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

train_df=pd.read_csv('/content/train.csv')
train_df
```

| | battery_power | blue | clock_speed | dual_sim | fc | four_g | int_memory | m_dep | mobile_wt | n_cores | ... | px_height | px_width | ram |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 842 | 0 | 2.2 | 0 | 1 | 0 | 7 | 0.6 | 188 | 2 | ... | 20 | 756 | 2549 |
| **1** | 1021 | 1 | 0.5 | 1 | 0 | 1 | 53 | 0.7 | 136 | 3 | ... | 905 | 1988 | 2631 |
| **2** | 563 | 1 | 0.5 | 1 | 2 | 1 | 41 | 0.9 | 145 | 5 | ... | 1263 | 1716 | 2603 |
| **3** | 615 | 1 | 2.5 | 0 | 0 | 0 | 10 | 0.8 | 131 | 6 | ... | 1216 | 1786 | 2769 |
| **4** | 1821 | 1 | 1.2 | 0 | 13 | 1 | 44 | 0.6 | 141 | 2 | ... | 1208 | 1212 | 1411 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **1995** | 794 | 1 | 0.5 | 1 | 0 | 1 | 2 | 0.8 | 106 | 6 | ... | 1222 | 1890 | 668 |
| **1996** | 1965 | 1 | 2.6 | 1 | 0 | 0 | 39 | 0.2 | 187 | 4 | ... | 915 | 1965 | 2032 |
| **1997** | 1911 | 0 | 0.9 | 1 | 1 | 1 | 36 | 0.7 | 108 | 8 | ... | 868 | 1632 | 3057 |
| **1998** | 1512 | 0 | 0.9 | 0 | 4 | 1 | 46 | 0.1 | 145 | 5 | ... | 336 | 670 | 869 |
| **1999** | 510 | 1 | 2.0 | 1 | 5 | 1 | 45 | 0.9 | 168 | 6 | ... | 483 | 754 | 3919 |

2000 rows × 21 columns

```
#first five rows
train_df.head()
```

| | battery_power | blue | clock_speed | dual_sim | fc | four_g | int_memory | m_dep | mobile_wt | n_cores | ... | px_height | px_width | ram | sc_ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 842 | 0 | 2.2 | 0 | 1 | 0 | 7 | 0.6 | 188 | 2 | ... | 20 | 756 | 2549 | |
| **1** | 1021 | 1 | 0.5 | 1 | 0 | 1 | 53 | 0.7 | 136 | 3 | ... | 905 | 1988 | 2631 | 1 |
| **2** | 563 | 1 | 0.5 | 1 | 2 | 1 | 41 | 0.9 | 145 | 5 | ... | 1263 | 1716 | 2603 | 1 |
| **3** | 615 | 1 | 2.5 | 0 | 0 | 0 | 10 | 0.8 | 131 | 6 | ... | 1216 | 1786 | 2769 | 1 |
| **4** | 1821 | 1 | 1.2 | 0 | 13 | 1 | 44 | 0.6 | 141 | 2 | ... | 1208 | 1212 | 1411 | |

5 rows × 21 columns

```
#last five rows
train_df.tail()
```

| | battery_power | blue | clock_speed | dual_sim | fc | four_g | int_memory | m_dep | mobile_wt | n_cores | ... | px_height | px_width | ram |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1995** | 794 | 1 | 0.5 | 1 | 0 | 1 | 2 | 0.8 | 106 | 6 | ... | 1222 | 1890 | 668 |
| **1996** | 1965 | 1 | 2.6 | 1 | 0 | 0 | 39 | 0.2 | 187 | 4 | ... | 915 | 1965 | 2032 |
| **1997** | 1911 | 0 | 0.9 | 1 | 1 | 1 | 36 | 0.7 | 108 | 8 | ... | 868 | 1632 | 3057 |
| **1998** | 1512 | 0 | 0.9 | 0 | 4 | 1 | 46 | 0.1 | 145 | 5 | ... | 336 | 670 | 869 |
| **1999** | 510 | 1 | 2.0 | 1 | 5 | 1 | 45 | 0.9 | 168 | 6 | ... | 483 | 754 | 3919 |

5 rows × 21 columns

```
#shape of the dataset
train_df.shape
```

```
(2000, 21)
```

```
#description of the dataset
train_df.describe()
```

| | battery_power | blue | clock_speed | dual_sim | fc | four_g | int_memory | m_dep | mobile_wt | n_cores |
|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 2000.000000 | 2000.0000 | 2000.000000 | 2000.000000 | 2000.000000 | 2000.000000 | 2000.000000 | 2000.000000 | 2000.000000 | 2000.000000 |
| **mean** | 1238.518500 | 0.4950 | 1.522250 | 0.509500 | 4.309500 | 0.521500 | 32.046500 | 0.501750 | 140.249000 | 4.520500 |
| **std** | 439.418206 | 0.5001 | 0.816004 | 0.500035 | 4.341444 | 0.499662 | 18.145715 | 0.288416 | 35.399655 | 2.287837 |
| **min** | 501.000000 | 0.0000 | 0.500000 | 0.000000 | 0.000000 | 0.000000 | 2.000000 | 0.100000 | 80.000000 | 1.000000 |
| **25%** | 851.750000 | 0.0000 | 0.700000 | 0.000000 | 1.000000 | 0.000000 | 16.000000 | 0.200000 | 109.000000 | 3.000000 |
| **50%** | 1226.000000 | 0.0000 | 1.500000 | 1.000000 | 3.000000 | 1.000000 | 32.000000 | 0.500000 | 141.000000 | 4.000000 |
| **75%** | 1615.250000 | 1.0000 | 2.200000 | 1.000000 | 7.000000 | 1.000000 | 48.000000 | 0.800000 | 170.000000 | 7.000000 |
| **max** | 1998.000000 | 1.0000 | 3.000000 | 1.000000 | 19.000000 | 1.000000 | 64.000000 | 1.000000 | 200.000000 | 8.000000 |

8 rows × 21 columns

```
#checking for missing values
train_df.isnull().sum()
```

| | 0 |
|---|---|
| **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

```
#info on the dataset
train_df.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
```

```
train_df.columns
```

```
Index(['battery_power', 'blue', 'clock_speed', 'dual_sim', 'fc', 'four_g',
       'int_memory', 'm_dep', 'mobile_wt', 'n_cores', 'pc', 'px_height',
       'px_width', 'ram', 'sc_h', 'sc_w', 'talk_time', 'three_g',
       'touch_screen', 'wifi', 'price_range'],
      dtype='object')
```

```
#change column names and data
train_df.rename(columns={'blue':"bluetooth",'dual_sim':"sim",'four_g':"fourgb"},inplace=True)
print(train_df)
```

```
      battery_power  bluetooth  clock_speed  sim  fc  fourgb  int_memory  \
0               842          0          2.2    0   1       0           7
1              1021          1          0.5    1   0       1          53
2               563          1          0.5    1   2       1          41
3               615          1          2.5    0   0       0          10
4              1821          1          1.2    0  13       1          44
...             ...        ...          ...  ...  ..     ...         ...
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  \
0       0.6        188        2  ...         20       756  2549     9     7
1       0.7        136        3  ...        905      1988  2631    17     3
2       0.9        145        5  ...       1263      1716  2603    11     2
3       0.8        131        6  ...       1216      1786  2769    16     8
4       0.6        141        2  ...       1208      1212  1411     8     2
...     ...        ...      ...  ...        ...       ...   ...   ...   ...
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
0            19        0             0     1            1
1             7        1             1     0            2
2             9        1             1     0            2
3            11        1             0     0            2
4            15        1             1     0            1
...         ...      ...           ...   ...          ...
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

[2000 rows x 21 columns]
```

```
#changing the values of bluetooth,sim,4gb,three_g,touch_screen,wifi with 0-No,1-Yes
changing_columns=['bluetooth','sim','fourgb','three_g','touch_screen','wifi']
train_df[changing_columns]=train_df[changing_columns].replace({0:'No',1:'Yes'})
print(train_df)
```

```
      battery_power bluetooth  clock_speed  sim  fc fourgb  int_memory  m_dep  \
0               842        No          2.2   No   1     No           7    0.6
1              1021       Yes          0.5  Yes   0    Yes          53    0.7
2               563       Yes          0.5  Yes   2    Yes          41    0.9
3               615       Yes          2.5   No   0     No          10    0.8
4              1821       Yes          1.2   No  13    Yes          44    0.6
...             ...       ...          ...  ...  ..    ...         ...    ...
1995            794       Yes          0.5  Yes   0    Yes           2    0.8
1996           1965       Yes          2.6  Yes   0     No          39    0.2
1997           1911        No          0.9  Yes   1    Yes          36    0.7
1998           1512        No          0.9   No   4    Yes          46    0.1
1999            510       Yes          2.0  Yes   5    Yes          45    0.9

      mobile_wt  n_cores  ...  px_height  px_width   ram  sc_h  sc_w  \
0           188        2  ...         20       756  2549     9     7
1           136        3  ...        905      1988  2631    17     3
2           145        5  ...       1263      1716  2603    11     2
```

```
3          131        6  ...       1216      1786  2769    16     8
4          141        2  ...       1208      1212  1411     8     2
...        ...      ...  ...        ...       ...   ...   ...   ...
1995       106        6  ...       1222      1890   668    13     4
1996       187        4  ...        915      1965  2032    11    10
1997       108        8  ...        868      1632  3057     9     1
1998       145        5  ...        336       670   869    18    10
1999       168        6  ...        483       754  3919    19     4

      talk_time  three_g touch_screen wifi price_range
0            19       No           No  Yes           1
1             7      Yes          Yes   No           2
2             9      Yes          Yes   No           2
3            11      Yes           No   No           2
4            15      Yes          Yes   No           1
...         ...      ...          ...  ...         ...
1995         19      Yes          Yes   No           0
1996         16      Yes          Yes  Yes           2
1997          5      Yes          Yes   No           3
1998         19      Yes          Yes  Yes           0
1999          2      Yes          Yes  Yes           3

[2000 rows x 21 columns]
```

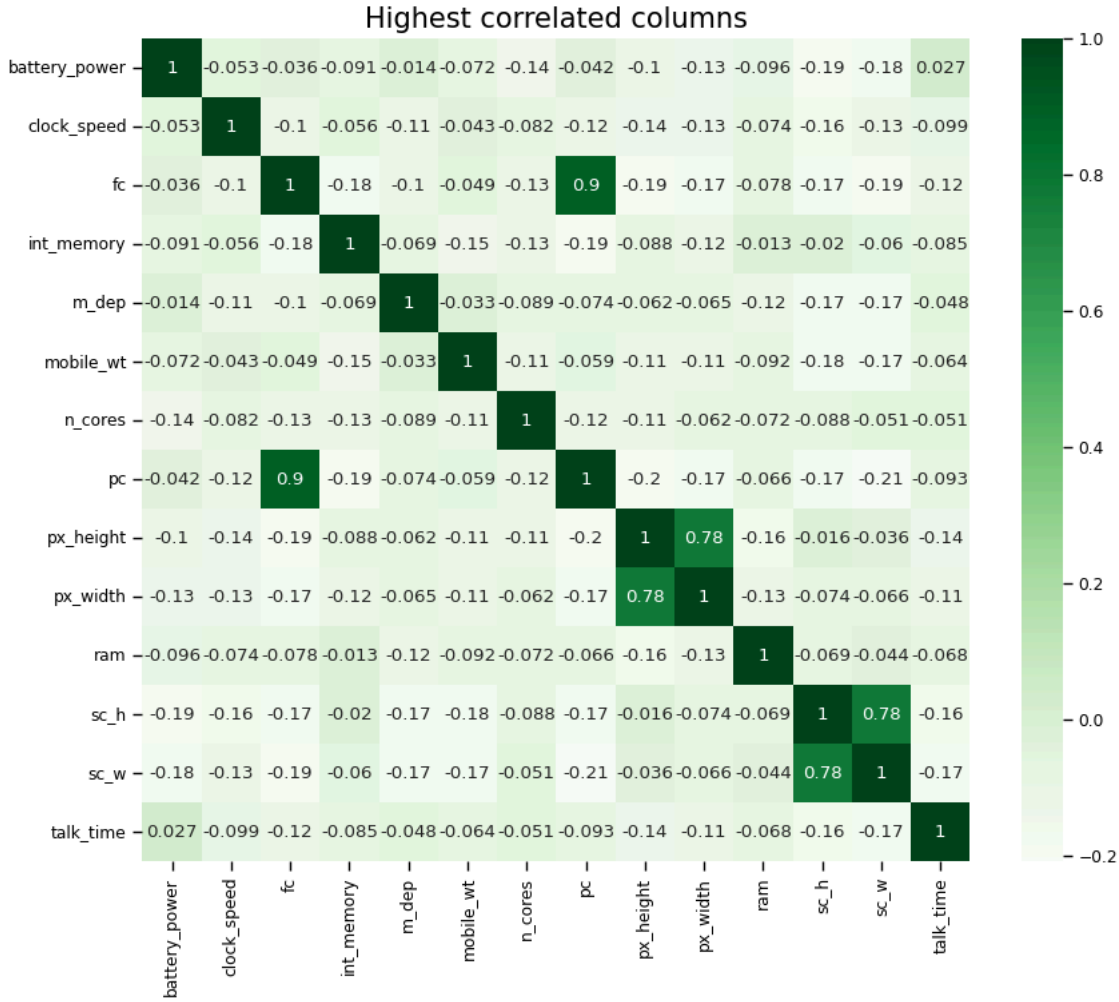
```
#changing price_range column

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

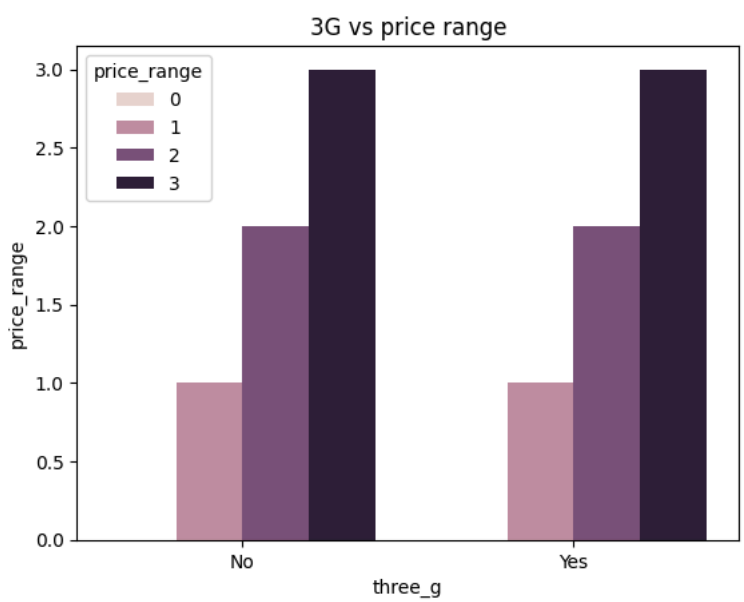
| | battery_power | bluetooth | clock_speed | sim | fc | fourgb | int_memory | m_dep | mobile_wt | n_cores | ... | px_height | px_width | ram |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 842 | No | 2.2 | No | 1 | No | 7 | 0.6 | 188 | 2 | ... | 20 | 756 | 2549 |
| **1** | 1021 | Yes | 0.5 | Yes | 0 | Yes | 53 | 0.7 | 136 | 3 | ... | 905 | 1988 | 2631 |
| **2** | 563 | Yes | 0.5 | Yes | 2 | Yes | 41 | 0.9 | 145 | 5 | ... | 1263 | 1716 | 2603 |
| **3** | 615 | Yes | 2.5 | No | 0 | No | 10 | 0.8 | 131 | 6 | ... | 1216 | 1786 | 2769 |
| **4** | 1821 | Yes | 1.2 | No | 13 | Yes | 44 | 0.6 | 141 | 2 | ... | 1208 | 1212 | 1411 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **1995** | 794 | Yes | 0.5 | Yes | 0 | Yes | 2 | 0.8 | 106 | 6 | ... | 1222 | 1890 | 668 |
| **1996** | 1965 | Yes | 2.6 | Yes | 0 | No | 39 | 0.2 | 187 | 4 | ... | 915 | 1965 | 2032 |
| **1997** | 1911 | No | 0.9 | Yes | 1 | Yes | 36 | 0.7 | 108 | 8 | ... | 868 | 1632 | 3057 |
| **1998** | 1512 | No | 0.9 | No | 4 | Yes | 46 | 0.1 | 145 | 5 | ... | 336 | 670 | 869 |
| **1999** | 510 | Yes | 2.0 | Yes | 5 | Yes | 45 | 0.9 | 168 | 6 | ... | 483 | 754 | 3919 |

2000 rows × 21 columns

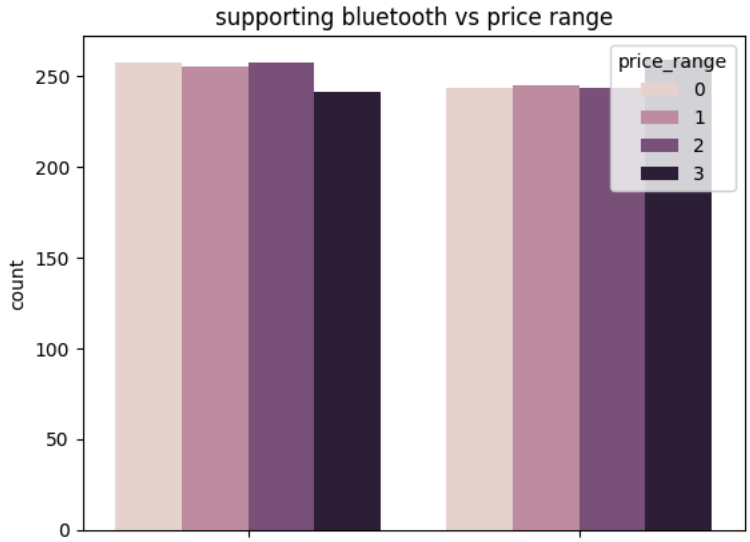
```
#VISUALIZATION
#highest correlated columns
plt.figure(figsize=(10,8))
sns.set_context('paper')
price_corr=train_df.corr(numeric_only=True)
sns.heatmap(price_corr.corr(),cmap='Greens',annot=True)
plt.title("Highest correlated columns",fontsize=15)
plt.show()
```



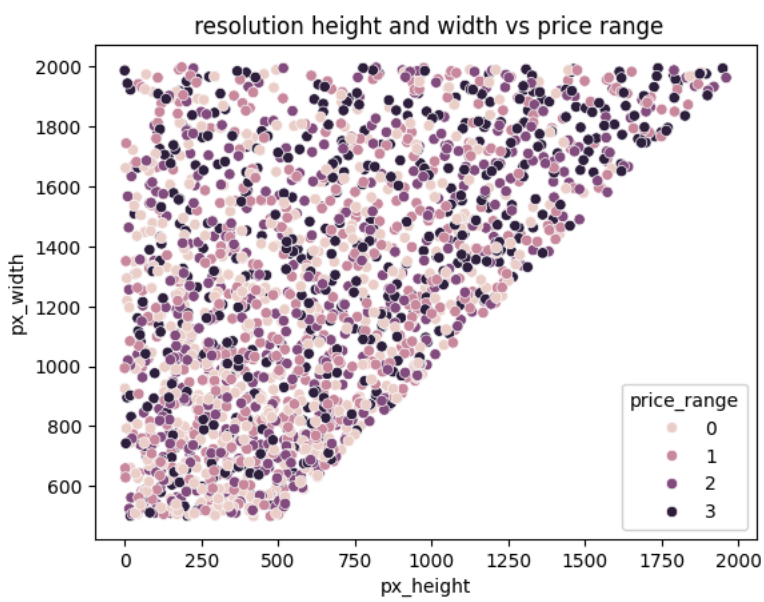

```
#three_g or not three_g mobile VS sale price
sns.barplot(x='three_g',y='price_range',hue='price_range',data=train_df)
plt.title('3G vs price range')
plt.show()
```




```
#countplot for supporting bluetooth vs price range
sns.countplot(x='bluetooth',hue='price_range',data=train_df)
plt.title('supporting bluetooth vs price range')
plt.show()
```




```
#scatterplot for pixel resolution height and width with price range
sns.scatterplot(x='px_height',y='px_width',hue='price_range',data=train_df)
plt.title(' resolution height and width vs price range')
plt.show()
```




```
#scatterplot for screen height and width with price range
sns.scatterplot(x='sc_h',y='sc_w',hue='price_range',data=train_df)
plt.title('screen height and width vs price range')
plt.show()
```