

# laptop-dataset-analysis-2

July 23, 2024

## 1 Milestone-1 Evaluation Project

### 1.1 Project Documentation: Exploratory Data Analysis of Laptop Dataset :

#### 1.1.1 Title : Laptop Data Analysis

#### 1.1.2 Name : Gaurav Kumar Singh

#### 1.1.3 DA/DS : Data Analytics (DA)

#### 1.1.4 Batch number : B4 (June - Online)(M) - DA & DS

#### 1.1.5 Online/Offline : Online

#### 1.1.6 Roll Number : 10624OLR018

### 1.2 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. Segmented Univariate Analysis
11. Bivariate Analysis
12. Multivariate Analysis
13. Overall Insights Obtained from Analysis
14. Conclusion

**NOTE : All the codes used for this are given after the documentation and displaying of results.**

### 1.3 1) Introduction :

*The laptop dataset comprises various attributes related to laptops, including manufacturer, specifications, and pricing information. The goal of this project is to conduct a comprehensive analysis of the dataset to derive insights into laptop characteristics, performance, and pricing, catering to both consumers and manufacturers in the computer industry..*

## Columns in the dataset related to Laptop:

1. Company: The manufacturer or brand of the laptop.
2. TypeName: The type or category of the laptop (e.g., gaming, ultrabook).
3. Inches: The size of the laptop screen in inches.
4. ScreenResolution: The resolution and display technology of the laptop screen.
5. Cpu: The processor (CPU) model of the laptop.
6. Ram: The amount of Random Access Memory (RAM) in the laptop.
7. Memory: The storage capacity (hard drive or SSD) of the laptop.
8. Gpu: The graphics processor (GPU) model of the laptop.
9. OpSys: The operating system installed on the laptop.
10. Weight: The weight of the laptop in kilograms.
11. Price: The price of the laptop in the local currency. \*\*

### 1.4 2) Aim :

*The aim of this project is to conduct a comprehensive analysis of the dataset to derive insights into laptop characteristics, performance, and pricing, catering to both consumers and manufacturers in the computer industry. \**

### 1.5 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.

### 1.6 Specifically, the problem is:

- 1) How do various laptop characteristics, such as screen size, RAM, CPU model, and brand reputation, impact pricing and consumer purchasing decisions?
- 2) 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?
- 3) How can manufacturers and retailers use data-driven insights to inform product development, marketing strategies, and pricing decisions to stay competitive in the market?

### 1.7 4) Project Workflow :

*Overview of the project workflow or methodology followed.* - Data Cleaning - Exploratory Data Analysis (EDA) - Data Visualization - Analysis and Interpretation - Documentation

### 1.8 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

```
[1]: #importing data and the required packages
```

```
import pandas as pd
import numpy as np
%matplotlib inline
import seaborn as sns
import matplotlib.pyplot as plt
```

```
[2]: data = pd.read_csv("C:/Users/DELL/Downloads/laptopData.csv")
```

```
[3]: data
```

```
[3]:      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]

```
[4]: data.head() #inspecting the First 5 Rows of the dataframe
```

```
[4]: Unnamed: 0 Company   TypeName Inches   ScreenResolution \
0      0.0   Apple  Ultrabook   13.3  IPS Panel Retina Display 2560x1600
1      1.0   Apple  Ultrabook   13.3                        1440x900
2      2.0     HP   Notebook   15.6                        Full HD 1920x1080
3      3.0   Apple  Ultrabook   15.4  IPS Panel Retina Display 2880x1800
4      4.0   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	

```
[5]: data.tail() # inspecting the last 5 Rows of the dataframe
```

```
[5]: Unnamed: 0 Company   TypeName Inches \
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	\
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

```
[6]: data.sample(5) #inspecting the Random 5 Rows of the dataframe
```

```
[6]: Unnamed: 0 Company TypeName Inches ScreenResolution \
739 739.0 Acer Notebook 15.6 1366x768
567 NaN NaN NaN NaN NaN
237 237.0 Asus Notebook 15.6 Full HD 1920x1080
1169 1169.0 MSI Gaming 15.6 IPS Panel Full HD 1920x1080
1163 1163.0 Lenovo Notebook 15.6 1366x768
```

			Cpu	Ram	Memory	\
739		Intel Core i3 6006U	2GHz	4GB	500GB HDD	
567			NaN	NaN	NaN	
237	AMD A10-Series A10-9620P	2.5GHz	4GB	128GB SSD +	1TB HDD	
1169	Intel Core i7 7500U	2.7GHz	8GB		1TB HDD	
1163	Intel Core i3 6006U	2.0GHz	8GB		1TB HDD	

		Gpu	OpSys	Weight	Price
739	Intel HD Graphics 520	Windows	10	2.4kg	21887.424
567		NaN	NaN	NaN	NaN
237	AMD Radeon R5 M430	Windows	10	2.2kg	30636.000
1169	Nvidia GeForce GTX 1050	Windows	10	2.2kg	44701.920
1163	Intel HD Graphics 520	Windows	10	2.2kg	24455.520

```
[7]: #Checking the shape of the Data :
print("Number of Rows:",data.shape[0])
print("Number of Columns:",data.shape[1])
```

```
Number of Rows: 1303
Number of Columns: 12
```

```
[8]: data.describe()
```

```
[8]:
```

	Unnamed: 0	Price
count	1273.000000	1273.000000
mean	652.674784	59955.814073
std	376.493027	37332.251005
min	0.000000	9270.720000
25%	327.000000	31914.720000
50%	652.000000	52161.120000
75%	980.000000	79333.387200
max	1302.000000	324954.720000

```
[9]: data.info() #checking if there is any missing value
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1303 entries, 0 to 1302
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Unnamed: 0            1273 non-null   float64
1   Company               1273 non-null   object
2   TypeName              1253 non-null   object
3   Inches               1220 non-null   object
4   ScreenResolution      1273 non-null   object
5   Cpu                  1273 non-null   object
6   Ram                  1259 non-null   object
7   Memory              1273 non-null   object
8   Gpu                  1273 non-null   object
9   OpSys                1273 non-null   object
10  Weight               1259 non-null   object
11  Price                1273 non-null   float64
dtypes: float64(2), object(10)
memory usage: 122.3+ KB
```

```
[10]: data.columns
```

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

```
[11]: data.drop(columns=['Unnamed: 0'], inplace=True)
data
```

```
[11]:      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

[1303 rows x 11 columns]

```
[12]: #Checking the shape of the Data :
print("Number of Rows:",data.shape[0])
print("Number of Columns:",data.shape[1])
```

Number of Rows: 1303  
Number of Columns: 11

[ ]:

[ ]:

## 1.9 Data Checks to Perform before Data Cleaning (Data Exploration)

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

```
[13]: # Check for Duplicate Values :
duplicates = data.duplicated().sum()
print("Number of Duplicate Values:", duplicates)
```

Number of Duplicate Values: 57

```
[14]: data_cleaned = data.drop_duplicates()
data_cleaned
```

```
[14]:
```

	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	



...	...	...	...
1270	Lenovo	2 in 1 Convertible	14
1271	Lenovo	2 in 1 Convertible	13.3
1272	Lenovo	Notebook	14
1273	HP	Notebook	15.6
1300	Lenovo	Notebook	65.4

		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

...	...
1270	IPS Panel Full HD / Touchscreen 1920x1080
1271	IPS Panel Quad HD+ / Touchscreen 3200x1800
1272	1366x768
1273	1366x768
1300	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
...	...	...	...	...
1270	Intel Core i7 6500U	2.5GHz	4GB	128GB SSD
1271	Intel Core i7 6500U	2.5GHz	16GB	512GB SSD
1272	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB Flash Storage
1273	Intel Core i7 6500U	2.5GHz	6GB	1TB HDD
1300	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB Flash Storage

		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
...	...	...	...	...	...
1270	Intel HD Graphics	520	Windows 10	1.8kg	33992.6400
1271	Intel HD Graphics	520	Windows 10	1.3kg	79866.7200
1272	Intel HD Graphics		Windows 10	1.5kg	12201.1200
1273	AMD Radeon R5	M330	Windows 10	2.19kg	40705.9200
1300	Intel HD Graphics		Windows 10	1.5kg	12201.1200

[1246 rows x 11 columns]

```
[15]: # Displaying the first few rows of the cleaned dataset
data_cleaned.head()
```

```
[15]: 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
```

```
[16]: data_cleaned
```

```
[16]: 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
...   ...      ...      ...
1270  Lenovo  2 in 1 Convertible   14
1271  Lenovo  2 in 1 Convertible  13.3
1272  Lenovo      Notebook   14
1273    HP        Notebook   15.6
1300  Lenovo      Notebook  65.4

      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
...   ...
1270  IPS Panel Full HD / Touchscreen 1920x1080
```

```

1271 IPS Panel Quad HD+ / Touchscreen 3200x1800
1272                               1366x768
1273                               1366x768
1300                               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
...
1270                Intel Core i7 6500U 2.5GHz   4GB           128GB SSD
1271                Intel Core i7 6500U 2.5GHz   16GB           512GB SSD
1272 Intel Celeron Dual Core N3050 1.6GHz   2GB   64GB Flash Storage
1273                Intel Core i7 6500U 2.5GHz   6GB           1TB HDD
1300 Intel Celeron Dual Core N3050 1.6GHz   2GB   64GB Flash Storage

```

```

                                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
...
1270          Intel HD Graphics 520  Windows 10  1.8kg  33992.6400
1271          Intel HD Graphics 520  Windows 10  1.3kg  79866.7200
1272          Intel HD Graphics      Windows 10  1.5kg  12201.1200
1273          AMD Radeon R5 M330  Windows 10  2.19kg  40705.9200
1300          Intel HD Graphics      Windows 10  1.5kg  12201.1200

```

[1246 rows x 11 columns]

```
[17]: data_cleaned.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Index: 1246 entries, 0 to 1300
Data columns (total 11 columns):
#   Column              Non-Null Count  Dtype
---  -
0   Company             1245 non-null   object
1   TypeName             1225 non-null   object
2   Inches              1192 non-null   object
3   ScreenResolution     1245 non-null   object
4   Cpu                 1245 non-null   object
5   Ram                 1231 non-null   object
6   Memory              1245 non-null   object

```

```

7   Gpu                1245 non-null   object
8   OpSys              1245 non-null   object
9   Weight             1231 non-null   object
10  Price              1245 non-null   float64
dtypes: float64(1), object(10)
memory usage: 116.8+ KB

```

```

[18]: # Checking the number of Unique Values in Every Column
print("Number of Unique Value in Every Columns :")
data_cleaned.nunique()

```

Number of Unique Value in Every Columns :

```

[18]: Company                19
      TypeName                6
      Inches                 33
      ScreenResolution       40
      Cpu                   118
      Ram                   10
      Memory                40
      Gpu                   110
      OpSys                  9
      Weight               196
      Price                777
      dtype: int64

```

```

[19]: data_cleaned.describe(include='all')

```

```

[19]:
      Company  TypeName  Inches  ScreenResolution  \
count      1245      1225   1192              1245
unique       19         6     33                40
top    Lenovo  Notebook   15.6  Full HD 1920x1080
freq       283       673    596                493
mean        NaN        NaN    NaN              NaN
std         NaN        NaN    NaN              NaN
min         NaN        NaN    NaN              NaN
25%         NaN        NaN    NaN              NaN
50%         NaN        NaN    NaN              NaN
75%         NaN        NaN    NaN              NaN
max         NaN        NaN    NaN              NaN

      Cpu  Ram  Memory  Gpu  \
count   1245 1231   1245 1245
unique   118  10    40   110
top Intel Core i5 7200U 2.5GHz 8GB 256GB SSD Intel HD Graphics 620
freq     183  588   401   269
mean     NaN  NaN   NaN   NaN
std      NaN  NaN   NaN   NaN

```

min	NaN	NaN	NaN	NaN
25%	NaN	NaN	NaN	NaN
50%	NaN	NaN	NaN	NaN
75%	NaN	NaN	NaN	NaN
max	NaN	NaN	NaN	NaN

	OpSys	Weight	Price
count	1245	1231	1245.000000
unique	9	196	NaN
top	Windows	10	2.2kg
freq	1023	104	NaN
mean	NaN	NaN	60567.344825
std	NaN	NaN	37434.736222
min	NaN	NaN	9270.720000
25%	NaN	NaN	32639.860800
50%	NaN	NaN	52693.920000
75%	NaN	NaN	79813.440000
max	NaN	NaN	324954.720000

**Insights gained from initial data exploration** - There are 1246 rows and 11 columns in the Dataset. - From the info we conclude that pricing column have Numerical Values, laptop characteristics, performance all are Categorical Values. - Farm Metadata has both the Values - Categoriactal and Numerical Values.

## 1.10 ) Data Cleaning

### 1.10.1 Handle missing values:

```
[20]: data_cleaned.isnull()
```

```
[20]:
```

	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	
...	...	...	...	...	...	...	...	
1270	False	False	False	False	False	False	False	
1271	False	False	False	False	False	False	False	
1272	False	False	False	False	False	False	False	
1273	False	False	False	False	False	False	False	
1300	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

```

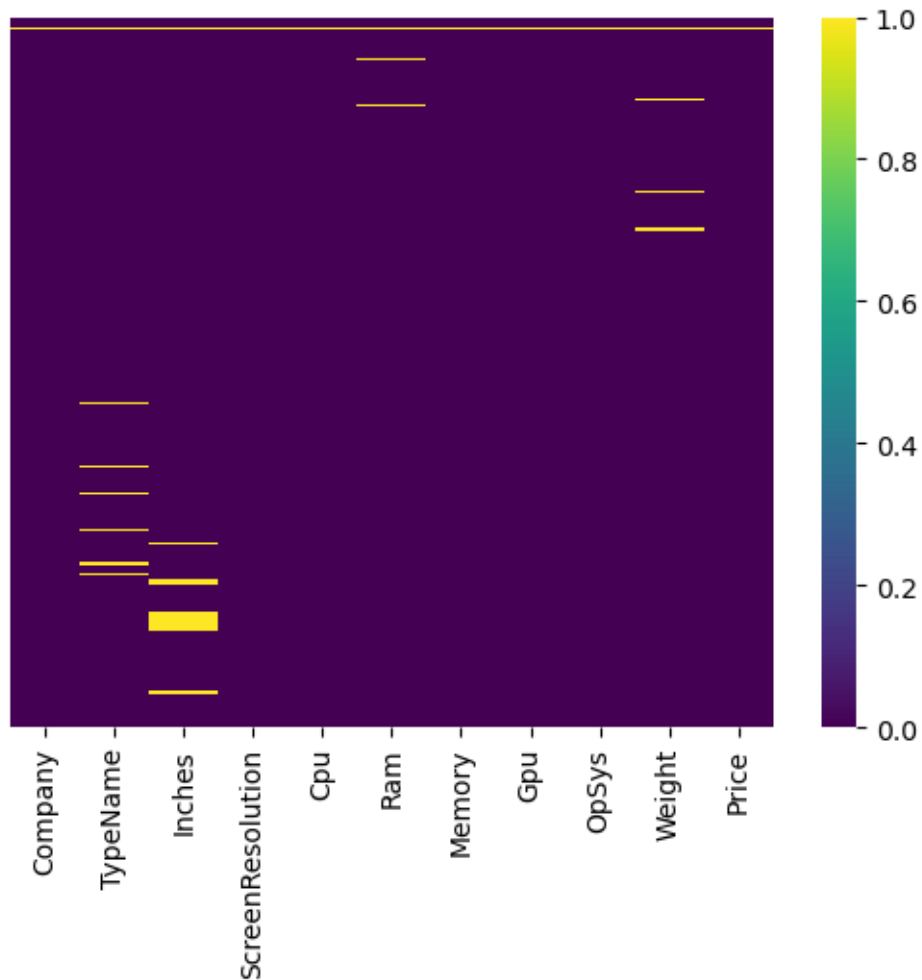
4      False  False   False  False
...    ...    ...    ...    ...
1270   False  False   False  False
1271   False  False   False  False
1272   False  False   False  False
1273   False  False   False  False
1300   False  False   False  False

```

[1246 rows x 11 columns]

```
[21]: sns.heatmap(data_cleaned.isnull(),yticklabels=False,cmap = 'viridis') #_
      ↳Checking the null values in the heatmap
```

[21]: <Axes: >



```
[22]: # Checking for missing values
data_cleaned.isna().sum()
```

```
[22]: Company          1
      TypeName        21
      Inches          54
      ScreenResolution 1
      Cpu              1
      Ram             15
      Memory           1
      Gpu              1
      OpSys            1
      Weight          15
      Price            1
      dtype: int64
```

#there is no column that need to be dropped

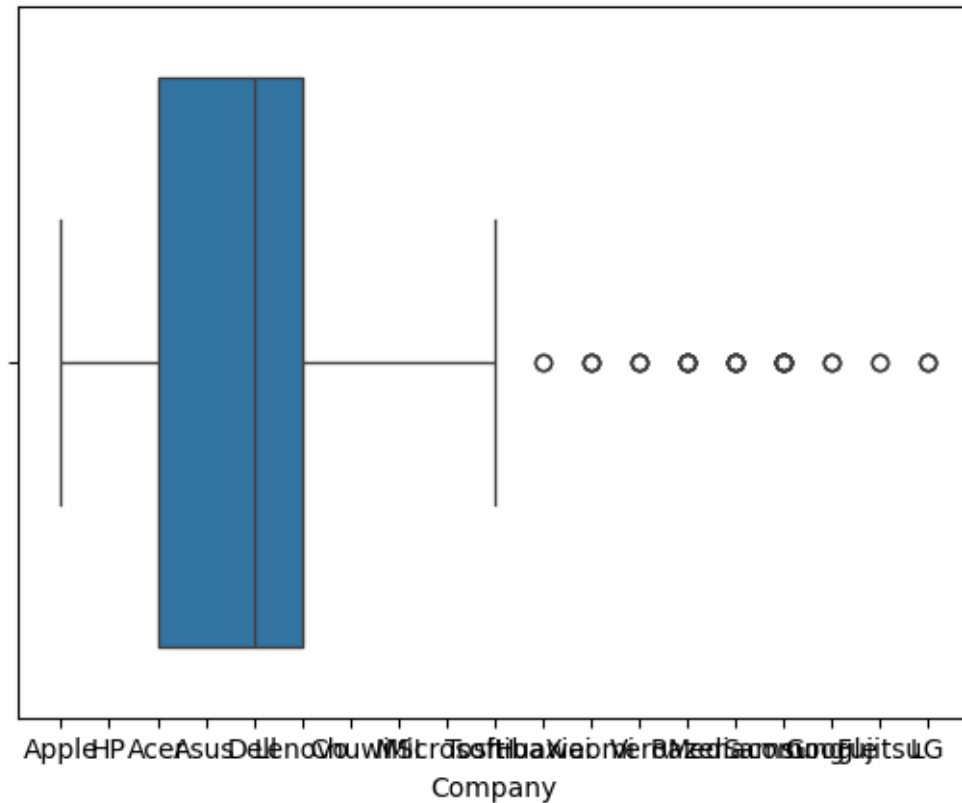
```
[23]: # Separating Numerical Columns and Non-Numerical Columns for Missing Value
      ↳ Imputation/Replacing :
Numerical_Data = data_cleaned.select_dtypes(include=['float64', 'int64']).
      ↳ columns
Non_Numerical_Data = data_cleaned.select_dtypes(include=['object']).columns
```

```
[24]: print(" No. of Numerical Columns =",len(Numerical_Data))
      print(Numerical_Data)
      print(" No. of Non-Numerical Columns =",len(Non_Numerical_Data))
      print(Non_Numerical_Data)
      print(' '*100)
```

```
      No. of Numerical Columns = 1
      Index(['Price'], dtype='object')
      No. of Non-Numerical Columns = 10
      Index(['Company', 'TypeName', 'Inches', 'ScreenResolution', 'Cpu', 'Ram',
            'Memory', 'Gpu', 'OpSys', 'Weight'],
            dtype='object')
```

```
[25]: sns.boxplot(x=data_cleaned['Company'])
```

```
[25]: <Axes: xlabel='Company'>
```



```
[26]: mode_value=data_cleaned['Company'].mode()
      mode_value
```

```
[26]: 0    Lenovo
      Name: Company, dtype: object
```

```
[27]: data_cleaned["Company"].fillna(mode_value[0], inplace=True)
      data_cleaned
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_3900\3320075380.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
data_cleaned["Company"].fillna(mode_value[0], inplace=True)
```

```
[27]:   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
...	...	...	...
1270	Lenovo	2 in 1 Convertible	14
1271	Lenovo	2 in 1 Convertible	13.3
1272	Lenovo	Notebook	14
1273	HP	Notebook	15.6
1300	Lenovo	Notebook	65.4

		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	
...		...	
1270	IPS Panel Full HD / Touchscreen	1920x1080	
1271	IPS Panel Quad HD+ / Touchscreen	3200x1800	
1272		1366x768	
1273		1366x768	
1300		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	
...		...	...	...	
1270	Intel Core i7 6500U	2.5GHz	4GB	128GB SSD	
1271	Intel Core i7 6500U	2.5GHz	16GB	512GB SSD	
1272	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB Flash Storage	
1273	Intel Core i7 6500U	2.5GHz	6GB	1TB HDD	
1300	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB Flash Storage	

		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
...		...	...	...	...
1270	Intel HD Graphics	520	Windows 10	1.8kg	33992.6400
1271	Intel HD Graphics	520	Windows 10	1.3kg	79866.7200
1272	Intel HD Graphics		Windows 10	1.5kg	12201.1200
1273	AMD Radeon R5 M330		Windows 10	2.19kg	40705.9200
1300	Intel HD Graphics		Windows 10	1.5kg	12201.1200

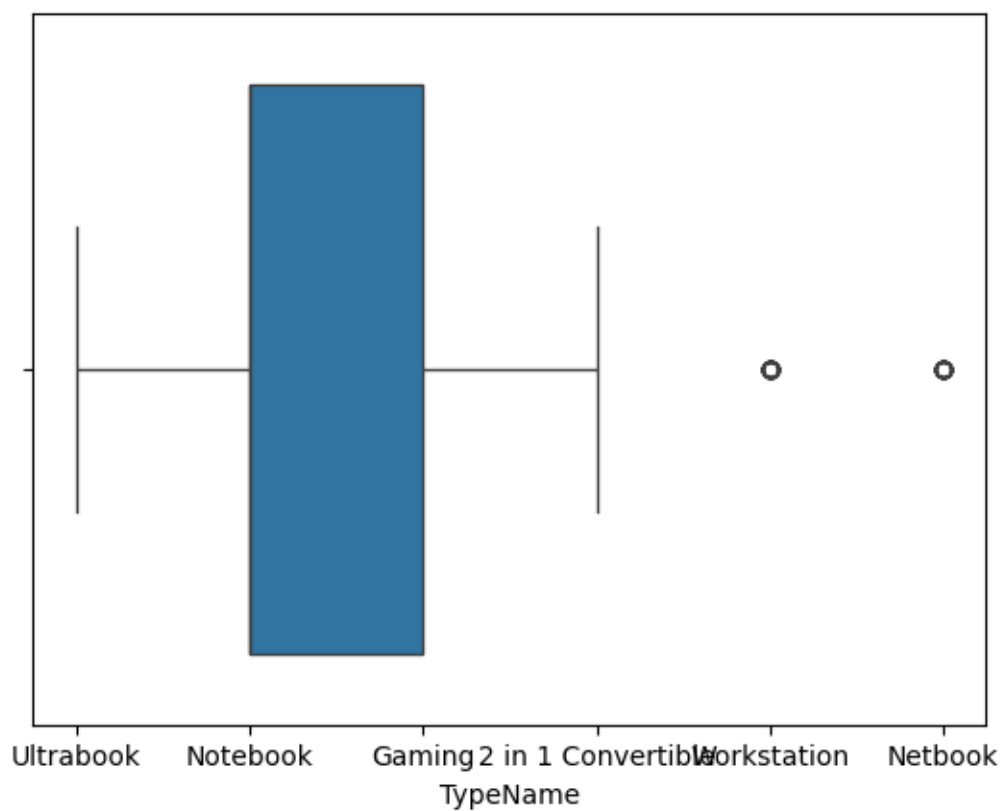
```
[1246 rows x 11 columns]
```

```
[28]: data_cleaned.isnull().sum()
```

```
[28]: Company          0
      TypeName        21
      Inches         54
      ScreenResolution  1
      Cpu             1
      Ram            15
      Memory          1
      Gpu             1
      OpSys           1
      Weight         15
      Price           1
      dtype: int64
```

```
[29]: sns.boxplot(x=data_cleaned['TypeName'])
```

```
[29]: <Axes: xlabel='TypeName'>
```



```
[30]: mode_v=data_cleaned['TypeName'].mode()
mode_v
```

```
[30]: 0    Notebook
      Name: TypeName, dtype: object
```

```
[31]: data_cleaned["TypeName"].fillna(mode_v[0], inplace=True)
data_cleaned
```

C:\Users\DELL\AppData\Local\Temp\ipykernel\_3900\2821992606.py:1:

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
data_cleaned["TypeName"].fillna(mode_v[0], inplace=True)
```

```
[31]:      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
...
1270  Lenovo  2 in 1 Convertible   14
1271  Lenovo  2 in 1 Convertible  13.3
1272  Lenovo          Notebook   14
1273    HP          Notebook   15.6
1300  Lenovo          Notebook  65.4
```

```
      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
...
1270  IPS Panel Full HD / Touchscreen 1920x1080
1271  IPS Panel Quad HD+ / Touchscreen 3200x1800
1272                                1366x768
1273                                1366x768
1300                                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
...		...	...		...
1270		Intel Core i7 6500U 2.5GHz	4GB		128GB SSD
1271		Intel Core i7 6500U 2.5GHz	16GB		512GB SSD
1272	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB Flash Storage	
1273		Intel Core i7 6500U 2.5GHz	6GB		1TB HDD
1300	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB Flash Storage	

		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
...		...	...	...	...
1270	Intel HD Graphics 520	Windows 10		1.8kg	33992.6400
1271	Intel HD Graphics 520	Windows 10		1.3kg	79866.7200
1272	Intel HD Graphics	Windows 10		1.5kg	12201.1200
1273	AMD Radeon R5 M330	Windows 10		2.19kg	40705.9200
1300	Intel HD Graphics	Windows 10		1.5kg	12201.1200

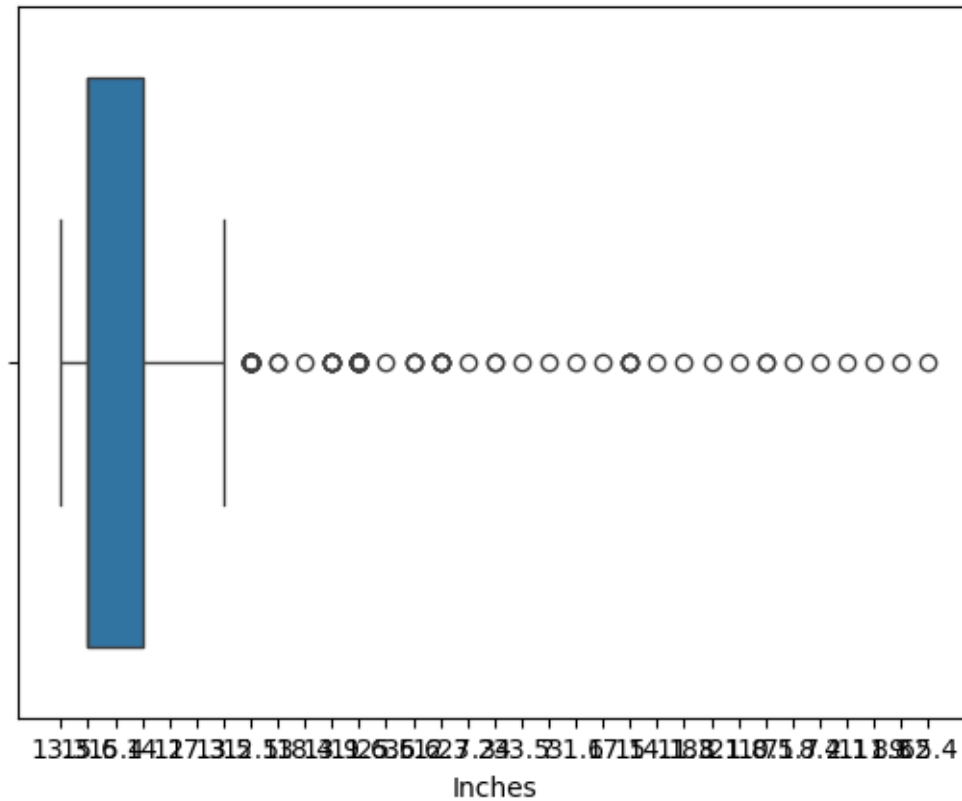
[1246 rows x 11 columns]

```
[32]: data_cleaned.isnull().sum()
```

```
[32]: Company          0
      TypeName         0
      Inches          54
      ScreenResolution  1
      Cpu              1
      Ram             15
      Memory           1
      Gpu              1
      OpSys            1
      Weight          15
      Price            1
      dtype: int64
```

```
[33]: sns.boxplot(x=data_cleaned['Inches'])
```

```
[33]: <Axes: xlabel='Inches'>
```



```
[34]: m3=data_cleaned['Inches'].mode()
      m3
```

```
[34]: 0    15.6
      Name: Inches, dtype: object
```

```
[35]: data_cleaned["Inches"].fillna(m3[0], inplace=True)
      data_cleaned
```

C:\Users\DELL\AppData\Local\Temp\ipykernel\_3900\3738339996.py:1:  
SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)  
data\_cleaned["Inches"].fillna(m3[0], inplace=True)

```
[35]:
```

	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
...	...	...	...
1270	Lenovo	2 in 1 Convertible	14
1271	Lenovo	2 in 1 Convertible	13.3
1272	Lenovo	Notebook	14
1273	HP	Notebook	15.6
1300	Lenovo	Notebook	65.4

		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	
...		...	
1270	IPS Panel Full HD / Touchscreen	1920x1080	
1271	IPS Panel Quad HD+ / Touchscreen	3200x1800	
1272		1366x768	
1273		1366x768	
1300		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		
...		...	...	...	
1270	Intel Core i7 6500U 2.5GHz	4GB	128GB SSD		
1271	Intel Core i7 6500U 2.5GHz	16GB	512GB SSD		
1272	Intel Celeron Dual Core N3050 1.6GHz	2GB	64GB Flash Storage		
1273	Intel Core i7 6500U 2.5GHz	6GB	1TB HDD		
1300	Intel Celeron Dual Core N3050 1.6GHz	2GB	64GB Flash Storage		

		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	
...		...	...	...	...
1270	Intel HD Graphics 520	Windows 10	1.8kg	33992.6400	
1271	Intel HD Graphics 520	Windows 10	1.3kg	79866.7200	
1272	Intel HD Graphics	Windows 10	1.5kg	12201.1200	
1273	AMD Radeon R5 M330	Windows 10	2.19kg	40705.9200	
1300	Intel HD Graphics	Windows 10	1.5kg	12201.1200	

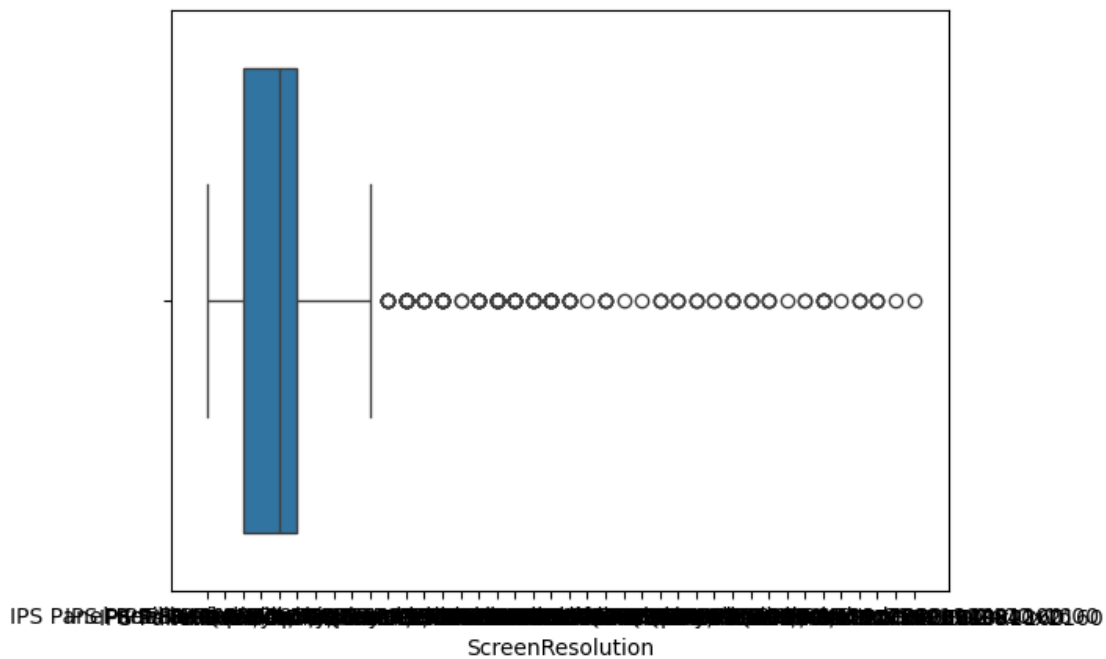
```
[1246 rows x 11 columns]
```

```
[36]: data_cleaned.isnull().sum()
```

```
[36]: Company          0
      TypeName         0
      Inches          0
      ScreenResolution  1
      Cpu             1
      Ram            15
      Memory          1
      Gpu             1
      OpSys           1
      Weight         15
      Price           1
      dtype: int64
```

```
[37]: sns.boxplot(x=data_cleaned['ScreenResolution'])
```

```
[37]: <Axes: xlabel='ScreenResolution'>
```



```
[38]: m4=data_cleaned['ScreenResolution'].mode()
      m4
```

```
[38]: 0    Full HD 1920x1080
      Name: ScreenResolution, dtype: object
```

```
[39]: data_cleaned["ScreenResolution"].fillna(m4[0], inplace=True)
      data_cleaned
```

C:\Users\DELL\AppData\Local\Temp\ipykernel\_3900\3215139936.py:1:

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
data_cleaned["ScreenResolution"].fillna(m4[0], inplace=True)
```

```
[39]:      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
...      ...      ...      ...
1270  Lenovo  2 in 1 Convertible   14
1271  Lenovo  2 in 1 Convertible  13.3
1272  Lenovo      Notebook      14
1273      HP      Notebook   15.6
1300  Lenovo      Notebook  65.4
```

```
      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
...      ...
1270  IPS Panel Full HD / Touchscreen 1920x1080
1271  IPS Panel Quad HD+ / Touchscreen 3200x1800
1272      1366x768
1273      1366x768
1300      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
...      ...      ...      ...
```



1270	Intel Core i7 6500U	2.5GHz	4GB	128GB SSD
1271	Intel Core i7 6500U	2.5GHz	16GB	512GB SSD
1272	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB Flash Storage
1273	Intel Core i7 6500U	2.5GHz	6GB	1TB HDD
1300	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB Flash Storage

		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
...	...	...	...	...	...
1270	Intel HD Graphics	520	Windows 10	1.8kg	33992.6400
1271	Intel HD Graphics	520	Windows 10	1.3kg	79866.7200
1272	Intel HD Graphics		Windows 10	1.5kg	12201.1200
1273	AMD Radeon R5 M330		Windows 10	2.19kg	40705.9200
1300	Intel HD Graphics		Windows 10	1.5kg	12201.1200

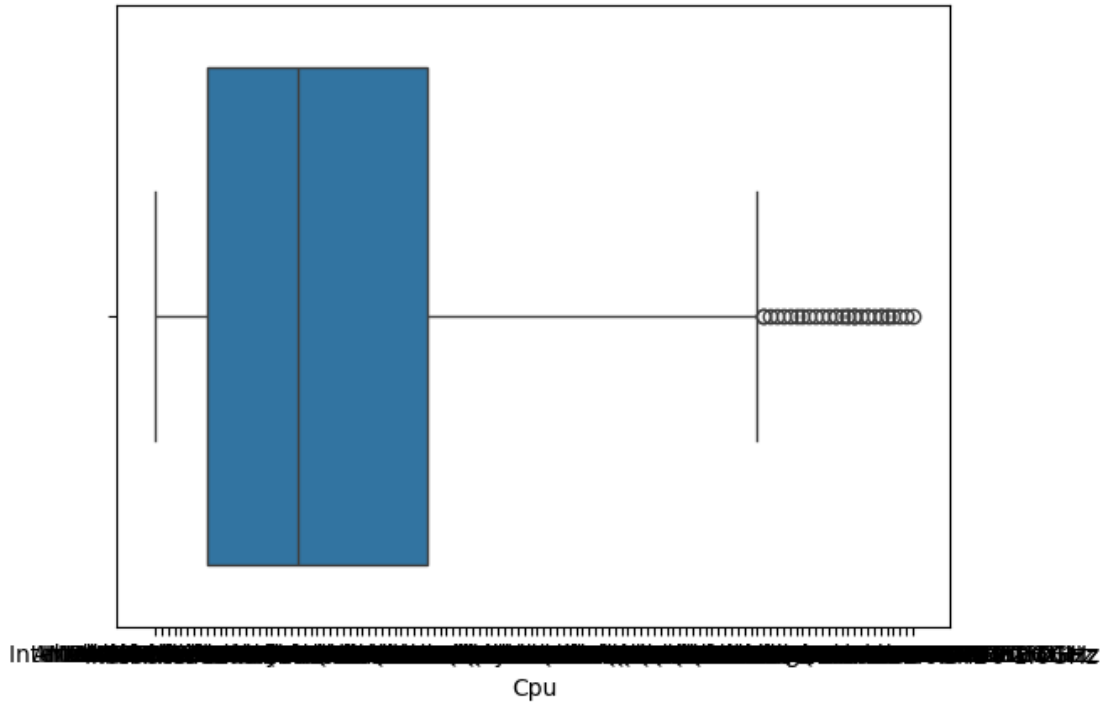
[1246 rows x 11 columns]

```
[40]: data_cleaned.isnull().sum()
```

```
[40]: Company          0
      TypeName         0
      Inches           0
      ScreenResolution  0
      Cpu              1
      Ram             15
      Memory           1
      Gpu              1
      OpSys            1
      Weight          15
      Price            1
      dtype: int64
```

```
[41]: sns.boxplot(x=data_cleaned['Cpu'])
```

```
[41]: <Axes: xlabel='Cpu'>
```



```
[42]: m5=data_cleaned['Cpu'].mode()
      m5
```

```
[42]: 0    Intel Core i5 7200U 2.5GHz
      Name: Cpu, dtype: object
```

```
[43]: data_cleaned["Cpu"].fillna(m5[0], inplace=True)
      data_cleaned
```

C:\Users\DELL\AppData\Local\Temp\ipykernel\_3900\1563642922.py:1:

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
data_cleaned["Cpu"].fillna(m5[0], inplace=True)
```

```
[43]:
```

	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	
...	...	...	...	

1270	Lenovo	2 in 1 Convertible	14
1271	Lenovo	2 in 1 Convertible	13.3
1272	Lenovo	Notebook	14
1273	HP	Notebook	15.6
1300	Lenovo	Notebook	65.4

		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	
...		...	
1270	IPS Panel Full HD / Touchscreen	1920x1080	
1271	IPS Panel Quad HD+ / Touchscreen	3200x1800	
1272		1366x768	
1273		1366x768	
1300		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	
...		...	...	...	
1270	Intel Core i7 6500U	2.5GHz	4GB	128GB SSD	
1271	Intel Core i7 6500U	2.5GHz	16GB	512GB SSD	
1272	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB Flash Storage	
1273	Intel Core i7 6500U	2.5GHz	6GB	1TB HDD	
1300	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB Flash Storage	

		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
...		...	...	...	...
1270	Intel HD Graphics	520	Windows 10	1.8kg	33992.6400
1271	Intel HD Graphics	520	Windows 10	1.3kg	79866.7200
1272	Intel HD Graphics		Windows 10	1.5kg	12201.1200
1273	AMD Radeon R5 M330		Windows 10	2.19kg	40705.9200
1300	Intel HD Graphics		Windows 10	1.5kg	12201.1200

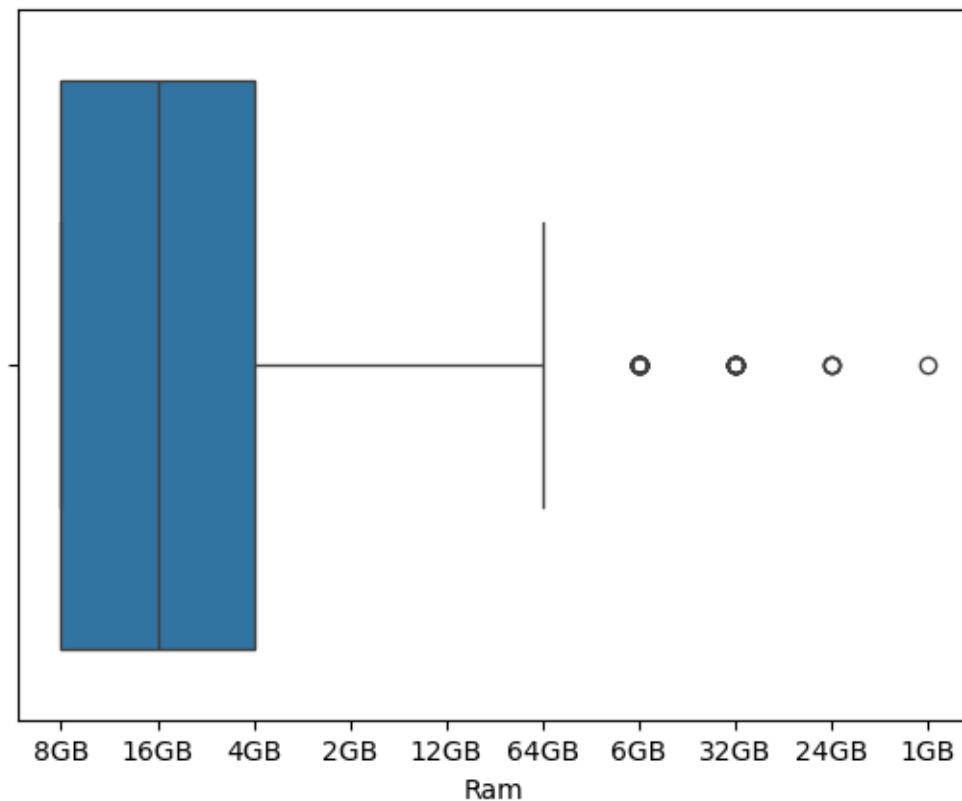
[1246 rows x 11 columns]

```
[44]: data_cleaned.isnull().sum()
```

```
[44]: Company          0
      TypeName        0
      Inches          0
      ScreenResolution 0
      Cpu             0
      Ram            15
      Memory          1
      Gpu             1
      OpSys           1
      Weight         15
      Price           1
      dtype: int64
```

```
[45]: sns.boxplot(x=data_cleaned['Ram'])
```

```
[45]: <Axes: xlabel='Ram'>
```



```
[46]: a=data_cleaned['Ram'].unique()
      a
```

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

```
[47]: len(a)
```

```
[47]: 11
```

```
[48]: m6=data_cleaned['Ram'].mode()
m6
```

```
[48]: 0      8GB
      Name: Ram, dtype: object
```

```
[49]: data_cleaned["Ram"].fillna(m6[0], inplace=True)
data_cleaned
```

C:\Users\DELL\AppData\Local\Temp\ipykernel\_3900\4102070387.py:1:

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
data_cleaned["Ram"].fillna(m6[0], inplace=True)
```

```
[49]:
```

	Company	Type	Name	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	
...	...	...	...	...	...
1270	Lenovo	2 in 1	Convertible	14	
1271	Lenovo	2 in 1	Convertible	13.3	
1272	Lenovo	Notebook		14	
1273	HP	Notebook		15.6	
1300	Lenovo	Notebook		65.4	

			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	
...	...	...	...	...
1270	IPS Panel	Full HD / Touchscreen	1920x1080	
1271	IPS Panel	Quad HD+ / Touchscreen	3200x1800	
1272			1366x768	

```

1273                                1366x768
1300                                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
...		...	...	...
1270		Intel Core i7 6500U 2.5GHz	4GB	128GB SSD
1271		Intel Core i7 6500U 2.5GHz	16GB	512GB SSD
1272	Intel	Celeron Dual Core N3050 1.6GHz	2GB	64GB Flash Storage
1273		Intel Core i7 6500U 2.5GHz	6GB	1TB HDD
1300	Intel	Celeron Dual Core N3050 1.6GHz	2GB	64GB Flash Storage

		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
...		...	...	...	...
1270		Intel HD Graphics 520	Windows 10	1.8kg	33992.6400
1271		Intel HD Graphics 520	Windows 10	1.3kg	79866.7200
1272		Intel HD Graphics	Windows 10	1.5kg	12201.1200
1273		AMD Radeon R5 M330	Windows 10	2.19kg	40705.9200
1300		Intel HD Graphics	Windows 10	1.5kg	12201.1200

[1246 rows x 11 columns]

```
[50]: data_cleaned.isnull().sum()
```

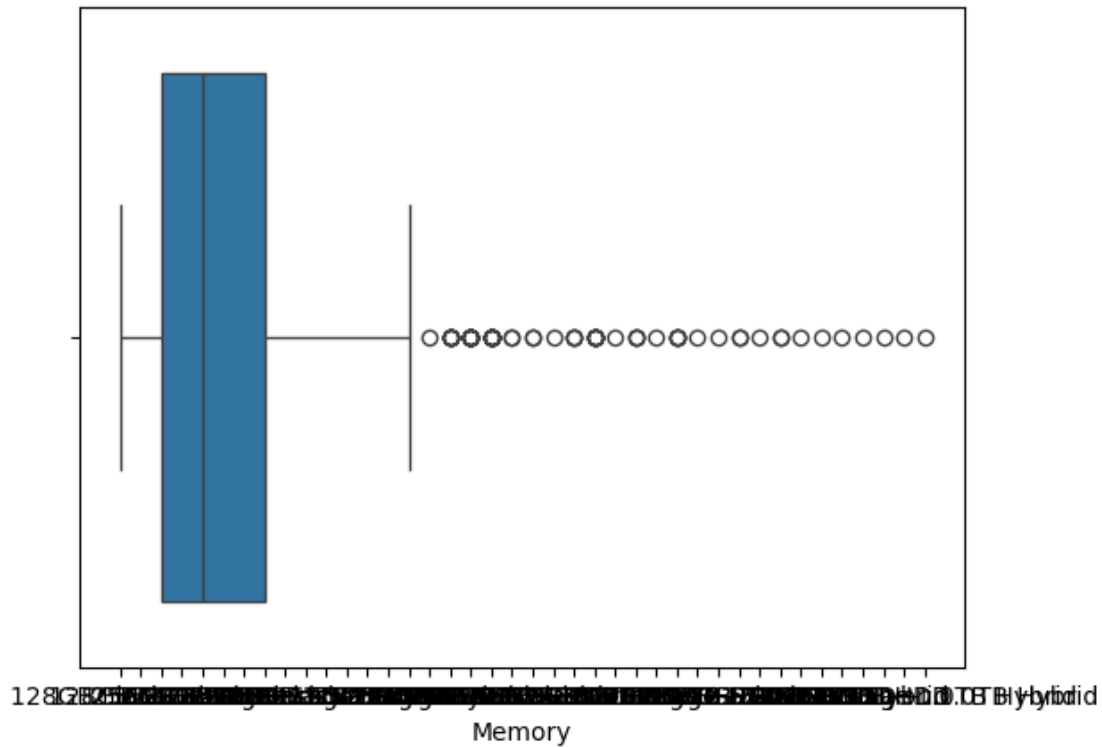
```

[50]: Company          0
      TypeName         0
      Inches           0
      ScreenResolution  0
      Cpu              0
      Ram              0
      Memory           1
      Gpu              1
      OpSys            1
      Weight           15
      Price            1
      dtype: int64

```

```
[51]: sns.boxplot(x=data_cleaned['Memory'])
```

```
[51]: <Axes: xlabel='Memory'>
```



```
[52]: m7=data_cleaned['Memory'].mode()
m7
```

```
[52]: 0    256GB SSD
      Name: Memory, dtype: object
```

```
[53]: data_cleaned["Memory"].fillna(m7[0], inplace=True)
data_cleaned
```

C:\Users\DELL\AppData\Local\Temp\ipykernel\_3900\3533751847.py:1:

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
data_cleaned["Memory"].fillna(m7[0], inplace=True)
```

```
[53]:
```

	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
...	...	...	...
1270	Lenovo	2 in 1 Convertible	14
1271	Lenovo	2 in 1 Convertible	13.3
1272	Lenovo	Notebook	14
1273	HP	Notebook	15.6
1300	Lenovo	Notebook	65.4

		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	
...		...	
1270	IPS Panel Full HD / Touchscreen	1920x1080	
1271	IPS Panel Quad HD+ / Touchscreen	3200x1800	
1272		1366x768	
1273		1366x768	
1300		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	
...		...	...	...	
1270	Intel Core i7 6500U	2.5GHz	4GB	128GB SSD	
1271	Intel Core i7 6500U	2.5GHz	16GB	512GB SSD	
1272	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB Flash Storage	
1273	Intel Core i7 6500U	2.5GHz	6GB	1TB HDD	
1300	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB Flash Storage	

		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
...		...	...	...	...
1270	Intel HD Graphics	520	Windows 10	1.8kg	33992.6400
1271	Intel HD Graphics	520	Windows 10	1.3kg	79866.7200
1272	Intel HD Graphics		Windows 10	1.5kg	12201.1200
1273	AMD Radeon R5 M330		Windows 10	2.19kg	40705.9200
1300	Intel HD Graphics		Windows 10	1.5kg	12201.1200



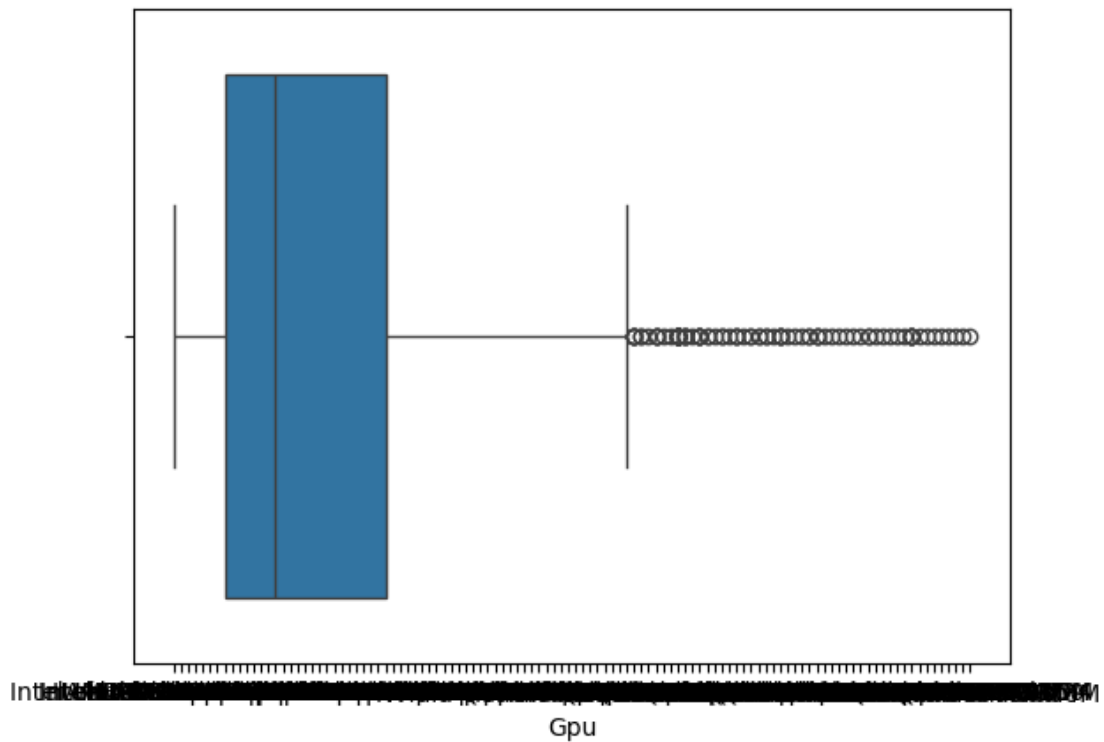
[1246 rows x 11 columns]

```
[54]: data_cleaned.isnull().sum()
```

```
[54]: Company          0
      TypeName         0
      Inches          0
      ScreenResolution 0
      Cpu             0
      Ram            0
      Memory          0
      Gpu             1
      OpSys           1
      Weight         15
      Price           1
      dtype: int64
```

```
[55]: sns.boxplot(x=data_cleaned['Gpu'])
```

```
[55]: <Axes: xlabel='Gpu'>
```



```
[56]: m8=data_cleaned['Gpu'].mode()
      m8
```

```
[56]: 0      Intel HD Graphics 620
      Name: Gpu, dtype: object
```

```
[57]: data_cleaned["Gpu"].fillna(m8[0], inplace=True)
      data_cleaned
```

C:\Users\DELL\AppData\Local\Temp\ipykernel\_3900\2568756894.py:1:

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
data_cleaned["Gpu"].fillna(m8[0], inplace=True)
```

```
[57]:      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
...
1270  Lenovo  2 in 1 Convertible   14
1271  Lenovo  2 in 1 Convertible   13.3
1272  Lenovo      Notebook      14
1273      HP      Notebook   15.6
1300  Lenovo      Notebook   65.4
```

```
      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
...
1270  IPS Panel Full HD / Touchscreen 1920x1080
1271  IPS Panel Quad HD+ / Touchscreen 3200x1800
1272      1366x768
1273      1366x768
1300      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
...		...	...		...
1270		Intel Core i7 6500U 2.5GHz	4GB		128GB SSD
1271		Intel Core i7 6500U 2.5GHz	16GB		512GB SSD
1272	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB Flash Storage	
1273		Intel Core i7 6500U 2.5GHz	6GB		1TB HDD
1300	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB Flash Storage	

		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
...	...	...	...	...	...
1270	Intel HD Graphics 520	Windows 10		1.8kg	33992.6400
1271	Intel HD Graphics 520	Windows 10		1.3kg	79866.7200
1272	Intel HD Graphics	Windows 10		1.5kg	12201.1200
1273	AMD Radeon R5 M330	Windows 10		2.19kg	40705.9200
1300	Intel HD Graphics	Windows 10		1.5kg	12201.1200

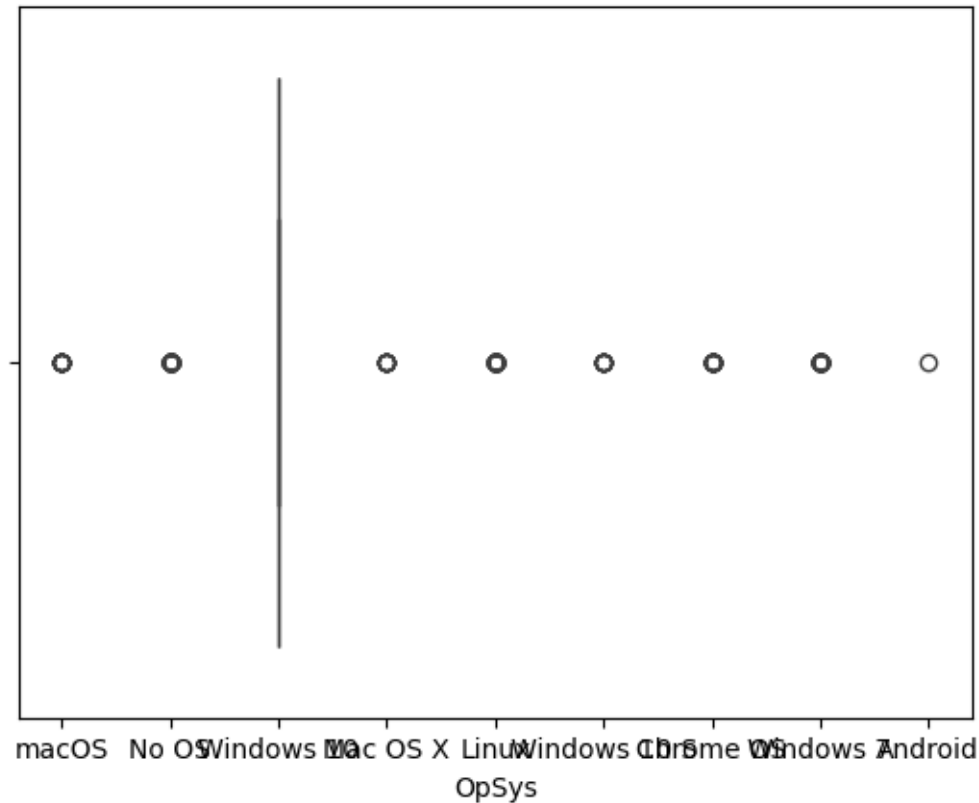
[1246 rows x 11 columns]

```
[58]: data_cleaned.isnull().sum()
```

```
[58]: Company          0
      TypeName        0
      Inches          0
      ScreenResolution 0
      Cpu             0
      Ram             0
      Memory          0
      Gpu             0
      OpSys           1
      Weight          15
      Price           1
      dtype: int64
```

```
[59]: sns.boxplot(x=data_cleaned['OpSys'])
```

```
[59]: <Axes: xlabel='OpSys'>
```



```
[60]: m9=data_cleaned['OpSys'].mode()
      m9
```

```
[60]: 0    Windows 10
      Name: OpSys, dtype: object
```

```
[61]: data_cleaned["OpSys"].fillna(m9[0], inplace=True)
      data_cleaned
```

C:\Users\DELL\AppData\Local\Temp\ipykernel\_3900\3098558997.py:1:

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
data_cleaned["OpSys"].fillna(m9[0], inplace=True)
```

```
[61]:
```

	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
...	...	...	...
1270	Lenovo	2 in 1 Convertible	14
1271	Lenovo	2 in 1 Convertible	13.3
1272	Lenovo	Notebook	14
1273	HP	Notebook	15.6
1300	Lenovo	Notebook	65.4

		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	
...		...	
1270	IPS Panel Full HD / Touchscreen	1920x1080	
1271	IPS Panel Quad HD+ / Touchscreen	3200x1800	
1272		1366x768	
1273		1366x768	
1300		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	
...		...	...	...	
1270	Intel Core i7 6500U	2.5GHz	4GB	128GB SSD	
1271	Intel Core i7 6500U	2.5GHz	16GB	512GB SSD	
1272	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB Flash Storage	
1273	Intel Core i7 6500U	2.5GHz	6GB	1TB HDD	
1300	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB Flash Storage	

		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
...		...	...	...	...
1270	Intel HD Graphics	520	Windows 10	1.8kg	33992.6400
1271	Intel HD Graphics	520	Windows 10	1.3kg	79866.7200
1272	Intel HD Graphics		Windows 10	1.5kg	12201.1200
1273	AMD Radeon R5 M330		Windows 10	2.19kg	40705.9200
1300	Intel HD Graphics		Windows 10	1.5kg	12201.1200

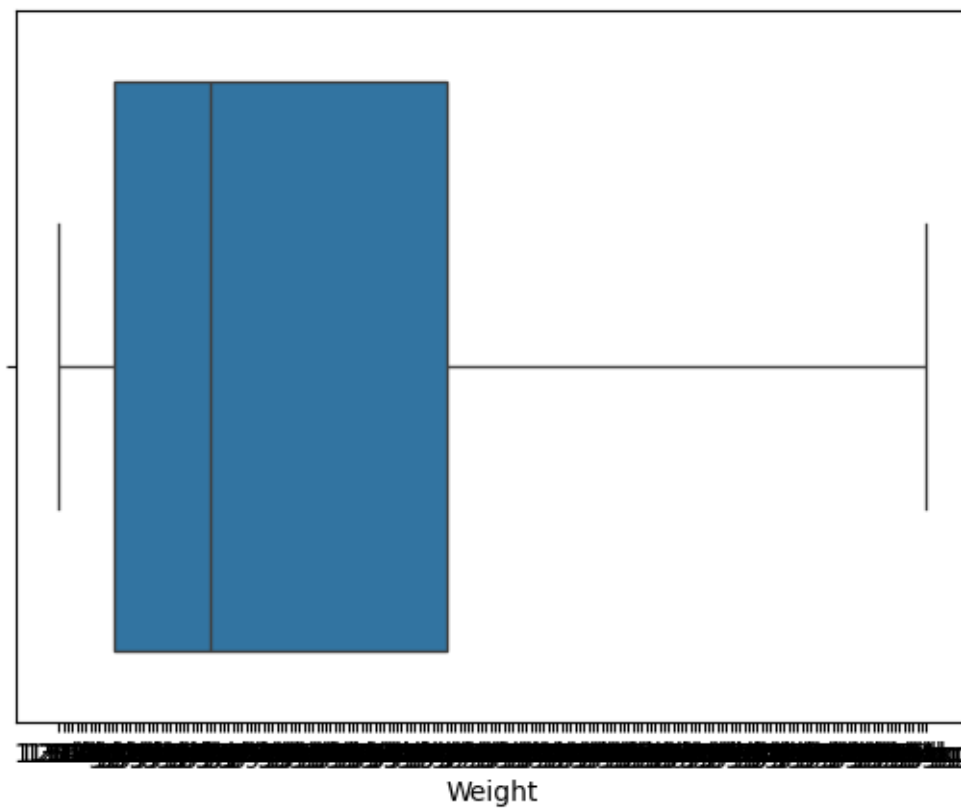
```
[1246 rows x 11 columns]
```

```
[62]: data_cleaned.isnull().sum()
```

```
[62]: Company          0
      TypeName         0
      Inches           0
      ScreenResolution  0
      Cpu              0
      Ram              0
      Memory           0
      Gpu              0
      OpSys            0
      Weight          15
      Price            1
      dtype: int64
```

```
[63]: sns.boxplot(x=data_cleaned['Weight'])
```

```
[63]: <Axes: xlabel='Weight'>
```



```
[64]: m10=data_cleaned['Weight'].mode()
m10
```

```
[64]: 0    2.2kg
      Name: Weight, dtype: object
```

```
[65]: data_cleaned["Weight"].fillna(m10[0], inplace=True)
data_cleaned
```

C:\Users\DELL\AppData\Local\Temp\ipykernel\_3900\108484281.py:1:

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
data_cleaned["Weight"].fillna(m10[0], inplace=True)
```

```
[65]:      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
...
1270  Lenovo  2 in 1 Convertible   14
1271  Lenovo  2 in 1 Convertible  13.3
1272  Lenovo          Notebook   14
1273      HP          Notebook  15.6
1300  Lenovo          Notebook  65.4
```

```
      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
...
1270  IPS Panel Full HD / Touchscreen 1920x1080
1271  IPS Panel Quad HD+ / Touchscreen 3200x1800
1272                                1366x768
1273                                1366x768
1300                                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
...		...	...		...
1270		Intel Core i7 6500U 2.5GHz	4GB		128GB SSD
1271		Intel Core i7 6500U 2.5GHz	16GB		512GB SSD
1272	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB Flash Storage	
1273	Intel Core i7 6500U	2.5GHz	6GB		1TB HDD
1300	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB Flash Storage	

		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
...		...	...	...	...
1270	Intel HD Graphics 520	Windows 10		1.8kg	33992.6400
1271	Intel HD Graphics 520	Windows 10		1.3kg	79866.7200
1272	Intel HD Graphics	Windows 10		1.5kg	12201.1200
1273	AMD Radeon R5 M330	Windows 10		2.19kg	40705.9200
1300	Intel HD Graphics	Windows 10		1.5kg	12201.1200

[1246 rows x 11 columns]

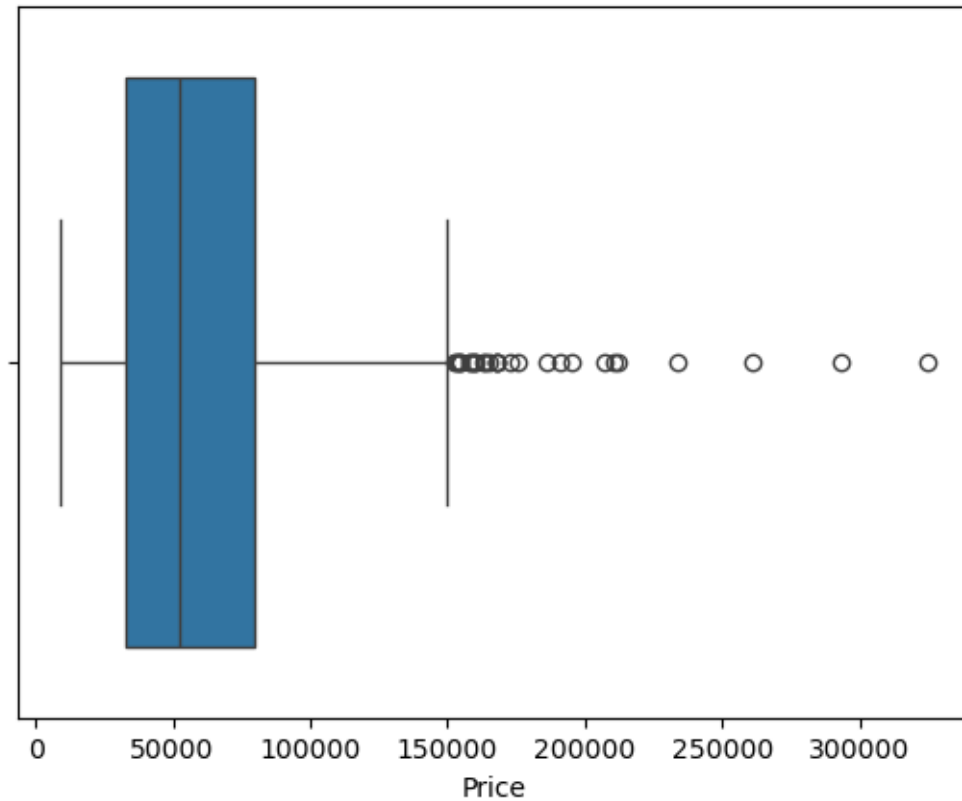
```
[66]: data_cleaned.isnull().sum()
```

```
[66]: Company      0
      TypeName    0
      Inches      0
      ScreenResolution  0
      Cpu         0
      Ram         0
      Memory      0
      Gpu         0
      OpSys       0
      Weight      0
      Price       1
      dtype: int64
```

```
[67]: #Boxplot for numerical column
      sns.boxplot(x=data_cleaned['Price'])
```

```
[67]: <Axes: xlabel='Price'>
```





```
[68]: med_value=data_cleaned['Price'].median()
      med_value
```

```
[68]: 52693.92
```

```
[69]: data_cleaned["Price"].fillna(med_value, inplace=True)
      data_cleaned
```

C:\Users\DELL\AppData\Local\Temp\ipykernel\_3900\2483598788.py:1:

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
data_cleaned["Price"].fillna(med_value, inplace=True)
```

```
[69]:
```

	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
...	...	...	...
1270	Lenovo	2 in 1 Convertible	14
1271	Lenovo	2 in 1 Convertible	13.3
1272	Lenovo	Notebook	14
1273	HP	Notebook	15.6
1300	Lenovo	Notebook	65.4

		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	
...		...	
1270	IPS Panel Full HD / Touchscreen	1920x1080	
1271	IPS Panel Quad HD+ / Touchscreen	3200x1800	
1272		1366x768	
1273		1366x768	
1300		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	
...		...	...	...	
1270	Intel Core i7 6500U	2.5GHz	4GB	128GB SSD	
1271	Intel Core i7 6500U	2.5GHz	16GB	512GB SSD	
1272	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB Flash Storage	
1273	Intel Core i7 6500U	2.5GHz	6GB	1TB HDD	
1300	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB Flash Storage	

		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
...		...	...	...	...
1270	Intel HD Graphics	520	Windows 10	1.8kg	33992.6400
1271	Intel HD Graphics	520	Windows 10	1.3kg	79866.7200
1272	Intel HD Graphics		Windows 10	1.5kg	12201.1200
1273	AMD Radeon R5	M330	Windows 10	2.19kg	40705.9200
1300	Intel HD Graphics		Windows 10	1.5kg	12201.1200

[1246 rows x 11 columns]

```
[70]: data_cleaned.isnull().sum()
```

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

#Outlier Detection

```
[71]: df1 = data_cleaned.select_dtypes(exclude=['object'])
      df1
```

```
[71]:          Price
0      71378.6832
1      47895.5232
2      30636.0000
3     135195.3360
4      96095.8080
...
1270    33992.6400
1271    79866.7200
1272    12201.1200
1273    40705.9200
1300    12201.1200
```

[1246 rows x 1 columns]

```
[72]: q1=df1.quantile(0.25)
      q3=df1.quantile(0.75)
      q1
```

```
[72]: Price      32645.0556
      Name: 0.25, dtype: float64
```

```
[73]: q3
```

```
[73]: Price      79773.48
      Name: 0.75, dtype: float64
```

```
[74]: Iqr=q3-q1
      Iqr
```

```
[74]: Price      47128.4244
      dtype: float64
```

```
[75]: b=((df1<q1-1.5*Iqr)|(df1>q3+1.5*Iqr)) #detecting the outlier through this and
      ↪then printing it True in row
      b
```

```
[75]:      Price
0      False
1      False
2      False
3      False
4      False
...      ...
1270 False
1271 False
1272 False
1273 False
1300 False

[1246 rows x 1 columns]
```

```
[76]: b.sum()
```

```
[76]: Price      28
      dtype: int64
```

```
[77]: #removing the outlier from original data
      filter=data_cleaned[~b.any(axis=1)]
      filter
```

```
[77]:      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
...      ...      ...      ...
1270 Lenovo  2 in 1 Convertible   14
1271 Lenovo  2 in 1 Convertible  13.3
1272 Lenovo          Notebook   14
```

1273	HP	Notebook	15.6
1300	Lenovo	Notebook	65.4

		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	
...		...	
1270	IPS Panel Full HD / Touchscreen	1920x1080	
1271	IPS Panel Quad HD+ / Touchscreen	3200x1800	
1272		1366x768	
1273		1366x768	
1300		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	
...		...	...		...	
1270	Intel Core i7 6500U	2.5GHz	4GB		128GB SSD	
1271	Intel Core i7 6500U	2.5GHz	16GB		512GB SSD	
1272	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB	Flash Storage	
1273	Intel Core i7 6500U	2.5GHz	6GB		1TB HDD	
1300	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB	Flash Storage	

		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
...		...	...	...	...
1270	Intel HD Graphics 520	Windows 10		1.8kg	33992.6400
1271	Intel HD Graphics 520	Windows 10		1.3kg	79866.7200
1272	Intel HD Graphics	Windows 10		1.5kg	12201.1200
1273	AMD Radeon R5 M330	Windows 10		2.19kg	40705.9200
1300	Intel HD Graphics	Windows 10		1.5kg	12201.1200

[1218 rows x 11 columns]

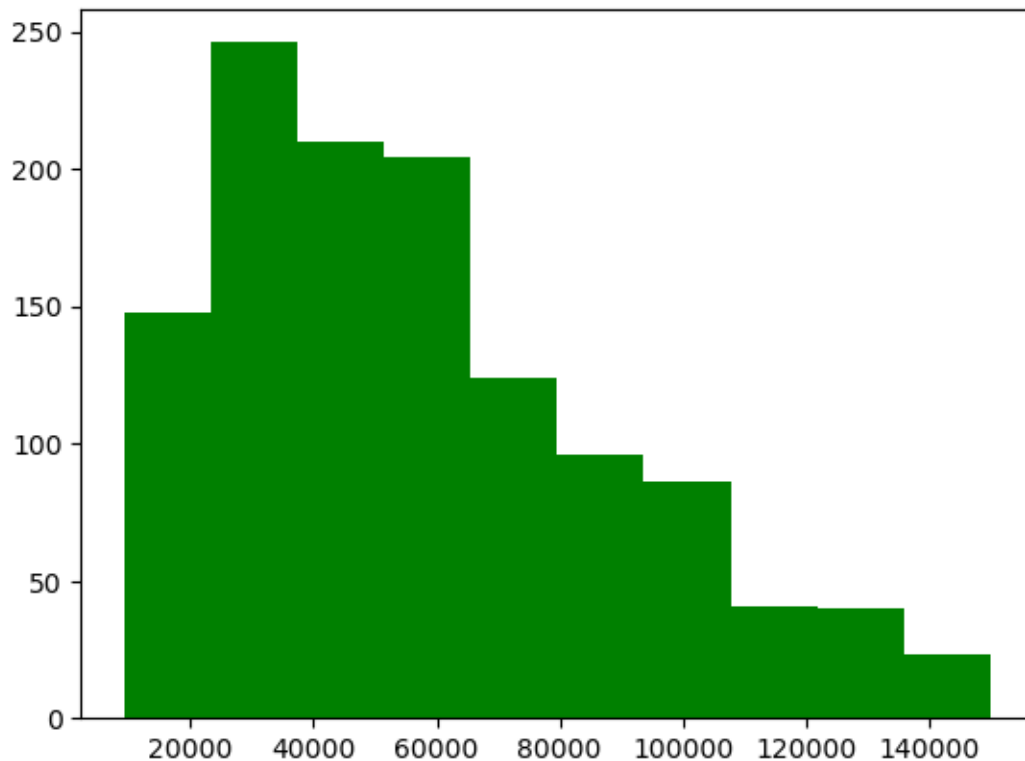
```
[78]: filter.shape
```

```
[78]: (1218, 11)
```

```
[79]: #Univariate analysis
```

```
[80]: plt.hist(filter['Price'],bins=10,color='green') #for numerical column we draw histogram
```

```
[80]: (array([148., 246., 210., 204., 124., 96., 86., 41., 40., 23.]),  
      array([ 9270.72 , 23335.308, 37399.896, 51464.484, 65529.072,  
            79593.66 , 93658.248, 107722.836, 121787.424, 135852.012,  
            149916.6 ]),  
      <BarContainer object of 10 artists>)
```



```
[81]: filter.head()
```

```
[81]: 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

```
[82]: #Summary statistics for screen size, RAM, and price
print("\nSummary Statistics:")
print(filter[['Inches', 'Ram', 'Price']].describe())
```

```
Summary Statistics:
              Price
count    1218.000000
mean     57665.397186
std      31879.277232
min       9270.720000
25%      31914.720000
50%      52054.560000
75%      77238.684000
max     149916.600000
```

```
[83]: filter.isnull().sum()
```

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

```
[84]: filter.groupby(['Inches']).count() #for categorical data count
```

```
[84]: Company  TypeName  ScreenResolution  Cpu  Ram  Memory  Gpu  OpSys  \
      Inches
```

10.1	2	2	2	2	2	2	2	2
11.3	1	1	1	1	1	1	1	1
11.6	29	29	29	29	29	29	29	29
111.8	1	1	1	1	1	1	1	1
12	6	6	6	6	6	6	6	6
12.3	5	5	5	5	5	5	5	5
12.5	30	30	30	30	30	30	30	30
13	2	2	2	2	2	2	2	2
13.3	151	151	151	151	151	151	151	151
13.5	5	5	5	5	5	5	5	5
13.9	6	6	6	6	6	6	6	6
14	179	179	179	179	179	179	179	179
14.1	1	1	1	1	1	1	1	1
15	4	4	4	4	4	4	4	4
15.4	3	3	3	3	3	3	3	3
15.6	642	642	642	642	642	642	642	642
17	1	1	1	1	1	1	1	1
17.3	132	132	132	132	132	132	132	132
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	29	29
111.8	1	1
12	6	6
12.3	5	5
12.5	30	30
13	2	2
13.3	151	151
13.5	5	5
13.9	6	6



14	179	179
14.1	1	1
15	4	4
15.4	3	3
15.6	642	642
17	1	1
17.3	132	132
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

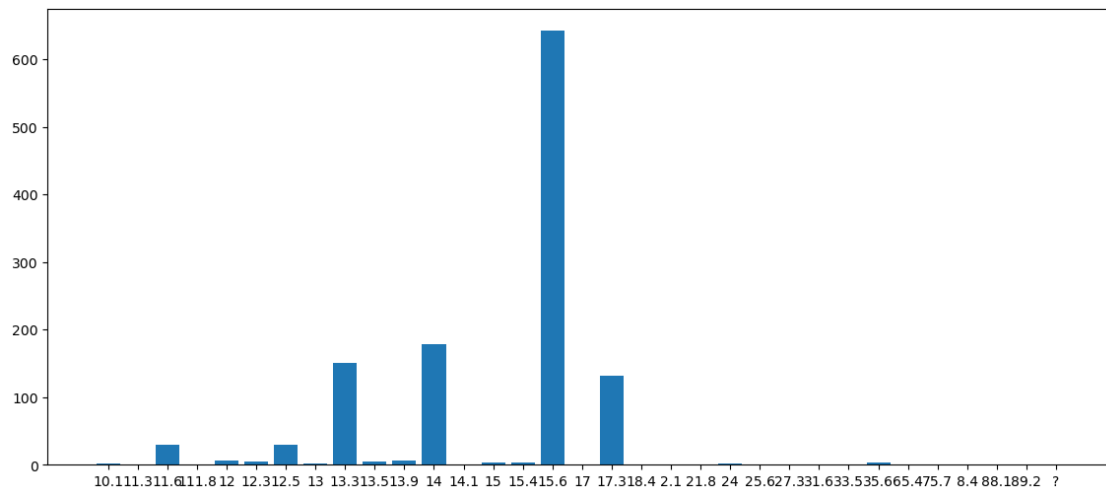
```
[85]: c=filter.groupby(['Inches']).size().reset_index(name="Count").
      ↪rename(columns={"Inches":"inches"})
      c
```

```
[85]:
```

	inches	Count
0	10.1	2
1	11.3	1
2	11.6	29
3	111.8	1
4	12	6
5	12.3	5
6	12.5	30
7	13	2
8	13.3	151
9	13.5	5
10	13.9	6
11	14	179
12	14.1	1
13	15	4
14	15.4	3
15	15.6	642
16	17	1
17	17.3	132
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

```
[86]: import matplotlib.pyplot as plt
plt.figure(figsize=(14, 6))
plt.bar(c["inches"],c["Count"])
plt.show()
```



```
[87]: c['Count%']=c['Count']/sum(c['Count'])*100
c
```

```
[87]:
```

	inches	Count	Count%
0	10.1	2	0.164204
1	11.3	1	0.082102
2	11.6	29	2.380952
3	11.8	1	0.082102
4	12	6	0.492611
5	12.3	5	0.410509

6	12.5	30	2.463054
7	13	2	0.164204
8	13.3	151	12.397373
9	13.5	5	0.410509
10	13.9	6	0.492611
11	14	179	14.696223
12	14.1	1	0.082102
13	15	4	0.328407
14	15.4	3	0.246305
15	15.6	642	52.709360
16	17	1	0.082102
17	17.3	132	10.837438
18	18.4	1	0.082102
19	2.1	1	0.082102
20	21.8	1	0.082102
21	24	2	0.164204
22	25.6	1	0.082102
23	27.3	1	0.082102
24	31.6	1	0.082102
25	33.5	1	0.082102
26	35.6	3	0.246305
27	65.4	1	0.082102
28	75.7	1	0.082102
29	8.4	1	0.082102
30	88.1	1	0.082102
31	89.2	1	0.082102
32	?	1	0.082102

```
[88]: filter.groupby(['Ram']).count() #for categorical data
```

```
[88]:
```

	Company	TypeName	Inches	ScreenResolution	Cpu	Memory	Gpu	OpSys	\
Ram									
12GB	24	24	24	24	24	24	24	24	
16GB	180	180	180	180	180	180	180	180	
1GB	1	1	1	1	1	1	1	1	
24GB	2	2	2	2	2	2	2	2	
2GB	17	17	17	17	17	17	17	17	
32GB	7	7	7	7	7	7	7	7	
4GB	353	353	353	353	353	353	353	353	
64GB	2	2	2	2	2	2	2	2	
6GB	33	33	33	33	33	33	33	33	
8GB	599	599	599	599	599	599	599	599	

	Weight	Price
Ram		
12GB	24	24
16GB	180	180

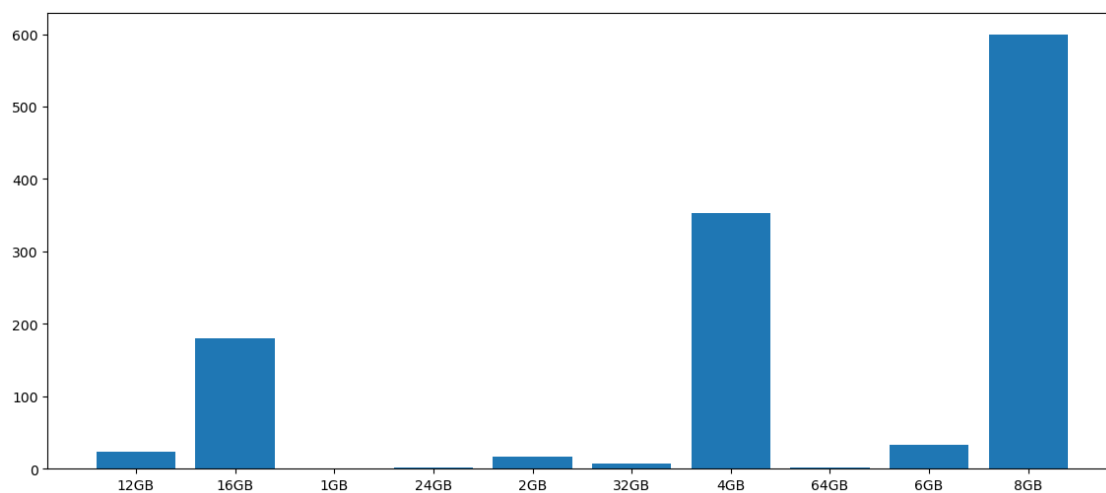
1GB	1	1
24GB	2	2
2GB	17	17
32GB	7	7
4GB	353	353
64GB	2	2
6GB	33	33
8GB	599	599

```
[89]: d=filter.groupby(['Ram']).size().reset_index(name="Count").
      ↪rename(columns={"Ram":"RAM"})
      d
```

```
[89]:
```

	RAM	Count
0	12GB	24
1	16GB	180
2	1GB	1
3	24GB	2
4	2GB	17
5	32GB	7
6	4GB	353
7	64GB	2
8	6GB	33
9	8GB	599

```
[90]: import matplotlib.pyplot as plt
      plt.figure(figsize=(14, 6))
      plt.bar(d["RAM"],d["Count"])
      plt.show()
```



```
[91]: d['Count%']=d['Count']/sum(d['Count'])*100
      d
```

```
[91]:
```

	RAM	Count	Count%
0	12GB	24	1.970443
1	16GB	180	14.778325
2	1GB	1	0.082102
3	24GB	2	0.164204
4	2GB	17	1.395731
5	32GB	7	0.574713
6	4GB	353	28.981938
7	64GB	2	0.164204
8	6GB	33	2.709360
9	8GB	599	49.178982

#Bivariate Analysis

2 Note- 1) A pairplot is typically used to visualize relationships between multiple numerical variables in a dataset by creating scatter plots for each pair of variables. In our case we have only one numerical column, so creating a pairplot doesn't make sense since there are no pairs of variables to plot.

3 2) Also we can not find correlation

4 For categorical to categorical data i.e screen size vs. weight

```
[ ]:
```

```
[92]: #chisquare Testing
```

```
[93]: filter
```

```
[93]:
```

	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	
...	...	...	...	
1270	Lenovo	2 in 1 Convertible	14	
1271	Lenovo	2 in 1 Convertible	13.3	
1272	Lenovo	Notebook	14	
1273	HP	Notebook	15.6	
1300	Lenovo	Notebook	65.4	

		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	
...			
1270	IPS Panel Full HD / Touchscreen	1920x1080	
1271	IPS Panel Quad HD+ / Touchscreen	3200x1800	
1272		1366x768	
1273		1366x768	
1300		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	
...					
1270	Intel Core i7 6500U	2.5GHz	4GB	128GB SSD	
1271	Intel Core i7 6500U	2.5GHz	16GB	512GB SSD	
1272	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB Flash Storage	
1273	Intel Core i7 6500U	2.5GHz	6GB	1TB HDD	
1300	Intel Celeron Dual Core N3050	1.6GHz	2GB	64GB Flash Storage	

		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
...					
1270	Intel HD Graphics	520	Windows 10	1.8kg	33992.6400
1271	Intel HD Graphics	520	Windows 10	1.3kg	79866.7200
1272	Intel HD Graphics		Windows 10	1.5kg	12201.1200
1273	AMD Radeon R5 M330		Windows 10	2.19kg	40705.9200
1300	Intel HD Graphics		Windows 10	1.5kg	12201.1200

[1218 rows x 11 columns]

```
[94]: filter.shape
```

```
[94]: (1218, 11)
```

```
[95]: filter['Inches'].unique() #screen size
```

```
[95]: 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', '14.1', '11.3', '88.1', '21.8',  
          '10.1', '75.7', '8.4', '2.1', '111.8', '89.2', '65.4'],  
          dtype=object)
```

```
[96]: filter["Weight"].unique()
```

```
[96]: array(['1.37kg', '1.34kg', '1.86kg', '1.83kg', '2.1kg', '2.04kg', '1.3kg',  
          '1.6kg', '2.2kg', '0.92kg', '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', '3.35kg',  
          '2.23kg', '?', '2.9kg', '4.42kg', '2.69kg', '2.37kg', '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', '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', '3.14kg', '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', '3.49kg', '29kg', '1.98kg', '1.17kg', '1.71kg', '2.32kg',  
          '4.2kg', '1.55kg', '0.81kg', '1.18kg', '2.72kg', '1.31kg', '18kg',  
          '0.920kg', '3.74kg', '4.36kg', '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)
```

```
[97]: filter["Ram"].unique()
```

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

```
[98]: filter["Cpu"].unique()
```

```
[98]: array(['Intel Core i5 2.3GHz', 'Intel Core i5 1.8GHz',
'Intel Core i5 7200U 2.5GHz', 'Intel Core i7 2.7GHz',
'Intel Core i5 3.1GHz', 'AMD A9-Series 9420 3GHz',
'Intel Core i7 2.2GHz', 'Intel Core i7 8550U 1.8GHz',
'Intel Core i5 8250U 1.6GHz', 'Intel Core i3 6006U 2GHz',
'Intel Core i7 2.8GHz', 'Intel Core M m3 1.2GHz',
'Intel Core i7 7500U 2.7GHz', 'Intel Core i3 7100U 2.4GHz',
'Intel Core i5 7300HQ 2.5GHz', 'AMD E-Series E2-9000e 1.5GHz',
'Intel Core i5 1.6GHz', 'Intel Core i7 8650U 1.9GHz',
'Intel Atom x5-Z8300 1.44GHz', 'AMD E-Series E2-6110 1.5GHz',
'AMD A6-Series 9220 2.5GHz',
'Intel Celeron Dual Core N3350 1.1GHz',
'Intel Core i3 7130U 2.7GHz', 'Intel Core i7 7700HQ 2.8GHz',
'Intel Core i5 2.0GHz', 'AMD Ryzen 1700 3GHz',
'Intel Pentium Quad Core N4200 1.1GHz',
'Intel Celeron Dual Core N3060 1.6GHz', 'Intel Core i5 1.3GHz',
'AMD FX 9830P 3GHz', 'Intel Core i7 7560U 2.4GHz',
'AMD E-Series 6110 1.5GHz', 'Intel Core i5 6200U 2.3GHz',
'Intel Core M 6Y75 1.2GHz', 'Intel Core i5 7500U 2.7GHz',
'Intel Core i3 6006U 2.2GHz', 'AMD A6-Series 9220 2.9GHz',
'Intel Core i7 6920HQ 2.9GHz', 'Intel Core i5 7Y54 1.2GHz',
'Intel Core i7 6500U 2.5GHz', 'AMD E-Series 9000e 1.5GHz',
'AMD A10-Series A10-9620P 2.5GHz', 'AMD A6-Series A6-9220 2.5GHz',
'Intel Core i5 2.9GHz', 'Intel Core i7 6600U 2.6GHz',
'Intel Core i3 6006U 2.0GHz',
'Intel Celeron Dual Core 3205U 1.5GHz',
'AMD A10-Series 9600P 2.4GHz', 'Intel Core i7 7600U 2.8GHz',
'AMD A8-Series 7410 2.2GHz',
'Intel Celeron Dual Core 3855U 1.6GHz',
'Intel Pentium Quad Core N3710 1.6GHz',
'AMD A12-Series 9720P 2.7GHz', 'Intel Core i5 7300U 2.6GHz',
'AMD A12-Series 9720P 3.6GHz',
'Intel Celeron Quad Core N3450 1.1GHz',
'Intel Celeron Dual Core N3060 1.60GHz',
'Intel Core i5 6440HQ 2.6GHz', 'Intel Core i7 6820HQ 2.7GHz',
'Intel Core i7 7820HQ 2.9GHz', 'Intel Core i7 7820HK 2.9GHz',
'AMD Ryzen 1600 3.2GHz', 'Intel Core i7 7Y75 1.3GHz',
'Intel Core i5 7440HQ 2.8GHz', 'Intel Xeon E3-1505M V6 3GHz',
'Intel Core i7 7660U 2.5GHz', 'Intel Core i7 7700HQ 2.7GHz',
'Intel Core M m3-7Y30 2.2GHz', 'Intel Core i5 7Y57 1.2GHz',
'Intel Core i7 6700HQ 2.6GHz', 'Intel Core i3 6100U 2.3GHz',
'Intel Atom x5-Z8350 1.44GHz', 'AMD A10-Series 9620P 2.5GHz',
'AMD E-Series 7110 1.8GHz', 'Intel Celeron Dual Core N3350 2.0GHz',
'AMD A9-Series A9-9420 3GHz', 'Intel Core i7 6820HK 2.7GHz',
'Intel Core M 7Y30 1.0GHz', 'Intel Celeron Quad Core N3160 1.6GHz',
'Intel Core i5 6300U 2.4GHz', 'Intel Core i3 6100U 2.1GHz',
'AMD E-Series E2-9000 2.2GHz',
```



```

'Intel Celeron Dual Core N3050 1.6GHz',
'Intel Core M M3-6Y30 0.9GHz', 'AMD A9-Series 9420 2.9GHz',
'Intel Core i5 6300HQ 2.3GHz', 'AMD A6-Series 7310 2GHz',
'Intel Atom Z8350 1.92GHz', 'Intel Core i5 6260U 1.8GHz',
'Intel Pentium Dual Core N4200 1.1GHz',
'Intel Celeron Quad Core N3710 1.6GHz', 'Intel Core M 1.2GHz',
'AMD A12-Series 9700P 2.5GHz', 'Intel Core i7 7500U 2.5GHz',
'Intel Pentium Dual Core 4405U 2.1GHz',
'AMD A4-Series 7210 2.2GHz', 'Intel Core i7 6560U 2.2GHz',
'Intel Core M m7-6Y75 1.2GHz', 'AMD FX 8800P 2.1GHz',
'Intel Core M M7-6Y75 1.2GHz', 'Intel Core i5 7200U 2.50GHz',
'Intel Core i5 7200U 2.70GHz', 'Intel Atom X5-Z8350 1.44GHz',
'Intel Core i5 7200U 2.7GHz', 'Intel Core M 1.1GHz',
'Intel Atom x5-Z8550 1.44GHz',
'Intel Pentium Dual Core 4405Y 1.5GHz',
'Intel Pentium Quad Core N3700 1.6GHz', 'Intel Core M 6Y54 1.1GHz',
'Intel Core i7 6500U 2.50GHz',
'Intel Celeron Dual Core N3350 2GHz',
'Samsung Cortex A72&A53 2.0GHz', 'AMD E-Series 9000 2.2GHz',
'Intel Core M 6Y30 0.9GHz', 'AMD A9-Series 9410 2.9GHz'],
dtype=object)

```

```
[99]: filter["Gpu"].unique()
```

```

[99]: array(['Intel Iris Plus Graphics 640', 'Intel HD Graphics 6000',
'Intel HD Graphics 620', 'AMD Radeon Pro 455',
'Intel Iris Plus Graphics 650', 'AMD Radeon R5',
'Intel Iris Pro Graphics', 'Nvidia GeForce MX150',
'Intel UHD Graphics 620', 'Intel HD Graphics 520',
'AMD Radeon Pro 555', 'AMD Radeon R5 M430',
'Intel HD Graphics 615', 'Nvidia GeForce 940MX',
'Nvidia GeForce GTX 1050', 'AMD Radeon R2', 'AMD Radeon 530',
'Nvidia GeForce 930MX', 'Intel HD Graphics',
'Intel HD Graphics 500', 'Nvidia GeForce 930MX ',
'Nvidia GeForce GTX 1060', 'Nvidia GeForce 150MX',
'Intel Iris Graphics 540', 'AMD Radeon RX 580',
'Nvidia GeForce 920MX', 'AMD Radeon R4 Graphics', 'AMD Radeon 520',
'Nvidia GeForce GTX 1070', 'Nvidia GeForce GTX 1050 Ti',
'Intel HD Graphics 400', 'Nvidia GeForce MX130', 'AMD R4 Graphics',
'Nvidia GeForce GTX 940MX', 'AMD Radeon RX 560',
'Nvidia GeForce 920M', 'AMD Radeon R7 M445', 'AMD Radeon RX 550',
'Nvidia GeForce GTX 1050M', 'Intel HD Graphics 515',
'AMD Radeon R5 M420', 'Intel HD Graphics 505',
'Nvidia GTX 980 SLI', 'AMD R17M-M1-70', 'Nvidia GeForce 920MX ',
'Nvidia GeForce GTX 950M', 'AMD FirePro W4190M ',
'Nvidia GeForce GTX 980M', 'Intel Iris Graphics 550',
'Nvidia GeForce 930M', 'Intel HD Graphics 630',

```

```
'AMD Radeon R5 430', 'Nvidia GeForce GTX 940M',
'Intel HD Graphics 510', 'Intel HD Graphics 405',
'AMD Radeon RX 540', 'Nvidia Quadro M1200',
'Nvidia GeForce GT 940MX', 'AMD FirePro W5130M',
'Nvidia Quadro M2200M', 'AMD Radeon R4', 'Nvidia Quadro M620',
'AMD Radeon R7 M460', 'Intel HD Graphics 530',
'Nvidia GeForce GTX 965M', 'Nvidia GeForce GTX1050 Ti',
'Nvidia GeForce GTX 960M', 'AMD Radeon R2 Graphics',
'Nvidia GeForce GTX 970M', 'Nvidia GeForce GTX 960<U+039C>',
'Intel Graphics 620', 'Nvidia GeForce GTX 1080',
'Nvidia GeForce GTX 960', 'AMD Radeon R5 520',
'AMD Radeon R7 M440', 'AMD Radeon R7', 'Nvidia Quadro M520M',
'Nvidia Quadro M2200', 'Intel HD Graphics 540',
'Nvidia Quadro M1000M', 'AMD Radeon 540', 'Nvidia Quadro M2000M',
'Nvidia GeForce GTX1060', 'Intel HD Graphics 5300',
'AMD Radeon R5 M420X', 'AMD Radeon R7 Graphics',
'Nvidia GeForce 920', 'Nvidia GeForce 940M',
'Nvidia GeForce GTX 930MX', 'AMD Radeon R7 M465', 'AMD Radeon R3',
'Nvidia GeForce GTX 1050Ti', 'AMD Radeon R7 M365X',
'AMD Radeon R9 M385', 'Intel HD Graphics 620 ',
'AMD Radeon R5 M330', 'AMD FirePro W4190M', 'AMD Radeon R5 M315',
'Nvidia Quadro M500M', 'AMD Radeon R7 M360', 'Nvidia GeForce 960M',
'ARM Mali T860 MP4'], dtype=object)
```

```
[100]: from sklearn.preprocessing import LabelEncoder
label=LabelEncoder()
filter["Inches"]=label.fit_transform(filter['Inches'])
filter['Weight']=label.fit_transform(filter['Weight'])
filter['Ram']=label.fit_transform(filter['Ram'])
filter['Gpu']=label.fit_transform(filter['Gpu'])
filter['Cpu']=label.fit_transform(filter['Cpu'])
filter
```

C:\Users\DELL\AppData\Local\Temp\ipykernel\_3900\1838295643.py:3:

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](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
filter["Inches"]=label.fit_transform(filter['Inches'])
```

C:\Users\DELL\AppData\Local\Temp\ipykernel\_3900\1838295643.py:4:

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
    filter['Weight']=label.fit_transform(filter['Weight'])
C:\Users\DELL\AppData\Local\Temp\ipykernel_3900\1838295643.py:5:
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
    filter['Ram']=label.fit_transform(filter['Ram'])
C:\Users\DELL\AppData\Local\Temp\ipykernel_3900\1838295643.py:6:
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
    filter['Gpu']=label.fit_transform(filter['Gpu'])
C:\Users\DELL\AppData\Local\Temp\ipykernel_3900\1838295643.py:7:
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
    filter['Cpu']=label.fit_transform(filter['Cpu'])
```

```
[100]:
```

	Company	TypeName	Inches	\
0	Apple	Ultrabook	8	
1	Apple	Ultrabook	8	
2	HP	Notebook	15	
3	Apple	Ultrabook	14	
4	Apple	Ultrabook	8	
...	...	...	...	...
1270	Lenovo	2 in 1 Convertible	11	
1271	Lenovo	2 in 1 Convertible	8	
1272	Lenovo	Notebook	11	
1273	HP	Notebook	15	
1300	Lenovo	Notebook	27	

		ScreenResolution	Cpu	Ram	\
0	IPS Panel	Retina Display	2560x1600	65	9
1			1440x900	63	9
2		Full HD	1920x1080	74	9
3	IPS Panel	Retina Display	2880x1800	85	1
4	IPS Panel	Retina Display	2560x1600	67	9
...			...	...	...

1270	IPS Panel Full HD / Touchscreen	1920x1080	88	6
1271	IPS Panel Quad HD+ / Touchscreen	3200x1800	88	1
1272		1366x768	34	4
1273		1366x768	88	8
1300		1366x768	34	4

		Memory	Gpu	OpSys	Weight	Price
0		128GB SSD	56	macOS	40	71378.6832
1	128GB Flash Storage		49	macOS	37	47895.5232
2		256GB SSD	51	No OS	75	30636.0000
3		512GB SSD	8	macOS	72	135195.3360
4		256GB SSD	57	macOS	40	96095.8080
...		...	...	...	...	...
1270		128GB SSD	45	Windows 10	79	33992.6400
1271		512GB SSD	45	Windows 10	43	79866.7200
1272	64GB Flash Storage		38	Windows 10	57	12201.1200
1273		1TB HDD	19	Windows 10	110	40705.9200
1300	64GB Flash Storage		38	Windows 10	57	12201.1200

[1218 rows x 11 columns]

```
[101]: df2=filter[['Inches', 'Weight', 'Ram', 'Gpu', 'Cpu']]
df2
```

```
[101]:
```

	Inches	Weight	Ram	Gpu	Cpu
0	8	40	9	56	65
1	8	37	9	49	63
2	15	75	9	51	74
3	14	72	1	8	85
4	8	40	9	57	67
...	...	...	...	...	...
1270	11	79	6	45	88
1271	8	43	1	45	88
1272	11	57	4	38	34
1273	15	110	8	19	88
1300	27	57	4	38	34

[1218 rows x 5 columns]

```
[102]: X=df2[['Inches', 'Ram', 'Gpu', 'Cpu']]
Y=df2['Weight']
X
```

```
[102]:
```

	Inches	Ram	Gpu	Cpu
0	8	9	56	65
1	8	9	49	63
2	15	9	51	74

3	14	1	8	85
4	8	9	57	67
...	...	...	...	...
1270	11	6	45	88
1271	8	1	45	88
1272	11	4	38	34
1273	15	8	19	88
1300	27	4	38	34

[1218 rows x 4 columns]

[ ]:

```
[103]: from sklearn.feature_selection import chi2
value=chi2(X,Y)
value
```

```
[103]: (array([1016.64755285, 476.90601084, 4261.45414119, 2917.62647653]),
array([1.07765455e-112, 1.98757538e-026, 0.00000000e+000, 0.00000000e+000]))
```

```
[104]: Y
```

```
[104]: 0      40
1      37
2      75
3      72
4      40

...
1270    79
1271    43
1272    57
1273   110
1300    57
Name: Weight, Length: 1218, dtype: int32
```

```
[105]: from sklearn.feature_selection import chi2
value=chi2(X,Y)
value
```

```
[105]: (array([1016.64755285, 476.90601084, 4261.45414119, 2917.62647653]),
array([1.07765455e-112, 1.98757538e-026, 0.00000000e+000, 0.00000000e+000]))
```

```
[106]: df2.groupby(['Inches', 'Weight']).count()
```

```
[106]:
```

		Ram	Gpu	Cpu
Inches	Weight			
0	2	2	2	2

1	23	1	1	1
2	11	1	1	1
	18	2	2	2
	19	1	1	1
...		...	...	...
28	182	1	1	1
29	135	1	1	1
30	46	1	1	1
31	124	1	1	1
32	69	1	1	1

[312 rows x 3 columns]

#this infer that weight of screen will be more related to inches out of Ram,Gpu and Cpu.

```
[107]: #Anova Testing
A=X[['Ram', 'Gpu', 'Cpu']]
B=filter['Price']
```

```
[108]: A
```

```
[108]:      Ram  Gpu  Cpu
0       9   56   65
1       9   49   63
2       9   51   74
3       1    8   85
4       9   57   67

...    ...  ...  ...
1270    6   45   88
1271    1   45   88
1272    4   38   34
1273    8   19   88
1300    4   38   34
```

[1218 rows x 3 columns]

```
[109]: B
```

```
[109]: 0      71378.6832
1      47895.5232
2      30636.0000
3     135195.3360
4      96095.8080

...
1270     33992.6400
1271     79866.7200
1272     12201.1200
```

```
1273      40705.9200
1300      12201.1200
Name: Price, Length: 1218, dtype: float64
```

```
[110]: from sklearn.feature_selection import f_classif, SelectKBest
p_values=f_classif(A,B)
p_values
```

```
[110]: (array([1.26424 , 1.4299228, 1.7314297], dtype=float32),
array([2.7565248e-03, 1.2727941e-05, 7.7857755e-11], dtype=float32))
```

```
[111]: selector=SelectKBest(score_func=f_classif,k=2)
new=selector.fit_transform(A,B)
new
```

```
[111]: array([[56, 65],
[49, 63],
[51, 74],
...,
[38, 34],
[19, 88],
[38, 34]])
```

```
[112]: import pandas as pd
p_value=pd.Series(p_values[1])
p_value.index=A.columns
p_value
```

```
[112]: Ram      2.756525e-03
Gpu       1.272794e-05
Cpu       7.785775e-11
dtype: float32
```

#Cpu is affecting more the price of Laptop as compared to Ram and Gpu

#This infer that we should or a industry should more focus improving cpu in the laptop out of Gpu and Ram so to increse the price

```
[ ]:
```