

## Project Documentation: Exploratory Data Analysis for laptop dataset

Exploratory Data Analysis of Laptop Dataset : Title:Laptop Data Analysis Name : K.Selvalakshmi  
DA/DS : Data Analytics (DA) Batch number : B4(June-Online)(M)-DA&DS Online/Offline : Online

### Table of Contents :

- 1.Introduction
- 2.Aim
- 3.Business Problem / Problem Statement
- 4.Project Workflow
- 5.Data Understanding
- 6.Data Cleaning - Missing Values Imputation, Outliers, Handling Inconsistent Values
- 7.Obtaining Derived Metrics
- 8.Filtering Data for Analysis
- 9.EDA-Univariate Analysis
- 10.Bivariate Analysis
- 11.Multivariate Analysis
- 12.Overall Insights Obtained from Analysis
- 13.Conclusion

1) Introduction : This dataset is a collection of features related to various laptops, such as brand, processor type, RAM, storage capacity, and other specifications. The dataset also includes the corresponding prices of these laptops. This dataset can be used for regression analysis to predict the prices of laptops based on their features.

Columns in the dataset related to Laptop: Company: The manufacturer or brand of the laptop. TypeName: The type or category of the laptop (e.g., gaming, ultrabook). Inches: The size of the laptop screen in inches. ScreenResolution: The resolution and display technology of the laptop screen. Cpu: The processor (CPU) model of the laptop. Ram: The amount of Random Access Memory (RAM) in the laptop. Memory: The storage capacity (hard drive or SSD) of the laptop. Gpu: The graphics processor (GPU) model of the laptop. OpSys: The operating system installed on the laptop. Weight: The weight of the laptop in kilograms. Price: The price of the laptop in the local currency. \*\*

2) Aim :cleaning the dataset.

3) Problem Statement: The laptop market is highly competitive, with numerous manufacturers and models available to consumers. With the rapid advancement of technology, consumer preferences and expectations are constantly evolving. To stay ahead in the market, manufacturers and retailers need to understand the key factors that influence laptop pricing, performance, and consumer preferences. Specifically, the problem is: How do various laptop characteristics, such as screen size, RAM, CPU model, and brand reputation, impact pricing and consumer purchasing decisions? What are the most important features that consumers look for when buying a laptop, and how do these preferences vary across different segments of the market? How can manufacturers and retailers use data-driven insights to inform product development, marketing strategies, and pricing decisions to stay competitive in the market?

4) Project Workflow : Overview of the project workflow or methodology followed.

## Data Cleaning Exploratory Data Analysis (EDA) Data Visualization Analysis and Interpretation Documentation

5) Data Understanding : > Description of the dataset, including structure, dimensions, and data types. > Summary statistics and insights gained from initial data exploration. Insights gained from initial data exploration There are 1303 rows and 12 columns in the Dataset. From the info we conclude that Price Columns have Numerical Values whereas column like Company, TypeName, Inches, ScreenResolution, Cpu, Ram, Memory, Gpu, OpSys, Weight have Categorical Values. Unnamed: 0 column should be dropped

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
df=pd.read_csv("C:/Users/ADMIN/Desktop/laptopData.csv")
```

df

	Unnamed: 0	Company	TypeName	Inches	\
0	0.0	Apple	Ultrabook	13.3	
1	1.0	Apple	Ultrabook	13.3	
2	2.0	HP	Notebook	15.6	
3	3.0	Apple	Ultrabook	15.4	
4	4.0	Apple	Ultrabook	13.3	
...	...	...	...	...	
1298	1298.0	Lenovo	2 in 1 Convertible	14	
1299	1299.0	Lenovo	2 in 1 Convertible	13.3	
1300	1300.0	Lenovo	Notebook	65.4	
1301	1301.0	HP	Notebook	15.6	
1302	1302.0	Asus	Notebook	15.6	

	ScreenResolution	\
0	IPS Panel Retina Display 2560x1600	
1	1440x900	
2	Full HD 1920x1080	
3	IPS Panel Retina Display 2880x1800	
4	IPS Panel Retina Display 2560x1600	
...	...	
1298	IPS Panel Full HD / Touchscreen 1920x1080	
1299	IPS Panel Quad HD+ / Touchscreen 3200x1800	
1300	1366x768	
1301	1366x768	
1302	1366x768	

	Cpu	Ram	Memory
\			
0	Intel Core i5 2.3GHz	8GB	128GB SSD
1	Intel Core i5 1.8GHz	8GB	128GB Flash Storage

2	Intel Core i5 7200U	2.5GHz	8GB	256GB SSD
3	Intel Core i7	2.7GHz	16GB	512GB SSD
4	Intel Core i5	3.1GHz	8GB	256GB SSD
...	...	...	...	...
1298	Intel Core i7 6500U	2.5GHz	4GB	128GB SSD
1299	Intel Core i7 6500U	2.5GHz	16GB	512GB SSD
1300	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB Flash Storage
1301	Intel Core i7 6500U	2.5GHz	6GB	1TB HDD
1302	Intel Celeron Dual Core N3050	1.6GHz	4GB	500GB HDD

		Gpu	OpSys	Weight	Price
0	Intel Iris Plus Graphics	640	macOS	1.37kg	71378.6832
1	Intel HD Graphics	6000	macOS	1.34kg	47895.5232
2	Intel HD Graphics	620	No OS	1.86kg	30636.0000
3	AMD Radeon Pro	455	macOS	1.83kg	135195.3360
4	Intel Iris Plus Graphics	650	macOS	1.37kg	96095.8080
...	...	...	...	...	...
1298	Intel HD Graphics	520	Windows 10	1.8kg	33992.6400
1299	Intel HD Graphics	520	Windows 10	1.3kg	79866.7200
1300	Intel HD Graphics		Windows 10	1.5kg	12201.1200
1301	AMD Radeon R5 M330		Windows 10	2.19kg	40705.9200
1302	Intel HD Graphics		Windows 10	2.2kg	19660.3200

[1303 rows x 12 columns]

```
type(df)
```

```
pandas.core.frame.DataFrame
```

```
df=df.iloc[:,1:]
```

```
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	1273 non-null	object
1	TypeName	1253 non-null	object
2	Inches	1220 non-null	object
3	ScreenResolution	1273 non-null	object

4	Cpu	1273	non-null	object
5	Ram	1259	non-null	object
6	Memory	1273	non-null	object
7	Gpu	1273	non-null	object
8	OpSys	1273	non-null	object
9	Weight	1259	non-null	object
10	Price	1273	non-null	float64

dtypes: float64(1), object(10)

memory usage: 112.1+ KB

df.head()

	Company	TypeName	Inches	ScreenResolution \			
0	Apple	Ultrabook	13.3	IPS Panel	Retina Display	2560x1600	
1	Apple	Ultrabook	13.3			1440x900	
2	HP	Notebook	15.6		Full HD	1920x1080	
3	Apple	Ultrabook	15.4	IPS Panel	Retina Display	2880x1800	
4	Apple	Ultrabook	13.3	IPS Panel	Retina Display	2560x1600	

		Cpu	Ram	Memory \
0	Intel Core i5	2.3GHz	8GB	128GB SSD
1	Intel Core i5	1.8GHz	8GB	128GB Flash Storage
2	Intel Core i5 7200U	2.5GHz	8GB	256GB SSD
3	Intel Core i7	2.7GHz	16GB	512GB SSD
4	Intel Core i5	3.1GHz	8GB	256GB SSD

		Gpu	OpSys	Weight	Price
0	Intel Iris Plus Graphics	640	macOS	1.37kg	71378.6832
1	Intel HD Graphics 6000		macOS	1.34kg	47895.5232
2	Intel HD Graphics 620		No OS	1.86kg	30636.0000
3	AMD Radeon Pro 455		macOS	1.83kg	135195.3360
4	Intel Iris Plus Graphics	650	macOS	1.37kg	96095.8080

df.tail()

	Company	TypeName	Inches \
1298	Lenovo	2 in 1 Convertible	14
1299	Lenovo	2 in 1 Convertible	13.3
1300	Lenovo	Notebook	65.4
1301	HP	Notebook	15.6
1302	Asus	Notebook	15.6

	ScreenResolution \	
1298	IPS Panel Full HD / Touchscreen	1920x1080
1299	IPS Panel Quad HD+ / Touchscreen	3200x1800
1300		1366x768
1301		1366x768
1302		1366x768

	Cpu	Ram
Memory \		

1298	Intel Core i7 6500U 2.5GHz	4GB	128GB SSD
1299	Intel Core i7 6500U 2.5GHz	16GB	512GB SSD
1300	Intel Celeron Dual Core N3050 1.6GHz	2GB	64GB Flash Storage
1301	Intel Core i7 6500U 2.5GHz	6GB	1TB HDD
1302	Intel Celeron Dual Core N3050 1.6GHz	4GB	500GB HDD

		Gpu	OpSys	Weight	Price
1298	Intel	HD Graphics 520	Windows 10	1.8kg	33992.64
1299	Intel	HD Graphics 520	Windows 10	1.3kg	79866.72
1300	Intel	HD Graphics	Windows 10	1.5kg	12201.12
1301	AMD	Radeon R5 M330	Windows 10	2.19kg	40705.92
1302	Intel	HD Graphics	Windows 10	2.2kg	19660.32

df.shape

(1303, 11)

df.describe()

	Price
count	1273.000000
mean	59955.814073
std	37332.251005
min	9270.720000
25%	31914.720000
50%	52161.120000
75%	79333.387200
max	324954.720000

df.isnull()

	Company	TypeName	Inches	ScreenResolution	Cpu	Ram
Memory \						
0	False	False	False	False	False	False
False						
1	False	False	False	False	False	False
False						
2	False	False	False	False	False	False
False						
3	False	False	False	False	False	False
False						
4	False	False	False	False	False	False
False						
...	...	...	...	...	...	...
.						
1298	False	False	False	False	False	False

```
False
1299      False      False      False      False      False      False      False
False
1300      False      False      False      False      False      False      False
False
1301      False      False      False      False      False      False      False
False
1302      False      False      False      False      False      False      False
False
```

```
      Gpu  OpSys  Weight  Price
0      False  False  False  False
1      False  False  False  False
2      False  False  False  False
3      False  False  False  False
4      False  False  False  False
...      ...      ...      ...      ...
1298  False  False  False  False
1299  False  False  False  False
1300  False  False  False  False
1301  False  False  False  False
1302  False  False  False  False
```

```
[1303 rows x 11 columns]
```

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

```
Company          30
TypeName         50
Inches           83
ScreenResolution  30
Cpu              30
Ram              44
Memory           30
Gpu              30
OpSys            30
Weight           44
Price            30
dtype: int64
```

```
df['Company'].unique()
```

```
array(['Apple', 'HP', 'Acer', 'Asus', 'Dell', 'Lenovo', nan, 'Chuwi',
      'MSI', 'Microsoft', 'Toshiba', 'Huawei', 'Xiaomi', 'Vero',
      'Razer',
      'Mediacom', 'Samsung', 'Google', 'Fujitsu', 'LG'],
      dtype=object)
```

```
for x in df['Company'].unique():
    print(x,df[df['Company'] == x]['TypeName'].unique())
```

```

Apple ['Ultrabook']
HP ['Notebook' 'Ultrabook' 'Gaming' '2 in 1 Convertible' 'Workstation'
nan
'Netbook']
Acer ['Notebook' 'Ultrabook' '2 in 1 Convertible' 'Gaming' 'Netbook'
nan]
Asus ['Ultrabook' 'Notebook' 'Gaming' '2 in 1 Convertible' 'Netbook'
nan]
Dell ['Notebook' 'Ultrabook' '2 in 1 Convertible' 'Gaming'
'Workstation'
'Netbook' nan]
Lenovo ['Notebook' 'Gaming' '2 in 1 Convertible' 'Netbook' 'Ultrabook'
'Workstation' nan]
nan []
Chuwi ['Notebook']
MSI ['Gaming']
Microsoft ['Ultrabook']
Toshiba ['Notebook' 'Ultrabook' nan]
Huawei ['Ultrabook']
Xiaomi ['Notebook' 'Ultrabook']
Vero ['Notebook']
Razer ['Gaming' 'Ultrabook']
Mediacom ['2 in 1 Convertible' 'Notebook']
Samsung ['Netbook' 'Notebook' 'Ultrabook' '2 in 1 Convertible']
Google ['Ultrabook']
Fujitsu ['Notebook' nan]
LG ['Ultrabook']

```

```
df.dropna(subset=['Company'], inplace=True)
```

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

```

Company          0
TypeName         20
Inches           53
ScreenResolution  0
Cpu              0
Ram              14
Memory           0
Gpu              0
OpSys            0
Weight           14
Price            0
dtype: int64

```

## 6) Data Cleaning:

- » Check for Duplicate Value
- » Check for Check for Missing Values
- » Check for DataType

- » Check the number of Unique Value in every Columns
- » Check Statistics of Dataset

```
TypeName_mode=df.TypeName.mode()[0]
```

```
TypeName_mode
```

```
'Notebook'
```

```
df.TypeName.fillna(TypeName_mode,inplace=True)
```

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

```
Company      0
TypeName     0
Inches       53
ScreenResolution  0
Cpu          0
Ram          14
Memory       0
Gpu          0
OpSys        0
Weight       14
Price        0
dtype: int64
```

```
df['Inches'].unique()
```

```
array(['13.3', '15.6', '15.4', '14', '12', '17.3', '13.5', '12.5',
      '13',
      '18.4', '13.9', '11.6', '25.6', '35.6', '12.3', '27.3', '24',
      '33.5', '?', '31.6', '17', '15', nan, '14.1', '11.3', '88.1',
      '21.8', '10.1', '75.7', '8.4', '2.1', '111.8', '89.2', '65.4'],
      dtype=object)
```

```
Inches_mode=df.Inches.mode()[0]
```

```
Inches_mode
```

```
'15.6'
```

```
df.Inches.fillna(Inches_mode,inplace=True)
```

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

```
Company      0
TypeName     0
Inches       0
ScreenResolution  0
Cpu          0
Ram          14
Memory       0
```



```

    Gpu          0
    OpSys        0
    Weight      14
    Price        0
    dtype: int64

df['Ram'].unique()

array(['8GB', '16GB', '4GB', nan, '2GB', '12GB', '64GB', '6GB',
      '32GB',
      '24GB', '1GB'], dtype=object)

Ram_mode=df.Ram.mode()[0]

Ram_mode
'8GB'

df.Ram.fillna(Ram_mode,inplace=True)

df.isnull().sum()

Company          0
TypeName         0
Inches           0
ScreenResolution 0
Cpu              0
Ram              0
Memory           0
Gpu              0
OpSys            0
Weight          14
Price            0
dtype: int64

df['Weight'].unique()

array(['1.37kg', '1.34kg', '1.86kg', '1.83kg', '2.1kg', '2.04kg',
      '1.3kg',
      '1.6kg', '2.2kg', '0.92kg', nan, '1.22kg', '2.5kg', '1.62kg',
      '1.91kg', '2.3kg', '1.35kg', '1.88kg', '1.89kg', '1.65kg',
      '2.71kg', '1.2kg', '1.44kg', '2.8kg', '2kg', '2.65kg',
      '2.77kg',
      '3.2kg', '1.49kg', '2.4kg', '2.13kg', '2.43kg', '1.7kg',
      '1.4kg',
      '1.8kg', '1.9kg', '3kg', '1.252kg', '2.7kg', '2.02kg',
      '1.63kg',
      '1.96kg', '1.21kg', '2.45kg', '1.25kg', '1.5kg', '2.62kg',
      '1.38kg', '1.58kg', '1.85kg', '1.23kg', '2.16kg', '2.36kg',
      '7.2kg', '2.05kg', '1.32kg', '1.75kg', '0.97kg', '2.56kg',
      '1.48kg', '1.74kg', '1.1kg', '1.56kg', '2.03kg', '1.05kg',

```

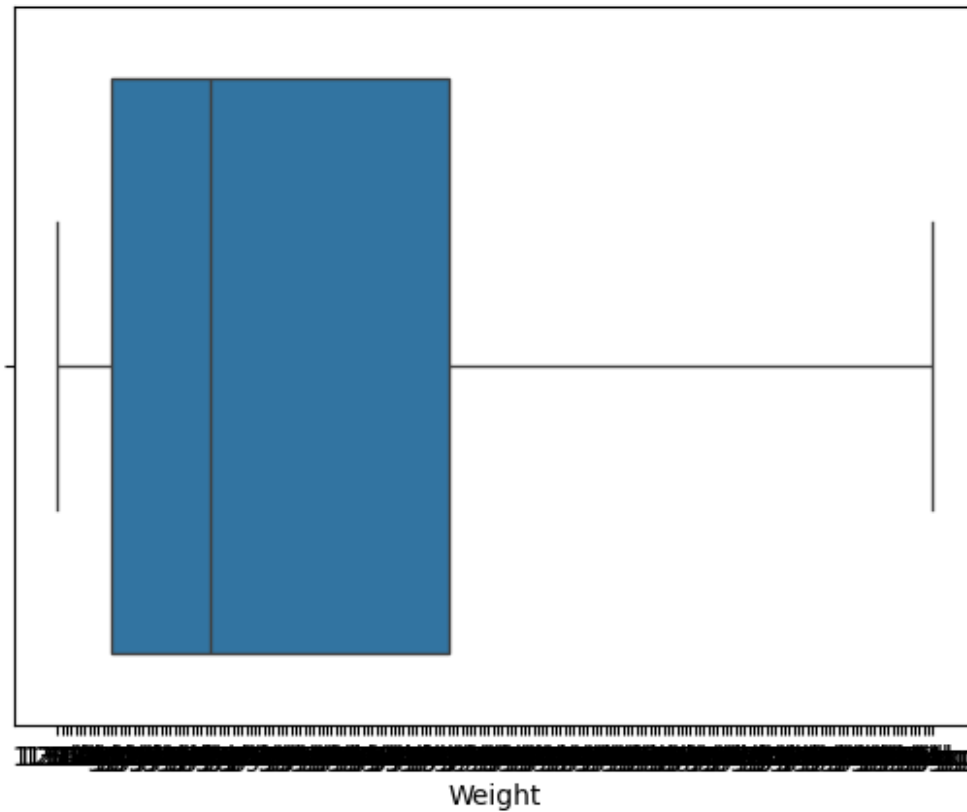
```

'5.4kg',
'4.4kg', '1.90kg', '1.29kg', '2.0kg', '1.95kg', '2.06kg',
'1.12kg',
'3.49kg', '3.35kg', '2.23kg', '?', '2.9kg', '4.42kg', '2.69kg',
'2.37kg', '4.7kg', '3.6kg', '2.08kg', '4.3kg', '1.68kg',
'1.41kg',
'4.14kg', '2.18kg', '2.24kg', '2.67kg', '4.1kg', '2.14kg',
'1.36kg', '2.25kg', '2.15kg', '2.19kg', '2.54kg', '3.42kg',
'5.8kg', '1.28kg', '2.33kg', '1.45kg', '2.79kg', '8.23kg',
'1.26kg', '0.0002kg', '1.84kg', '2.6kg', '2.26kg', '3.25kg',
'11.5kg', '1.13kg', '1.42kg', '0.00008kg', '1.78kg', '1.10kg',
'22.1kg', '1.15kg', '1.27kg', '1.43kg', '2.31kg', '1.16kg',
'1.64kg', '2.17kg', '1.47kg', '3.78kg', '1.79kg', '0.91kg',
'1.99kg', '4.33kg', '9kg', '1.93kg', '10kg', '1.87kg',
'2.63kg',
'3.4kg', '3.14kg', '1.94kg', '1.24kg', '4.6kg', '14kg',
'4.5kg',
'8.4kg', '2.73kg', '1.39kg', '2.29kg', '2.59kg', '2.94kg',
'11.1kg', '1.14kg', '3.8kg', '6.2kg', '3.31kg', '1.09kg',
'3.21kg',
'1.19kg', '29kg', '1.98kg', '1.17kg', '4.36kg', '1.71kg',
'2.32kg',
'4.2kg', '1.55kg', '0.81kg', '1.18kg', '2.72kg', '1.31kg',
'18kg',
'0.920kg', '3.74kg', '1.76kg', '1.54kg', '2.83kg', '2.07kg',
'2.38kg', '3.58kg', '1.08kg', '2.20kg', '0.98kg', '2.75kg',
'1.70kg', '2.99kg', '1.11kg', '2.09kg', '4kg', '3.0kg',
'0.99kg',
'0.69kg', '3.52kg', '2.591kg', '2.21kg', '3.3kg', '2.191kg',
'2.34kg', '4.0kg'], dtype=object)

```

```
sns.boxplot(x=df.Weight)
```

```
<Axes: xlabel='Weight'>
```



```
Weight_mode=df.Weight.mode()[0]
Weight_mode
'2.2kg'
df.Weight.fillna(Weight_mode,inplace=True)
df.isnull().sum()
Company          0
TypeName         0
Inches           0
ScreenResolution 0
Cpu              0
Ram              0
Memory           0
Gpu              0
OpSys            0
Weight           0
Price            0
dtype: int64
df
```

	Company		TypeName	Inches	\
0	Apple		Ultrabook	13.3	
1	Apple		Ultrabook	13.3	
2	HP		Notebook	15.6	
3	Apple		Ultrabook	15.4	
4	Apple		Ultrabook	13.3	
...	...		...	...	
1298	Lenovo	2 in 1	Convertible	14	
1299	Lenovo	2 in 1	Convertible	13.3	
1300	Lenovo		Notebook	65.4	
1301	HP		Notebook	15.6	
1302	Asus		Notebook	15.6	
				ScreenResolution	\
0			IPS Panel Retina Display	2560x1600	
1				1440x900	
2			Full HD	1920x1080	
3			IPS Panel Retina Display	2880x1800	
4			IPS Panel Retina Display	2560x1600	
...			...	...	
1298			IPS Panel Full HD / Touchscreen	1920x1080	
1299			IPS Panel Quad HD+ / Touchscreen	3200x1800	
1300				1366x768	
1301				1366x768	
1302				1366x768	
			Cpu	Ram	Memory
\					
0			Intel Core i5 2.3GHz	8GB	128GB SSD
1			Intel Core i5 1.8GHz	8GB	128GB Flash Storage
2			Intel Core i5 7200U 2.5GHz	8GB	256GB SSD
3			Intel Core i7 2.7GHz	16GB	512GB SSD
4			Intel Core i5 3.1GHz	8GB	256GB SSD
...			...	...	...
1298			Intel Core i7 6500U 2.5GHz	4GB	128GB SSD
1299			Intel Core i7 6500U 2.5GHz	16GB	512GB SSD
1300			Intel Celeron Dual Core N3050 1.6GHz	2GB	64GB Flash Storage
1301			Intel Core i7 6500U 2.5GHz	6GB	1TB HDD
1302			Intel Celeron Dual Core N3050 1.6GHz	4GB	500GB HDD

		Gpu	OpSys	Weight	Price
0	Intel Iris Plus Graphics	640	macOS	1.37kg	71378.6832
1	Intel HD Graphics	6000	macOS	1.34kg	47895.5232
2	Intel HD Graphics	620	No OS	1.86kg	30636.0000
3	AMD Radeon Pro	455	macOS	1.83kg	135195.3360
4	Intel Iris Plus Graphics	650	macOS	1.37kg	96095.8080
...		...	...	...	...
1298	Intel HD Graphics	520	Windows 10	1.8kg	33992.6400
1299	Intel HD Graphics	520	Windows 10	1.3kg	79866.7200
1300	Intel HD Graphics		Windows 10	1.5kg	12201.1200
1301	AMD Radeon R5 M330		Windows 10	2.19kg	40705.9200
1302	Intel HD Graphics		Windows 10	2.2kg	19660.3200

[1273 rows x 11 columns]

df['Ram']

0	8GB
1	8GB
2	8GB
3	16GB
4	8GB
...	
1298	4GB
1299	16GB
1300	2GB
1301	6GB
1302	4GB

Name: Ram, Length: 1273, dtype: object

df['Ram']=pd.Series(df['Ram'].str.replace('GB',''))

df['Weight']

0	1.37kg
1	1.34kg
2	1.86kg
3	1.83kg
4	1.37kg
...	
1298	1.8kg
1299	1.3kg
1300	1.5kg
1301	2.19kg
1302	2.2kg

Name: Weight, Length: 1273, dtype: object

df['Weight']=pd.Series(df['Weight'].str.replace('kg',''))

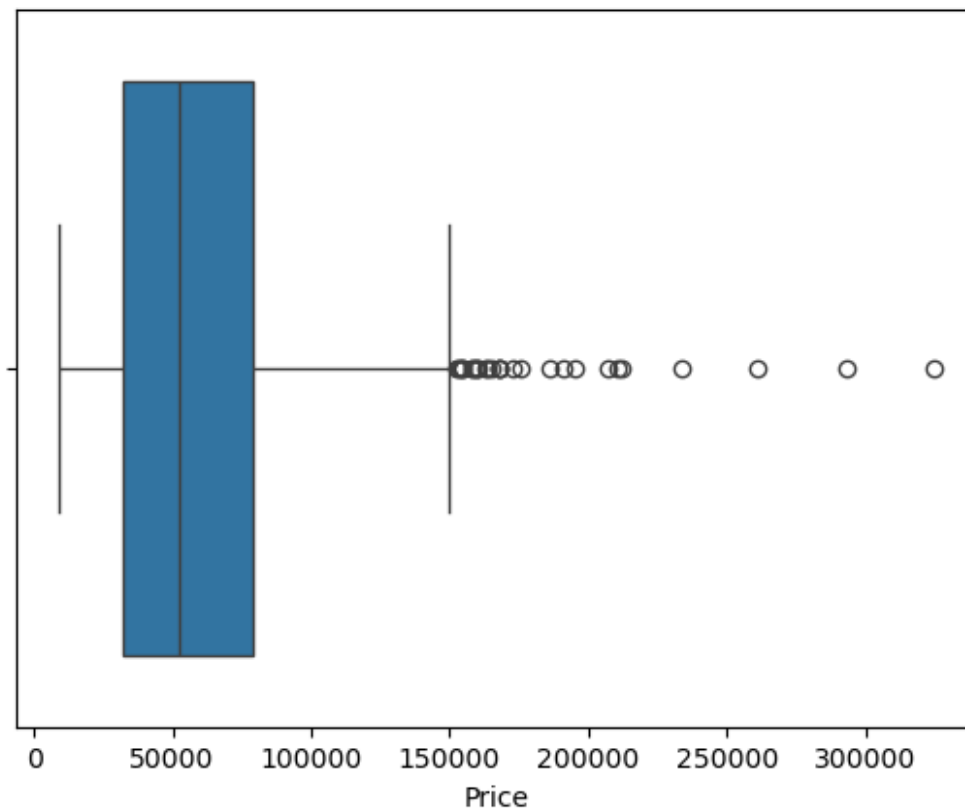
df

	Company		TypeName	Inches	\	
0	Apple		Ultrabook	13.3		
1	Apple		Ultrabook	13.3		
2	HP		Notebook	15.6		
3	Apple		Ultrabook	15.4		
4	Apple		Ultrabook	13.3		
...	...		...	...		
1298	Lenovo	2 in 1	Convertible	14		
1299	Lenovo	2 in 1	Convertible	13.3		
1300	Lenovo		Notebook	65.4		
1301	HP		Notebook	15.6		
1302	Asus		Notebook	15.6		
			ScreenResolution	\		
0		IPS Panel	Retina Display	2560x1600		
1				1440x900		
2			Full HD	1920x1080		
3		IPS Panel	Retina Display	2880x1800		
4		IPS Panel	Retina Display	2560x1600		
...				...		
1298		IPS Panel	Full HD / Touchscreen	1920x1080		
1299		IPS Panel	Quad HD+ / Touchscreen	3200x1800		
1300				1366x768		
1301				1366x768		
1302				1366x768		
			Cpu	Ram	Memory	\
0		Intel Core i5	2.3GHz	8	128GB SSD	
1		Intel Core i5	1.8GHz	8	128GB Flash Storage	
2		Intel Core i5 7200U	2.5GHz	8	256GB SSD	
3		Intel Core i7	2.7GHz	16	512GB SSD	
4		Intel Core i5	3.1GHz	8	256GB SSD	
...			...	...	...	
1298		Intel Core i7 6500U	2.5GHz	4	128GB SSD	
1299		Intel Core i7 6500U	2.5GHz	16	512GB SSD	
1300	Intel	Celeron Dual Core N3050	1.6GHz	2	64GB Flash Storage	
1301		Intel Core i7 6500U	2.5GHz	6	1TB HDD	
1302	Intel	Celeron Dual Core N3050	1.6GHz	4	500GB HDD	
			Gpu	OpSys	Weight	Price
0	Intel	Iris Plus Graphics 640		macOS	1.37	71378.6832
1		Intel HD Graphics 6000		macOS	1.34	47895.5232
2		Intel HD Graphics 620		No OS	1.86	30636.0000
3		AMD Radeon Pro 455		macOS	1.83	135195.3360
4	Intel	Iris Plus Graphics 650		macOS	1.37	96095.8080
...			...	...	...	...
1298		Intel HD Graphics 520	Windows	10	1.8	33992.6400
1299		Intel HD Graphics 520	Windows	10	1.3	79866.7200
1300		Intel HD Graphics	Windows	10	1.5	12201.1200
1301		AMD Radeon R5 M330	Windows	10	2.19	40705.9200

```

1302          Intel HD Graphics  Windows 10    2.2    19660.3200
[1273 rows x 11 columns]
df.shape
(1273, 11)
sns.boxplot(x=df['Price'])
<Axes: xlabel='Price'>

```



#### 7) Obtaining Derived Metrics:

Using Boxplot method to find the IQR and remove the outlier in the original dataset

```
df1=df.select_dtypes(exclude=['object'])
```

```
df1
```

	Price
0	71378.6832
1	47895.5232
2	30636.0000
3	135195.3360

```

4      96095.8080
...
1298   33992.6400
1299   79866.7200
1300   12201.1200
1301   40705.9200
1302   19660.3200

[1273 rows x 1 columns]

q1=df1.quantile(0.25)

q1
Price      31914.72
Name: 0.25, dtype: float64

q3=df1.quantile(0.75)

q3
Price      79333.3872
Name: 0.75, dtype: float64

IQR=q3-q1

IQR
Price      47418.6672
dtype: float64

a=((df1<(q1-1.5*IQR)|(df1>q3+1.5*IQR)))

a
   Price
0  False
1  False
2  False
3  False
4  False
...
1298 False
1299 False
1300 False
1301 False
1302 False

[1273 rows x 1 columns]

```

## 8) Filtering Data for Analysis

```
filter=df[(~a).all(axis=1)]
```



filter

	Company	TypeName	Inches	\
0	Apple	Ultrabook	13.3	
1	Apple	Ultrabook	13.3	
2	HP	Notebook	15.6	
3	Apple	Ultrabook	15.4	
4	Apple	Ultrabook	13.3	
...	...	...	...	
1298	Lenovo	2 in 1 Convertible	14	
1299	Lenovo	2 in 1 Convertible	13.3	
1300	Lenovo	Notebook	65.4	
1301	HP	Notebook	15.6	
1302	Asus	Notebook	15.6	

	ScreenResolution	\
0	IPS Panel Retina Display 2560x1600	
1	1440x900	
2	Full HD 1920x1080	
3	IPS Panel Retina Display 2880x1800	
4	IPS Panel Retina Display 2560x1600	
...	...	
1298	IPS Panel Full HD / Touchscreen 1920x1080	
1299	IPS Panel Quad HD+ / Touchscreen 3200x1800	
1300	1366x768	
1301	1366x768	
1302	1366x768	

	Cpu	Ram	Memory	\
0	Intel Core i5 2.3GHz	8	128GB SSD	
1	Intel Core i5 1.8GHz	8	128GB Flash Storage	
2	Intel Core i5 7200U 2.5GHz	8	256GB SSD	
3	Intel Core i7 2.7GHz	16	512GB SSD	
4	Intel Core i5 3.1GHz	8	256GB SSD	
...	...	...	...	
1298	Intel Core i7 6500U 2.5GHz	4	128GB SSD	
1299	Intel Core i7 6500U 2.5GHz	16	512GB SSD	
1300	Intel Celeron Dual Core N3050 1.6GHz	2	64GB Flash Storage	
1301	Intel Core i7 6500U 2.5GHz	6	1TB HDD	
1302	Intel Celeron Dual Core N3050 1.6GHz	4	500GB HDD	

	Gpu	OpSys	Weight	Price
0	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832
1	Intel HD Graphics 6000	macOS	1.34	47895.5232
2	Intel HD Graphics 620	No OS	1.86	30636.0000
3	AMD Radeon Pro 455	macOS	1.83	135195.3360
4	Intel Iris Plus Graphics 650	macOS	1.37	96095.8080
...	...	...	...	...
1298	Intel HD Graphics 520	Windows 10	1.8	33992.6400
1299	Intel HD Graphics 520	Windows 10	1.3	79866.7200

1300	Intel HD Graphics	Windows 10	1.5	12201.1200
1301	AMD Radeon R5 M330	Windows 10	2.19	40705.9200
1302	Intel HD Graphics	Windows 10	2.2	19660.3200

[1273 rows x 11 columns]

`filter.shape`

(1273, 11)

9)EDA-Univariate Analysis

`filter.groupby(['Inches']).count()`

OpSys \ Inches	Company	TypeName	ScreenResolution	Cpu	Ram	Memory	Gpu
10.1	2	2	2	2	2	2	2
11.3	1	1	1	1	1	1	1
11.6	31	31	31	31	31	31	31
11.8	1	1	1	1	1	1	1
12	6	6	6	6	6	6	6
12.3	5	5	5	5	5	5	5
12.5	31	31	31	31	31	31	31
13	2	2	2	2	2	2	2
13.3	155	155	155	155	155	155	155
13.5	5	5	5	5	5	5	5
13.9	6	6	6	6	6	6	6
14	184	184	184	184	184	184	184
14.1	1	1	1	1	1	1	1
15	4	4	4	4	4	4	4
15.4	4	4	4	4	4	4	4
15.6	668	668	668	668	668	668	668
17	1	1	1	1	1	1	1

1								
17.3	148	148		148	148	148	148	148
148								
18.4	1	1		1	1	1	1	1
1								
2.1	1	1		1	1	1	1	1
1								
21.8	1	1		1	1	1	1	1
1								
24	2	2		2	2	2	2	2
2								
25.6	1	1		1	1	1	1	1
1								
27.3	1	1		1	1	1	1	1
1								
31.6	1	1		1	1	1	1	1
1								
33.5	1	1		1	1	1	1	1
1								
35.6	3	3		3	3	3	3	3
3								
65.4	1	1		1	1	1	1	1
1								
75.7	1	1		1	1	1	1	1
1								
8.4	1	1		1	1	1	1	1
1								
88.1	1	1		1	1	1	1	1
1								
89.2	1	1		1	1	1	1	1
1								
?	1	1		1	1	1	1	1
1								

	Weight	Price
Inches		
10.1	2	2
11.3	1	1
11.6	31	31
111.8	1	1
12	6	6
12.3	5	5
12.5	31	31
13	2	2
13.3	155	155
13.5	5	5
13.9	6	6
14	184	184
14.1	1	1

15	4	4
15.4	4	4
15.6	668	668
17	1	1
17.3	148	148
18.4	1	1
2.1	1	1
21.8	1	1
24	2	2
25.6	1	1
27.3	1	1
31.6	1	1
33.5	1	1
35.6	3	3
65.4	1	1
75.7	1	1
8.4	1	1
88.1	1	1
89.2	1	1
?	1	1

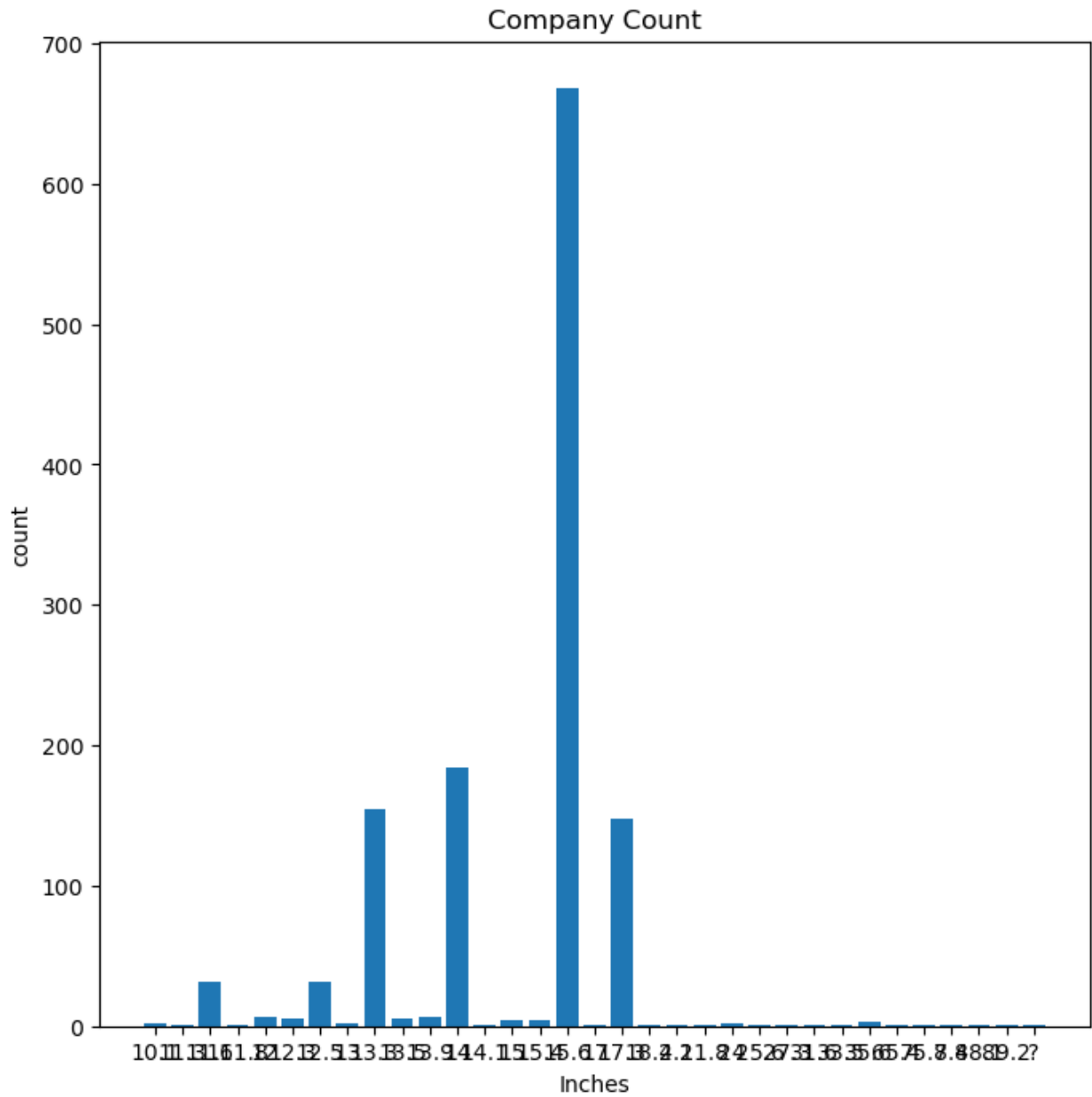
```
f=filter.groupby(['Inches']).size().reset_index(name='count').rename(columns={'Inches':'Inches'})
```

f

	Inches	count
0	10.1	2
1	11.3	1
2	11.6	31
3	11.8	1
4	12	6
5	12.3	5
6	12.5	31
7	13	2
8	13.3	155
9	13.5	5
10	13.9	6
11	14	184
12	14.1	1
13	15	4
14	15.4	4
15	15.6	668
16	17	1
17	17.3	148
18	18.4	1
19	2.1	1
20	21.8	1
21	24	2
22	25.6	1

23	27.3	1
24	31.6	1
25	33.5	1
26	35.6	3
27	65.4	1
28	75.7	1
29	8.4	1
30	88.1	1
31	89.2	1
32	?	1

```
plt.figure(figsize=(8,8))
plt.bar(f['Inches'],f['count'])
plt.title('Company Count')
plt.xlabel('Inches')
plt.ylabel('count')
plt.show()
```



```
filter.groupby(['Price']).count()
```

	Company	TypeName	Inches	ScreenResolution	Cpu	Ram
Memory \ Price						
9270.720	1	1	1	1	1	1
10442.880	1	1	1	1	1	1
10602.720	2	2	2	2	2	2

10810.512	1	1	1	1	1	1
1						
11135.520	4	4	4	4	4	4
4						
...	...	...	...	...	...	...
...						
211788.000	1	1	1	1	1	1
1						
233845.920	1	1	1	1	1	1
1						
261018.720	1	1	1	1	1	1
1						
292986.720	1	1	1	1	1	1
1						
324954.720	1	1	1	1	1	1
1						

	Gpu	OpSys	Weight
Price			
9270.720	1	1	1
10442.880	1	1	1
10602.720	2	2	2
10810.512	1	1	1
11135.520	4	4	4
...	...	...	...
211788.000	1	1	1
233845.920	1	1	1
261018.720	1	1	1
292986.720	1	1	1
324954.720	1	1	1

[777 rows x 10 columns]

```
g=filter.groupby(['Price']).size().reset_index(name='count').rename(columns={'Price':'price'})
```

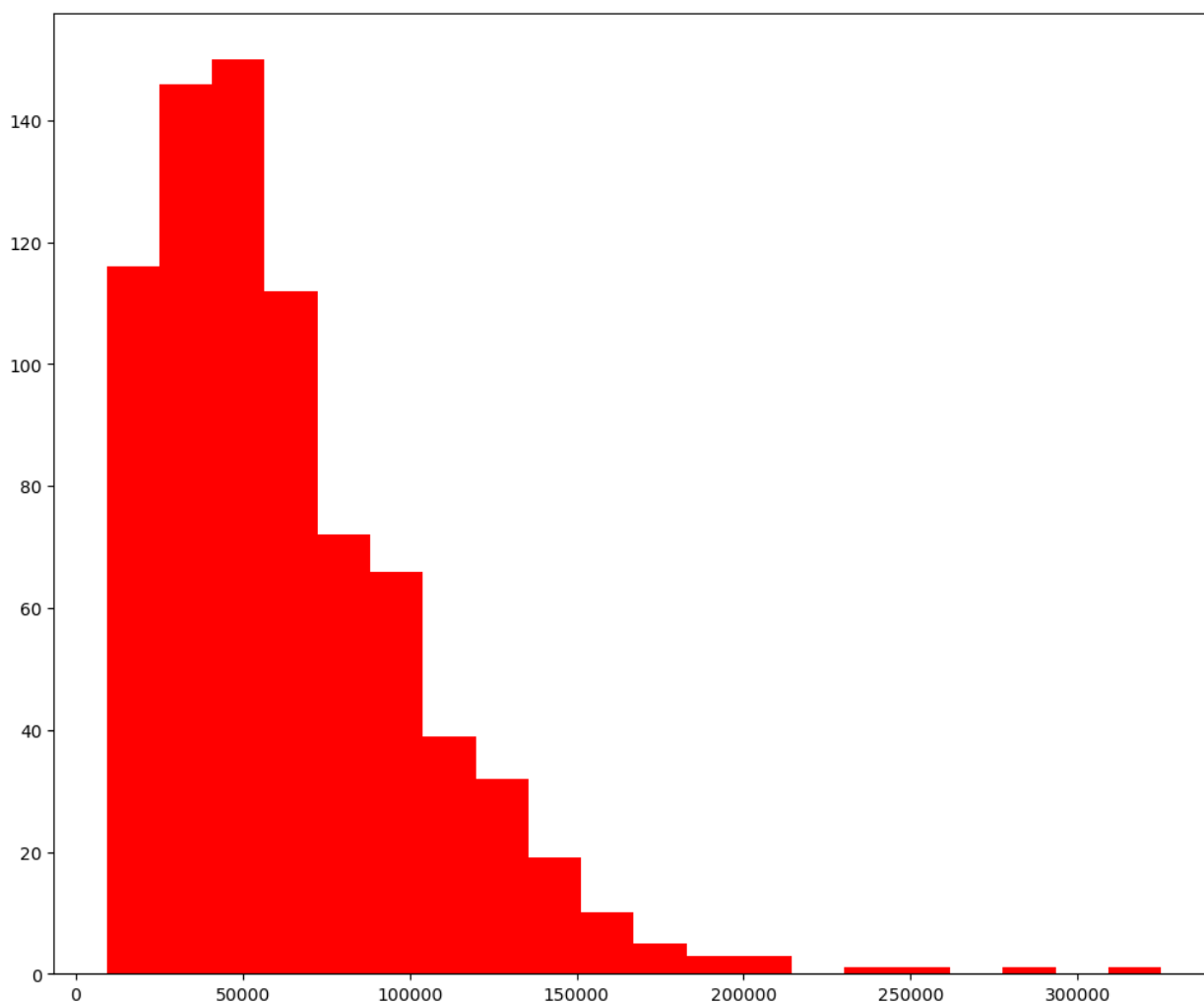
g

	price	count
0	9270.720	1
1	10442.880	1
2	10602.720	2
3	10810.512	1
4	11135.520	4
...	...	...
772	211788.000	1
773	233845.920	1
774	261018.720	1
775	292986.720	1
776	324954.720	1

```
[777 rows x 2 columns]
```

```
#The price is numerical data so we do the histogram
```

```
plt.figure(figsize=(12,10))  
plt.hist(g['price'],bins=20,color='Red')  
plt.show()
```



```
newdf=filter[['Price']]  
newdf
```

```
      Price  
0    71378.6832  
1    47895.5232  
2    30636.0000  
3   135195.3360  
4    96095.8080  
...         ...
```



```
1298    33992.6400
1299    79866.7200
1300    12201.1200
1301    40705.9200
1302    19660.3200
```

```
[1273 rows x 1 columns]
```

```
newdf['Price'].max()
```

```
324954.72
```

```
newdf['Price'].min()
```

```
9270.72
```

```
newdf['Price'].max()
```

```
324954.72
```

```
newdf['Price'].mean()
```

```
59955.81407321288
```

```
newdf['Price'].median()
```

```
52161.12
```

```
np.percentile(newdf,25)
```

```
31914.72
```

```
np.percentile(newdf,50)
```

```
52161.12
```

```
np.percentile(newdf,75)
```

```
79333.3872
```

```
# BIVARIATE ANALYSIS
```

```
filter
```

	Company	TypeName	Inches	\
0	Apple	Ultrabook	13.3	
1	Apple	Ultrabook	13.3	
2	HP	Notebook	15.6	
3	Apple	Ultrabook	15.4	
4	Apple	Ultrabook	13.3	
...	...	...	...	
1298	Lenovo	2 in 1 Convertible	14	

1299	Lenovo	2 in 1 Convertible	13.3
1300	Lenovo	Notebook	65.4
1301	HP	Notebook	15.6
1302	Asus	Notebook	15.6

		ScreenResolution	\
0	IPS Panel Retina Display	2560x1600	
1		1440x900	
2	Full HD	1920x1080	
3	IPS Panel Retina Display	2880x1800	
4	IPS Panel Retina Display	2560x1600	
...			
1298	IPS Panel Full HD / Touchscreen	1920x1080	
1299	IPS Panel Quad HD+ / Touchscreen	3200x1800	
1300		1366x768	
1301		1366x768	
1302		1366x768	

		Cpu	Ram		Memory	\
0		Intel Core i5	2.3GHz	8	128GB SSD	
1		Intel Core i5	1.8GHz	8	128GB Flash Storage	
2	Intel	Core i5 7200U	2.5GHz	8	256GB SSD	
3		Intel Core i7	2.7GHz	16	512GB SSD	
4		Intel Core i5	3.1GHz	8	256GB SSD	
...						
1298		Intel Core i7 6500U	2.5GHz	4	128GB SSD	
1299		Intel Core i7 6500U	2.5GHz	16	512GB SSD	
1300	Intel	Celeron Dual Core N3050	1.6GHz	2	64GB Flash Storage	
1301		Intel Core i7 6500U	2.5GHz	6	1TB HDD	
1302	Intel	Celeron Dual Core N3050	1.6GHz	4	500GB HDD	

		Gpu	OpSys	Weight	Price
0	Intel	Iris Plus Graphics 640	macOS	1.37	71378.6832
1		Intel HD Graphics 6000	macOS	1.34	47895.5232
2		Intel HD Graphics 620	No OS	1.86	30636.0000
3		AMD Radeon Pro 455	macOS	1.83	135195.3360
4	Intel	Iris Plus Graphics 650	macOS	1.37	96095.8080
...					
1298		Intel HD Graphics 520	Windows 10	1.8	33992.6400
1299		Intel HD Graphics 520	Windows 10	1.3	79866.7200
1300		Intel HD Graphics	Windows 10	1.5	12201.1200
1301		AMD Radeon R5 M330	Windows 10	2.19	40705.9200
1302		Intel HD Graphics	Windows 10	2.2	19660.3200

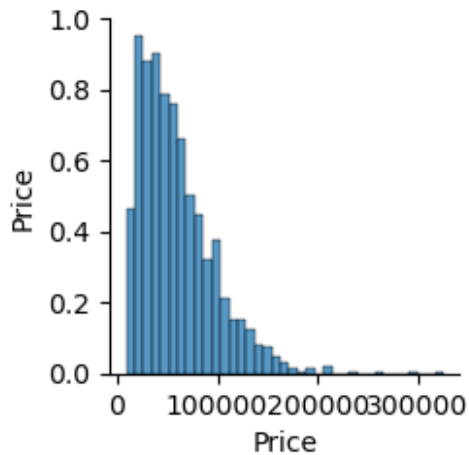
[1273 rows x 11 columns]

filter.shape

(1273, 11)

```
sns.pairplot(filter)
```

```
<seaborn.axisgrid.PairGrid at 0x161367aa090>
```



```
filter.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
Index: 1273 entries, 0 to 1302
```

```
Data columns (total 11 columns):
```

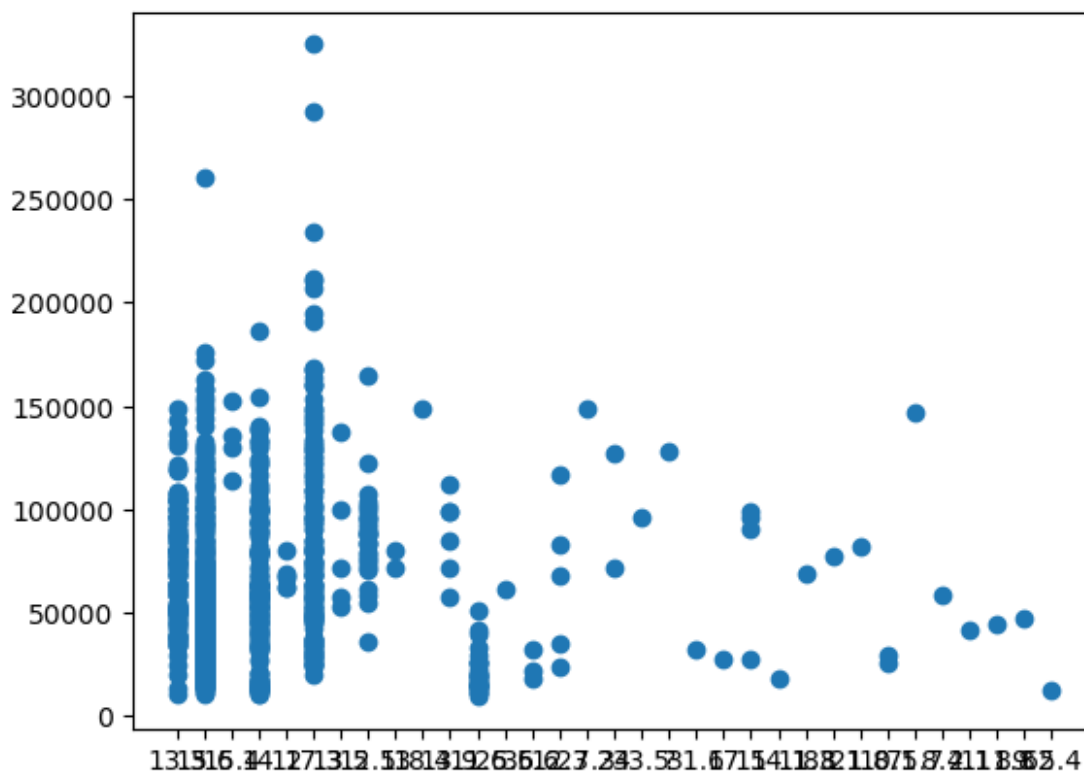
#	Column	Non-Null Count	Dtype
0	Company	1273 non-null	object
1	TypeName	1273 non-null	object
2	Inches	1273 non-null	object
3	ScreenResolution	1273 non-null	object
4	Cpu	1273 non-null	object
5	Ram	1273 non-null	object
6	Memory	1273 non-null	object
7	Gpu	1273 non-null	object
8	OpSys	1273 non-null	object
9	Weight	1273 non-null	object
10	Price	1273 non-null	float64

```
dtypes: float64(1), object(10)
```

```
memory usage: 119.3+ KB
```

```
plt.scatter(filter['Inches'], filter['Price'])
```

```
<matplotlib.collections.PathCollection at 0x161366c5390>
```



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

	Company	TypeName	Inches	\
0	Apple	Ultrabook	13.3	
1	Apple	Ultrabook	13.3	
2	HP	Notebook	15.6	
3	Apple	Ultrabook	15.4	
4	Apple	Ultrabook	13.3	
...	...	...	...	...
1298	Lenovo	2 in 1 Convertible	14	
1299	Lenovo	2 in 1 Convertible	13.3	
1300	Lenovo	Notebook	65.4	
1301	HP	Notebook	15.6	
1302	Asus	Notebook	15.6	

	ScreenResolution	\
0	IPS Panel Retina Display 2560x1600	
1	1440x900	
2	Full HD 1920x1080	
3	IPS Panel Retina Display 2880x1800	
4	IPS Panel Retina Display 2560x1600	
...	...	...
1298	IPS Panel Full HD / Touchscreen 1920x1080	
1299	IPS Panel Quad HD+ / Touchscreen 3200x1800	

1300	1366x768
1301	1366x768
1302	1366x768

		Cpu	Ram		Memory	\
0		Intel Core i5 2.3GHz	8		128GB SSD	
1		Intel Core i5 1.8GHz	8	128GB Flash Storage		
2	Intel	Core i5 7200U 2.5GHz	8		256GB SSD	
3		Intel Core i7 2.7GHz	16		512GB SSD	
4		Intel Core i5 3.1GHz	8		256GB SSD	
...			...	..		...
1298		Intel Core i7 6500U 2.5GHz	4		128GB SSD	
1299		Intel Core i7 6500U 2.5GHz	16		512GB SSD	
1300	Intel	Celeron Dual Core N3050 1.6GHz	2	64GB Flash Storage		
1301		Intel Core i7 6500U 2.5GHz	6		1TB HDD	
1302	Intel	Celeron Dual Core N3050 1.6GHz	4		500GB HDD	

		Gpu	OpSys	Weight	Price
0	Intel	Iris Plus Graphics 640	macOS	1.37	71378.6832
1		Intel HD Graphics 6000	macOS	1.34	47895.5232
2		Intel HD Graphics 620	No OS	1.86	30636.0000
3		AMD Radeon Pro 455	macOS	1.83	135195.3360
4	Intel	Iris Plus Graphics 650	macOS	1.37	96095.8080
...			...	...	...
1298		Intel HD Graphics 520	Windows 10	1.8	33992.6400
1299		Intel HD Graphics 520	Windows 10	1.3	79866.7200
1300		Intel HD Graphics	Windows 10	1.5	12201.1200
1301		AMD Radeon R5 M330	Windows 10	2.19	40705.9200
1302		Intel HD Graphics	Windows 10	2.2	19660.3200

[1273 rows x 11 columns]

*#To perform chisquare based on the price*

x=filter[['Ram']]

x

	Ram
0	8
1	8
2	8
3	16
4	8
...	..
1298	4
1299	16
1300	2
1301	6
1302	4

```
[1273 rows x 1 columns]
```

```
y=df[['Company']]
```

```
y
```

```
      Company
0      Apple
1      Apple
2        HP
3      Apple
4      Apple
...      ...
1298  Lenovo
1299  Lenovo
1300  Lenovo
1301      HP
1302     Asus
```

```
[1273 rows x 1 columns]
```

```
from sklearn.feature_selection import chi2
values=chi2(x,y)
values
```

```
(array([482.41557886]), array([5.16396297e-91]))
```

```
10)Conclusion
```

```
#Based on the Ram and Cpu the price will be increased
```