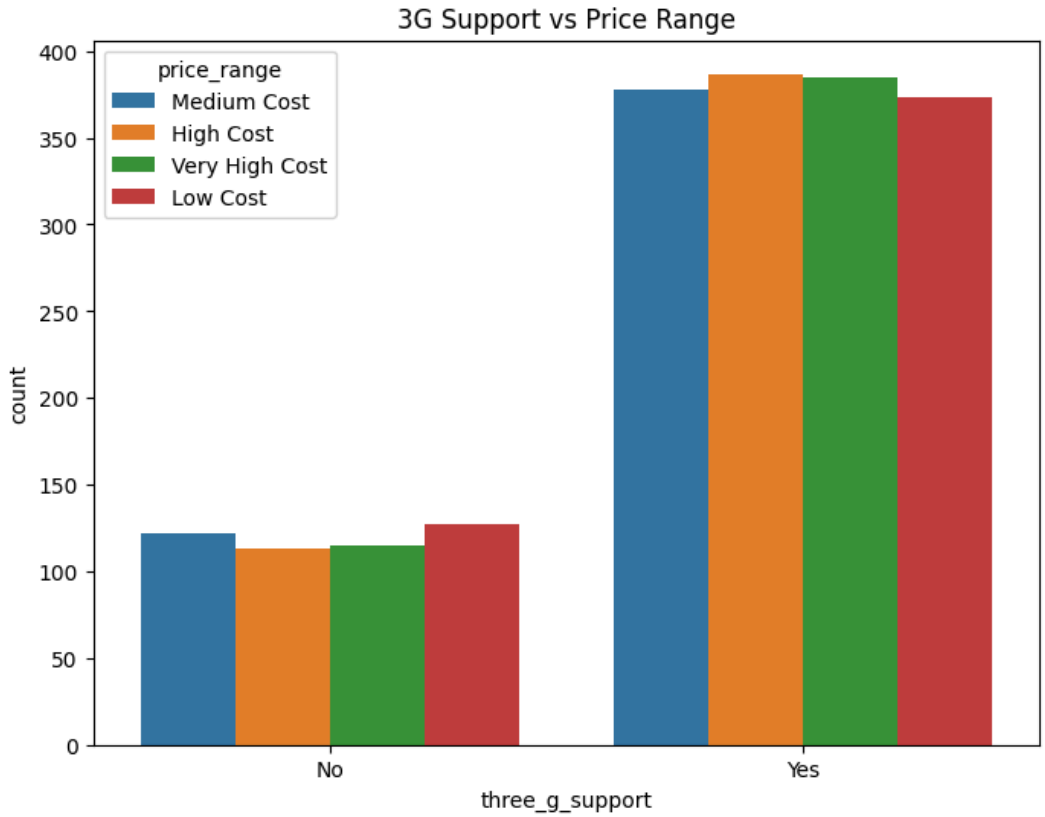
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns


#LOAD THE DATASET

data=pd.read_csv("/content/train.csv")


#1st 5 rows

print("first 5 rows:")
print(data.head())


#last 5 rows

print("last 5 rows")
print(data.tail())


#SHAPE

print("shape of the dataset:")
print(data.shape)


#DISCRIPTION

print("discription of the dataset:")
print(data.describe())


#INFO

print("info of the dataset:")
print(data.info())


#CLEANING THE DATA FOR MISSING AND NULL

print("checking for missing values:")
print(data.isnull().sum())


#changing column name

data.rename(columns={
    'blue': 'bluetooth',
    'dual_sim': 'dual_sim_support',
    'four_g': 'four_g_support',
    'three_g': 'three_g_support',
    'touch_screen': 'touch_screen_support',
    'wifi': 'wifi_support'
}, inplace=True)


#Changing the categorical data for specified columns

data['bluetooth'].replace({0: 'No', 1: 'Yes'}, inplace=True)
data['dual_sim_support'].replace({0: 'No', 1: 'Yes'}, inplace=True)
data['four_g_support'].replace({0: 'No', 1: 'Yes'}, inplace=True)
data['three_g_support'].replace({0: 'No', 1: 'Yes'}, inplace=True)
data['touch_screen_support'].replace({0: 'No', 1: 'Yes'}, inplace=True)
data['wifi_support'].replace({0: 'No', 1: 'Yes'}, inplace=True)


#Changing price_range values

data['price_range'].replace({0: 'Low Cost', 1: 'Medium Cost', 2: 'High Cost', 3: 'Very High Cost'}, inplace=True)
```

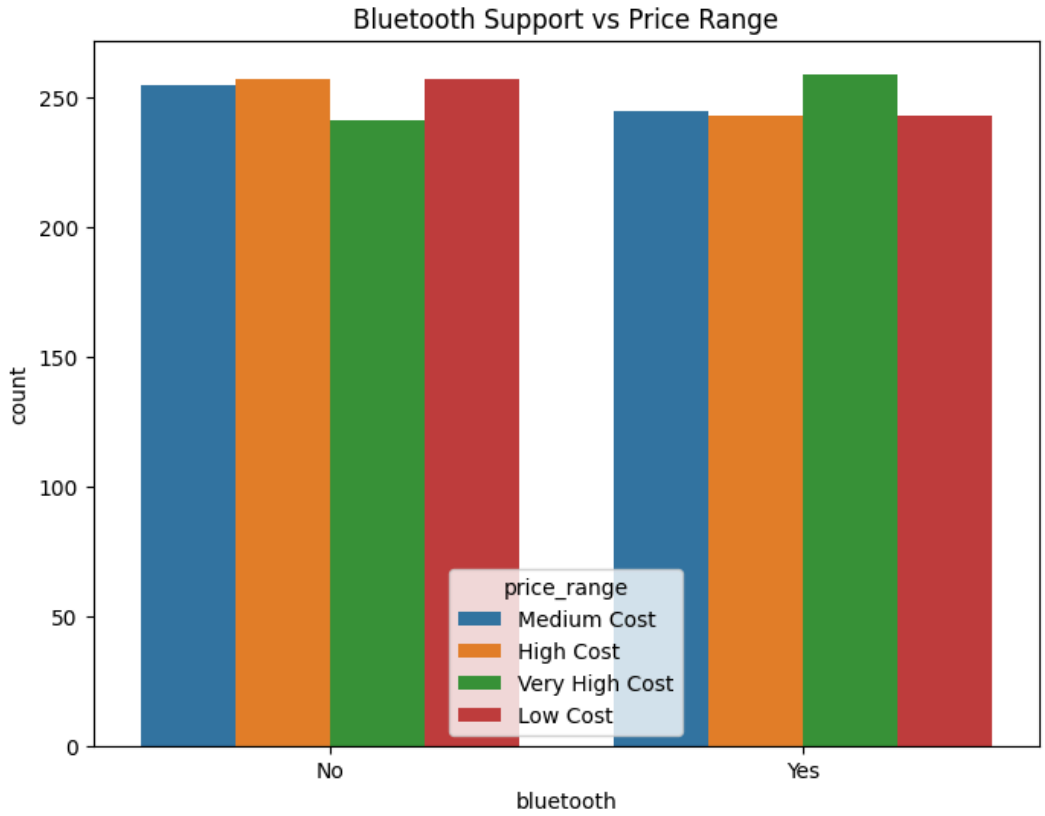
```
# Verify changes

print("Data after transformations:")
print(data.head())



#Visualization and Analysis


#3G or Not 3G Mobile VS Sale Price using bar plot

plt.figure(figsize=(8, 6))
sns.countplot(x='three_g_support', hue='price_range', data=data)
plt.title('3G Support vs Price Range')
plt.show()


#Count Plot for Supporting Bluetooth or Not vs Price

plt.figure(figsize=(8, 6))
sns.countplot(x='bluetooth', hue='price_range', data=data)
plt.title('Bluetooth Support vs Price Range')
plt.show()


#Scatterplot showing relation between pixel resolution height/width and price range

plt.figure(figsize=(8, 6))
sns.scatterplot(x='px_height', y='px_width', hue='price_range', data=data)
plt.title('Pixel Resolution Height vs Width with Price Range')
plt.show()


#Scatterplot showing relation between screen height/width and price range

plt.figure(figsize=(8, 6))
sns.scatterplot(x='sc_h', y='sc_w', hue='price_range', data=data)
plt.title('Screen Height vs Screen Width with Price Range')
plt.show()
```

```
first 5 rows:
   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

   mobile_wt  n_cores  ...  px_height  px_width   ram  sc_h  sc_w  talk_time  \
0        188        2  ...         20       756  2549     9     7         19
1        136        3  ...        905      1988  2631    17     3          7
2        145        5  ...       1263      1716  2603    11     2          9
3        131        6  ...       1216      1786  2769    16     8         11
4        141        2  ...       1208      1212  1411     8     2         15

   three_g  touch_screen  wifi  price_range
0        0             0     1            1
1        1             1     0            2
2        1             1     0            2
3        1             0     0            2
4        1             1     0            1

[5 rows x 21 columns]
last 5 rows
      battery_power  blue  clock_speed  dual_sim  fc  four_g  int_memory  \
1995            794     1          0.5         1   0       1           2
1996           1965     1          2.6         1   0       0          39
1997           1911     0          0.9         1   1       1          36
1998           1512     0          0.9         0   4       1          46
1999            510     1          2.0         1   5       1          45

      m_dep  mobile_wt  n_cores  ...  px_height  px_width   ram  sc_h  sc_w  \
1995    0.8        106        6  ...       1222      1890   668    13     4
1996    0.2        187        4  ...        915      1965  2032    11    10
1997    0.7        108        8  ...        868      1632  3057     9     1
1998    0.1        145        5  ...        336       670   869    18    10
1999    0.9        168        6  ...        483       754  3919    19     4

      talk_time  three_g  touch_screen  wifi  price_range
1995         19        1             1     0            0
1996         16        1             1     1            2
1997          5        1             1     0            3
1998         19        1             1     1            0
1999          2        1             1     1            3

[5 rows x 21 columns]
shape of the dataset:
(2000, 21)
discription of the dataset:
       battery_power       blue  clock_speed     dual_sim           fc  \
count    2000.000000  2000.0000  2000.000000  2000.000000  2000.000000
mean     1238.518500     0.4950     1.522250     0.509500     4.309500
std       439.418206     0.5001     0.816004     0.500035     4.341444
min       501.000000     0.0000     0.500000     0.000000     0.000000
25%       851.750000     0.0000     0.700000     0.000000     1.000000
50%      1226.000000     0.0000     1.500000     1.000000     3.000000
75%      1615.250000     1.0000     2.200000     1.000000     7.000000
max      1998.000000     1.0000     3.000000     1.000000    19.000000

            four_g   int_memory        m_dep    mobile_wt      n_cores  ...  \
count  2000.000000  2000.000000  2000.000000  2000.000000  2000.000000  ...
mean      0.521500    32.046500     0.501750   140.249000     4.520500  ...
std       0.499662    18.145715     0.288416    35.399655     2.287837  ...
min       0.000000     2.000000     0.100000    80.000000     1.000000  ...
25%       0.000000    16.000000     0.200000   109.000000     3.000000  ...
50%       1.000000    32.000000     0.500000   141.000000     4.000000  ...
75%       1.000000    48.000000     0.800000   170.000000     7.000000  ...
max       1.000000    64.000000     1.000000   200.000000     8.000000  ...

         px_height     px_width          ram         sc_h         sc_w  \
count  2000.000000  2000.000000  2000.000000  2000.000000  2000.000000
mean    645.108000  1251.515500  2124.213000    12.306500     5.767000
std     443.780811   432.199447  1084.732044     4.213245     4.356398
min       0.000000   500.000000   256.000000     5.000000     0.000000
25%     282.750000   874.750000  1207.500000     9.000000     2.000000
50%     564.000000  1247.000000  2146.500000    12.000000     5.000000
75%     947.250000  1633.000000  3064.500000    16.000000     9.000000
max    1960.000000  1998.000000  3998.000000    19.000000    18.000000
```

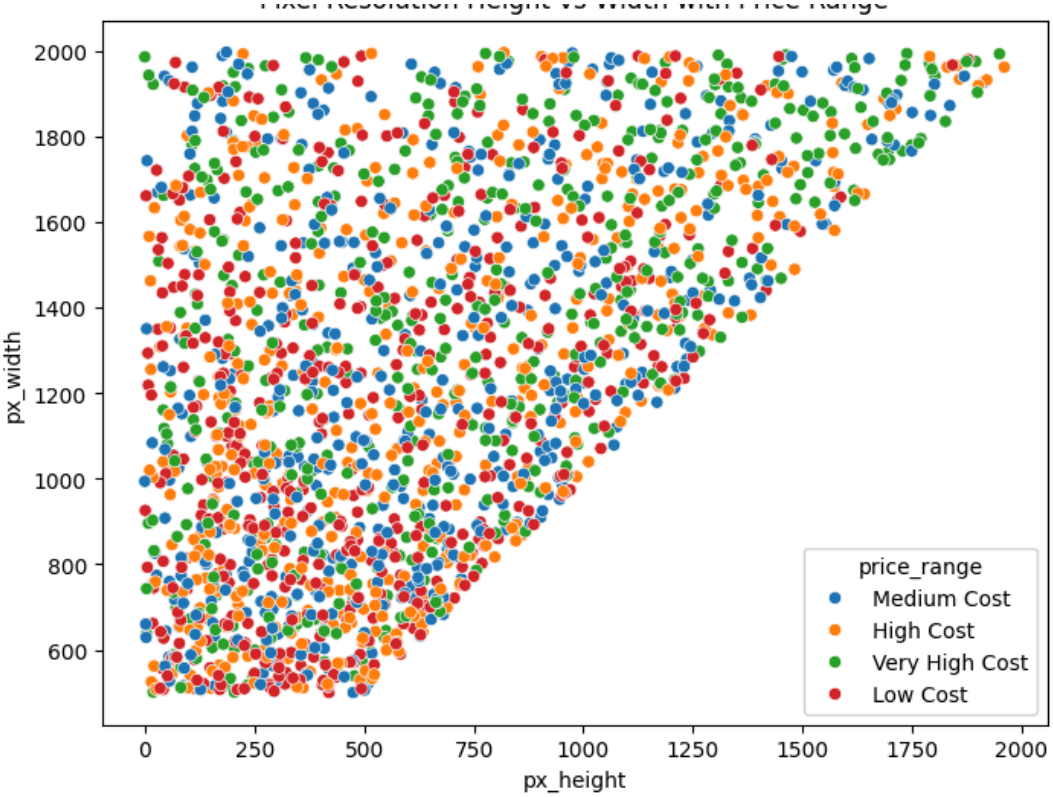
```
         talk_time      three_g  touch_screen         wifi  price_range
count  2000.000000  2000.000000   2000.000000  2000.000000  2000.000000
mean     11.011000     0.761500      0.503000     0.507000     1.500000
std       5.463955     0.426273      0.500116     0.500076     1.118314
min       2.000000     0.000000      0.000000     0.000000     0.000000
25%       6.000000     1.000000      0.000000     0.000000     0.750000
50%      11.000000     1.000000      1.000000     1.000000     1.500000
75%      16.000000     1.000000      1.000000     1.000000     2.250000
max      20.000000     1.000000      1.000000     1.000000     3.000000

[8 rows x 21 columns]
info of the dataset:
<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
None
checking for missing values:
battery_power    0
blue             0
clock_speed      0
dual_sim         0
fc               0
four_g           0
int_memory       0
m_dep            0
mobile_wt        0
n_cores          0
pc               0
px_height        0
px_width         0
ram              0
sc_h             0
sc_w             0
talk_time        0
three_g          0
touch_screen     0
wifi             0
price_range      0
dtype: int64
Data after transformations:
   battery_power bluetooth  clock_speed dual_sim_support  fc four_g_support  \
0            842        No          2.2               No   1             No
1           1021       Yes          0.5              Yes   0            Yes
2            563       Yes          0.5              Yes   2            Yes
3            615       Yes          2.5               No   0             No
4           1821       Yes          1.2               No  13            Yes

   int_memory  m_dep  mobile_wt  n_cores  ...  px_height  px_width   ram  \
0           7    0.6        188        2  ...         20       756  2549
1          53    0.7        136        3  ...        905      1988  2631
2          41    0.9        145        5  ...       1263      1716  2603
3          10    0.8        131        6  ...       1216      1786  2769
4          44    0.6        141        2  ...       1208      1212  1411
```

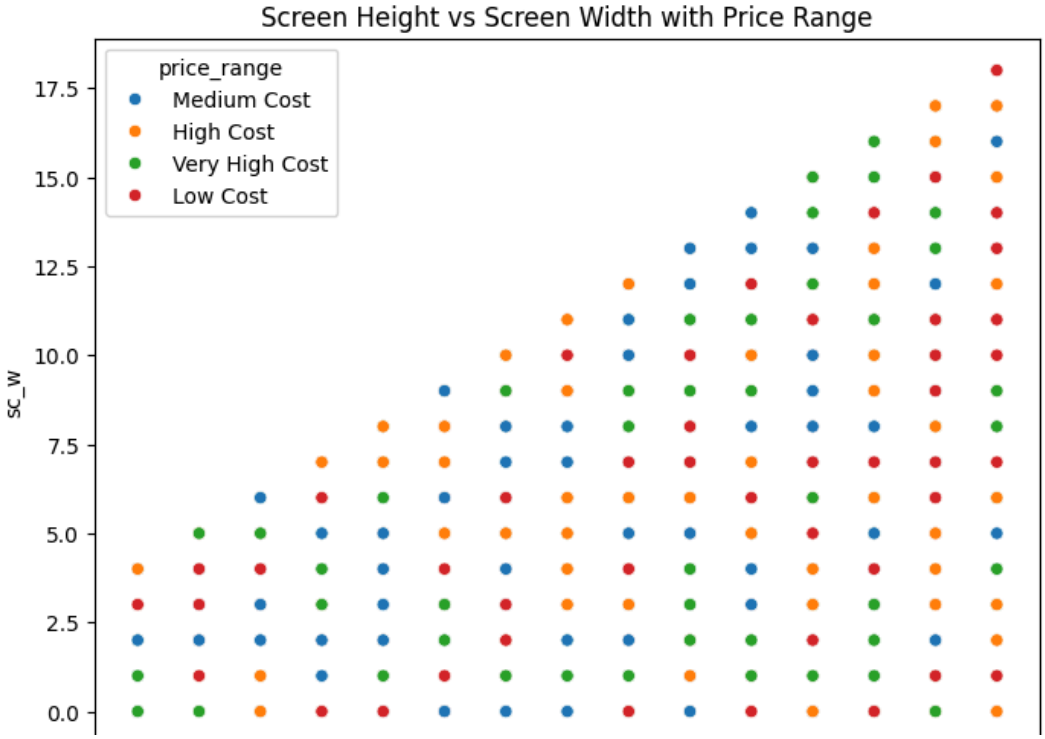
```
   sc_h  sc_w  talk_time  three_g_support touch_screen_support wifi_support  \
0     9     7         19               No                  No          Yes
1    17     3          7              Yes                 Yes           No
2    11     2          9              Yes                 Yes           No
3    16     8         11              Yes                  No           No
4     8     2         15              Yes                 Yes           No

   price_range
0  Medium Cost
1    High Cost
2    High Cost
3    High Cost
4  Medium Cost

[5 rows x 21 columns]
```





Pixel Resolution Height vs Width with Price Range


px_width
px_height
price_range
Medium Cost
High Cost
Very High Cost
Low Cost


Screen Height vs Screen Width with Price Range
sc_w
price_range
Medium Cost
High Cost
Very High Cost
Low Cost