

```
In [12]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [13]: datapad_read_csv('laptop_prices.csv')
data.head()
```

| | Company | Product | TypeName | Inches | Ram | OS | Weight | Price_euros | Screen | ScreenW | ... | RetinaDisplay | CPU_company | CPU_freq | CPU_model | PrimaryStorage | SecondaryStorage | PrimaryStorageType | SecondaryStorageType |
|---|---------|-------------|-----------|--------|-----|-------|--------|-------------|----------|---------|-----|---------------|-------------|----------|---------------|----------------|------------------|--------------------|----------------------|
| 0 | Apple | MacBook Pro | Ultrabook | 13.3 | 8 | macOS | 1.37 | 1339.69 | Standard | 2560 | ... | Yes | Intel | 2.3 | Core i5 | 128 | 0 | SSD | |
| 1 | Apple | MacBook Air | Ultrabook | 13.3 | 8 | macOS | 1.34 | 895.94 | Standard | 1440 | ... | No | Intel | 1.8 | Core i5 | 128 | 0 | Flash Storage | |
| 2 | HP | 250 G6 | Notebook | 15.6 | 8 | No OS | 1.86 | 575.00 | Full HD | 1920 | ... | No | Intel | 2.5 | Core i5 7200U | 256 | 0 | SSD | |
| 3 | Apple | MacBook Pro | Ultrabook | 15.4 | 16 | macOS | 1.83 | 2537.45 | Standard | 2880 | ... | Yes | Intel | 2.7 | Core i7 | 512 | 0 | SSD | |
| 4 | Apple | MacBook Pro | Ultrabook | 13.3 | 8 | macOS | 1.37 | 1803.60 | Standard | 2560 | ... | Yes | Intel | 3.1 | Core i6 | 256 | 0 | SSD | |

```
5 rows x 23 columns
```

```
In [14]: data.shape
```

```
Out[14]: (1275, 23)
```

```
In [15]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1275 entries, 0 to 1274
Data columns (total 23 columns):
 #   Column                Non-Null Count  Dtype
---  --
 0   Company               1275 non-null   object
 1   Product              1275 non-null   object
 2   TypeName              1275 non-null   object
 3   Inches               1275 non-null   float64
 4   Ram                  1275 non-null   int64
 5   OS                   1275 non-null   object
 6   Weight               1275 non-null   float64
 7   Price_euros          1275 non-null   float64
 8   Screen               1275 non-null   object
 9   ScreenW              1275 non-null   int64
10   ScreenH              1275 non-null   int64
11   Touchscreen          1275 non-null   object
12   IPSpanel             1275 non-null   object
13   RetinaDisplay        1275 non-null   object
14   CPU_company          1275 non-null   object
15   CPU_freq             1275 non-null   float64
16   CPU_model            1275 non-null   object
17   PrimaryStorage       1275 non-null   int64
18   SecondaryStorage     1275 non-null   int64
19   PrimaryStorageType   1275 non-null   object
20   SecondaryStorageType 1275 non-null   object
21   GPU_company          1275 non-null   object
22   GPU_model            1275 non-null   object
dtypes: float64(6), int64(5), object(14)
memory usage: 225.2+ MB
```

```
In [16]: data.info().sum()
```

```
Out[16]: Company      0
Product      0
TypeName      0
Inches      0
Ram          0
OS           0
Weight       0
Price_euros  0
Screen       0
ScreenW      0
ScreenH      0
Touchscreen  0
IPSpanel     0
RetinaDisplay 0
CPU_company  0
CPU_freq     0
CPU_model    0
PrimaryStorage 0
SecondaryStorage 0
PrimaryStorageType 0
SecondaryStorageType 0
GPU_company  0
GPU_model    0
dtype: int64
```

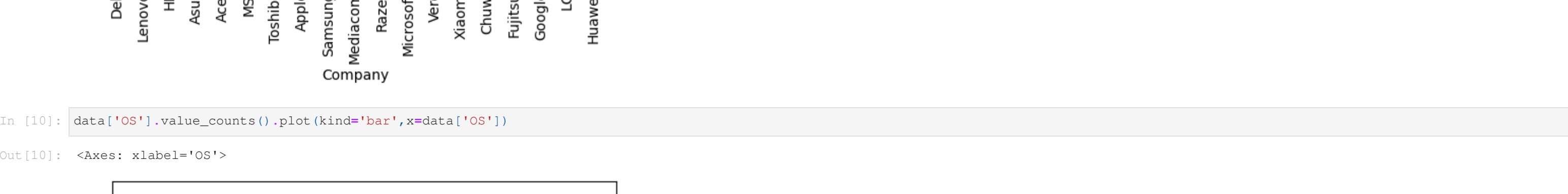
```
In [17]: data.describe()
```

```
Out[17]:
```

| | Inches | Ram | Weight | Price_euros | ScreenW | ScreenH | CPU_freq | PrimaryStorage | SecondaryStorage |
|-------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|----------------|------------------|
| count | 1275.000000 | 1275.000000 | 1275.000000 | 1275.000000 | 1275.000000 | 1275.000000 | 1275.000000 | 1275.000000 | 1275.000000 |
| mean | 15.022802 | 8.440784 | 2.040525 | 1134.969059 | 1900.043922 | 1073.904314 | 2.302980 | 444.517647 | 176.069020 |
| std | 1.429470 | 5.097899 | 0.668196 | 700.792504 | 493.348196 | 283.883940 | 0.503846 | 365.937726 | 415.960605 |
| min | 10.100000 | 2.000000 | 0.690000 | 174.000000 | 1366.000000 | 768.000000 | 0.800000 | 8.000000 | 0.000000 |
| 25% | 14.000000 | 4.000000 | 1.500000 | 609.000000 | 1920.000000 | 1080.000000 | 2.000000 | 256.000000 | 0.000000 |
| 50% | 15.600000 | 8.000000 | 2.040000 | 969.000000 | 1920.000000 | 1080.000000 | 2.500000 | 256.000000 | 0.000000 |
| 75% | 15.600000 | 8.000000 | 2.310000 | 1498.500000 | 1920.000000 | 1080.000000 | 2.700000 | 512.000000 | 0.000000 |
| max | 16.400000 | 64.000000 | 4.700000 | 6089.000000 | 3840.000000 | 2160.000000 | 3.600000 | 2048.000000 | 2048.000000 |

```
In [18]: data['Company'].value_counts().plot(kind='bar')
```

```
Out[18]: <Axes: xlabel='Company'>
```



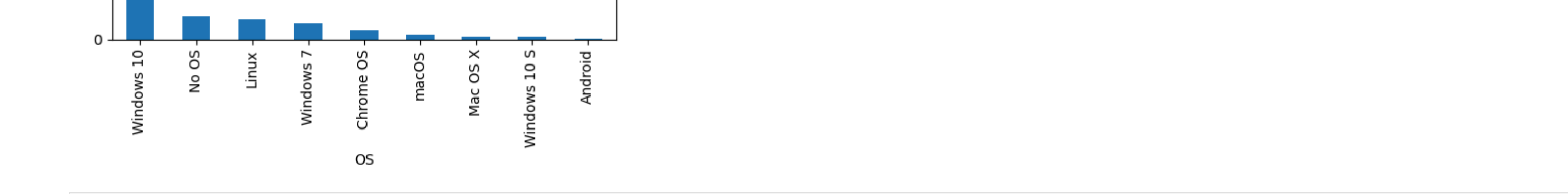
```
In [19]: data['OS'].value_counts().plot(kind='bar')
```

```
Out[19]: <Axes: xlabel='OS'>
```



```
In [20]: data['Ram'].value_counts().plot(kind='bar')
```

```
Out[20]: <Axes: xlabel='Ram'>
```



```
In [21]: data['Inches'].value_counts().plot(kind='bar')
```

```
Out[21]: <Axes: xlabel='Inches'>
```



```
In [22]: data['Screen'].value_counts().plot(kind='bar')
```

```
Out[22]: <Axes: xlabel='Screen'>
```



```
In [24]: data['SecondaryStorageType'].value_counts().plot(kind='bar')
```

```
Out[24]: <Axes: xlabel='SecondaryStorageType'>
```



```
In [26]: data['Touchscreen'].value_counts().plot(kind='pie', autopct='%2f%%', title='Touchscreen')
```

```
Out[26]: <Axes: title='center': 'Touchscreen', ylabel='count'>
```



```
In [44]: data['CPU_company'].value_counts().plot(kind='pie', autopct='%2f%%', title='CPU_company')
```

```
Out[44]: <Axes: title='center': 'CPU_company', ylabel='count'>
```



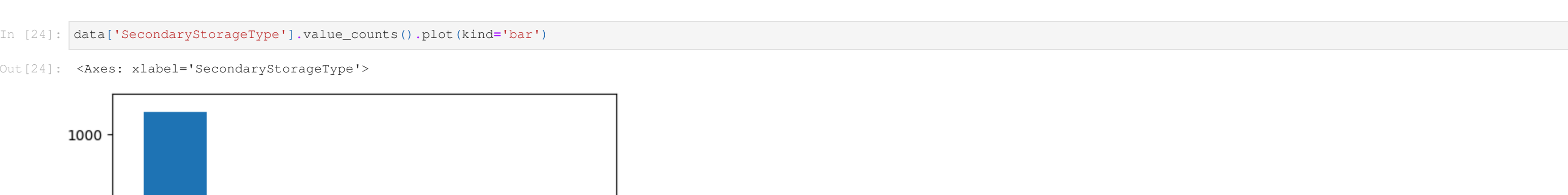
```
In [15]: data['GPU_company'].value_counts().plot(kind='pie', autopct='%2f%%', title='GPU_company')
```

```
Out[15]: <Axes: title='center': 'GPU_company', ylabel='count'>
```



```
In [49]: data['IPSpanel'].value_counts().plot(kind='pie', autopct='%2f%%')
```

```
Out[49]: <Axes: ylabel='count'>
```



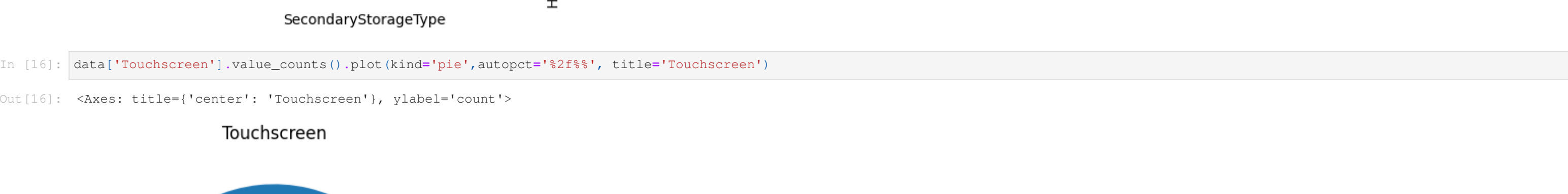
```
In [20]: data['Inches'].value_counts().plot(kind='bar')
```

```
Out[20]: <Axes: xlabel='Inches'>
```

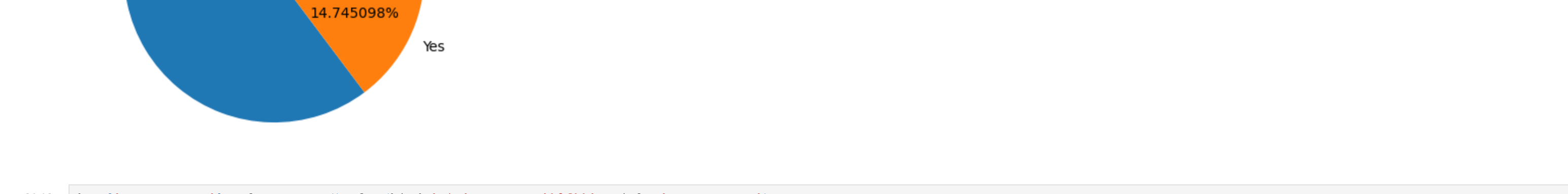


```
In [22]: data['PrimaryStorageType'].value_counts().plot(kind='pie', autopct='%2f%%')
```

```
Out[22]: <Axes: ylabel='count'>
```



```
In [27]: plt.figure(figsize=(10,8))
sns.boxplot(x=data['OS'], y=data['Price_euros'])
plt.title('OS VS CPU_frequency')
plt.show()
```



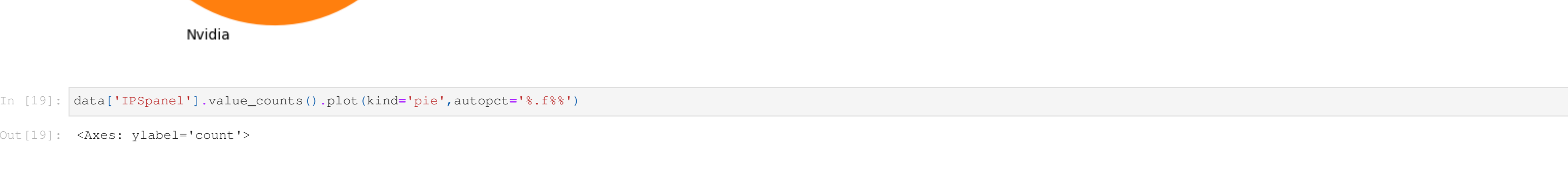
```
In [27]: plt.figure(figsize=(10,8))
sns.boxplot(x=data['Touchscreen'], y=data['Price_euros'],
hue=data['Screen'])
plt.show()
```



```
In [28]: plt.figure(figsize=(10,8))
sns.boxplot(x=data['OS'], y=data['Price_euros'], hue=data['Touchscreen'])
plt.show()
```



```
In [29]: plt.figure(figsize=(10,8))
sns.boxplot(x=data['PrimaryStorage'], y=data['Price_euros'], hue=data['SecondaryStorage'])
plt.show()
```



```
In [30]: plt.figure(figsize=(10,8))
sns.boxplot(x=data['RetinaDisplay'], y=data['Price_euros'], hue=data['PrimaryStorage'])
plt.show()
```



```
In [31]: sns.scatterplot(data=data, x=data['CPU_freq'], y=data['Ram'], hue=data['OS'])
```

```
Out[31]: <Axes: xlabel='CPU_freq', ylabel='Ram'>
```

