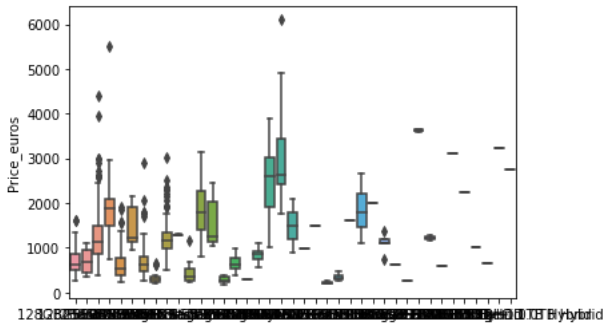

```
sns.boxplot(x="Ram", y="Price_euros", data=laptopPrice)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f080920d4f0>
```



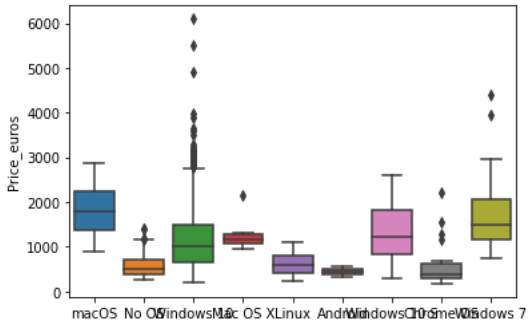

```
sns.boxplot(x="Memory", y="Price_euros", data=laptopPrice)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f080d952e20>
```




```
sns.boxplot(x="OpSys", y="Price_euros", data=laptopPrice)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0808942100>
```




```
laptopPrice.drop(['Memory', 'ScreenResolution', 'Inches'], axis = 1, inplace = True)
```

```
laptopPrice.shape
```

```
(1292, 10)
```

```
laptopPrice.describe()
```

	laptop_ID	Price_euros
count	1292.000000	1292.000000
mean	659.464396	1123.850093
std	379.169131	700.047034
min	1.000000	174.000000
25%	332.750000	599.000000
50%	657.500000	977.000000
75%	987.250000	1483.942500

```
laptopPrice['Price_euros']
```

```
0       1339.69
1        898.94
2        575.00
3       2537.45
4       1803.60
         ...
1298     638.00
1299    1499.00
1300     229.00
1301     764.00
1302     369.00
Name: Price_euros, Length: 1292, dtype: float64
```

```
laptopPrice.describe(include=['object'])
```

	Company	Product	TypeName	Cpu	Ram	Gpu	OpSys	Weigh
count	1292	1292	1292	1292	1292	1292	1292	129
unique	19	616	6	117	9	110	9	17
top	Dell	XPS 13	Notebook	Intel Core i5	8GB	Intel HD Graphics	Windows	2.2k

```
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()
laptopPrice.Company = labelencoder.fit_transform(laptopPrice.Company)
laptopPrice.TypeName = labelencoder.fit_transform(laptopPrice.TypeName)
laptopPrice.Cpu = labelencoder.fit_transform(laptopPrice.Cpu)
laptopPrice.Ram = labelencoder.fit_transform(laptopPrice.Ram)
laptopPrice.Gpu  = labelencoder.fit_transform(laptopPrice.Gpu   )
laptopPrice.OpSys = labelencoder.fit_transform(laptopPrice.OpSys)
laptopPrice.Weight = labelencoder.fit_transform(laptopPrice.Weight)
laptopPrice.Product = labelencoder.fit_transform(laptopPrice.Product)
```

```
laptopPrice.head(10)
```

	laptop_ID	Company	Product	TypeName	Cpu	Ram	Gpu	OpSys	Weight	Pr
0	1	1	300	4	64	8	58	8	38	
1	2	1	301	4	62	8	51	8	35	
2	3	7	50	3	73	8	53	4	74	
3	4	1	300	4	84	1	9	8	71	
4	5	1	300	4	66	8	59	8	38	
5	6	0	58	3	14	5	17	5	105	
6	7	1	300	4	83	1	60	3	90	
7	8	1	301	4	62	8	51	8	35	
8	9	2	606	4	105	1	98	5	41	

```
import scipy.stats as stats
laptopPrice = stats.zscore(laptopPrice)
```

```
laptopPrice
```

	laptop_ID	Company	Product	TypeName	Cpu	Ram	G
0	-1.737271	-1.336816	-0.043382	1.140718	-0.490363	0.866545	0.1526
1	-1.734632	-1.336816	-0.037543	1.140718	-0.570711	0.866545	-0.1704
2	-1.731994	0.132071	-1.503146	0.335359	-0.128794	0.866545	-0.0781
3	-1.729355	-1.336816	-0.043382	1.140718	0.313123	-1.795491	-2.1086
4	-1.726717	-1.336816	-0.043382	1.140718	-0.410014	0.866545	0.1987
...	...	...	...	...	...	...	
1298	1.732182	0.866515	1.579876	-2.080719	0.473820	-0.274327	-0.3550
1299	1.734820	0.866515	1.626588	-2.080719	0.473820	-1.795491	-0.3550
1300	1.737458	0.866515	-0.650643	0.335359	-1.695591	-1.034909	-0.6780
1301	1.740097	0.132071	-1.783420	0.335359	0.473820	0.486254	-1.5548

```
x_train=laptopPrice.iloc[:,0:7]
y_train=laptopPrice.iloc[:,8]
x_test=laptopPrice.iloc[:,0:7]
y_test=laptopPrice.iloc[:,8]
```

```
x_train.head()
```

	laptop_ID	Company	Product	TypeName	Cpu	Ram	Gpu
0	-1.737271	-1.336816	-0.043382	1.140718	-0.490363	0.866545	0.152628
1	-1.734632	-1.336816	-0.037543	1.140718	-0.570711	0.866545	-0.170416
2	-1.731994	0.132071	-1.503146	0.335359	-0.128794	0.866545	-0.078118
3	-1.729355	-1.336816	-0.043382	1.140718	0.313123	-1.795491	-2.108675
4	-1.726717	-1.336816	-0.043382	1.140718	-0.410014	0.866545	0.198777

```
y_train.head()
```

```
0   -1.222830
1   -1.292554
2   -0.386131
3   -0.455856
4   -1.222830
Name: Weight, dtype: float64
```

```
from sklearn.model_selection import train_test_split
# splitting the data
x_train, x_test, y_train, y_test = train_test_split(x_train, y_train, test_size = 0.2, random_state = 0) #20% testing data
```

```
#print the shape of train and test data after spltting
print (x_train.shape)
print (x_test.shape)
```

```
(1033, 7)
(259, 7)
```

```
from sklearn.linear_model import LinearRegression

mlr = LinearRegression()
model_mlr = mlr.fit(x_train,y_train)


y_pred1 = model_mlr.predict(x_test)



MSE1 = mean_squared_error(y_test,y_pred1)
print('MSE is ', MSE1)
```

```
     MSE is  0.8545716932620815
```

```
rf = RandomForestRegressor()
modelrf=rf.fit(x_train,y_train)


y_pred2 = modelrf.predict(x_test)



MSE2 = mean_squared_error(y_test,y_pred2)
print('MSE is ', MSE2)
```

```
     MSE is  0.22296974098400177
```

```
scores = [('MLR', MSE1),
          ('Random Forest', MSE2)]


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

	Model	MSE Score
0	MLR	0.854572
1	Random Forest	0.222970

```
MSE.sort_values(by=(['MSE Score']), ascending=False, inplace=True)

f, axe = plt.subplots(1,1, figsize=(10,7))
sns.barplot(x = MSE['Model'], y=MSE['MSE Score'], ax = axe)
axe.set_xlabel('Model', size=20)
axe.set_ylabel('Mean Squared Error', size=20)

plt.show()
```