

project-1-explorative-data-analysis

July 29, 2024

1 Milestone-1 Evaluation

2 *Project Documentation: Exploratory Data Analysis of Laptop Dataset :*

2.0.1 Title : Laptop Data Analysis

2.0.2 Name : Esakiappan E

2.0.3 DA/DS : Data Analytics (DA)

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

2.0.5 Online/Offline : Online

2.1 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.

2.2 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. **

2.3 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. **

2.4 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.

2.5 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?

2.6 4) Project Workflow :

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

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

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

```
[521]:
```

	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]

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

```
[523]: df
```

```
[523]:
```

	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]

[524]: 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
```

```
[525]: df.describe()
```

```
[525]:
```

	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

3 Data Checks to Perform before Data Cleaning (Data Exploration)

1. Check for Duplicate Value

2. Check for Check for Missing Values

3. Check for DataType

4. Check the number of Unique Value in every Columns

5. Check Statistics of Dataset

```
[ ]:
```

```
[526]: duplicates_value = df.duplicated().sum()  
duplicates_value
```

```
[526]: 57
```

```
[527]: df=df.drop_duplicates()  
df
```

```
[527]:
```

	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]

4) Data Cleaning

4.0.1 * Handle missing values:

5 MISSING VALUE IMPUTATION

[389]: df

```
[389]:      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]

```
[390]: df.isnull()
```

```
[390]:
```

	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]

```
[391]: df.isnull().sum()
```

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

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

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

```
[393]: company_mode=df.Company.mode()[0]
```

```
[394]: company_mode
```

```
[394]: 'Lenovo'
```

```
[528]: df.Company.fillna(company_mode,inplace=True)
```

C:\Users\DELL\AppData\Local\Temp\ipykernel_17368\80910062.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

```
df.Company.fillna(company_mode,inplace=True)
```

```
[396]: df.isnull().sum()
```

```
[396]: Company          0
      TypeName        21
      Inches          54
      ScreenResolution  1
      Cpu              1
      Ram             15
      Memory          1
      Gpu              1
      OpSys            1
      Weight          15
```

```
Price          1
dtype: int64
```

```
[397]: df['TypeName'].unique()
```

```
[397]: array(['Ultrabook', 'Notebook', nan, 'Gaming', '2 in 1 Convertible',
           'Workstation', 'Netbook'], dtype=object)
```

```
[398]: typename_mode=df.TypeName.mode()[0]
typename_mode
```

```
[398]: 'Notebook'
```

```
[399]: df.TypeName.fillna(typename_mode,inplace=True)
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_17368\2877775559.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
```

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

```
[400]: df.isnull().sum()
```

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

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

```
[401]: array(['13.3', '15.6', '15.4', '14', '12', nan, '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)
```

```
[402]: inches_mode=df.Inches.mode()[0]
inches_mode
```

```
[402]: '15.6'
```

```
[403]: df.Inches.fillna(inches_mode,inplace=True)
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_17368\3128342732.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
df.Inches.fillna(inches_mode,inplace=True)
```

```
[404]: df.isnull().sum()
```

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

```
[405]: a=df['ScreenResolution'].unique()
```

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

```
[406]: 41
```

```
[407]: screen_resolution_mode=df.ScreenResolution.mode()[0]
screen_resolution_mode
```

```
[407]: 'Full HD 1920x1080'
```

```
[408]: df.ScreenResolution.fillna(screen_resolution_mode,inplace=True)
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_17368\1393818328.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
df.ScreenResolution.fillna(screen_resolution_mode,inplace=True)
```

```
[409]: df.isnull().sum()
```

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

```
[410]: df['Cpu'].unique()
```

```
[410]: 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 i7 2.9GHz',
      'Intel Core i3 7100U 2.4GHz', nan, '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 7820HK 2.9GHz', 'Intel Xeon E3-1505M V6 3GHz',
      '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',
```

```

'Intel Core i7 7820HQ 2.9GHz', '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',
'AMD Ryzen 1600 3.2GHz', 'Intel Core i7 7Y75 1.3GHz',
'Intel Core i5 7440HQ 2.8GHz', '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 Xeon E3-1535M v6 3.1GHz',
'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 Xeon E3-1535M v5 2.9GHz',
'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)

```

```

[411]: b=df['Cpu'].unique()
len(b)

```

```
[411]: 119
```

```
[412]: cpu_mode=df.Cpu.mode()[0]  
cpu_mode
```

```
[412]: 'Intel Core i5 7200U 2.5GHz'
```

```
[413]: df.Cpu.fillna(cpu_mode,inplace=True)
```

C:\Users\DELL\AppData\Local\Temp\ipykernel_17368\2794462538.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

```
df.Cpu.fillna(cpu_mode,inplace=True)
```

```
[414]: df.isnull().sum()
```

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

```
[415]: df['Ram'].unique()
```

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

```
[416]: ram_mode=df.Ram.mode()[0]  
ram_mode
```

```
[416]: '8GB'
```

```
[417]: df.Ram.fillna(ram_mode,inplace=True)
```

C:\Users\DELL\AppData\Local\Temp\ipykernel_17368\1067653301.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
`df.Ram.fillna(ram_mode,inplace=True)`

```
[418]: df.isnull().sum()
```

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

```
[419]: df['Memory'].unique()
```

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

```
[420]: memory_mode=df.Memory.mode()[0]
      memory_mode
```

```
[420]: '256GB SSD'
```

```
[421]: df.Memory.fillna(memory_mode,inplace=True)
```

C:\Users\DELL\AppData\Local\Temp\ipykernel_17368\1470385812.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
df.Memory.fillna(memory_mode,inplace=True)
```

```
[422]: df.isnull().sum()
```

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

```
[423]: c=df['Gpu'].unique()
```

```
[424]: len(c)
```

```
[424]: 111
```

```
[425]: gpu_mode=df.Gpu.mode()[0]
      gpu_mode
```

```
[425]: 'Intel HD Graphics 620'
```

```
[426]: df.Gpu.fillna(gpu_mode,inplace=True)
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_17368\4038479589.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
df.Gpu.fillna(gpu_mode,inplace=True)
```

```
[427]: df.isnull().sum()
```

```
[427]: Company          0
      TypeName        0
      Inches          0
      ScreenResolution 0
      Cpu             0
      Ram             0
      Memory          0
```

```
Gpu          0
OpSys        1
Weight       15
Price        1
dtype: int64
```

```
[428]: df['OpSys'].unique()
```

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

```
[429]: opsys_mode=df.OpSys.mode()[0]
opsys_mode
```

```
[429]: 'Windows 10'
```

```
[430]: df.OpSys.fillna(opsys_mode,inplace=True)
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_17368\2770944402.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
df.OpSys.fillna(opsys_mode,inplace=True)
```

```
[431]: df.isnull().sum()
```

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

```
[432]: d=df['Weight'].unique()
len(d)
```

```
[432]: 197
```

```
[433]: weight_mode=df.Weight.mode()[0]
weight_mode
```

```
[433]: '2.2kg'
```

```
[434]: df.Weight.fillna(weight_mode,inplace=True)
```

C:\Users\DELL\AppData\Local\Temp\ipykernel_17368\3899060447.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

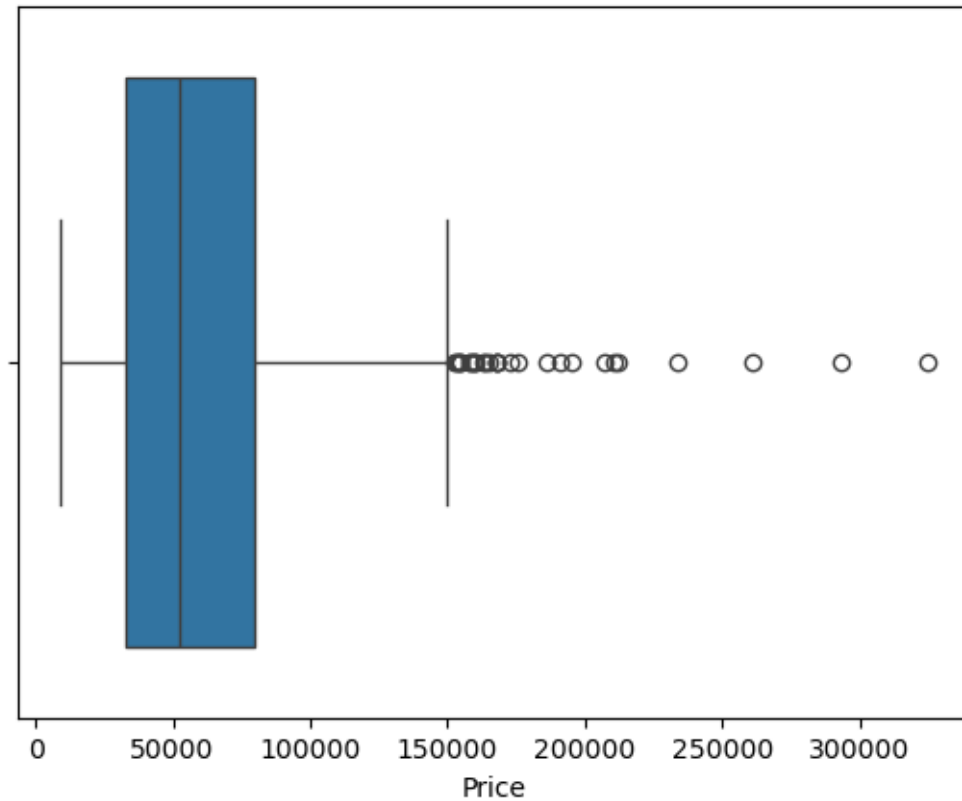
```
df.Weight.fillna(weight_mode,inplace=True)
```

```
[435]: df.isnull().sum()
```

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

```
[529]: sns.boxplot(x=df['Price'])
```

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



```
[436]: df['Price'].unique()
```

```
[436]: array([ 71378.6832,  47895.5232,  30636.      , 135195.336 ,  96095.808 ,
        21312.      , 114017.6016,  61735.536 ,  79653.6   ,  41025.6   ,
        20986.992 ,  18381.0672, 130001.6016,  26581.392 ,  67260.672 ,
        80908.344 ,  39693.6   , 152274.24  ,  26586.72 ,  52161.12 ,
         nan,    53226.72 ,  13746.24  ,  43636.32 ,  35111.52 ,
        22305.1392,  58554.72 ,  42624.   ,  69157.44 ,  47738.88 ,
        13053.0672,  10602.72 ,  23389.92 ,  99580.32 ,  53173.44 ,
        13266.72  ,  19553.76 ,  26037.4032,  46833.12 ,  20725.92 ,
        79866.72 ,  27864.9072,  36336.96 ,  75604.32 ,  69210.72 ,
        34045.92 ,  24828.48 ,  44808.48 ,  21231.5472,  58767.84 ,
        20459.52 ,  40908.384 ,  31232.2032, 130482.72 ,  22111.2   ,
        31914.72 ,  50136.48 ,  36763.2   , 105654.24 ,  23373.4032,
        12201.12 ,  29250.72 ,  50562.72 ,  58021.92 ,  50882.4   ,
        46353.6   ,  58341.6   ,  27652.32 ,  45554.4   ,  28238.4   ,
        52054.56 ,  58403.4048,  80452.8   ,  45820.8   ,  21258.72 ,
        21045.6   ,  71874.72 ,  37242.72 ,  31914.1872,  77202.72 ,
        87858.72 ,  36709.92 ,  63776.16 ,  63669.6   ,  55890.72 ,
        45128.16 ,  31962.672 ,  25840.8   ,  30742.56 ,  66546.72 ,
        38308.32 ,  18594.72 ,  34472.16 ,  59620.32 ,  71395.2   ,
```

22105.872 , 63563.04 , 78854.4 , 67239.36 , 73473.12 ,
 74538.72 , 38468.16 , 86793.12 , 57755.52 , 60223.9824 ,
 30049.92 , 59567.04 , 25521.12 , 119427.12 , 33513.12 ,
 67718.88 , 24029.28 , 43263.36 , 14811.3072 , 74378.88 ,
 49443.84 , 34045.3872 , 23922.72 , 47099.52 , 30476.16 ,
 31861.44 , 52640.64 , 13445.7408 , 49976.64 , 34898.4 ,
 59461.5456 , 46300.32 , 32074.56 , 19660.32 , 107305.92 ,
 18328.32 , 23816.16 , 66560.5728 , 47898.72 , 26533.44 ,
 100699.2 , 57648.96 , 32980.32 , 70063.2 , 21471.84 ,
 42890.4 , 38787.84 , 57489.12 , 18541.44 , 95850.72 ,
 19367.8128 , 56502.9072 , 45501.12 , 40173.12 , 16463.52 ,
 26053.92 , 49177.44 , 24455.52 , 149130.72 , 43316.64 ,
 98514.72 , 42251.04 , 63882.72 , 82530.72 , 127712.16 ,
 41505.12 , 52693.92 , 57808.8 , 13852.8 , 53274.672 ,
 37189.44 , 44701.92 , 48697.92 , 324954.72 , 51095.52 ,
 55677.6 , 98301.6 , 26267.04 , 39533.76 , 93186.72 ,
 162770.4 , 74485.44 , 103842.72 , 74964.96 , 49650.5664 ,
 31381.92 , 54931.68 , 61218.72 , 68145.12 , 36089.208 ,
 72620.64 , 42304.32 , 130873.7952 , 44328.96 , 45768.0528 ,
 40972.32 , 47472.48 , 67612.32 , 21258.1872 , 17582.4 ,
 45767.52 , 20779.2 , 207259.2 , 45074.88 , 61005.6 ,
 47365.92 , 52480.8 , 29783.52 , 159786.72 , 35964. ,
 108691.2 , 24988.32 , 37402.56 , 23757.552 , 56423.52 ,
 133146.72 , 90522.72 , 60845.76 , 23656.32 , 35004.96 ,
 30103.2 , 42570.72 , 54239.04 , 46886.4 , 104370.192 ,
 39164.5296 , 37992.3696 , 45234.72 , 22803.84 , 44169.12 ,
 30849.12 , 50669.28 , 58448.16 , 62817.12 , 35112.0528 ,
 63243.36 , 97449.12 , 39373.92 , 153705.3408 , 78215.04 ,
 27119.52 , 113060.16 , 34578.72 , 67399.2 , 19180.2672 ,
 105228. , 55571.04 , 111834.72 , 14652. , 44968.32 ,
 24503.472 , 52214.4 , 68837.76 , 58288.32 , 48058.56 ,
 15557.76 , 55938.672 , 71128.8 , 140605.92 , 50243.04 ,
 71075.52 , 67559.04 , 60952.32 , 14651.4672 , 60885.72 ,
 14646.672 , 38148.48 , 84129.12 , 60153.12 , 14865.12 ,
 85672.1088 , 19980. , 35324.64 , 69477.12 , 75071.52 ,
 92615.0256 , 74751.84 , 51729.552 , 17155.6272 , 29696.6736 ,
 76030.56 , 50349.6 , 43103.52 , 93240. , 22697.28 ,
 117162.72 , 46300.8528 , 26053.3872 , 29463.84 , 15238.08 ,
 63456.48 , 21498.48 , 88178.4 , 93181.392 , 121584.96 ,
 72940.32 , 113752.8 , 133679.52 , 55357.92 , 84768.48 ,
 36975.7872 , 41498.1936 , 65510.9568 , 144495.36 , 139860. ,
 16303.68 , 81465.12 , 60978.96 , 119826.72 , 99793.44 ,
 93080.16 , 89510.4 , 21791.52 , 16221.096 , 102564. ,
 103523.04 , 42038.4528 , 31909.392 , 79920. , 28768.536 ,
 64755.4464 , 101178.72 , 31808.16 , 61751.52 , 60867.072 ,
 106506.72 , 15930.72 , 14332.32 , 53812.8 , 130269.6 ,
 90309.6 , 18488.16 , 126912.96 , 39906.72 , 76137.12 ,

23539.104 , 106187.04 , 54757.9872, 137941.92 , 81731.52 ,
 71661.6 , 109010.88 , 34093.872 , 28984.32 , 23176.8 ,
 111355.2 , 16197.12 , 40439.52 , 98994.24 , 61485.12 ,
 67932. , 98133.768 , 39427.2 , 128298.24 , 72673.92 ,
 89084.16 , 31254.048 , 38681.28 , 13261.392 , 124568.64 ,
 95797.44 , 103896. , 53918.8272, 71928. , 47893.392 ,
 48538.08 , 18115.2 , 32979.7872, 85194.72 , 14119.2 ,
 65214.72 , 32660.64 , 70489.44 , 44542.08 , 64961.1072,
 74589.336 , 13053.6 , 27783.9216, 158135.04 , 80133.12 ,
 68184.0144, 31168.8 , 73366.56 , 160520.3856, 42486.0048,
 39207.1536, 42517.44 , 37589.04 , 41824.8 , 24634.008 ,
 21152.16 , 92121.12 , 59513.2272, 143802.72 , 28992.312 ,
 68198.4 , 11934.72 , 13586.4 , 30310.992 , 32921.712 ,
 33566.4 , 104695.2 , 52747.2 , 99153.5472, 44222.4 ,
 36496.8 , 99367.2 , 141884.64 , 145401.12 , 39907.2528,
 53733.9456, 81912.1392, 15717.6 , 125154.72 , 79813.44 ,
 89137.44 , 32447.52 , 94305.6 , 32127.84 , 28185.12 ,
 107892. , 78534.72 , 88977.6 , 93932.64 , 64948.32 ,
 35616.6144, 17529.12 , 24775.2 , 122490.72 , 261018.72 ,
 46833.6528, 23650.992 , 19127.52 , 62231.04 , 74005.92 ,
 120831.5808, 20193.12 , 59886.72 , 78055.2 , 41345.28 ,
 49656.96 , 12733.92 , 24935.04 , 34046.4528, 101232. ,
 78801.12 , 48304.7136, 93635.3376, 127818.72 , 59087.52 ,
 27753.552 , 130536. , 62284.32 , 23976. , 14598.72 ,
 48964.32 , 138474.72 , 64628.64 , 84395.52 , 37775.52 ,
 77250.672 , 63499.104 , 19441.872 , 56689.92 , 60472.8 ,
 63722.88 , 167691.8736, 65481.12 , 22324.32 , 28504.8 ,
 28717.92 , 79215.1056, 67026.24 , 21951.36 , 99519.048 ,
 43580.376 , 55091.52 , 68944.32 , 26373.6 , 53168.112 ,
 109277.28 , 149916.6 , 32639.8608, 29073.2976, 30316.32 ,
 16943.04 , 101657.7072, 137995.2 , 51841.44 , 35644.32 ,
 100006.56 , 22857.12 , 33110.856 , 42357.6 , 98834.4 ,
 34898.9328, 15877.44 , 76012.4448, 45664.6896, 63936. ,
 89864.1792, 44574.048 , 194972.832 , 71847.0144, 26101.872 ,
 31409.6256, 103096.8 , 34632. , 51148.8 , 128884.32 ,
 111593.8944, 21887.424 , 64308.96 , 35431.2 , 81784.8 ,
 175770.72 , 60031.1088, 233845.92 , 25308. , 45282.672 ,
 83170.08 , 152859.7872, 55837.44 , 99047.52 , 83063.52 ,
 25059.7152, 62938.0656, 54345.6 , 126273.6 , 33886.08 ,
 154458.72 , 31003.632 , 191211.264 , 125208. , 93985.92 ,
 100752.48 , 71341.92 , 10810.512 , 104961.6 , 62071.2 ,
 124142.4 , 43156.8 , 110017.872 , 39640.32 , 41558.4 ,
 89457.12 , 128671.2 , 51202.08 , 42081.0768, 20512.8 ,
 133467.4656, 40226.4 , 58075.2 , 99900. , 72354.24 ,
 18914.4 , 292986.72 , 63190.08 , 87912. , 147832.2864,
 17316. , 31435.2 , 163723.5792, 48484.8 , 38041.3872,
 99633.6 , 32767.2 , 54665.28 , 121318.56 , 46087.2 ,

```

101391.84 , 41931.36 , 136343.52 , 45323.1648, 68464.8 ,
54185.76 , 119347.2 , 94731.84 , 56210.4 , 73952.64 ,
39160.8 , 90043.2 , 49816.8 , 48618. , 90576. ,
36486.144 , 24279.696 , 72988.272 , 15824.16 , 141138.72 ,
78588. , 49497.12 , 47952. , 32713.92 , 107257.968 ,
109170.72 , 104588.1072, 79014.24 , 42037.92 , 64202.4 ,
165168. , 54291.7872, 46939.68 , 59668.8048, 36496.2672,
69103.6272, 97236. , 38889.072 , 37725.4368, 15392.592 ,
58874.4 , 88924.32 , 29762.208 , 146946.24 , 78438.816 ,
94572. , 77788.8 , 79333.3872, 168045.12 , 41292. ,
93772.8 , 53759.52 , 95371.2 , 45101.52 , 24808.2336,
43956. , 167778.72 , 37029.6 , 104587.5744, 149184. ,
62870.4 , 109218.672 , 53386.56 , 48751.2 , 42943.68 ,
63349.92 , 101658.24 , 14418.6336, 61272. , 20246.4 ,
111301.92 , 67132.8 , 29144.16 , 58607.4672, 55904.5728,
36443.52 , 81997.92 , 84715.2 , 100550.5488, 64468.8 ,
41025.0672, 79387.2 , 118761.12 , 42410.3472, 63159.7104,
55754.3232, 56476.8 , 86526.72 , 57542.4 , 65480.5872,
91908. , 56633.976 , 53839.9728, 11231.424 , 75924. ,
55922.688 , 53280. , 112065.9552, 78268.32 , 82351.6992,
122010.6672, 15339.312 , 139593.6 , 120093.12 , 37570.392 ,
96916.32 , 52746.6672, 102777.12 , 77682.24 , 211788. ,
42517.9728, 70809.12 , 69264. , 25679.8944, 91294.7472,
28771.2 , 50083.2 , 172627.2 , 34433.2656, 39960. ,
54931.1472, 20619.36 , 20965.1472, 18434.3472, 43601.688 ,
34035.264 , 26640. , 60480.792 , 62176.1616, 11135.52 ,
75289.968 , 80516.2032, 46193.76 , 21205.44 , 40980.312 ,
104908.32 , 146519.4672, 54825.12 , 118601.28 , 69929.4672,
10442.88 , 80612.64 , 27899.0064, 100965.6 , 109244.2464,
14811.84 , 40066.56 , 32820.48 , 210424.032 , 41771.52 ,
115709.2416, 130003.2 , 60888.384 , 122381.496 , 53807.472 ,
124621.92 , 18061.92 , 119916.2304, 25515.2592, 79536.384 ,
42010.7472, 108744.48 , 94252.32 , 25414.0272, 74059.2 ,
36177.12 , 69530.4 , 114731.5536, 87219.36 , 44275.68 ,
47686.1328, 17742.24 , 48431.52 , 36816.48 , 61964.64 ,
70702.56 , 19607.04 , 114552. , 14492.16 , 109165.392 ,
142790.4 , 103842.1872, 9270.72 , 67772.16 , 78647.1408,
91288.3536, 78694.56 , 27804.7008, 186426.72 , 24988.8528,
85141.44 , 25467.84 , 117119.5632, 39267.36 , 31838.5296,
19276.704 , 46620. , 123876. , 30529.44 , 96596.64 ,
17262.72 , 57116.16 , 23655.7872, 26107.2 , 47685.6 ,
44382.7728, 38841.12 , 15397.92 , 29303.4672, 42943.1472,
38378.6496, 33992.64 , 40705.92 ] )

```

```

[437]: price_mode=df.Price.median()
price_mode

```

[437]: 52693.92

```
[438]: df.Price.fillna(price_mode,inplace=True)
```

C:\Users\DELL\AppData\Local\Temp\ipykernel_17368\1251914371.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

```
df.Price.fillna(price_mode,inplace=True)
```

```
[439]: df.isnull().sum()
```

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

```
[440]: df
```

```
[440]:
```

	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]

```
[441]: df.isnull().sum()
```

```
[441]: Company      0
      TypeName    0
      Inches      0
      ScreenResolution  0
      Cpu          0
      Ram          0
      Memory      0
      Gpu          0
      OpSys        0
```

```
Weight          0
Price           0
dtype: int64
```

6 OUTLIER DETECTION AND REMOVAL USING IQR

```
[442]: df
```

```
[442]:      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

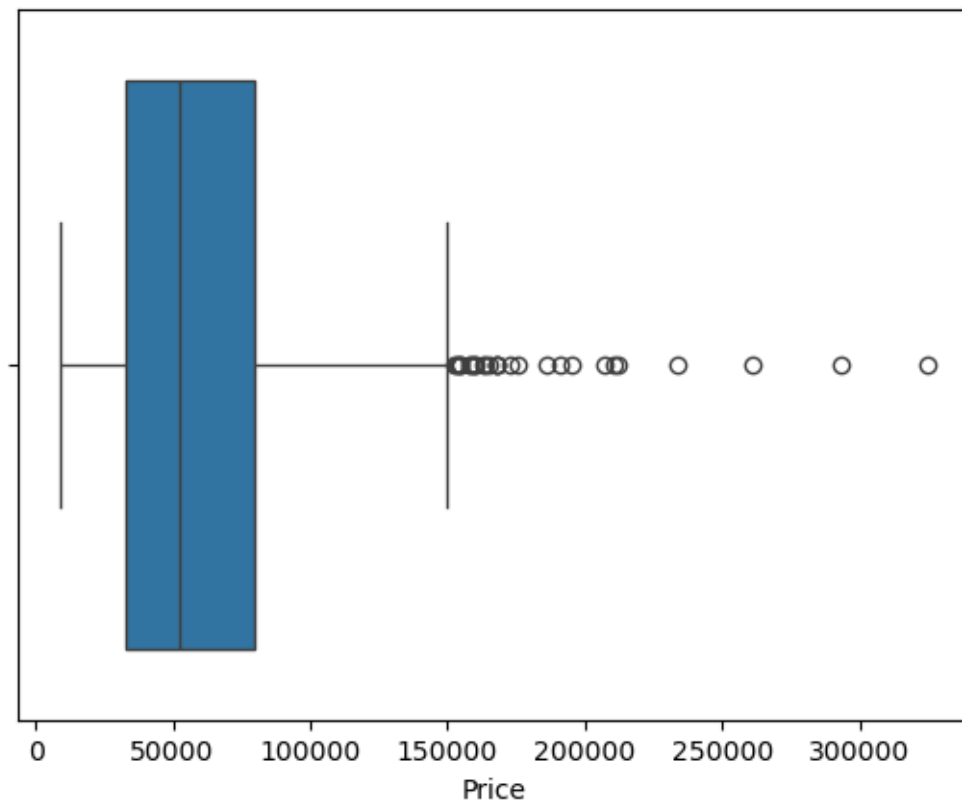
[1246 rows x 11 columns]

```
[443]: df.shape
```

```
[443]: (1246, 11)
```

```
[444]: sns.boxplot(x=df['Price'])
```

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



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

```
[445]:          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]
```

```
[446]: q1=df1.quantile(0.25)
q1
```

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

```
[447]: q3=df1.quantile(0.75)
q3
```

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

```
[448]: IQR=q3-q1
IQR
```

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

```
[449]: e=((df1<q1-1.5*IQR)|(df1>q3+1.5*IQR))
e
```

```
[449]:          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]
```

```
[479]: e.sum()
```

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

```
[451]: filter=df[(~e).any(axis=1)]
```

```
[452]: filter
```

```
[452]:      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]

```
[453]: filter.shape
```

```
[453]: (1218, 11)
```

7 UNIVARIATE ANALYSIS

```
[480]: filter
```

```
[480]:
```

	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]

[]:

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

[481]:

	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

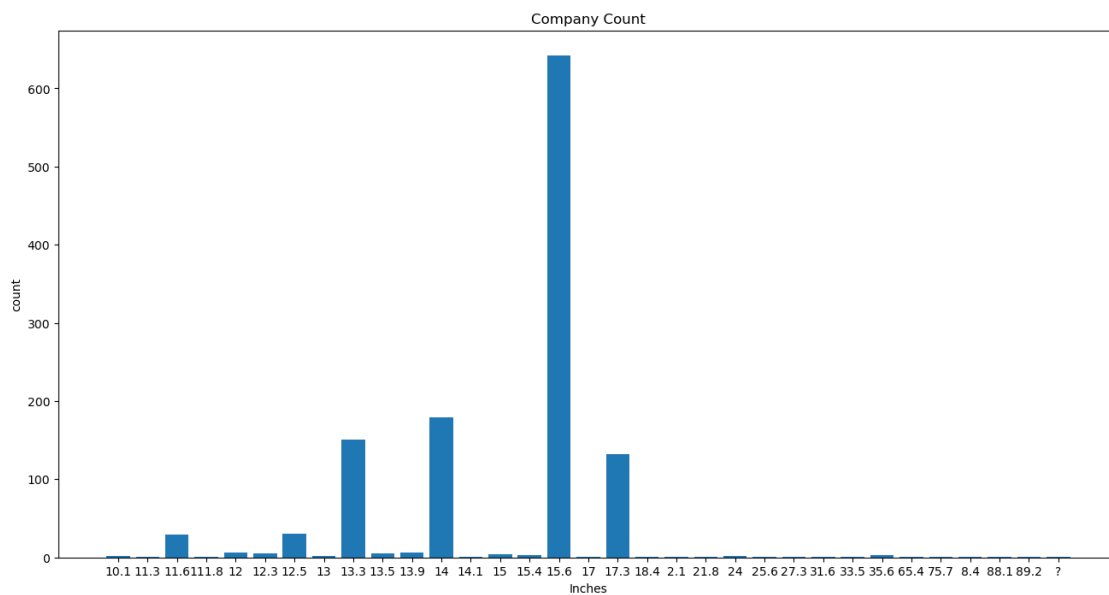
```
[482]: f=filter.groupby(['Inches']).size().reset_index(name='count').
        ↪rename(columns={'Inches':'Inches'})
        f
```

```
[482]:
```

	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

```
[483]: plt.figure(figsize=(16,8))
plt.bar(f['Inches'],f['count'])
plt.show()
```



```
[484]: f['count%']=(f["count"]/sum(f["count"]))*100
f
```

	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

[485]: filter

[485]:

	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]

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

```
[486]:
```

Price	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory \
9270.7200	1	1	1	1	1	1	1
10442.8800	1	1	1	1	1	1	1
10602.7200	2	2	2	2	2	2	2
10810.5120	1	1	1	1	1	1	1

11135.5200	2	2	2		2	2	2	2
...
146946.2400	1	1	1		1	1	1	1
147832.2864	1	1	1		1	1	1	1
149130.7200	4	4	4		4	4	4	4
149184.0000	1	1	1		1	1	1	1
149916.6000	1	1	1		1	1	1	1

	Gpu	OpSys	Weight
Price			
9270.7200	1	1	1
10442.8800	1	1	1
10602.7200	2	2	2
10810.5120	1	1	1
11135.5200	2	2	2
...
146946.2400	1	1	1
147832.2864	1	1	1
149130.7200	4	4	4
149184.0000	1	1	1
149916.6000	1	1	1

[752 rows x 10 columns]

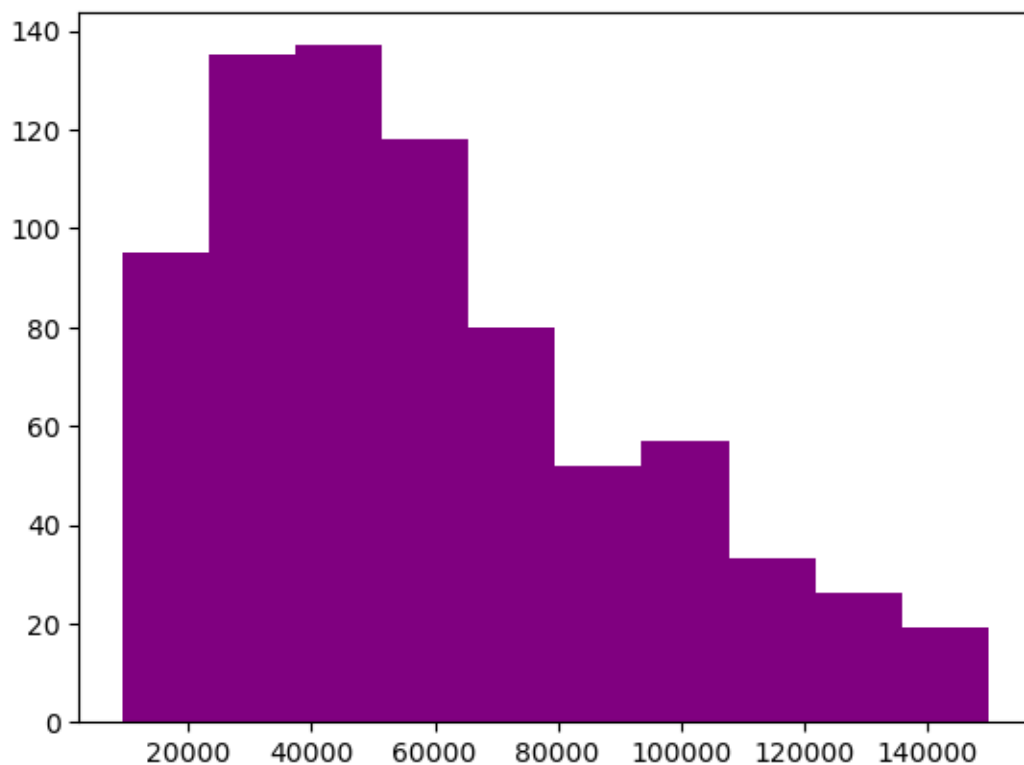
```
[487]: g=filter.groupby(['Price']).size().reset_index(name='count').
        ↪rename(columns={'Price':'price'})
g
```

```
[487]:
```

	price	count
0	9270.7200	1
1	10442.8800	1
2	10602.7200	2
3	10810.5120	1
4	11135.5200	2
..
747	146946.2400	1
748	147832.2864	1
749	149130.7200	4
750	149184.0000	1
751	149916.6000	1

[752 rows x 2 columns]

```
[488]: plt.hist(g['price'],bins=10,color='purple')
plt.show()
```



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

```
[489]:      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

[1218 rows x 1 columns]
```

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

```
[490]: 149916.6
```

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

```
[491]: 9270.72
```

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

```
[492]: 57665.397186206894
```

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

```
[493]: 52054.56
```

```
[494]: np.percentile(newdf,25)
```

```
[494]: 31914.72
```

```
[495]: np.percentile(newdf,50)
```

```
[495]: 52054.56
```

```
[496]: np.percentile(newdf,75)
```

```
[496]: 77238.684000000001
```

```
[497]: filter.groupby(['Ram']).count()
```

```
[497]:      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

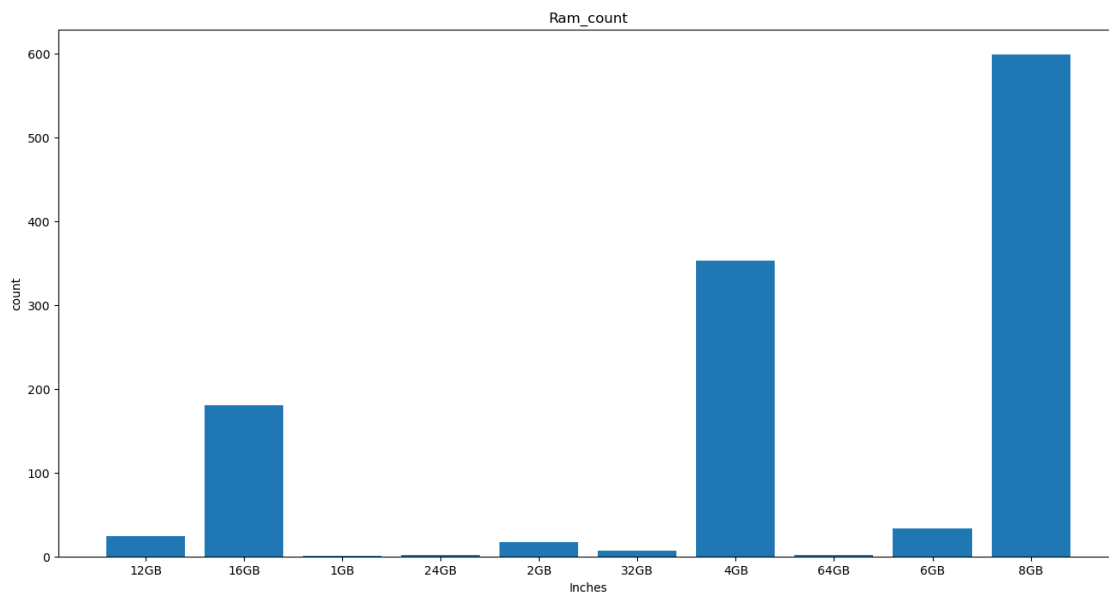
      Weight  Price
Ram
12GB      24      24
16GB     180     180
1GB        1        1
24GB        2        2
2GB       17       17
32GB        7        7
```

4GB	353	353
64GB	2	2
6GB	33	33
8GB	599	599

```
[498]: h=filter.groupby(['Ram']).size().reset_index(name='count').
      ↪rename(columns={'Ram':'Ram'})
      h
```

```
[498]:   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
```

```
[499]: plt.figure(figsize=(16,8))
      plt.bar(h['Ram'],h['count'])
      plt.title('Ram_count')
      plt.xlabel('Inches')
      plt.ylabel('count')
      plt.show()
```



[]:

8 BIVARIATE ANALYSIS

9 # Correlation

10 Bivariate Analysis

10.1 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.

10.2 2) Also we can not find correlation

[500]: `filter`

```
[500]:
```

	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

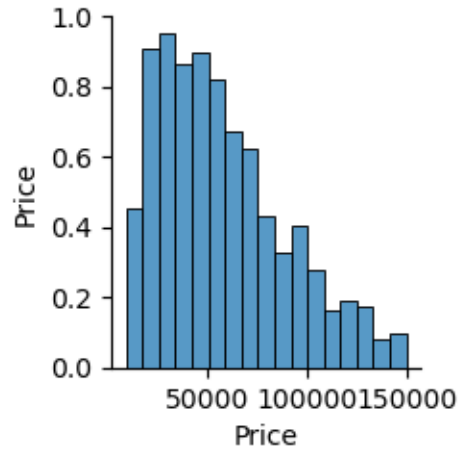
		Gpu	OpSys	Weight	Price
0	Intel Iris Plus Graphics	640	macOS	1.37kg	71378.6832
1	Intel HD Graphics	6000	macOS	1.34kg	47895.5232
2	Intel HD Graphics	620	No OS	1.86kg	30636.0000
3	AMD Radeon Pro	455	macOS	1.83kg	135195.3360
4	Intel Iris Plus Graphics	650	macOS	1.37kg	96095.8080
...	
1270	Intel HD Graphics	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]

```
[501]: sns.pairplot(filter) # Correlation is for Numerical - Numerical in bivariant
      ↪analysis
```

C:\Users\DELL\anaconda3\Lib\site-packages\seaborn\axisgrid.py:123: UserWarning:
The figure layout has changed to tight
self._figure.tight_layout(*args, **kwargs)

```
[501]: <seaborn.axisgrid.PairGrid at 0x1ac952e2250>
```



11 # Chi square

```
[502]: filter.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1218 entries, 0 to 1300
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Company               1218 non-null   object
1   TypeName              1218 non-null   object
2   Inches                1218 non-null   object
3   ScreenResolution      1218 non-null   object
4   Cpu                   1218 non-null   object
5   Ram                   1218 non-null   object
6   Memory                1218 non-null   object
7   Gpu                   1218 non-null   object
8   OpSys                 1218 non-null   object
9   Weight                1218 non-null   object
10  Price                 1218 non-null   float64
dtypes: float64(1), object(10)
memory usage: 114.2+ KB
```

```
[503]: filter.columns
```

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

```
[504]: filter['Inches'].unique()
```

```
[504]: 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)
```

```
[505]: filter['Weight'].unique()
```

```
[505]: 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)
```

```
[506]: from sklearn.preprocessing import LabelEncoder
label=LabelEncoder()
filter['Inches']=label.fit_transform(filter['Inches'])
filter['ScreenResolution']=label.fit_transform(filter['ScreenResolution'])
filter['Memory']=label.fit_transform(filter['Memory'])
filter['Weight']=label.fit_transform(filter['Weight'])
filter
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_17368\1318132180.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

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

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_17368\1318132180.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['ScreenResolution']=label.fit_transform(filter['ScreenResolution'])
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_17368\1318132180.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['Memory']=label.fit_transform(filter['Memory'])
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_17368\1318132180.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['Weight']=label.fit_transform(filter['Weight'])
```

```
[506]:
```

	Company	TypeName	Inches	ScreenResolution	\
0	Apple	Ultrabook	8	23	
1	Apple	Ultrabook	8	1	
2	HP	Notebook	15	8	
3	Apple	Ultrabook	14	25	
4	Apple	Ultrabook	8	23	
...	
1270	Lenovo	2 in 1 Convertible	11	13	
1271	Lenovo	2 in 1 Convertible	8	19	
1272	Lenovo	Notebook	11	0	
1273	HP	Notebook	15	0	
1300	Lenovo	Notebook	27	0	

```
Cpu  Ram  Memory \
```

0		Intel Core i5 2.3GHz	8GB	4
1		Intel Core i5 1.8GHz	8GB	2
2		Intel Core i5 7200U 2.5GHz	8GB	14
3		Intel Core i7 2.7GHz	16GB	27
4		Intel Core i5 3.1GHz	8GB	14
...	
1270		Intel Core i7 6500U 2.5GHz	4GB	4
1271		Intel Core i7 6500U 2.5GHz	16GB	27
1272		Intel Celeron Dual Core N3050 1.6GHz	2GB	32
1273		Intel Core i7 6500U 2.5GHz	6GB	10
1300		Intel Celeron Dual Core N3050 1.6GHz	2GB	32

		Gpu	OpSys	Weight	Price
0	Intel Iris Plus Graphics 640		macOS	40	71378.6832
1	Intel HD Graphics 6000		macOS	37	47895.5232
2	Intel HD Graphics 620		No OS	75	30636.0000
3	AMD Radeon Pro 455		macOS	72	135195.3360
4	Intel Iris Plus Graphics 650		macOS	40	96095.8080
...
1270	Intel HD Graphics 520	Windows 10		79	33992.6400
1271	Intel HD Graphics 520	Windows 10		43	79866.7200
1272	Intel HD Graphics	Windows 10		57	12201.1200
1273	AMD Radeon R5 M330	Windows 10		110	40705.9200
1300	Intel HD Graphics	Windows 10		57	12201.1200

[1218 rows x 11 columns]

```
[507]: newdf2=filter[['Inches','ScreenResolution','Memory','Weight']]
newdf2
```

```
[507]:
```

	Inches	ScreenResolution	Memory	Weight
0	8	23	4	40
1	8	1	2	37
2	15	8	14	75
3	14	25	27	72
4	8	23	14	40
...
1270	11	13	4	79
1271	8	19	27	43
1272	11	0	32	57
1273	15	0	10	110
1300	27	0	32	57

[1218 rows x 4 columns]

```
[508]: x=newdf2[['Inches','ScreenResolution','Memory']]
x
```

```
[508]:
```

	Inches	ScreenResolution	Memory
0	8	23	4
1	8	1	2
2	15	8	14
3	14	25	27
4	8	23	14
...
1270	11	13	4
1271	8	19	27
1272	11	0	32
1273	15	0	10
1300	27	0	32

[1218 rows x 3 columns]

```
[509]: y=newdf2[['Weight']]
y
```

```
[509]:
```

	Weight
0	40
1	37
2	75
3	72
4	40
...	...
1270	79
1271	43
1272	57
1273	110
1300	57

[1218 rows x 1 columns]

```
[510]: from sklearn.feature_selection import chi2
values=chi2(x,y)
values
```

```
[510]: (array([1016.64755285, 2893.78268552, 992.54186422]),
array([1.07765455e-112, 0.00000000e+000, 1.92541515e-108]))
```

```
[511]: newdf2.groupby(['Inches','Weight']).count()
```

```
[511]:
```

	Inches	Weight	ScreenResolution	Memory
0	2		2	2
1	23		1	1
2	11		1	1

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

[312 rows x 2 columns]

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

12 # Anova

```
[512]: filter
```

```
[512]:
```

	Company		TypeName	Inches	ScreenResolution	\
0	Apple		Ultrabook	8	23	
1	Apple		Ultrabook	8	1	
2	HP		Notebook	15	8	
3	Apple		Ultrabook	14	25	
4	Apple		Ultrabook	8	23	
...	
1270	Lenovo	2 in 1	Convertible	11	13	
1271	Lenovo	2 in 1	Convertible	8	19	
1272	Lenovo		Notebook	11	0	
1273	HP		Notebook	15	0	
1300	Lenovo		Notebook	27	0	

			Cpu	Ram	Memory	\
0			Intel Core i5 2.3GHz	8GB	4	
1			Intel Core i5 1.8GHz	8GB	2	
2		Intel	Core i5 7200U 2.5GHz	8GB	14	
3			Intel Core i7 2.7GHz	16GB	27	
4			Intel Core i5 3.1GHz	8GB	14	
...			
1270		Intel	Core i7 6500U 2.5GHz	4GB	4	

1271	Intel Core i7 6500U	2.5GHz	16GB	27
1272	Intel Celeron Dual Core N3050	1.6GHz	2GB	32
1273	Intel Core i7 6500U	2.5GHz	6GB	10
1300	Intel Celeron Dual Core N3050	1.6GHz	2GB	32

		Gpu	OpSys	Weight	Price
0	Intel Iris Plus Graphics	640	macOS	40	71378.6832
1	Intel HD Graphics	6000	macOS	37	47895.5232
2	Intel HD Graphics	620	No OS	75	30636.0000
3	AMD Radeon Pro	455	macOS	72	135195.3360
4	Intel Iris Plus Graphics	650	macOS	40	96095.8080
...
1270	Intel HD Graphics	520	Windows 10	79	33992.6400
1271	Intel HD Graphics	520	Windows 10	43	79866.7200
1272	Intel HD Graphics		Windows 10	57	12201.1200
1273	AMD Radeon R5 M330		Windows 10	110	40705.9200
1300	Intel HD Graphics		Windows 10	57	12201.1200

[1218 rows x 11 columns]

```
[513]: from sklearn.preprocessing import LabelEncoder
label=LabelEncoder()
filter['Ram']=label.fit_transform(filter['Ram'])
filter['Cpu']=label.fit_transform(filter['Cpu'])
filter['Gpu']=label.fit_transform(filter['Gpu'])
filter['Price']=label.fit_transform(filter['Price'])
filter
```

C:\Users\DELL\AppData\Local\Temp\ipykernel_17368\560081952.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

```
filter['Ram']=label.fit_transform(filter['Ram'])
```

C:\Users\DELL\AppData\Local\Temp\ipykernel_17368\560081952.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['Cpu']=label.fit_transform(filter['Cpu'])
```

C:\Users\DELL\AppData\Local\Temp\ipykernel_17368\560081952.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['Gpu']=label.fit_transform(filter['Gpu'])
```

C:\Users\DELL\AppData\Local\Temp\ipykernel_17368\560081952.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['Price']=label.fit_transform(filter['Price'])
```

```
[513]:
```

	Company		TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	\
0	Apple		Ultrabook	8	23	65	9	4	
1	Apple		Ultrabook	8	1	63	9	2	
2	HP		Notebook	15	8	74	9	14	
3	Apple		Ultrabook	14	25	85	1	27	
4	Apple		Ultrabook	8	23	67	9	14	
...
1270	Lenovo	2 in 1	Convertible	11	13	88	6	4	
1271	Lenovo	2 in 1	Convertible	8	19	88	1	27	
1272	Lenovo		Notebook	11	0	34	4	32	
1273	HP		Notebook	15	0	88	8	10	
1300	Lenovo		Notebook	27	0	34	4	32	

	Gpu	OpSys	Weight	Price
0	56	macOS	40	517
1	49	macOS	37	338
2	51	No OS	75	164
3	8	macOS	72	732
4	57	macOS	40	627
...
1270	45	Windows 10	79	196
1271	45	Windows 10	43	567
1272	38	Windows 10	57	7
1273	19	Windows 10	110	261
1300	38	Windows 10	57	7

[1218 rows x 11 columns]

```
[514]: newdf3=filter[['Ram','Cpu','Gpu','Price']]
newdf3
```

```
[514]:
```

	Ram	Cpu	Gpu	Price
0	9	65	56	517

1	9	63	49	338
2	9	74	51	164
3	1	85	8	732
4	9	67	57	627
...
1270	6	88	45	196
1271	1	88	45	567
1272	4	34	38	7
1273	8	88	19	261
1300	4	34	38	7

[1218 rows x 4 columns]

```
[515]: X=newdf3[['Ram','Cpu','Gpu']]
X
```

```
[515]:
```

	Ram	Cpu	Gpu
0	9	65	56
1	9	63	49
2	9	74	51
3	1	85	8
4	9	67	57
...
1270	6	88	45
1271	1	88	45
1272	4	34	38
1273	8	88	19
1300	4	34	38

[1218 rows x 3 columns]

```
[516]: Y=newdf3[['Price']]
Y
```

```
[516]:
```

	Price
0	517
1	338
2	164
3	732
4	627
...	...
1270	196
1271	567
1272	7
1273	261
1300	7

[1218 rows x 1 columns]

```
[517]: from sklearn.feature_selection import f_classif, SelectKBest
P_values=f_classif(X,Y)
P_values
```

```
C:\Users\DELL\anaconda3\Lib\site-packages\sklearn\utils\validation.py:1184:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
```

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

```
[518]: import pandas as pd
P_value=pd.Series(P_values[1])
P_value.index=X.columns
P_value
```

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

```
[ ]:
```

```
[ ]:
```