In [1]:
```
#importing necessary libraries

import pandas as pd
import numpy as np


import matplotlib.pyplot as plt
%matplotlib inline
import plotly as px
```

In [2]:
```
df=pd.read_csv("D:/projects/udemy/laptop price predictor/laptop_data.csv")
```

In [3]:
```
df
```

Out[3]:

	Unnamed: 0	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price
0	0	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8GB	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37kg	71378.6832
1	1	Apple	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8GB	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34kg	47895.5232
2	2	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8GB	256GB SSD	Intel HD Graphics 620	No OS	1.86kg	30636.0000
3	3	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16GB	512GB SSD	AMD Radeon Pro 455	macOS	1.83kg	135195.3360
4	4	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8GB	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37kg	96095.8080
...	...	...	...	...	...	...	...	...	...	...	...	...
1298	1298	Lenovo	2 in 1 Convertible	14.0	IPS Panel Full HD / Touchscreen 1920x1080	Intel Core i7 6500U 2.5GHz	4GB	128GB SSD	Intel HD Graphics 520	Windows 10	1.8kg	33992.6400
1299	1299	Lenovo	2 in 1 Convertible	13.3	IPS Panel Quad HD+ / Touchscreen 3200x1800	Intel Core i7 6500U 2.5GHz	16GB	512GB SSD	Intel HD Graphics 520	Windows 10	1.3kg	79866.7200
1300	1300	Lenovo	Notebook	14.0	1366x768	Intel Celeron Dual Core N3050 1.6GHz	2GB	64GB Flash Storage	Intel HD Graphics	Windows 10	1.5kg	12201.1200
1301	1301	HP	Notebook	15.6	1366x768	Intel Core i7 6500U 2.5GHz	6GB	1TB HDD	AMD Radeon R5 M330	Windows 10	2.19kg	40705.9200
1302	1302	Asus	Notebook	15.6	1366x768	Intel Celeron Dual Core N3050 1.6GHz	4GB	500GB HDD	Intel HD Graphics	Windows 10	2.2kg	19660.3200

1303 rows × 12 columns

In [4]:
```
df.columns
```

Out[4]:
```
Index(['Unnamed: 0', 'Company', 'TypeName', 'Inches', 'ScreenResolution',
       'Cpu', 'Ram', 'Memory', 'Gpu', 'OpSys', 'Weight', 'Price'],
      dtype='object')
```

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
Unnamed: 0          0
Company             0
TypeName            0
Inches              0
ScreenResolution    0
Cpu                 0
Ram                 0
Memory              0
Gpu                 0
OpSys               0
Weight              0
Price               0
dtype: int64
```

In [6]:
```
df.duplicated().sum()
```

Out[6]:
```
0
```

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1303 entries, 0 to 1302
Data columns (total 12 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Unnamed: 0        1303 non-null   int64  
 1   Company           1303 non-null   object 
 2   TypeName          1303 non-null   object 
 3   Inches            1303 non-null   float64
 4   ScreenResolution  1303 non-null   object 
 5   Cpu               1303 non-null   object 
 6   Ram               1303 non-null   object 
 7   Memory            1303 non-null   object 
 8   Gpu               1303 non-null   object 
 9   OpSys             1303 non-null   object 
 10  Weight            1303 non-null   object 
 11  Price             1303 non-null   float64
dtypes: float64(2), int64(1), object(9)
memory usage: 122.3+ KB
```

In [10]:
```
for i in df.columns:
    print(f'Details of particular {i} is : {df[i].unique()}')
    print('-'*75)
```

```
Details of particular Unnamed: 0 is : [   0    1    2 ... 1300 1301 1302]
---------------------------------------------------------------------------
Details of particular Company is : ['Apple' 'HP' 'Acer' 'Asus' 'Dell' 'Lenovo' 'Chuwi' 'MSI' 'Microsoft'
 'Toshiba' 'Huawei' 'Xiaomi' 'Vero' 'Razer' 'Mediacom' 'Samsung' 'Google'
 'Fujitsu' 'LG']
---------------------------------------------------------------------------
Details of particular TypeName is : ['Ultrabook' 'Notebook' 'Netbook' 'Gaming' '2 in 1 Convertible'
 'Workstation']
---------------------------------------------------------------------------
Details of particular Inches is : [13.3 15.6 15.4 14.  12.  11.6 17.3 10.1 13.5 12.5 13.  18.4 13.9 12.3
 17.  15.  14.1 11.3]
---------------------------------------------------------------------------
Details of particular ScreenResolution is : ['IPS Panel Retina Display 2560x1600' '1440x900' 'Full HD 1920x1080'
 'IPS Panel Retina Display 2880x1800' '1366x768'
 'IPS Panel Full HD 1920x1080' 'IPS Panel Retina Display 2304x1440'
 'IPS Panel Full HD / Touchscreen 1920x1080'
 'Full HD / Touchscreen 1920x1080' 'Touchscreen / Quad HD+ 3200x1800'
 'IPS Panel Touchscreen 1920x1200' 'Touchscreen 2256x1504'
 'Quad HD+ / Touchscreen 3200x1800' 'IPS Panel 1366x768'
```

In [11]:
```
df=df[['Company', 'TypeName', 'Inches', 'ScreenResolution',
       'Cpu', 'Ram', 'Memory', 'Gpu', 'OpSys', 'Weight', 'Price']]
```

In [12]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1303 entries, 0 to 1302
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Company           1303 non-null   object
 1   TypeName          1303 non-null   object
 2   Inches            1303 non-null   float64
 3   ScreenResolution  1303 non-null   object
 4   Cpu               1303 non-null   object
 5   Ram               1303 non-null   object
 6   Memory            1303 non-null   object
 7   Gpu               1303 non-null   object
 8   OpSys             1303 non-null   object
 9   Weight            1303 non-null   object
 10  Price             1303 non-null   float64
dtypes: float64(2), object(9)
memory usage: 112.1+ KB
```

In [13]:
```
catvar=df.select_dtypes(include=['object']).columns
numvar=df.select_dtypes(include=['int32','float32','float64']).columns
```

In [14]:
```
catvar,numvar
```

Out[14]:
```
(Index(['Company', 'TypeName', 'ScreenResolution', 'Cpu', 'Ram', 'Memory',
        'Gpu', 'OpSys', 'Weight'],
       dtype='object'),
 Index(['Inches', 'Price'], dtype='object'))
```

In [16]:
```
df['Weight']=df['Weight'].str.replace('kg',"")
df['Ram']=df['Ram'].str.replace('GB',"")
```

```
C:\Users\basha\AppData\Local\Temp\ipykernel_30028\1311007701.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Weight']=df['Weight'].str.replace('kg',"")
C:\Users\basha\AppData\Local\Temp\ipykernel_30028\1311007701.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Ram']=df['Ram'].str.replace('GB',"")
```

In [17]:
```
df.head()
```

Out[17]:

	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price
0	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832
1	Apple	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895.5232
2	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	30636.0000
3	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83	135195.3360
4	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37	96095.8080

In [18]:
```
df['Ram'] = df['Ram'].astype('int32')
```

```
C:\Users\basha\AppData\Local\Temp\ipykernel_30028\1977328478.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Ram'] = df['Ram'].astype('int32')
```

In [19]:
```
df['Weight'] = df['Weight'].astype('float32')
```

```
C:\Users\basha\AppData\Local\Temp\ipykernel_30028\528318470.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Weight'] = df['Weight'].astype('float32')
```

In [20]:
```
df.info()
```

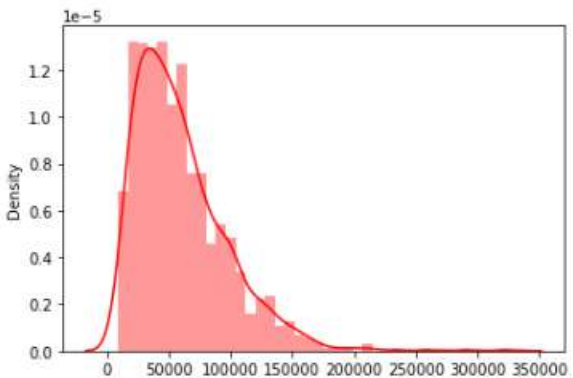
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1303 entries, 0 to 1302
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Company           1303 non-null   object
 1   TypeName          1303 non-null   object
 2   Inches            1303 non-null   float64
 3   ScreenResolution  1303 non-null   object
 4   Cpu               1303 non-null   object
 5   Ram               1303 non-null   int32
 6   Memory            1303 non-null   object
 7   Gpu               1303 non-null   object
 8   OpSys             1303 non-null   object
 9   Weight            1303 non-null   float32
 10  Price             1303 non-null   float64
dtypes: float32(1), float64(2), int32(1), object(7)
memory usage: 101.9+ KB
```

EXPLORATORY DATA ANALYSIS

In [21]:
```
import seaborn as sn
sn.distplot(x = df['Price'],color='red',bins = 40)
```

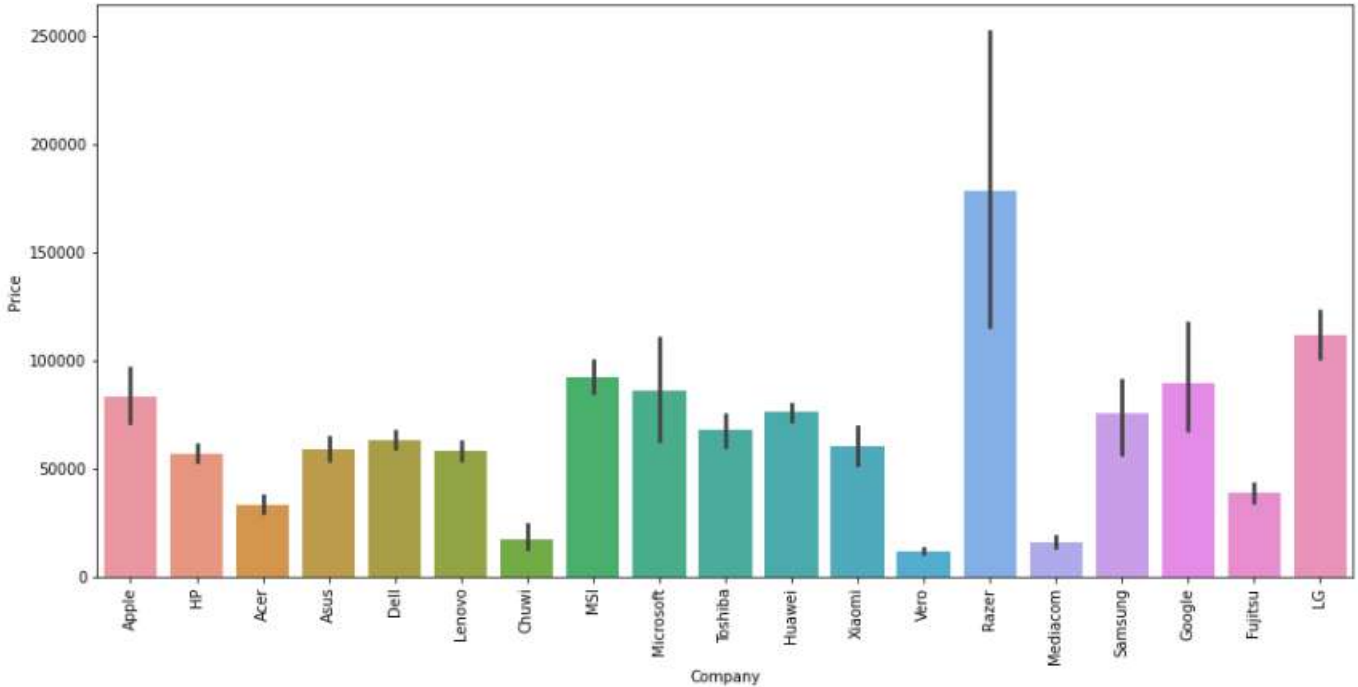
```
C:\Users\basha\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function an
d will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar fle
xibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[21]: <AxesSubplot:ylabel='Density'>



In [22]:
```
def drawplot(col):
    plt.figure(figsize=(15,7))
    sn.countplot(df[col],palette='plasma')
    plt.xticks(rotation='vertical')
```

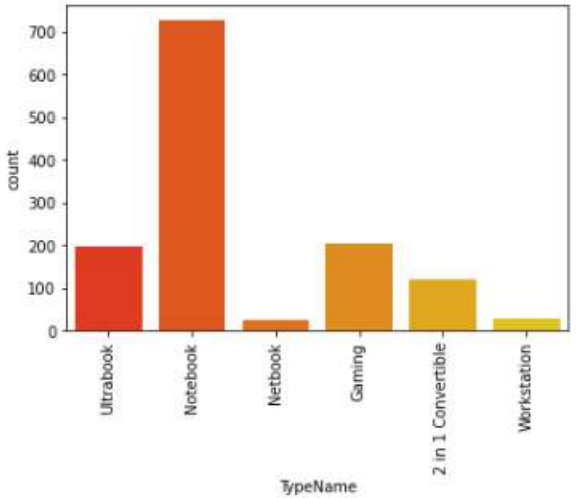
In [24]:
```
#average price of laptop
plt.figure(figsize=(15,7))
sn.barplot(x=df['Company'],y=df['Price'])
plt.xticks(rotation='vertical')
plt.show()
```



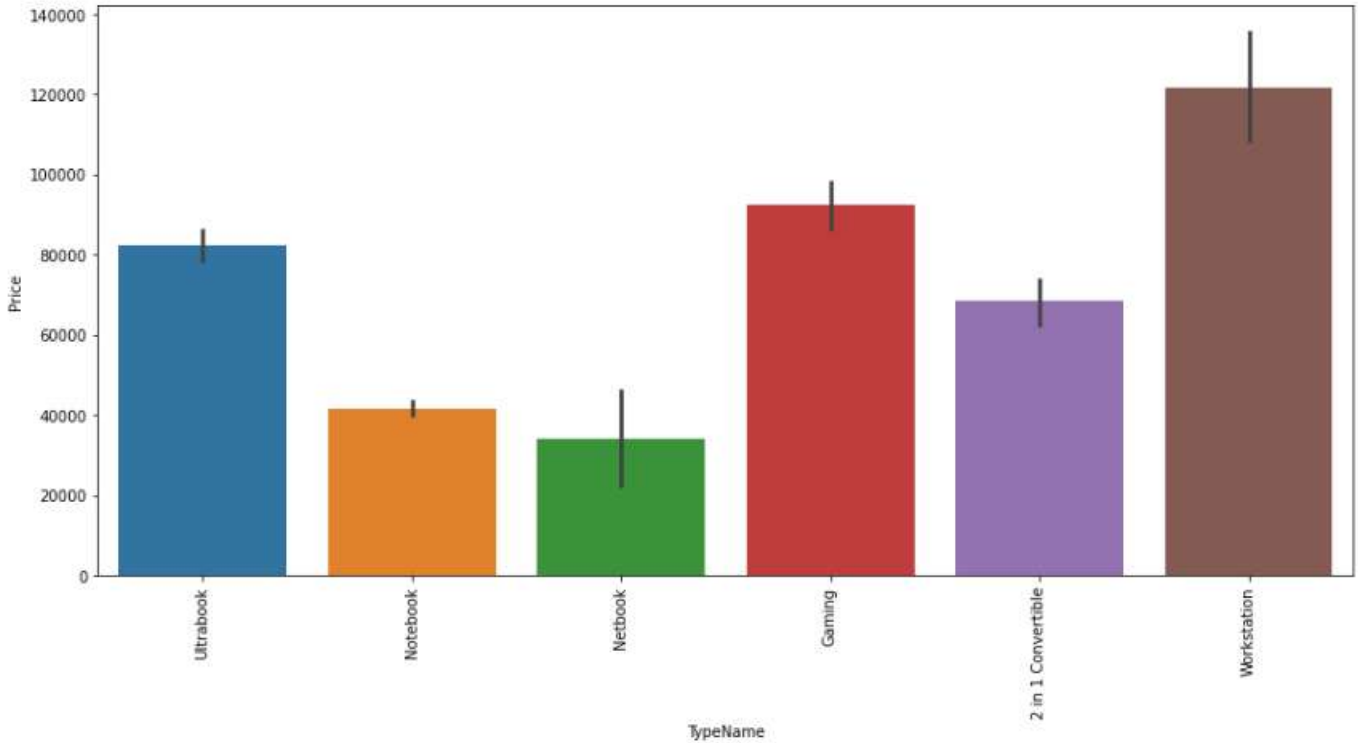
In [25]:
```
#average types of laptop

sn.countplot(df['TypeName'],palette='autumn')
plt.xticks(rotation='vertical')
plt.show()
```

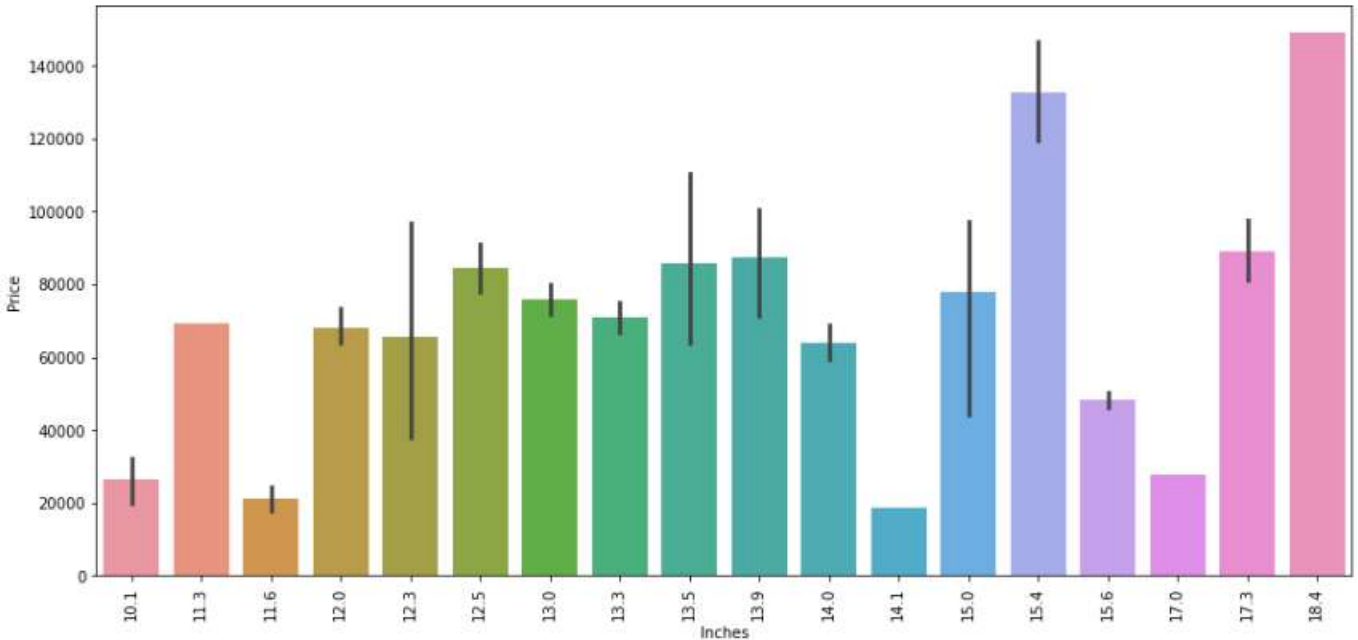
```
C:\Users\basha\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword a
rg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
```



In [26]:
```
#average price of laptop
plt.figure(figsize=(15,7))
sn.barplot(x=df['TypeName'],y=df['Price'])
plt.xticks(rotation='vertical')
plt.show()
```



In [27]:
```
#average price of laptop
plt.figure(figsize=(15,7))
sn.barplot(x=df['Inches'],y=df['Price'])
plt.xticks(rotation='vertical')
plt.show()
```



In [28]:
```
df['ScreenResolution'].value_counts()
```

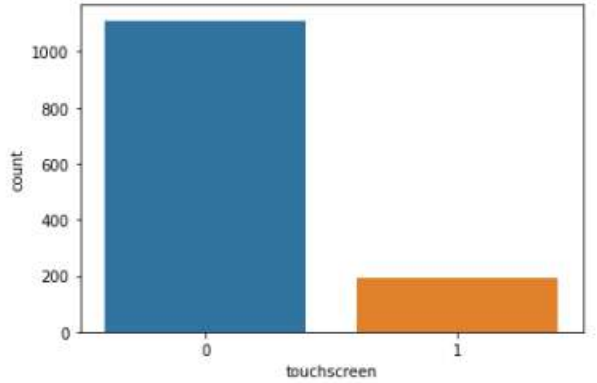
Out[28]:
```
Full HD 1920x1080                                507
1366x768                                         281
IPS Panel Full HD 1920x1080                      230
IPS Panel Full HD / Touchscreen 1920x1080         53
Full HD / Touchscreen 1920x1080                   47
1600x900                                          23
Touchscreen 1366x768                              16
Quad HD+ / Touchscreen 3200x1800                  15
IPS Panel 4K Ultra HD 3840x2160                   12
IPS Panel 4K Ultra HD / Touchscreen 3840x2160     11
4K Ultra HD / Touchscreen 3840x2160               10
4K Ultra HD 3840x2160                              7
Touchscreen 2560x1440                              7
IPS Panel 1366x768                                 7
IPS Panel Quad HD+ / Touchscreen 3200x1800         6
IPS Panel Retina Display 2560x1600                 6
IPS Panel Retina Display 2304x1440                 6
Touchscreen 2256x1504                              6
IPS Panel Touchscreen 2560x1440                    5
IPS Panel Retina Display 2880x1800                 4
IPS Panel Touchscreen 1920x1200                    4
1440x900                                           4
IPS Panel 2560x1440                                4
IPS Panel Quad HD+ 2560x1440                       3
Quad HD+ 3200x1800                                 3
1920x1080                                          3
Touchscreen 2400x1600                              3
2560x1440                                          3
IPS Panel Touchscreen 1366x768                     3
IPS Panel Touchscreen / 4K Ultra HD 3840x2160      2
IPS Panel Full HD 2160x1440                        2
IPS Panel Quad HD+ 3200x1800                       2
IPS Panel Retina Display 2736x1824                 1
IPS Panel Full HD 1920x1200                        1
IPS Panel Full HD 2560x1440                        1
IPS Panel Full HD 1366x768                         1
Touchscreen / Full HD 1920x1080                    1
Touchscreen / Quad HD+ 3200x1800                   1
Touchscreen / 4K Ultra HD 3840x2160                1
IPS Panel Touchscreen 2400x1600                    1
Name: ScreenResolution, dtype: int64
```

In [29]:
```
df['touchscreen']=df['ScreenResolution'].apply(lambda element:1 if 'Touchscreen' in element else 0)
```

In [30]:
```
sn.countplot(df['touchscreen'])
```

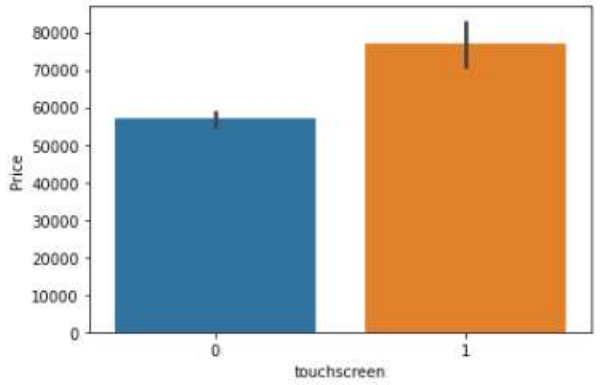
```
C:\Users\basha\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword a
rg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
```

Out[30]: <AxesSubplot:xlabel='touchscreen', ylabel='count'>



In [31]:
```
sn.barplot(x=df['touchscreen'],y=df['Price'])
```

Out[31]: <AxesSubplot:xlabel='touchscreen', ylabel='Price'>

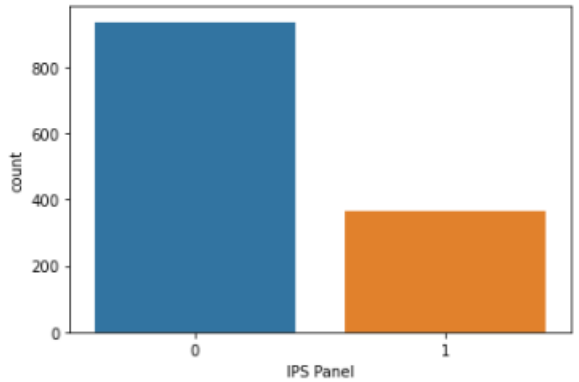


In [32]:
```
df['IPS Panel']=df['ScreenResolution'].apply(lambda element:1 if 'IPS Panel' in element else 0)
```

In [33]:
```
sn.countplot(df['IPS Panel'])
```

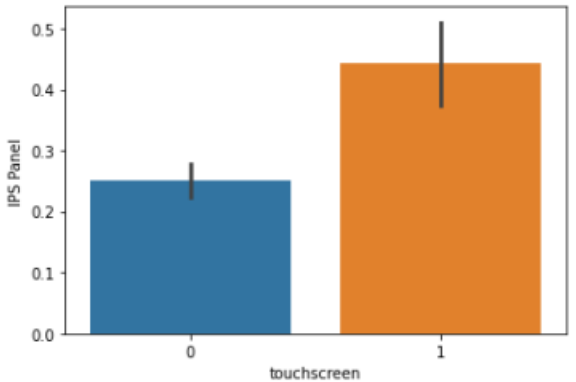
```
C:\Users\basha\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword a
rg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
```

Out[33]: <AxesSubplot:xlabel='IPS Panel', ylabel='count'>



In [34]:
```
sn.barplot(x=df['touchscreen'],y=df['IPS Panel'])
```

Out[34]: <AxesSubplot:xlabel='touchscreen', ylabel='IPS Panel'>



In [35]:
```
splitdf=df['ScreenResolution'].str.split('x',expand=True)
splitdf
```

Out[35]:

	0	1
0	IPS Panel Retina Display 2560	1600
1	1440	900
2	Full HD 1920	1080
3	IPS Panel Retina Display 2880	1800
4	IPS Panel Retina Display 2560	1600
...	...	...
1298	IPS Panel Full HD / Touchscreen 1920	1080
1299	IPS Panel Quad HD+ / Touchscreen 3200	1800
1300	1366	768
1301	1366	768
1302	1366	768

1303 rows × 2 columns

In [36]:
```
df['Xres']=splitdf[0]
df['Yres']=splitdf[1]
```

In [37]:
```
df.head()
```

Out[37]:

	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price	touchscreen	IPS Panel	Xres	Yres
0	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832	0	1	IPS Panel Retina Display 2560	1600
1	Apple	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895.5232	0	0	1440	900
2	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	30636.0000	0	0	Full HD 1920	1080

In [38]:
```
#findall(r"(\d+\.?\d+)"): The findall function is being used with a regular expression pattern as its argument.
#The pattern r"(\d+\.?\d+)" is used to match numerical values in the input string. Here's a breakdown of the pattern:

#\d+: Matches one or more digits.
#\.?: Matches an optional decimal point (the backslash is used to escape the special meaning of the dot).
#\d+: Matches one or more digits after the decimal point.
#The parentheses ( ) are used to create a capturing group, which allows accessing the matched value later.

df['Xres']=df['Xres'].str.replace(',','').str.findall(r"(\d+\.?\d+)").apply(lambda x:x[0])
```

In [39]:
```
df.head()
```

Out[39]:

	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price	touchscreen	IPS Panel	Xres	Yres
0	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832	0	1	2560	1600
1	Apple	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895.5232	0	0	1440	900
2	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	30636.0000	0	0	1920	1080
3	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83	135195.3360	0	1	2880	1800
4	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37	96095.8080	0	1	2560	1600

In [40]:
```
df['Xres']=df['Xres'].astype('int')
df['Yres']=df['Yres'].astype('int')
```

In [42]:
```
plt.figure(figsize=(15,7))
sn.heatmap(df.corr(),annot=True,cmap='plasma')
```

Out[42]: <AxesSubplot:>

In [43]:
```
#we found multi collinearity Xres & Yres
#pixel per inch
#PPI is calculated by dividing the number of pixels in the horizontal and vertical directions of a screen by the screen's physica
#The formula to calculate PPI is:

#
PPI = √(Pixel width^2 + Pixel height^2) / Screen size in inches

df['PPI']=(((df['Xres']**2+df['Yres']**2))*0.5/df['Inches']).astype('float')
```

In [45]:
```
df.corr()['Price']
```

Out[45]:
```
Inches         0.068197
Ram            0.743007
Weight         0.210370
Price          1.000000
touchscreen    0.191226
IPS Panel      0.252208
Xres           0.556529
Yres           0.552809
PPI            0.480017
Name: Price, dtype: float64
```

In [46]:
```
#Drop unnecessary column as we get PPI from Xres,Yres,Inches
df.drop(columns=['Xres','Yres','Inches'],inplace=True)
df.head()
```

Out[46]:

	Company	TypeName	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price	touchscreen	IPS Panel	PPI
0	Apple	Ultrabook	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832	0	1	342616.541353
1	Apple	Ultrabook	1440x900	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895.5232	0	0	108406.015038
2	HP	Notebook	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	30636.0000	0	0	155538.461538
3	Apple	Ultrabook	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83	135195.3360	0	1	374493.506494
4	Apple	Ultrabook	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37	96095.8080	0	1	342616.541353

In [47]:
```
df['Cpu'].value_counts()
```

Out[47]:
```
Intel Core i5 7200U 2.5GHz       190
Intel Core i7 7700HQ 2.8GHz      146
Intel Core i7 7500U 2.7GHz       134
Intel Core i7 8550U 1.8GHz        73
Intel Core i5 8250U 1.6GHz        72
                                 ...
Intel Core M M3-6Y30 0.9GHz        1
AMD A9-Series 9420 2.9GHz          1
Intel Core i3 6006U 2.2GHz         1
AMD A6-Series 7310 2GHz            1
Intel Xeon E3-1535M v6 3.1GHz      1
Name: Cpu, Length: 118, dtype: int64
```

In [48]:
```
text='Intel Core i7 8550U 1.8GHz'
" ".join(text.split()[:3])
```

Out[48]:
```
'Intel Core i7'
```

In [49]:
```
df['Cpu_name']=df['Cpu'].apply(lambda text:" ".join(text.split()[:3]))
df.head()
```

Out[49]:

	Company	TypeName	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price	touchscreen	IPS Panel	PPI	Cpu_name
0	Apple	Ultrabook	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832	0	1	342616.541353	Intel Core i5
1	Apple	Ultrabook	1440x900	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895.5232	0	0	108406.015038	Intel Core i5
2	HP	Notebook	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	30636.0000	0	0	155538.461538	Intel Core i5
3	Apple	Ultrabook	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83	135195.3360	0	1	374493.506494	Intel Core i7
4	Apple	Ultrabook	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37	96095.8080	0	1	342616.541353	Intel Core i5

In [50]:
```
df['Cpu_name']
```

Out[50]:
```
0            Intel Core i5
1            Intel Core i5
2            Intel Core i5
3            Intel Core i7
4            Intel Core i5
               ...
1298         Intel Core i7
1299         Intel Core i7
1300    Intel Celeron Dual
1301         Intel Core i7
1302    Intel Celeron Dual
Name: Cpu_name, Length: 1303, dtype: object
```

In [51]:
```
def processortype(text):
    if text=='Intel Core i7' or text=='Intel Core i5' or text=='Intel Core i3':
        return text
    else:
        if text.split()[0]=='Intel':
            return 'Intel processor'
        else:
            return'AMD processor'

df['Cpu_name']=df['Cpu_name'].apply(lambda text:processortype(text))
df.head()
```

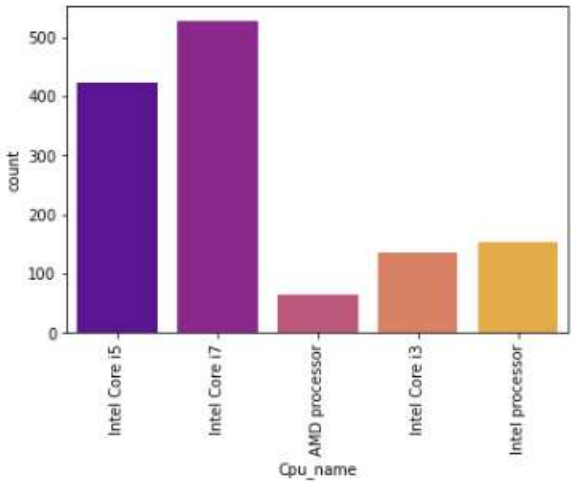
Out[51]:

	Company	TypeName	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price	touchscreen	IPS Panel	PPI	Cpu_name
0	Apple	Ultrabook	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832	0	1	342616.541353	Intel Core i5
1	Apple	Ultrabook	1440x900	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895.5232	0	0	108406.015038	Intel Core i5
2	HP	Notebook	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	30636.0000	0	0	155538.461538	Intel Core i5
3	Apple	Ultrabook	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83	135195.3360	0	1	374493.506494	Intel Core i7
4	Apple	Ultrabook	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37	96095.8080	0	1	342616.541353	Intel Core i5

In [52]:
```
sn.countplot(df['Cpu_name'],palette='plasma')
plt.xticks(rotation='vertical')
```

```
C:\Users\basha\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword a
rg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
```

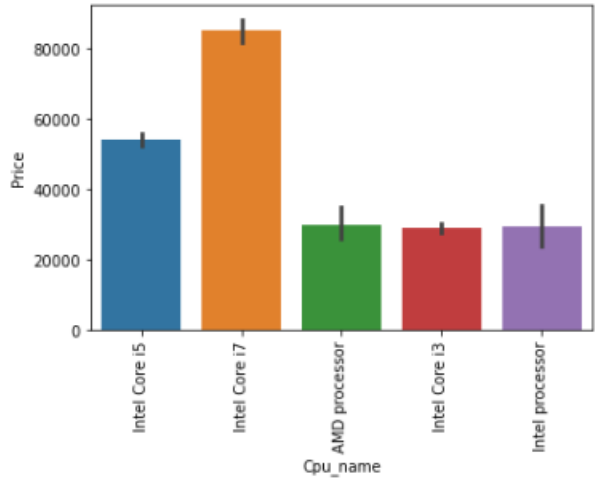
Out[52]:
```
(array([0, 1, 2, 3, 4]),
 [Text(0, 0, 'Intel Core i5'),
  Text(1, 0, 'Intel Core i7'),
  Text(2, 0, 'AMD processor'),
  Text(3, 0, 'Intel Core i3'),
  Text(4, 0, 'Intel processor')])
```



In [53]:
```
#price vs process variation
sn.barplot(df['Cpu_name'],df['Price'])
plt.xticks(rotation='vertical')
```

```
C:\Users\basha\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword ar
gs: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[53]:
```
(array([0, 1, 2, 3, 4]),
 [Text(0, 0, 'Intel Core i5'),
  Text(1, 0, 'Intel Core i7'),
  Text(2, 0, 'AMD processor'),
  Text(3, 0, 'Intel Core i3'),
  Text(4, 0, 'Intel processor')])
```



In [54]:
```
df.drop(columns=['Cpu'],inplace=True)
df.head()
```

Out[54]:

	Company	TypeName	ScreenResolution	Ram	Memory	Gpu	OpSys	Weight	Price	touchscreen	IPS Panel	PPI	Cpu_name
0	Apple	Ultrabook	IPS Panel Retina Display 2560x1600	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832	0	1	342616.541353	Intel Core i5
1	Apple	Ultrabook	1440x900	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895.5232	0	0	108406.015038	Intel Core i5
2	HP	Notebook	Full HD 1920x1080	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	30636.0000	0	0	155538.461538	Intel Core i5

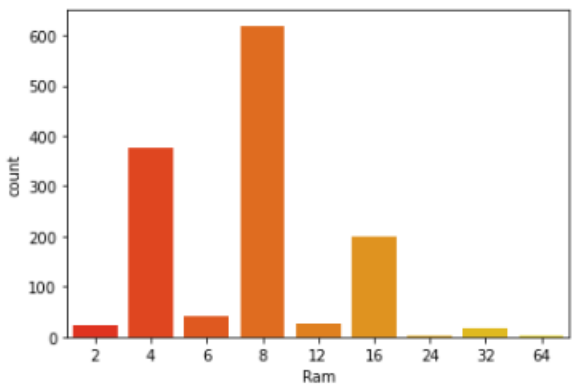
In [55]:
```
#Analysis on RAM column
df['Ram'].unique()
```

Out[55]:
```
array([ 8, 16,  4,  2, 12,  6, 32, 24, 64])
```

In [56]:
```
sn.countplot(df['Ram'],palette='autumn')
# sn.countplot(df['Cpu_name'],palette='plasma')
# plt.xticks(rotation='vertical')
```

```
C:\Users\basha\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword a
rg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
```

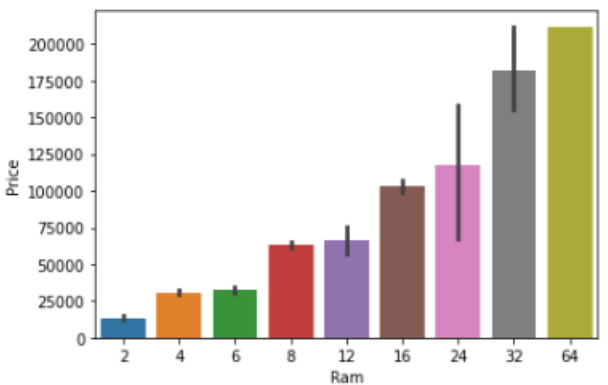
Out[56]:
```
<AxesSubplot:xlabel='Ram', ylabel='count'>
```



In [57]:
```
sn.barplot(df['Ram'],df['Price'])
```

```
C:\Users\basha\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword ar
gs: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[57]:
```
<AxesSubplot:xlabel='Ram', ylabel='Price'>
```



In [58]:
```
#Memory column
df['Memory'].unique()
```

Out[58]:
```
array(['128GB SSD', '128GB Flash Storage', '256GB SSD', '512GB SSD',
       '500GB HDD', '256GB Flash Storage', '1TB HDD',
       '32GB Flash Storage', '128GB SSD +  1TB HDD',
       '256GB SSD +  256GB SSD', '64GB Flash Storage',
       '256GB SSD +  1TB HDD', '256GB SSD +  2TB HDD', '32GB SSD',
       '2TB HDD', '64GB SSD', '1.0TB Hybrid', '512GB SSD +  1TB HDD',
       '1TB SSD', '256GB SSD +  500GB HDD', '128GB SSD +  2TB HDD',
       '512GB SSD +  512GB SSD', '16GB SSD', '16GB Flash Storage',
       '512GB SSD +  256GB SSD', '512GB SSD +  2TB HDD',
       '64GB Flash Storage +  1TB HDD', '180GB SSD', '1TB HDD +  1TB HDD',
       '32GB HDD', '1TB SSD +  1TB HDD', '512GB Flash Storage',
       '128GB HDD', '240GB SSD', '8GB SSD', '508GB Hybrid', '1.0TB HDD',
       '512GB SSD +  1.0TB Hybrid', '256GB SSD +  1.0TB Hybrid'],
      dtype=object)
```

In [59]:
```
df['Memory'].value_counts()
```

Out[59]:
```
256GB SSD                        412
1TB HDD                          223
500GB HDD                        132
512GB SSD                        118
128GB SSD +  1TB HDD              94
128GB SSD                         76
256GB SSD +  1TB HDD              73
32GB Flash Storage                38
2TB HDD                           16
64GB Flash Storage                15
512GB SSD +  1TB HDD              14
1TB SSD                           14
256GB SSD +  2TB HDD              10
1.0TB Hybrid                       9
256GB Flash Storage                8
16GB Flash Storage                 7
32GB SSD                           6
180GB SSD                          5
128GB Flash Storage                4
512GB SSD +  2TB HDD               3
16GB SSD                           3
512GB Flash Storage                2
1TB SSD +  1TB HDD                 2
256GB SSD +  500GB HDD             2
128GB SSD +  2TB HDD               2
256GB SSD +  256GB SSD             2
512GB SSD +  256GB SSD             1
512GB SSD +  512GB SSD             1
64GB Flash Storage +  1TB HDD      1
1TB HDD +  1TB HDD                 1
32GB HDD                           1
64GB SSD                           1
128GB HDD                          1
240GB SSD                          1
8GB SSD                            1
508GB Hybrid                       1
1.0TB HDD                          1
512GB SSD +  1.0TB Hybrid          1
256GB SSD +  1.0TB Hybrid          1
Name: Memory, dtype: int64
```

In [60]:
```
df['Memory']=df['Memory'].astype(str).replace('\.0',"",regex=True) #whatever comes after decimal is truncated
```

In [61]:
```
df['Memory']=df['Memory'].str.replace('GB',"")
df['Memory']=df['Memory'].str.replace('TB',"000")
```

In [62]:
```
df['Memory'].value_counts()
```

Out[62]:
```
256  SSD                        412
1000 HDD                        224
500  HDD                        132
512  SSD                        118
128  SSD +  1000 HDD             94
128  SSD                         76
256  SSD +  1000 HDD             73
32  Flash Storage                38
2000 HDD                        16
64  Flash Storage                15
512  SSD +  1000 HDD             14
1000 SSD                        14
256  SSD +  2000 HDD             10
1000 Hybrid                      9
256  Flash Storage               8
16  Flash Storage                7
32  SSD                         6
180  SSD                        5
128  Flash Storage              4
512  SSD +  2000 HDD            3
16  SSD                         3
512  Flash Storage              2
1000 SSD +  1000 HDD            2
256  SSD +  500  HDD            2
128  SSD +  2000 HDD            2
256  SSD +  256  SSD            2
512  SSD +  256  SSD            1
512  SSD +  512  SSD            1
64  Flash Storage +  1000 HDD    1
1000 HDD +  1000 HDD            1
32  HDD                         1
64  SSD                         1
128  HDD                        1
240  SSD                        1
8  SSD                          1
508  Hybrid                     1
512  SSD +  1000 Hybrid         1
256  SSD +  1000 Hybrid         1
Name: Memory, dtype: int64
```

In [63]:
```
#now split the word accross the "+" character

new_df=df['Memory'].str.split('+',expand=True,n=1)
```

In [64]:
```
new_df
```

Out[64]:

	0	1
0	128 SSD	None
1	128 Flash Storage	None
2	256 SSD	None
3	512 SSD	None
4	256 SSD	None
...	...	...
1298	128 SSD	None
1299	512 SSD	None
1300	64 Flash Storage	None
1301	1000 HDD	None
1302	500 HDD	None

1303 rows × 2 columns

In [65]:
```
df['first']=new_df[0].str.strip()
df.head()
```

Out[65]:

	Company	TypeName	ScreenResolution	Ram	Memory	Gpu	OpSys	Weight	Price	touchscreen	IPS Panel	PPI	Cpu_name	first
0	Apple	Ultrabook	IPS Panel Retina Display 2560x1600	8	128 SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832	0	1	342616.541353	Intel Core i5	128 SSD
1	Apple	Ultrabook	1440x900	8	128 Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895.5232	0	0	108406.015038	Intel Core i5	128 Flash Storage
2	HP	Notebook	Full HD 1920x1080	8	256 SSD	Intel HD Graphics 620	No OS	1.86	30636.0000	0	0	155538.461538	Intel Core i5	256 SSD
3	Apple	Ultrabook	IPS Panel Retina Display 2880x1800	16	512 SSD	AMD Radeon Pro 455	macOS	1.83	135195.3360	0	1	374493.506494	Intel Core i7	512 SSD

In [66]:
```
df['first'].unique()
```

Out[66]:
```
array(['128  SSD', '128  Flash Storage', '256  SSD', '512  SSD',
       '500  HDD', '256  Flash Storage', '1000 HDD', '32  Flash Storage',
       '64  Flash Storage', '32  SSD', '2000 HDD', '64  SSD',
       '1000 Hybrid', '1000 SSD', '16  SSD', '16  Flash Storage',
       '180  SSD', '32  HDD', '512  Flash Storage', '128  HDD',
       '240  SSD', '8  SSD', '508  Hybrid'], dtype=object)
```

In [67]:
```
def applychange(value):
    df['layer1'+value]=df['first'].apply(lambda x:1 if value in x else 0)
listtoapply=['SSD','Flash Storage','HDD','Hybrid']

for value in listtoapply:
    applychange(value)
df.head()
```

Out[67]:

Ram	Memory	Gpu	OpSys	Weight	Price	touchscreen	IPS Panel	PPI	Cpu_name	first	layer1SSD	layer1Flash Storage	layer1HDD	layer1Hybrid
8	128 SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832	0	1	342616.541353	Intel Core i5	128 SSD	1	0	0	0
8	128 Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895.5232	0	0	108406.015038	Intel Core i5	128 Flash Storage	0	1	0	0
8	256 SSD	Intel HD Graphics 620	No OS	1.86	30636.0000	0	0	155538.461538	Intel Core i5	256 SSD	1	0	0	0
16	512 SSD	AMD Radeon Pro 455	macOS	1.83	135195.3360	0	1	374493.506494	Intel Core i7	512 SSD	1	0	0	0
8	256 SSD	Intel Iris Plus Graphics 650	macOS	1.37	96095.8080	0	1	342616.541353	Intel Core i5	256 SSD	1	0	0	0

In [68]:
```
#In first column we will remove all text and keep numbers

df['first']=df['first'].str.replace('\D','')
df['first'].value_counts()
```

```
C:\Users\basha\AppData\Local\Temp\ipykernel_30028\2715463296.py:3: FutureWarning: The default value of regex will change from T
rue to False in a future version.
  df['first']=df['first'].str.replace('\D','')
```

Out[68]:
```
256     508
1000    250
128     177
```

In [69]:
```
new_df
```

Out[69]:

	0	1
0	128 SSD	None
1	128 Flash Storage	None
2	256 SSD	None
3	512 SSD	None
4	256 SSD	None
...	...	...
1298	128 SSD	None
1299	512 SSD	None
1300	64 Flash Storage	None
1301	1000 HDD	None
1302	500 HDD	None

1303 rows × 2 columns

In [70]:
```
df['second']=new_df[1]
df.head()
```

Out[70]:

	Company	TypeName	ScreenResolution	Ram	Memory	Gpu	OpSys	Weight	Price	touchscreen	IPS Panel	PPI	Cpu_name	first	layer1
0	Apple	Ultrabook	IPS Panel Retina Display 2560x1600	8	128 SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832	0	1	342616.541353	Intel Core i5	128	
1	Apple	Ultrabook	1440x900	8	128 Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895.5232	0	0	108406.015038	Intel Core i5	128	
2	HP	Notebook	Full HD 1920x1080	8	256 SSD	Intel HD Graphics 620	No OS	1.86	30636.0000	0	0	155538.461538	Intel Core i5	256	
3	Apple	Ultrabook	IPS Panel Retina Display 2880x1800	16	512 SSD	AMD Radeon Pro 455	macOS	1.83	135195.3360	0	1	374493.506494	Intel Core i7	512	
4	Apple	Ultrabook	IPS Panel Retina Display 2560x1600	8	256 SSD	Intel Iris Plus Graphics 650	macOS	1.37	96095.8080	0	1	342616.541353	Intel Core i5	256	

In [72]:
```
def applychange1(value):
    df['layer2'+value]=df['second'].apply(lambda x:1 if value in x else 0)
listtoapply=['SSD','Flash Storage','HDD','Hybrid']
df['second']=df['second'].fillna('0')

for value in listtoapply:
    applychange1(value)
df.head()
```

Out[72]:

OpSys	Weight	Price	touchscreen	...	first	layer1SSD	layer1Flash Storage	layer1HDD	layer1Hybrid	second	layer2SSD	layer2Flash Storage	layer2HDD	layer2Hybrid
macOS	1.37	71378.6832	0	...	128	1	0	0	0	0	0	0	0	0
macOS	1.34	47895.5232	0	...	128	0	1	0	0	0	0	0	0	0
No OS	1.86	30636.0000	0	...	256	1	0	0	0	0	0	0	0	0
macOS	1.83	135195.3360	0	...	512	1	0	0	0	0	0	0	0	0
macOS	1.37	96095.8080	0	...	256	1	0	0	0	0	0	0	0	0

In [73]:
```
df['second'].unique()
```

Out[73]:
```
array(['0', '  1000 HDD', '  256  SSD', '  2000 HDD', '  500  HDD',
       '  512  SSD', '  1000 Hybrid'], dtype=object)
```

In [74]:
```
df['second']
```

Out[74]:
```
0       0
1       0
2       0
3       0
4       0
       ..
1298    0
1299    0
1300    0
1301    0
```

In [75]:
```
df['second']=df['second'].str.replace('\D',"")
df['second'].value_counts()
```

```
C:\Users\basha\AppData\Local\Temp\ipykernel_30028\1244662679.py:1: FutureWarning: The default value of regex will change from T
rue to False in a future version.
  df['second']=df['second'].str.replace('\D',"")
```

Out[75]:
```
0       1095
1000     187
2000      15
256        3
500        2
512        1
Name: second, dtype: int64
```

In [76]:
```
df['first'] = df['first'].astype('int')
df['second'] = df['second'].astype('int')
df.head()
```

Out[76]:

	Company	TypeName	ScreenResolution	Ram	Memory	Gpu	OpSys	Weight	Price	touchscreen	...	first	layer1SSD	layer1Flash Storage	layer1HDD	la
0	Apple	Ultrabook	IPS Panel Retina Display 2560x1600	8	128 SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832	0	...	128	1	0	0	
1	Apple	Ultrabook	1440x900	8	128 Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895.5232	0	...	128	0	1	0	
2	HP	Notebook	Full HD 1920x1080	8	256 SSD	Intel HD Graphics 620	No OS	1.86	30636.0000	0	...	256	1	0	0	
3	Apple	Ultrabook	IPS Panel Retina Display 2880x1800	16	512 SSD	AMD Radeon Pro 455	macOS	1.83	135195.3360	0	...	512	1	0	0	
4	Apple	Ultrabook	IPS Panel Retina Display 2560x1600	8	256 SSD	Intel Iris Plus Graphics 650	macOS	1.37	96095.8080	0	...	256	1	0	0	

5 rows × 23 columns

In [80]:
```
#drop unnecesary columns
df.drop(columns=['first', 'layer1SSD', 'layer1Flash Storage', 'layer1HDD',
       'layer1Hybrid', 'second', 'layer2SSD', 'layer2Flash Storage',
       'layer2HDD', 'layer2Hybrid'],inplace=True)
```

In [81]:
```
df.sample(5)
```

Out[81]:

TypeName	ScreenResolution	Ram	Memory	Gpu	OpSys	Weight	Price	touchscreen	IPS Panel	PPI	Cpu_name	SSD	HDD	Flash Storage	Hybrid
2 in 1 Convertible	Full HD / Touchscreen 1920x1080	4	256 SSD	Intel HD Graphics 620	Windows 10	1.28	90576.000	1	0	182436.090226	Intel Core i5	256	0	0	0
Notebook	1366x768	4	256 SSD	AMD Radeon R5	Windows 10	2.10	24029.280	0	0	78710.897436	AMD processor	256	0	0	0
Notebook	1366x768	4	1000 HDD	Intel HD Graphics 620	No OS	1.90	26101.872	0	0	78710.897436	Intel Core i5	0	1000	0	0
Notebook	IPS Panel Full HD 1920x1080	8	1000 HDD	Nvidia GeForce 930MX	Windows 10	2.50	48697.920	0	1	140254.335260	Intel Core i5	0	1000	0	0
2 in 1 Convertible	Touchscreen 1366x768	8	64 Flash Storage	Intel HD Graphics 500	Chrome OS	1.40	26373.600	1	0	105852.586207	Intel processor	0	0	64	0

In [82]:
```
df.corr()['Price']
```

Out[82]:
```
Ram             0.743007
Weight          0.210370
Price           1.000000
touchscreen     0.191226
IPS Panel       0.252208
PPI             0.480017
SSD             0.670799
HDD            -0.096441
Flash Storage  -0.040511
Hybrid          0.007989
Name: Price, dtype: float64
```

In [83]:
```
df.drop(columns=['Hybrid','Flash Storage'],inplace=True)
```

GPU

In [86]:
```
df['Gpu'].value_counts()
```

Out[86]:
```
Intel HD Graphics 620      281
Intel HD Graphics 520      185
Intel UHD Graphics 620      68
Nvidia GeForce GTX 1050     66
Nvidia GeForce GTX 1060     48
                          ...
AMD Radeon R5 520            1
AMD Radeon R7                1
Intel HD Graphics 540        1
AMD Radeon 540               1
ARM Mali T860 MP4            1
Name: Gpu, Length: 110, dtype: int64
```

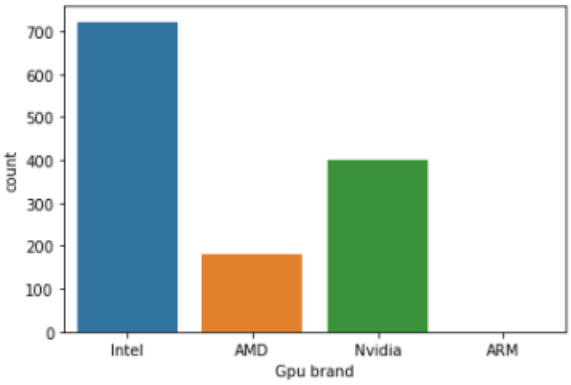
In [92]:
```
for i in df['Gpu'].unique():
    print(i.split()[0])
```

```
Intel
Intel
Intel
AMD
Intel
AMD
Intel
Nvidia
Intel
Intel
AMD
AMD
Intel
AMD
Nvidia
Intel
Nvidia
AMD
AMD
```

In [93]:
```
df['Gpu brand']=df['Gpu'].apply(lambda x:x.split()[0])
sn.countplot(df['Gpu brand'])
```

```
C:\Users\basha\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword a
rg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
```

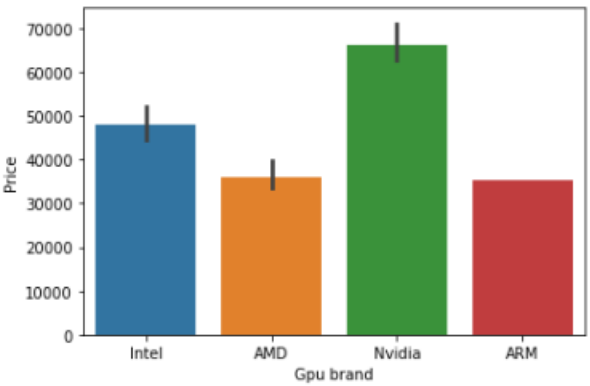
Out[93]: <AxesSubplot:xlabel='Gpu brand', ylabel='count'>



In [96]:
```
sn.barplot(df['Gpu brand'],df['Price'],estimator=np.median)
```

```
C:\Users\basha\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword ar
gs: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[96]: <AxesSubplot:xlabel='Gpu brand', ylabel='Price'>



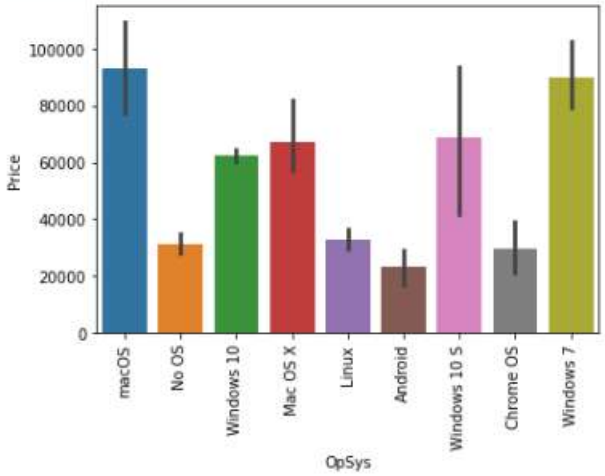
OPS

In [102]:
```
df['OpSys'].value_counts()
```

Out[102]:
```
Windows 10      1072
No OS             66
Linux             62
Windows 7         45
Chrome OS         27
macOS             13
Mac OS X           8
Windows 10 S       8
Android            2
Name: OpSys, dtype: int64
```

In [104]:
```
sn.barplot(df['OpSys'],df['Price'])
plt.xticks(rotation='vertical')
plt.show()
```

```
C:\Users\basha\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword ar
gs: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [105]:
```
df['OpSys'].unique()
```

Out[105]:
```
array(['macOS', 'No OS', 'Windows 10', 'Mac OS X', 'Linux', 'Android',
       'Windows 10 S', 'Chrome OS', 'Windows 7'], dtype=object)
```

In [106]:
```
def club(text):
    if text=='Windows 10 S' or text =='Windows 10' or text == 'Windows 7':
        return 'Windows'
    elif text=='macOS' or text=='Mac OS X' :
        return 'Mac'
    else:
        return 'Other'
df['OpSys']=df['OpSys'].apply(lambda x:club(x))
df.head()
```

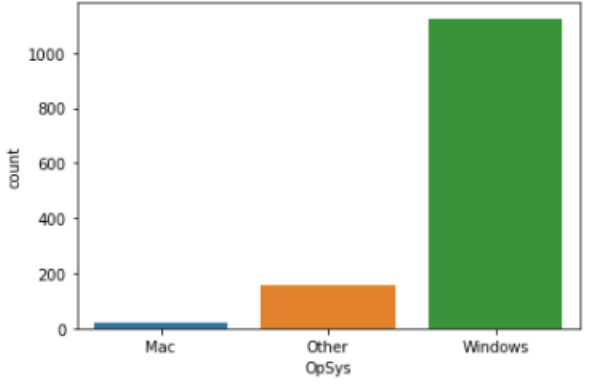
Out[106]:

	Company	TypeName	Ram	OpSys	Weight	Price	touchscreen	IPS Panel	PPI	Cpu_name	SSD	HDD	Gpu brand
0	Apple	Ultrabook	8	Mac	1.37	71378.6832	0	1	342616.541353	Intel Core i5	128	0	Intel
1	Apple	Ultrabook	8	Mac	1.34	47895.5232	0	0	108406.015038	Intel Core i5	0	0	Intel
2	HP	Notebook	8	Other	1.86	30636.0000	0	0	155538.461538	Intel Core i5	256	0	Intel
3	Apple	Ultrabook	16	Mac	1.83	135195.3360	0	1	374493.506494	Intel Core i7	512	0	AMD
4	Apple	Ultrabook	8	Mac	1.37	96095.8080	0	1	342616.541353	Intel Core i5	256	0	Intel

In [107]:
```
sn.countplot(df['OpSys'])
```

```
C:\Users\basha\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword a
rg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
```

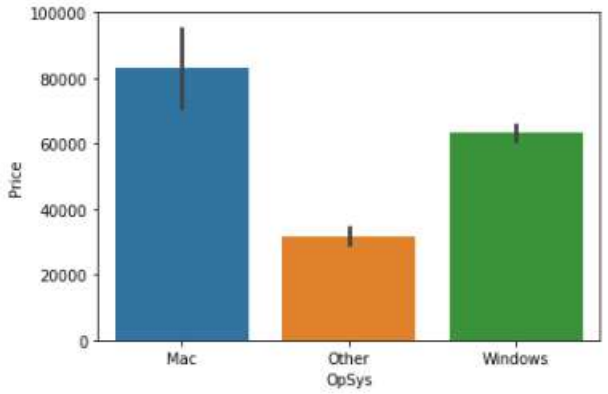
Out[107]: <AxesSubplot:xlabel='OpSys', ylabel='count'>



In [108]:
```
sn.barplot(df['OpSys'],df['Price'])
```

```
C:\Users\basha\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword ar
gs: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[108]: <AxesSubplot:xlabel='OpSys', ylabel='Price'>

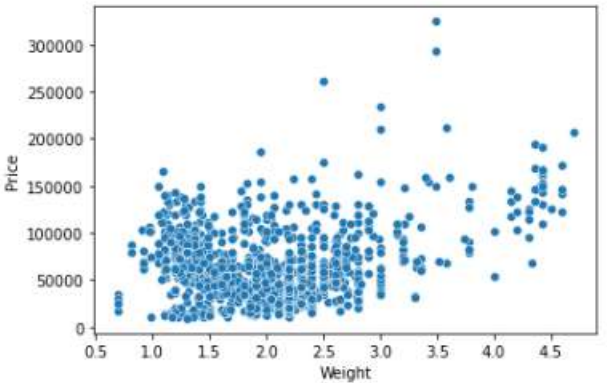


weight Analysis

In [110]:
```
sn.scatterplot(df['Weight'],df['Price'])
```

```
C:\Users\basha\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword ar
gs: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[110]: <AxesSubplot:xlabel='Weight', ylabel='Price'>



In [113]:
```
plt.figure(figsize=(15,7))
sn.heatmap(df.corr(),annot=True,cmap='plasma')
```

Out[113]: <AxesSubplot:>