```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
data=pd.read_csv("/content/archive (1).zip")
```

```python
data.head()
```

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep
0	1	1043	1	1.8	1	14	0	5	0.1
1	2	841	1	0.5	1	4	1	61	0.8
2	3	1807	1	2.8	0	1	0	27	0.9
3	4	1546	0	0.5	1	18	1	25	0.5
4	5	1434	0	1.4	0	11	1	49	0.5

5 rows × 21 columns

```python
data.tail()
```

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep
1995	794	1	0.5	1	0	1	2	0.8
1996	1965	1	2.6	1	0	0	39	0.2
1997	1911	0	0.9	1	1	1	36	0.7
1998	1512	0	0.9	0	4	1	46	0.1
1999	510	1	2.0	1	5	1	45	0.9

5 rows × 21 columns

```python
print(data.shape)
```
```
(2000, 21)
```

```python
data.info()
```
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   battery_power  2000 non-null   int64
 1   blue           2000 non-null   int64
 2   clock_speed    2000 non-null   float64
 3   dual_sim       2000 non-null   int64
 4   fc             2000 non-null   int64
```

```
 5    four_g           2000 non-null    int64
 6    int_memory       2000 non-null    int64
 7    m_dep            2000 non-null    float64
 8    mobile_wt        2000 non-null    int64
 9    n_cores          2000 non-null    int64
10    pc               2000 non-null    int64
11    px_height        2000 non-null    int64
12    px_width         2000 non-null    int64
13    ram              2000 non-null    int64
14    sc_h             2000 non-null    int64
15    sc_w             2000 non-null    int64
16    talk_time        2000 non-null    int64
17    three_g          2000 non-null    int64
18    touch_screen     2000 non-null    int64
19    wifi             2000 non-null    int64
20    price_range      2000 non-null    int64
dtypes: float64(2), int64(19)
memory usage: 328.2 KB
```

data.describe()

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_m
count	2000.000000	2000.0000	2000.000000	2000.000000	2000.000000	2000.000000	2000.
mean	1238.518500	0.4950	1.522250	0.509500	4.309500	0.521500	32.
std	439.418206	0.5001	0.816004	0.500035	4.341444	0.499662	18
min	501.000000	0.0000	0.500000	0.000000	0.000000	0.000000	2.
25%	851.750000	0.0000	0.700000	0.000000	1.000000	0.000000	16.
50%	1226.000000	0.0000	1.500000	1.000000	3.000000	1.000000	32.
75%	1615.250000	1.0000	2.200000	1.000000	7.000000	1.000000	48.
max	1998.000000	1.0000	3.000000	1.000000	19.000000	1.000000	64.

8 rows × 21 columns

data.isnull().sum()

```
                            0

         battery_pwwer      0
                  blue      0
           clvck_speed      0
              dual_sim      0
                    fc      0
                fvur_g      0
            int_memvry      0
                 m_dep      0
             mvbile_wt      0
               n_cvres      0
                    pc      0
             px_height      0
              px_width      0
                   ram      0
                  sc_h      0
                  sc_w      0
             talk_time      0
               three_g      0
          tvuch_screen      0
                  wifi      0
           price_range      0
```

dtype: int64

data.dropna()

3

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep
0	842	0	2.2	0	1	0	7	0.6
1	1021	1	0.5	1	0	1	53	0.7
2	563	1	0.5	1	2	1	41	0.9
3	615	1	2.5	0	0	0	10	0.8
4	1821	1	1.2	0	13	1	44	0.6
...	...	...	...	...	...	...	...	...
1995	794	1	0.5	1	0	1	2	0.8
1996	1965	1	2.6	1	0	0	39	0.2
1997	1911	0	0.9	1	1	1	36	0.7
1998	1512	0	0.9	0	4	1	46	0.1
1999	510	1	2.0	1	5	1	45	0.9

2000 rows × 21 columns

```
data.count()
```

	0
battery_pwwer	2000
blue	2000
clvck_speed	2000
dual_sim	2000
fc	2000
fvur_g	2000
int_memvry	2000
m_dep	2000
mvbile_wt	2000
n_cvres	2000
pc	2000
px_height	2000
px_width	2000
ram	2000
sc_h	2000
sc_w	2000
talk_time	2000
three_g	2000
tvuch_screen	2000
wifi	2000
price_range	2000

dtype: int64

```
data=data.drop_duplicates()
```

```
#missing value
data=data.T.drop_duplicates().T
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   battery_power  2000 non-null   float64
 1   blue           2000 non-null   float64
```

```
 2   clock_speed    2000 non-null   float64
 3   dual_sim       2000 non-null   float64
 4   fc             2000 non-null   float64
 5   four_g         2000 non-null   float64
 6   int_memory     2000 non-null   float64
 7   m_dep          2000 non-null   float64
 8   mobile_wt      2000 non-null   float64
 9   n_cores        2000 non-null   float64
10   pc             2000 non-null   float64
11   px_height      2000 non-null   float64
12   px_width       2000 non-null   float64
13   ram            2000 non-null   float64
14   sc_h           2000 non-null   float64
15   sc_w           2000 non-null   float64
16   talk_time      2000 non-null   float64
17   three_g        2000 non-null   float64
18   touch_screen   2000 non-null   float64
19   wifi           2000 non-null   float64
20   price_range    2000 non-null   float64
dtypes: float64(21)
memory usage: 328.2 KB
```

```python
#In the following columns : bluetooth, dual_sim, four_g, three_g, touch_screen, wifi [cha
columns=['bluetooth','dual_sim', 'four_g', 'three_g', 'touch_screen', 'wifi']
data.rename(columns={0:'No',1:'Yes'},inplace=True)
```

```python
data = data.rename(columns={'blue': 'Bluetooth', 'dual_sim': 'Dual_SIM', 'four_g': 'Four_
                    'touch_screen': 'Touch_Screen', 'wifi': 'WiFi', 'px_height': 'Pi:
                    'px_width': 'Pixel_Resolution_Width', 'sc_h': 'Screen_Height', ':
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 21 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   battery_power            2000 non-null   float64
 1   Bluetooth                2000 non-null   float64
 2   clock_speed              2000 non-null   float64
 3   Dual_SIM                 2000 non-null   float64
 4   fc                       2000 non-null   float64
 5   Four_G                   2000 non-null   float64
 6   int_memory               2000 non-null   float64
 7   m_dep                    2000 non-null   float64
 8   mobile_wt                2000 non-null   float64
 9   n_cores                  2000 non-null   float64
10   pc                       2000 non-null   float64
11   Pixel_Resolution_Height  2000 non-null   float64
12   Pixel_Resolution_Width   2000 non-null   float64
13   ram                      2000 non-null   float64
14   Screen_Height            2000 non-null   float64
15   Screen_Width             2000 non-null   float64
16   talk_time                2000 non-null   float64
17   Three_G                  2000 non-null   float64
18   Touch_Screen             2000 non-null   float64
19   WiFi                     2000 non-null   float64
20   price_range              2000 non-null   float64
dtypes: float64(21)
memory usage: 328.2 KB
```

```python
data.head()
```

	battery_power	Bluetooth	clock_speed	Dual_SIM	fc	Four_G	int_memory	m_d
0	842.0	0.0	2.2	0.0	1.0	0.0	7.0	
1	1021.0	1.0	0.5	1.0	0.0	1.0	53.0	
2	563.0	1.0	0.5	1.0	2.0	1.0	41.0	
3	615.0	1.0	2.5	0.0	0.0	0.0	10.0	
4	1821.0	1.0	1.2	0.0	13.0	1.0	44.0	

5 rows × 21 columns

```python
data['Bluetooth'] = data['Bluetooth'].replace({0: 'No', 1: 'Yes'})
data['Dual_SIM'] = data['Dual_SIM'].replace({0: 'No', 1: 'Yes'})
data['Four_G'] = data['Four_G'].replace({0: 'No', 1: 'Yes'})
data['Three_G'] = data['Three_G'].replace({0: 'No', 1: 'Yes'})
data['Touch_Screen'] = data['Touch_Screen'].replace({0: 'No', 1: 'Yes'})
data['WiFi'] = data['WiFi'].replace({0: 'No', 1: 'Yes'})
```

```python
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 21 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   battery_power           2000 non-null   float64
 1   Bluetooth               2000 non-null   object
 2   clock_speed             2000 non-null   float64
 3   Dual_SIM                2000 non-null   object
 4   fc                      2000 non-null   float64
 5   Four_G                  2000 non-null   object
 6   int_memory              2000 non-null   float64
 7   m_dep                   2000 non-null   float64
 8   mobile_wt               2000 non-null   float64
 9   n_cores                 2000 non-null   float64
 10  pc                      2000 non-null   float64
 11  Pixel_Resolution_Height 2000 non-null   float64
 12  Pixel_Resolution_Width  2000 non-null   float64
 13  ram                     2000 non-null   float64
 14  Screen_Height           2000 non-null   float64
 15  Screen_Width            2000 non-null   float64
 16  talk_time               2000 non-null   float64
 17  Three_G                 2000 non-null   object
 18  Touch_Screen            2000 non-null   object
 19  WiFi                    2000 non-null   object
 20  price_range             2000 non-null   float64
dtypes: float64(15), object(6)
memory usage: 328.2+ KB
```

```python
data.head()
```

	battery_power	Bluetooth	clock_speed	Dual_SIM	fc	Four_G	int_memory	m_d
0	842.0	Nv	2.2	Nv	1.0	Nv	7.0	
1	1021.0	Yes	0.5	Yes	0.0	Yes	53.0	
2	563.0	Yes	0.5	Yes	2.0	Yes	41.0	
3	615.0	Yes	2.5	Nv	0.0	Nv	10.0	
4	1821.0	Yes	1.2	Nv	13.0	Yes	44.0	

5 rvws × 21 cvlumns

```python
#data['price_range'] = data['price_range'].replace({0: 'Low Cost', 1: 'Medium Cost', 2:
```

```python
data.head()
```

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mo
0	842	0	2.2	0	1	0	7	0.6	
1	1021	1	0.5	1	0	1	53	0.7	
2	563	1	0.5	1	2	1	41	0.9	
3	615	1	2.5	0	0	0	10	0.8	
4	1821	1	1.2	0	13	1	44	0.6	

5 rvws × 21 cvlumns

```python
corr = data.corr()
print(corr['price_range'].sort_values(ascending=False))
```

```
price_range       1.000000
ram               0.917046
battery_power     0.200723
px_width          0.165818
px_height         0.148858
int_memory        0.044435
sc_w              0.038711
pc                0.033599
three_g           0.023611
sc_h              0.022986
fc                0.021998
talk_time         0.021859
blue              0.020573
wifi              0.018785
dual_sim          0.017444
four_g            0.014772
n_cores           0.004399
m_dep             0.000853
clock_speed      -0.006606
```
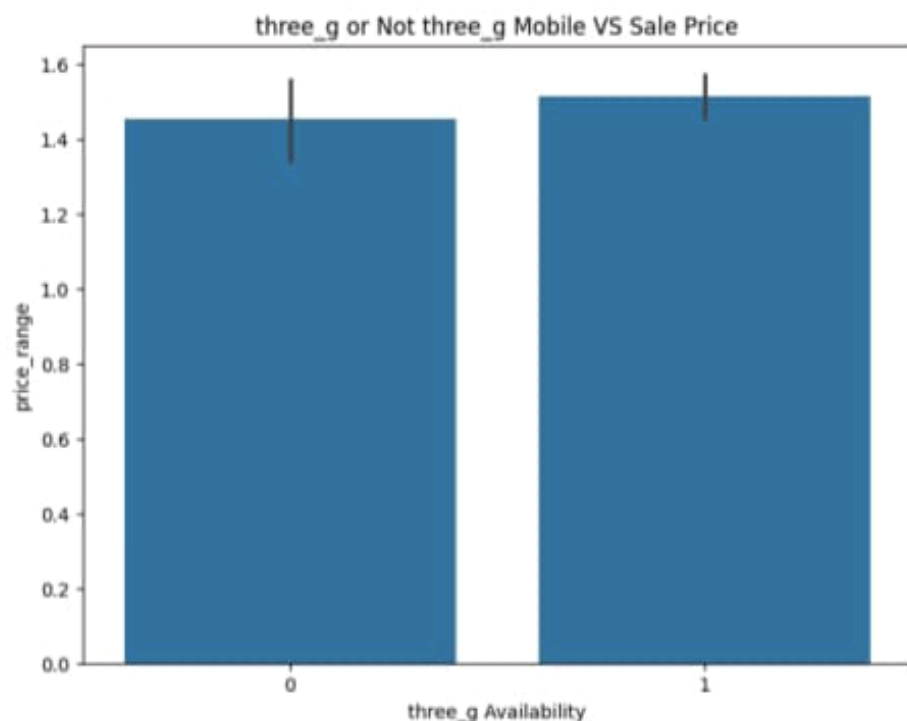
```python
plt.figure(figsize=(8, 6))
sns.barplot(x='three_g', y='price_ra
plt.title('three_g or Not three_g Mo
plt.xlabel('three_g Availability')
plt.ylabel('price_range')
plt.show()
```


three_g or Not three_g Mobile VS Sale Price

```python
plt.figure(figsize=(10, 8))
sns.countplot(x='price_range', hue='
plt.title('Supporting Bluetooth or N
plt.xlabel('price_range')
plt.ylabel('Count')
```

```
plt.legend(title='Bluetooth Availabi
plt.show()
```


Supporting Bluetooth or NOT vs Price
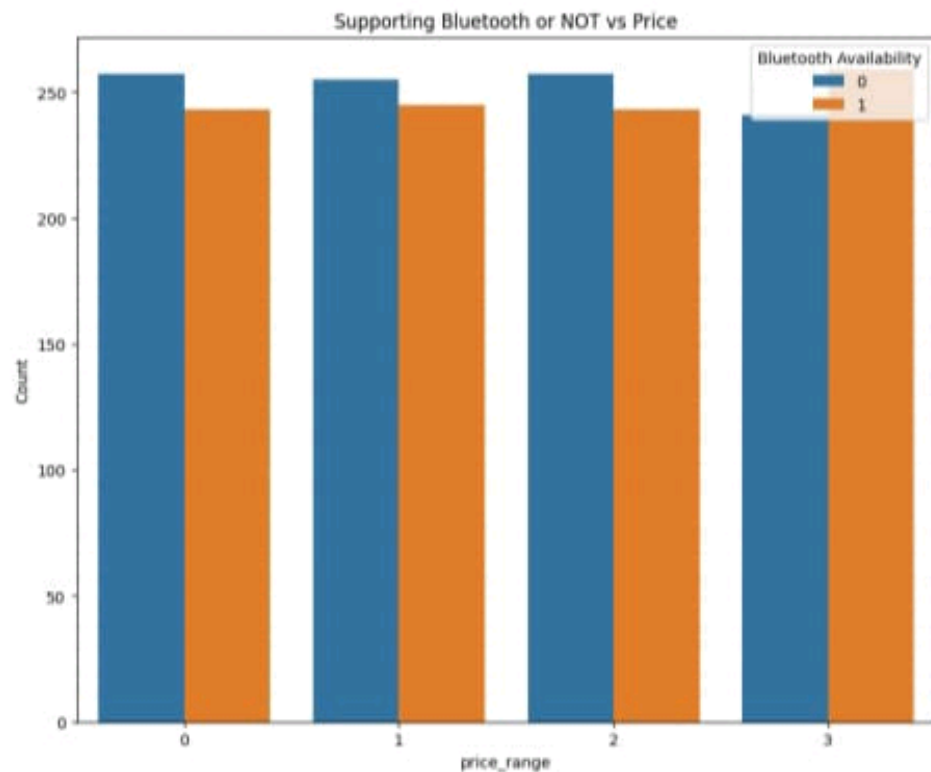
```
plt.figure(figsize=(10, 8))
sns.scatterplot(x='px_width', y='px_
plt.title('Pixel Resolution Height a
plt.xlabel('Pixel Resolution Height'
plt.ylabel('Pixel Resolution Width')
plt.legend(title='Price Range')
plt.show()
```

Pixel Resolution Height and Pixel Resolution Width with Price Range
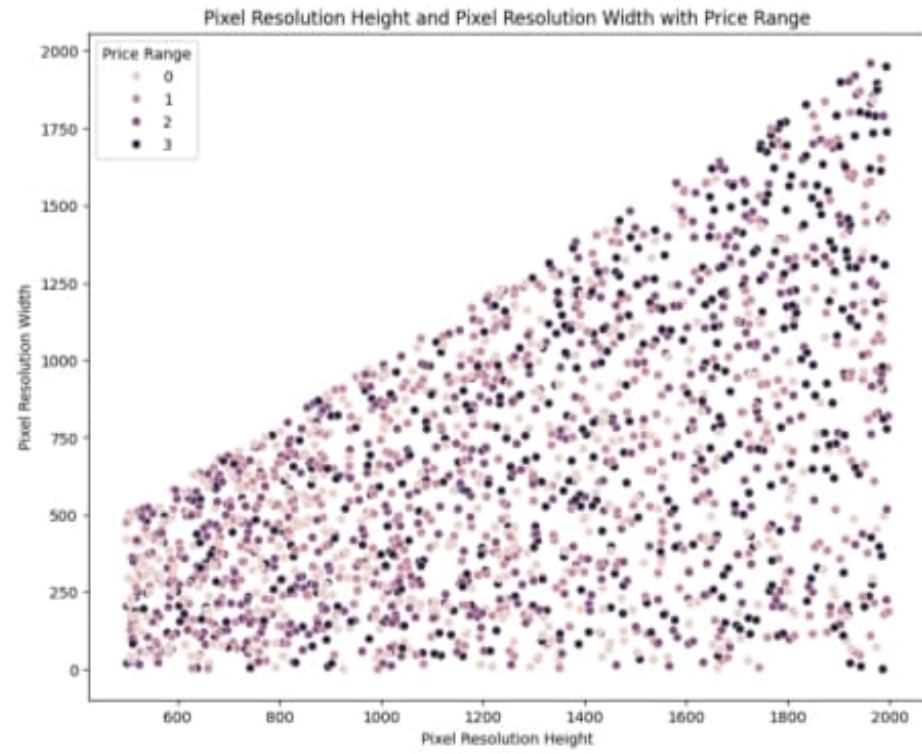
```
plt.figure(figsize=(10, 8))
sns.scatterplot(x='sc_h', y='sc_w',
plt.title('Screen Height and Screen
plt.xlabel('Screen Height')
plt.ylabel('Screen Width')
plt.legend(title='Price Range')
plt.show()
```

Screen Height and Screen Width with Price Range