

```
In [1]: import pandas as pd  
import seaborn as sns
```

```
In [2]: df=pd.read_csv("C:/Users/Rutuja/OneDrive/Desktop/most_used_beauty_cosmetics_product  
df.head()
```

	Product_Name	Brand	Category	Usage_Frequency	Price_USD	Rating	Number_of_Reviews
0	Ultra Face Mask	Drunk Elephant	Blush	Weekly	67.85	1.4	686
1	Ultra Lipstick	Laura Mercier	Makeup Remover	Occasional	116.43	4.2	5483
2	Ultra Serum	Natasha Denona	Highlighter	Daily	90.84	1.6	5039
3	Divine Serum	Ilia Beauty	Face Mask	Occasional	55.17	3.2	6202
4	Super Foundation	Charlotte Tilbury	Highlighter	Occasional	140.56	1.7	297

```
In [3]: df[df['Brand']=='Make Up For Ever']
```

Out[3]:

		Product_Name	Brand	Category	Usage_Frequency	Price_USD	Rating	Number_of_Review
20	Perfect Blush	Make Up For Ever		Contour	Monthly	35.33	3.4	423
253	Perfect Serum	Make Up For Ever		Primer	Monthly	101.91	1.9	442
254	Divine Mascara	Make Up For Ever		Face Oil	Occasional	10.20	1.8	781
272	Magic Mascara	Make Up For Ever		Mascara	Weekly	148.03	1.8	85
377	Magic Face Mask	Make Up For Ever		Cleanser	Daily	62.29	4.8	107
...
14747	Ultra Lip Gloss	Make Up For Ever		Highlighter	Weekly	58.21	2.6	298
14792	Ultra Highlighter	Make Up For Ever		Serum	Monthly	141.75	2.0	145
14850	Divine Powder	Make Up For Ever		Primer	Monthly	78.40	4.9	146
14900	Divine Powder	Make Up For Ever		Lip Gloss	Occasional	142.63	4.1	920
14936	Ultra CC Cream	Make Up For Ever		Makeup Remover	Weekly	140.56	2.1	119

414 rows × 14 columns

In [4]: df.shape

Out[4]: (15000, 14)

In [37]: df['Packaging_Type'].nunique()

Out[37]: 6

```
In [5]: df.isnull().sum()
```

```
Out[5]: Product_Name      0  
Brand          0  
Category        0  
Usage_Frequency 0  
Price_USD       0  
Rating          0  
Number_of_Reviews 0  
Product_Size     0  
Skin_Type        0  
Gender_Target    0  
Packaging_Type   0  
Main_Ingredient  0  
Cruelty_Free     0  
Country_of-Origin 0  
dtype: int64
```

```
In [6]: df.columns
```

```
Out[6]: Index(['Product_Name', 'Brand', 'Category', 'Usage_Frequency', 'Price_USD',  
               'Rating', 'Number_of_Reviews', 'Product_Size', 'Skin_Type',  
               'Gender_Target', 'Packaging_Type', 'Main_Ingredient', 'Cruelty_Free',  
               'Country_of-Origin'],  
              dtype='object')
```

```
In [7]: df['Product_Name']
```

```
Out[7]: 0      Ultra Face Mask  
1      Ultra Lipstick  
2      Ultra Serum  
3      Divine Serum  
4      Super Foundation  
      ...  
14995    Magic Eyeliner  
14996    Perfect Powder  
14997    Magic Serum  
14998    Magic Mascara  
14999    Divine Lip Liner  
Name: Product_Name, Length: 15000, dtype: object
```

```
In [9]: df['Product_Name'].unique()
```

```
Out[9]: array(['Ultra Face Mask', 'Ultra Lipstick', 'Ultra Serum', 'Divine Serum',
   'Super Foundation', 'Super Eye Shadow', 'Super Lip Gloss',
   'Ultra Cleanser', 'Magic Highlighter', 'Perfect Bronzer',
   'Ultra Moisturizer', 'Divine CC Cream', 'Ultra Setting Spray',
   'Divine Primer', 'Divine Exfoliator', 'Perfect Blush',
   'Perfect Highlighter', 'Super Cleanser', 'Ultra Face Oil',
   'Divine Lip Gloss', 'Divine Foundation', 'Divine Cleanser',
   'Perfect Eyeliner', 'Super Serum', 'Divine Setting Spray',
   'Magic Eye Shadow', 'Ultra Exfoliator', 'Magic Lip Liner',
   'Super Makeup Remover', 'Perfect BB Cream', 'Magic BB Cream',
   'Super Lip Liner', 'Ultra Lip Gloss', 'Super Primer',
   'Divine Eyeliner', 'Super Contour', 'Magic Makeup Remover',
   'Perfect Face Mask', 'Perfect Contour', 'Perfect Lip Liner',
   'Magic Setting Spray', 'Magic CC Cream', 'Super Exfoliator',
   'Perfect Eye Shadow', 'Super Highlighter', 'Perfect Serum',
   'Super Eyeliner', 'Super Blush', 'Magic Primer', 'Perfect Mascara',
   'Ultra Concealer', 'Magic Contour', 'Magic Mascara',
   'Ultra Eyeliner', 'Super BB Cream', 'Perfect Concealer',
   'Divine Makeup Remover', 'Super Bronzer', 'Magic Lipstick',
   'Divine Contour', 'Ultra Blush', 'Ultra Makeup Remover',
   'Super Mascara', 'Divine Concealer', 'Ultra Foundation',
   'Divine Face Mask', 'Divine BB Cream', 'Perfect Cleanser',
   'Super Lipstick', 'Magic Eyeliner', 'Ultra Eye Shadow',
   'Super Face Mask', 'Divine Mascara', 'Magic Serum',
   'Divine Face Oil', 'Ultra Lip Liner', 'Magic Cleanser',
   'Ultra BB Cream', 'Super Powder', 'Perfect Lipstick',
   'Divine Highlighter', 'Ultra Powder', 'Perfect Setting Spray',
   'Ultra Highlighter', 'Super Setting Spray', 'Magic Powder',
   'Perfect Moisturizer', 'Magic Lip Gloss', 'Ultra Primer',
   'Perfect Exfoliator', 'Divine Moisturizer', 'Super Concealer',
   'Magic Face Oil', 'Ultra CC Cream', 'Super Face Oil',
   'Perfect Foundation', 'Perfect Lip Gloss', 'Divine Eye Shadow',
   'Perfect Primer', 'Ultra Mascara', 'Perfect Powder',
   'Divine Bronzer', 'Magic Blush', 'Ultra Bronzer', 'Magic Bronzer',
   'Divine Lipstick', 'Magic Moisturizer', 'Magic Exfoliator',
   'Magic Face Mask', 'Perfect Face Oil', 'Ultra Contour',
   'Perfect Makeup Remover', 'Magic Foundation', 'Magic Concealer',
   'Perfect CC Cream', 'Super CC Cream', 'Super Moisturizer',
   'Divine Blush', 'Divine Powder', 'Divine Lip Liner'], dtype=object)
```

In [10]: `df['Product_Name'].nunique()`

Out[10]: 120

In [11]: `df['Product_Name'].value_counts()[0:11]`

```
Out[11]: Product_Name
Super Setting Spray    154
Magic Face Oil        151
Magic Lip Liner       149
Perfect Lip Liner     147
Divine Exfoliator     146
Ultra Eye Shadow      144
Magic Bronzer         143
Super Foundation      142
Divine Lip Gloss       142
Super Cleanser        141
Super Blush           141
Name: count, dtype: int64
```

In [12]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15000 entries, 0 to 14999
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Product_Name     15000 non-null   object  
 1   Brand            15000 non-null   object  
 2   Category          15000 non-null   object  
 3   Usage_Frequency  15000 non-null   object  
 4   Price_USD         15000 non-null   float64 
 5   Rating            15000 non-null   float64 
 6   Number_of_Reviews 15000 non-null   int64  
 7   Product_Size      15000 non-null   object  
 8   Skin_Type          15000 non-null   object  
 9   Gender_Target      15000 non-null   object  
 10  Packaging_Type    15000 non-null   object  
 11  Main_Ingredient   15000 non-null   object  
 12  Cruelty_Free       15000 non-null   bool   
 13  Country_of-Origin 15000 non-null   object  
dtypes: bool(1), float64(2), int64(1), object(10)
memory usage: 1.5+ MB
```

```
In [11]: df['Brand']
```

```
Out[11]: 0           Drunk Elephant
          1           Laura Mercier
          2           Natasha Denona
          3           Ilia Beauty
          4           Charlotte Tilbury
          ...
          14995        Patrick Ta
          14996        Farsali
          14997        Kiehl's
          14998        Perricone MD
          14999        Tatcha
Name: Brand, Length: 15000, dtype: object
```

```
In [12]: df['Brand'].value_counts()
```

```
Out[12]:
```

Brand	count
Milk Makeup	426
Make Up For Ever	414
Kiehl's	411
NARS	400
E.l.f.	399
Farsali	399
Yves Saint Laurent	394
Morphe	392
Bite Beauty	392
Sisley	392
KVD Beauty	391
Glossier	388
Becca	388
Huda Beauty	384
RMS Beauty	382
Bobby Brown	380
Ilia Beauty	379
Clinique	377
Rare Beauty	375
Tatcha	374
Too Faced	371
Kylie Cosmetics	370
Shiseido	370
Anastasia Beverly Hills	369
Juvia's Place	368
Drunk Elephant	368
Hourglass	366
Natasha Denona	366
Fenty Beauty	366
Charlotte Tilbury	364
Tarte	361
Danessa Myricks	358
Bourjois	357
Urban Decay	356
Laura Mercier	355
Perricone MD	349
IT Cosmetics	346
Patrick Ta	345
Pat McGrath Labs	330
ColourPop	328

Name: count, dtype: int64

```
In [13]: df['Brand'].unique()
```

```
Out[13]: array(['Drunk Elephant', 'Laura Mercier', 'Natasha Denona', 'Ilia Beauty',
   'Charlotte Tilbury', 'Danessa Myricks', 'Bourjois', 'IT Cosmetics',
   'Fenty Beauty', 'Sisley', 'Juvia's Place', 'NARS', 'ColourPop',
   'Huda Beauty', 'Tatcha', 'Kiehl's', 'Tarte', 'Glossier',
   'Make Up For Ever', 'Anastasia Beverly Hills', 'E.l.f.',
   'Hourglass', 'Pat McGrath Labs', 'Too Faced', 'Perricone MD',
   'RMS Beauty', 'Urban Decay', 'Rare Beauty', 'Becca', 'Patrick Ta',
   'Shiseido', 'Kylie Cosmetics', 'Bite Beauty', 'Yves Saint Laurent',
   'Bobby Brown', 'Farsali', 'Morphe', 'Milk Makeup', 'Clinique',
   'KVD Beauty'], dtype=object)
```

```
In [14]: df['Brand'].nunique()
```

```
Out[14]: 40
```

```
In [18]: df['Category']
```

```
Out[18]: 0          Blush
         1      Makeup Remover
         2      Highlighter
         3      Face Mask
         4      Highlighter
         ...
        14995     Face Mask
        14996     Serum
        14997     Highlighter
        14998     Powder
        14999     Blush
Name: Category, Length: 15000, dtype: object
```

```
In [19]: df['Category'].unique()
```

```
Out[19]: array(['Blush', 'Makeup Remover', 'Highlighter', 'Face Mask',
       'Foundation', 'Powder', 'Lip Gloss', 'CC Cream', 'Eye Shadow',
       'Concealer', 'Eyeliner', 'Lipstick', 'Setting Spray', 'Cleanser',
       'Bronzer', 'Primer', 'Face Oil', 'Contour', 'Mascara', 'Serum',
       'BB Cream', 'Exfoliator', 'Lip Liner', 'Moisturizer'], dtype=object)
```

```
In [20]: df['Category'].nunique()
```

```
Out[20]: 24
```

```
In [21]: df['Category'].value_counts()
```

```
Out[21]: Category
Serum           710
Mascara         674
Face Oil        671
Highlighter     653
Face Mask       653
Moisturizer     638
Makeup Remover  636
Lipstick         633
Concealer        633
Bronzer          631
Eyeliner         629
Blush            628
Setting Spray   622
Cleanser         614
Foundation       613
BB Cream         613
Lip Liner        610
Primer           606
Contour          604
Powder           599
CC Cream         594
Exfoliator       583
Eye Shadow       582
Lip Gloss         571
Name: count, dtype: int64
```

```
In [22]: df[(df['Brand'] == 'Milk Makeup') & (df['Usage_Frequency'] == 'Daily')]
```

Out[22]:

	Product_Name	Brand	Category	Usage_Frequency	Price_USD	Rating	Number_of_Reviews
89	Divine Contour	Milk Makeup	Mascara	Daily	105.93	1.7	5
140	Super Makeup Remover	Milk Makeup	Contour	Daily	55.52	1.2	5
163	Perfect Highlighter	Milk Makeup	Primer	Daily	103.84	3.3	1
538	Divine Makeup Remover	Milk Makeup	Powder	Daily	112.34	3.8	6
548	Ultra Moisturizer	Milk Makeup	Concealer	Daily	104.07	1.9	1
...							
14261	Ultra Contour	Milk Makeup	Mascara	Daily	87.23	4.3	8
14317	Magic Concealer	Milk Makeup	Face Mask	Daily	147.21	2.3	7
14404	Magic Bronzer	Milk Makeup	Lipstick	Daily	125.80	4.3	8
14550	Super Bronzer	Milk Makeup	Lip Liner	Daily	80.37	1.9	9
14990	Magic Concealer	Milk Makeup	Foundation	Daily	97.20	3.3	1

104 rows × 14 columns

In [85]:

```
df[(df['Brand'] == 'Milk Makeup') & (df['Usage_Frequency'] == 'Weekly')]
```

Out[85]:

	Product_Name	Brand	Category	Usage_Frequency	Price_USD	Rating	Number_of_Reviews
657	Magic Powder	Milk Makeup	Exfoliator	Weekly	74.95	1.2	7
827	Super Serum	Milk Makeup	Eyeliner	Weekly	87.76	4.2	
937	Perfect Lip Gloss	Milk Makeup	Setting Spray	Weekly	27.55	1.3	2
1183	Super CC Cream	Milk Makeup	Foundation	Weekly	75.58	5.0	5
1214	Super Concealer	Milk Makeup	Mascara	Weekly	35.85	2.0	9
...
14023	Ultra Powder	Milk Makeup	Moisturizer	Weekly	97.28	4.3	6
14082	Divine Eyeliner	Milk Makeup	Contour	Weekly	87.31	3.5	4
14425	Perfect Face Oil	Milk Makeup	Concealer	Weekly	149.86	2.6	1
14573	Ultra Setting Spray	Milk Makeup	Eyeliner	Weekly	70.54	1.1	
14682	Magic Lipstick	Milk Makeup	Mascara	Weekly	97.19	4.6	3

109 rows × 14 columns

In [86]:

```
df[(df['Brand'] == 'Milk Makeup') & (df['Gender_Target'] == 'Male')]
```

Out[86]:

	Product_Name	Brand	Category	Usage_Frequency	Price_USD	Rating	Number_of_Reviews
89	Divine Contour	Milk Makeup	Mascara	Daily	105.93	1.7	5
140	Super Makeup Remover	Milk Makeup	Contour	Daily	55.52	1.2	5
574	Super Exfoliator	Milk Makeup	Bronzer	Monthly	58.22	2.2	20
625	Divine Mascara	Milk Makeup	Moisturizer	Daily	94.88	1.6	9
637	Perfect Concealer	Milk Makeup	CC Cream	Monthly	108.03	4.9	3
...
14272	Super Powder	Milk Makeup	Setting Spray	Occasional	55.18	3.5	-
14317	Magic Concealer	Milk Makeup	Face Mask	Daily	147.21	2.3	7
14573	Ultra Setting Spray	Milk Makeup	Eyeliner	Weekly	70.54	1.1	-
14608	Super BB Cream	Milk Makeup	Serum	Occasional	52.21	1.3	6
14682	Magic Lipstick	Milk Makeup	Mascara	Weekly	97.19	4.6	3

142 rows × 14 columns



In [87]:

df.describe()

Out[87]:

	Price_USD	Rating	Number_of_Reviews
count	15000.000000	15000.000000	15000.000000
mean	80.134108	3.002327	5014.231333
std	40.402983	1.168029	2855.665464
min	10.000000	1.000000	52.000000
25%	45.480000	2.000000	2562.000000
50%	80.040000	3.000000	5002.000000
75%	114.760000	4.000000	7497.000000
max	149.990000	5.000000	10000.000000

In [88]:

df.columns

Out[88]:

```
Index(['Product_Name', 'Brand', 'Category', 'Usage_Frequency', 'Price_USD',
       'Rating', 'Number_of_Reviews', 'Product_Size', 'Skin_Type',
       'Gender_Target', 'Packaging_Type', 'Main_Ingredient', 'Cruelty_Free',
       'Country_of_Origin'],
      dtype='object')
```

In [89]:

df['Brand'].value_counts()[0:10]

```
Out[89]: Brand
Milk Makeup      426
Make Up For Ever 414
Kiehl's          411
NARS             400
E.l.f.           399
Farsali          399
Yves Saint Laurent 394
Morphe           392
Bite Beauty      392
Sisley           392
Name: count, dtype: int64
```

which is the top 10 most selling brand and there products?

```
In [6]: def product_details(Brand):
    data = df[df['Brand'] == Brand]
    print('No of products', data['Product_Name'].value_counts()[0:10])
```

```
In [7]: product_details('Milk Makeup')
```

```
No of products Product_Name
Super Concealer      10
Super Setting Spray   10
Magic Lip Liner       9
Ultra Foundation     8
Divine Mascara       8
Magic Highlighter    7
Perfect Concealer    7
Divine Contour       6
Perfect Contour      6
Magic Eyeliner       6
Name: count, dtype: int64
```

```
In [92]: product_details('Make Up For Ever')
```

```
No of products Product_Name
Divine Exfoliator    10
Magic Concealer      8
Perfect Eye Shadow   8
Super BB Cream        8
Divine Blush          7
Super Face Mask      7
Name: count, dtype: int64
```

```
In [93]: product_details('Kiehl's')
```

```
No of products Product_Name
Super Lip Liner       9
Perfect Eye Shadow   8
Perfect Primer        7
Magic Face Oil        7
Perfect Mascara       6
Ultra Foundation      6
Name: count, dtype: int64
```

```
In [94]: product_details('NARS')
```

```
No of products Product_Name
Ultra Lipstick      8
Super Lipstick     8
Super Highlighter  7
Magic Cleanser     7
Magic Lip Liner    7
Ultra Cleanser     7
Name: count, dtype: int64
```

In [95]: `product_details('E.l.f.')`

```
No of products Product_Name
Ultra Face Mask    9
Magic Face Oil     8
Magic Setting Spray 7
Perfect Face Mask   7
Magic Makeup Remover 7
Perfect Cleanser    7
Name: count, dtype: int64
```

In [96]: `l1=[
 'Milk Makeup',
 'Make Up For Ever',
 'Kiehl's',
 'NARS',
 'E.l.f.',
 'Farsali',
 'Yves Saint Laurent',
 'Morphe',
 'Bite Beauty',
 'Sisley'
]`

In [97]: `l1`

Out[97]: `['Milk Makeup',
 'Make Up For Ever',
 'Kiehl's',
 'NARS',
 'E.l.f.',
 'Farsali',
 'Yves Saint Laurent',
 'Morphe',
 'Bite Beauty',
 'Sisley']`

In [98]: `l2=['Super Setting Spray',
 'Magic Face Oil',
 'Magic Lip Liner',
 'Perfect Lip Liner',
 'Divine Exfoliator'
]`

In [99]: `def Brand_prod(brand,product):
 data = df[(df['Brand']==brand) & (df['Product_Name']==product)]
 return data;`

In [100...]: `for a in l1:
 for b in l2:
 print(f"brand={a} and product={b}")
 data=Brand_prod(a,b)
 print(data.shape, '\n')`

brand=Milk Makeup and product=Super Setting Spray
(10, 14)

brand=Milk Makeup and product=Magic Face Oil
(1, 14)

brand=Milk Makeup and product=Magic Lip Liner
(9, 14)

brand=Milk Makeup and product=Perfect Lip Liner
(3, 14)

brand=Milk Makeup and product=Divine Exfoliator
(4, 14)

brand=Make Up For Ever and product=Super Setting Spray
(4, 14)

brand=Make Up For Ever and product=Magic Face Oil
(3, 14)

brand=Make Up For Ever and product=Magic Lip Liner
(7, 14)

brand=Make Up For Ever and product=Perfect Lip Liner
(4, 14)

brand=Make Up For Ever and product=Divine Exfoliator
(10, 14)

brand=Kiehl's and product=Super Setting Spray
(6, 14)

brand=Kiehl's and product=Magic Face Oil
(7, 14)

brand=Kiehl's and product=Magic Lip Liner
(4, 14)

brand=Kiehl's and product=Perfect Lip Liner
(3, 14)

brand=Kiehl's and product=Divine Exfoliator
(1, 14)

brand=NARS and product=Super Setting Spray
(5, 14)

brand=NARS and product=Magic Face Oil
(3, 14)

brand=NARS and product=Magic Lip Liner
(7, 14)

brand=NARS and product=Perfect Lip Liner
(3, 14)

brand=NARS and product=Divine Exfoliator
(4, 14)

brand=E.l.f. and product=Super Setting Spray
(1, 14)

brand=E.l.f. and product=Magic Face Oil

(8, 14)

brand=E.l.f. and product=Magic Lip Liner
(6, 14)

brand=E.l.f. and product=Perfect Lip Liner
(6, 14)

brand=E.l.f. and product=Divine Exfoliator
(2, 14)

brand=Farsali and product=Super Setting Spray
(8, 14)

brand=Farsali and product=Magic Face Oil
(4, 14)

brand=Farsali and product=Magic Lip Liner
(6, 14)

brand=Farsali and product=Perfect Lip Liner
(1, 14)

brand=Farsali and product=Divine Exfoliator
(3, 14)

brand=Yves Saint Laurent and product=Super Setting Spray
(5, 14)

brand=Yves Saint Laurent and product=Magic Face Oil
(3, 14)

brand=Yves Saint Laurent and product=Magic Lip Liner
(3, 14)

brand=Yves Saint Laurent and product=Perfect Lip Liner
(2, 14)

brand=Yves Saint Laurent and product=Divine Exfoliator
(2, 14)

brand=Morphe and product=Super Setting Spray
(4, 14)

brand=Morphe and product=Magic Face Oil
(4, 14)

brand=Morphe and product=Magic Lip Liner
(2, 14)

brand=Morphe and product=Perfect Lip Liner
(4, 14)

brand=Morphe and product=Divine Exfoliator
(6, 14)

brand=Bite Beauty and product=Super Setting Spray
(2, 14)

brand=Bite Beauty and product=Magic Face Oil
(3, 14)

brand=Bite Beauty and product=Magic Lip Liner
(6, 14)

```
brand=Bite Beauty and product=Perfect Lip Liner  
(3, 14)  
  
brand=Bite Beauty and product=Divine Exfoliator  
(4, 14)  
  
brand=Sisley and product=Super Setting Spray  
(6, 14)  
  
brand=Sisley and product=Magic Face Oil  
(2, 14)  
  
brand=Sisley and product=Magic Lip Liner  
(6, 14)  
  
brand=Sisley and product=Perfect Lip Liner  
(2, 14)  
  
brand=Sisley and product=Divine Exfoliator  
(6, 14)
```

```
In [101]: df['Usage_Frequency'].value_counts()
```

```
Out[101]: Usage_Frequency  
Occasional    3794  
Monthly       3781  
Weekly        3757  
Daily         3668  
Name: count, dtype: int64
```

```
In [ ]:
```

```
In [102]: def usage_brand_rating(usage,Brand,r):  
    data=  
    return data;
```

```
In [103]: l1=[  
    'Occasional',  
    'Monthly',  
    'Weekly',  
    'Daily'  
]
```

```
In [104]: l2=['Milk Makeup',  
    'Make Up For Ever',  
    'Kiehl's',  
    'NARS',  
    'E.l.f.'  
  
]
```

```
In [105]: df[(df['Usage_Frequency']=='Occasional') & (df['Brand']=='Milk Makeup') & (df['Rating'])]  
Out[105]: (56, 14)
```

```
In [106]: df[(df['Usage_Frequency']=='Monthly') & (df['Brand']=='Milk Makeup') & (df['Rating'])]  
Out[106]: (65, 14)
```

In [107...]

```
for a in l1:  
    for b in l2:  
        r=5  
        print(f"usage F={a} and brand={b}")  
        data=usage_brand_rating(a,b,r)  
        print(data.shape[0],'\n')
```

```
usage F=Occasional and brand=Milk Makeup
(2, 14)

usage F=Occasional and brand=Make Up For Ever
(1, 14)

usage F=Occasional and brand=Kiehl's
(0, 14)

usage F=Occasional and brand=NARS
(0, 14)

usage F=Occasional and brand=E.l.f.
(0, 14)

usage F=Monthly and brand=Milk Makeup
(1, 14)

usage F=Monthly and brand=Make Up For Ever
(0, 14)

usage F=Monthly and brand=Kiehl's
(1, 14)

usage F=Monthly and brand=NARS
(0, 14)

usage F=Monthly and brand=E.l.f.
(0, 14)

usage F=Weekly and brand=Milk Makeup
(2, 14)

usage F=Weekly and brand=Make Up For Ever
(1, 14)

usage F=Weekly and brand=Kiehl's
(0, 14)

usage F=Weekly and brand=NARS
(1, 14)

usage F=Weekly and brand=E.l.f.
(0, 14)

usage F=Daily and brand=Milk Makeup
(0, 14)

usage F=Daily and brand=Make Up For Ever
(3, 14)

usage F=Daily and brand=Kiehl's
(1, 14)

usage F=Daily and brand=NARS
(1, 14)

usage F=Daily and brand=E.l.f.
(0, 14)
```

what is the uses frequency for top 5 brands and there ratings?

```
In [1]: data=df[(df['Usage_Frequency']=='Occasional') & (df['Brand']=='Milk Makeup') & (df[percentage = (data / total_count) * 100
```

```
NameError Traceback (most recent call last)
Cell In[1], line 1
----> 1 data=df[(df['Usage_Frequency']=='Occasional') & (df['Brand']=='Milk Makeup') & (df['Rating']>3)].shape
      2 percentage = (data / total_count) * 100

NameError: name 'df' is not defined
```

```
In [2]: 'data'
```

```
NameError Traceback (most recent call last)
Cell In[2], line 1
----> 1 data

NameError: name 'data' is not defined
```

```
In [77]: data=df[(df['Usage_Frequency']=='Monthly') & (df['Brand']=='Milk Makeup') & (df['Ra
```

```
In [78]: data
```

```
Out[78]: (65, 14)
```

```
In [83]: data=df[(df['Usage_Frequency']=='Weekly') & (df['Brand']=='Milk Makeup') & (df['Ra
```

```
In [84]: data
```

```
Out[84]: (0, 14)
```

```
In [85]: data=df[(df['Usage_Frequency']=='Daily') & (df['Brand']=='Milk Makeup') & (df['Ra
```

```
In [86]: data
```

```
Out[86]: (52, 14)
```

```
In [116... data=df[(df['Usage_Frequency']=='Occasional') & (df['Brand']=='Make Up For Ever') &
```

```
In [117... data
```

```
Out[117]: (60, 14)
```

```
In [118... data=df[(df['Usage_Frequency']=='Monthly') & (df['Brand']=='Make Up For Ever') & (df['Ra
```

```
In [119... data
```

```
Out[119]: (53, 14)
```

```
In [88]: data=df[(df['Usage_Frequency']=='Weekly') & (df['Brand']=='Make Up For Ever') & (df['Ra
```

```
data
```

```
Out[88]: (50, 14)
```

```
In [90]: data=df[(df['Usage_Frequency']=='Daily') & (df['Brand']=='Make Up For Ever') & (df[
```

```
Out[90]: (47, 14)
```

```
In [120... data=df[(df['Usage_Frequency']=='Occasional') & (df['Brand']=='Kiehl's') & (df['Rat
```

```
In [121... data
```

```
Out[121]: (54, 14)
```

```
In [122... data=df[(df['Usage_Frequency']=='Monthly') & (df['Brand']=='Kiehl's') & (df['Rating'
```

```
In [123... data
```

```
Out[123]: (48, 14)
```

```
In [92]: data=df[(df['Usage_Frequency']=='Weekly') & (df['Brand']=='Kiehl's') & (df['Rating'
```

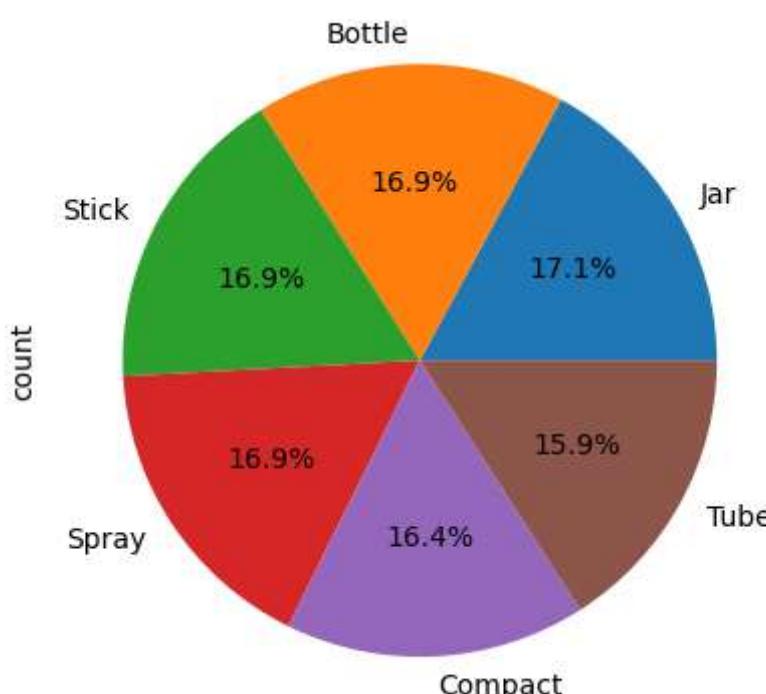
```
Out[92]: (38, 14)
```

```
In [94]: data=df[(df['Usage_Frequency']=='Daily') & (df['Brand']=='Kiehl's') & (df['Rating'
```

```
Out[94]: (55, 14)
```

```
In [39]: df['Packaging_Type'].value_counts().plot(kind='pie', autopct="%0.1f%%")
```

```
Out[39]: <Axes: ylabel='count'>
```



```
In [126]: pd.crosstab(df['Usage_Frequency'], df['Gender_Target'])
```

```
Out[126]: Gender_Target  Female  Male  Unisex
```

		Usage_Frequency			
		Daily	1236	1248	1184
		Monthly	1248	1283	1250
		Occasional	1274	1290	1230
		Weekly	1244	1196	1317

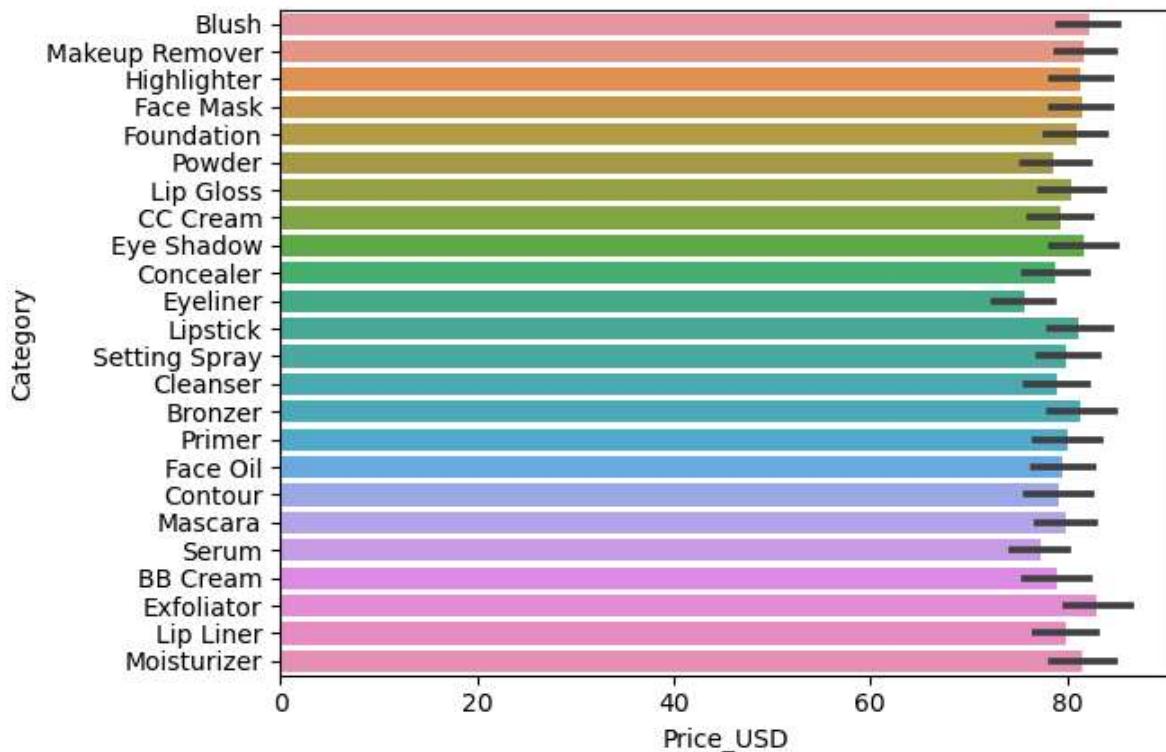
```
In [127]: sns.heatmap(pd.crosstab(df['Usage_Frequency'], df['Gender_Target']))
```

```
Out[127]: <Axes: xlabel='Gender_Target', ylabel='Usage_Frequency'>
```



```
In [128]: sns.barplot(x=df['Price_USD'], y=df['Category'])
```

```
Out[128]: <Axes: xlabel='Price_USD', ylabel='Category'>
```



which product are uses daily and there rating>3 and from which brand are they use?

```
In [1]: data=df[(df['Usage_Frequency']=='Daily') & (df['Rating']>=5)].shape
```

```
NameError: name 'df' is not defined
```

```
In [130... df['Brand'].value_counts()
```

```
Out[130]:
```

Brand	count
Milk Makeup	426
Make Up For Ever	414
Kiehl's	411
NARS	400
E.l.f.	399
Farsali	399
Yves Saint Laurent	394
Morphe	392
Bite Beauty	392
Sisley	392
KVD Beauty	391
Glossier	388
Becca	388
Huda Beauty	384
RMS Beauty	382
Bobby Brown	380
Ilia Beauty	379
Clinique	377
Rare Beauty	375
Tatcha	374
Too Faced	371
Kylie Cosmetics	370
Shiseido	370
Anastasia Beverly Hills	369
Juvia's Place	368
Drunk Elephant	368
Hourglass	366
Natasha Denona	366
Fenty Beauty	366
Charlotte Tilbury	364
Tarte	361
Danessa Myricks	358
Bourjois	357
Urban Decay	356
Laura Mercier	355
Perricone MD	349
IT Cosmetics	346
Patrick Ta	345
Pat McGrath Labs	330
ColourPop	328

Name: count, dtype: int64

```
In [31]: df['Packaging_Type'].value_counts()
```

```
Out[31]:
```

Packaging_Type	count
Jar	2567
Bottle	2532
Stick	2529
Spray	2528
Compact	2454
Tube	2390

Name: count, dtype: int64

Top 5 brand suit on which skin type and there gender?

```
In [14]: data10=df[(df['Brand']=='Milk Makeup') & (df['Skin_Type']=='Combination') & (df['Ge
```

```
In [38]: data11=df[(df['Packaging_Type']=='Spray')]
```

```
In [39]: data11['Price_USD'].median()
```

```
Out[39]: 80.25
```

```
In [8]: data10=df[(df['Brand']=='Milk Makeup') & (df['Skin_Type']=='Oily') & (df['Gender_Target']=='Female')]
```

```
Out[8]: (34, 14)
```

```
In [ ]:
```

```
In [9]: df[(df['Brand']=='Milk Makeup') & (df['Skin_Type']=='Normal') & (df['Gender_Target']=='Female')]
```

```
Out[9]: (24, 14)
```

```
In [12]: df[(df['Brand']=='Milk Makeup') & (df['Skin_Type']=='Sensitive') & (df['Gender_Target']=='Female')]
```

```
Out[12]: (27, 14)
```

```
In [13]: df[(df['Brand']=='Milk Makeup') & (df['Skin_Type']=='Dry') & (df['Gender_Target']=='Female')]
```

```
Out[13]: (26, 14)
```

```
In [16]: df[(df['Brand']=='Make Up For Ever') & (df['Skin_Type']=='Dry') & (df['Gender_Target']=='Female')]
```

```
Out[16]: (39, 14)
```

```
In [22]: df[(df['Brand']=='Make Up For Ever') & (df['Skin_Type']=='Combination') & (df['Gender_Target']=='Female')]
```

```
Out[22]: (26, 14)
```

```
In [24]: df[(df['Brand']=='Make Up For Ever') & (df['Skin_Type']=='Sensitive') & (df['Gender_Target']=='Female')]
```

```
Out[24]: (23, 14)
```

```
In [25]: df[(df['Brand']=='Make Up For Ever') & (df['Skin_Type']=='Normal') & (df['Gender_Target']=='Female')]
```

```
Out[25]: (31, 14)
```

```
In [5]: data1=df[(df['Brand']=='Make Up For Ever') & (df['Skin_Type']=='Oily') & (df['Gender_Target']=='Female')]
```

```
In [8]: data1['Price_USD'].median()
```

```
Out[8]: 95.7
```

```
In [13]: df[(df['Brand']=='Make Up For Ever') & (df['Skin_Type']=='Combination') & (df['Gender_Target']=='Female')]
```

```
Out[13]: Product_Name    Brand    Category    Usage_Frequency    Price_USD    Rating    Number_of_Reviews    Prod
```

```
In [12]: data2['Price_USD'].median()
```

```
Out[12]: nan
```

```
In [11]: df[(df['Brand']=='Make Up For Ever') & (df['Skin_Type']=='Normal') & (df['Gender_Target']=='Female')]
```

```
Out[11]: (0, 14)
```

```
In [173... df[(df['Brand']=='Make Up For Ever') & (df['Skin_Type']=='Combination') & (df['Gender_Target']=='Female')]]  
Out[173]: (0, 14)
```

```
In [174... df[(df['Brand']=='Milk Makeup') & (df['Skin_Type']=='Oily') & (df['Gender_Target']=='Male')]]  
Out[174]: (0, 14)
```

```
In [175... df[(df['Brand']=='Milk Makeup') & (df['Skin_Type']=='Combination') & (df['Gender_Target']=='Unspecified')]]  
Out[175]: (0, 14)
```

```
In [176... df[(df['Brand']=='Kiehl's') & (df['Skin_Type']=='Combination') & (df['Gender_Target']=='Female')]]  
Out[176]: (0, 14)
```

```
In [178... df[(df['Brand']=='Milk Makeup') & (df['Skin_Type']=='Combination') & (df['Gender_Target']=='Male')]]  
Out[178]: (0, 14)
```

```
In [ ]:
```

which product size sales most what are there number of reviews and ratings?

```
In [9]: df['Product_Size'].value_counts().numeric_only=True
```

```
In [ ]:
```

```
In [42]: data=df[(df['Product_Size']=='100ml') & (df['Rating']>4)]  
data
```

Out[42]:

	Product_Name	Brand	Category	Usage_Frequency	Price_USD	Rating	Number_of_Reviews
22	Divine Cleanser	Fenty Beauty	Face Mask	Monthly	119.26	4.6	
46	Ultra Moisturizer	Hourglass	Contour	Daily	79.42	4.7	
59	Super Blush	Yves Saint Laurent	Face Mask	Weekly	110.51	4.2	
102	Ultra Eye Shadow	NARS	Foundation	Monthly	136.12	4.6	
180	Perfect Concealer	Becca	Powder	Monthly	142.44	4.2	
...
14895	Magic Powder	Yves Saint Laurent	Primer	Occasional	125.05	4.8	
14900	Divine Powder	Make Up For Ever	Lip Gloss	Occasional	142.63	4.1	
14908	Perfect Contour	Bourjois	Face Oil	Daily	85.81	4.8	
14947	Ultra Highlighter	Laura Mercier	Powder	Daily	123.54	4.3	
14950	Perfect Primer	Danessa Myricks	Serum	Daily	25.85	4.9	

587 rows × 14 columns

In [46]: `filtered_data = df[(df['Product_Size'] == '100ml') & (df['Rating'] > 4)]
percentage = (len(filtered_data) / len(df)) * 100`

In [47]: `print(f"Percentage of products with '100ml' size and rating greater than 4: {percentage}%")`
Percentage of products with '100ml' size and rating greater than 4: 3.91%

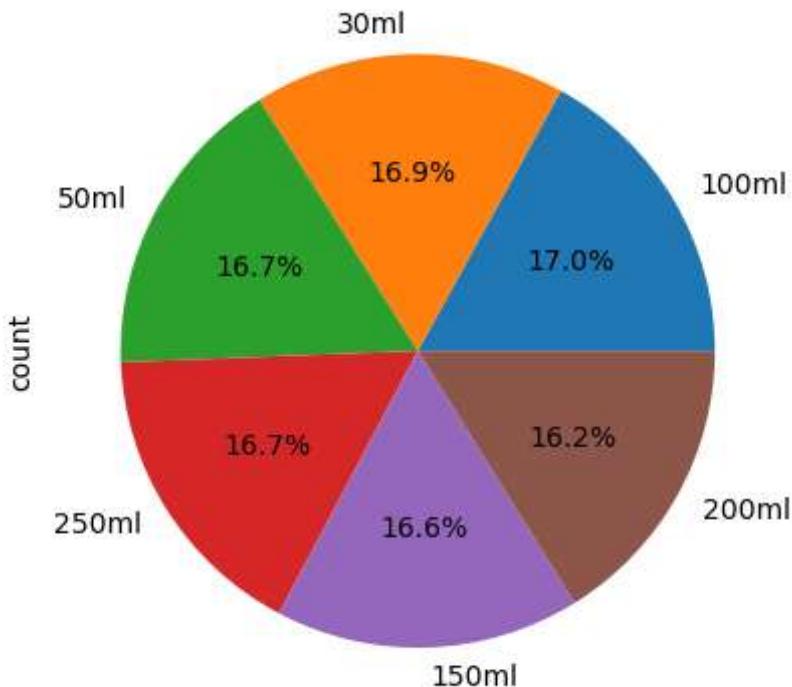
In [29]: `data`

Out[29]: `True`

In [58]: `import numpy as np`

In [65]: `df['Product_Size'].value_counts().plot(kind='pie', autopct="%0.1f%%")`

Out[65]: `<Axes: ylabel='count'>`



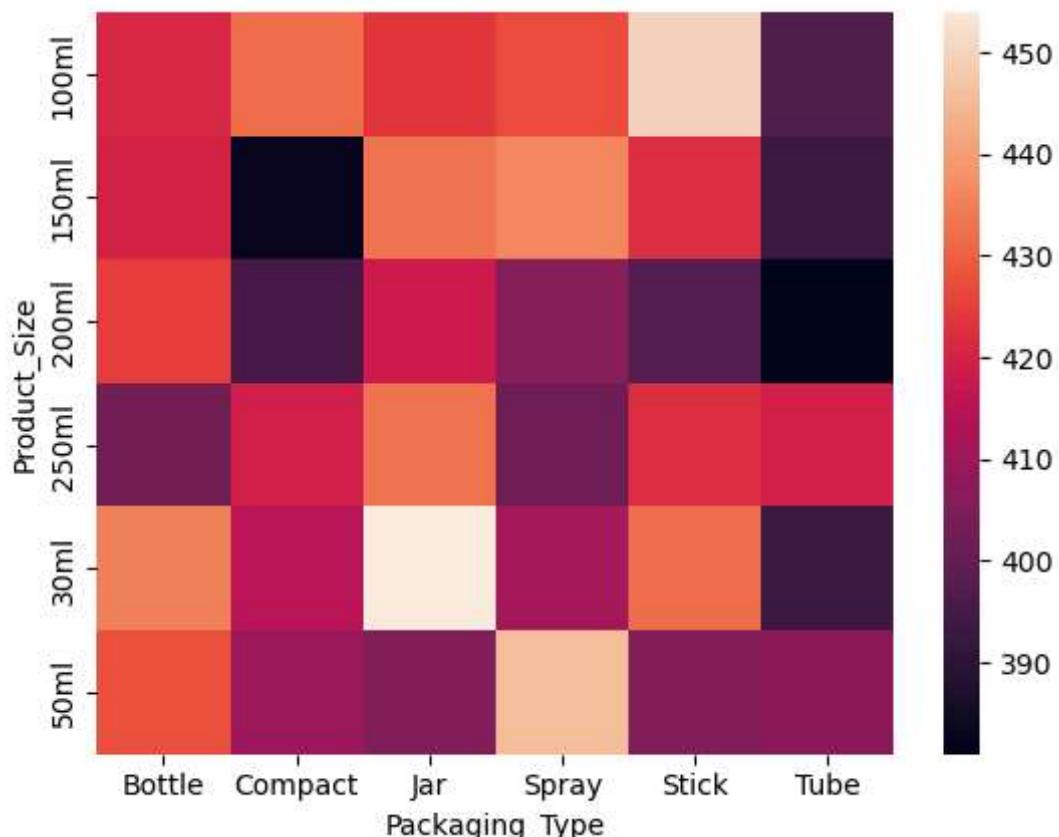
```
In [94]: pd.crosstab(df['Product_Size'], df['Packaging_Type'])
```

```
Out[94]: Packaging_Type  Bottle  Compact  Jar  Spray  Stick  Tube
```

		Product_Size					
		100ml	150ml	200ml	250ml	30ml	50ml
Packaging_Type		421	420	425	403	435	428
Bottle	432	383	395	419	415	410	405
Compact	424	433	418	433	454	405	405
Jar	427	436	406	402	411	446	405
Spray	450	422	398	422	432	405	407
Stick	397	393	381	419	393	393	
Tube							

```
In [18]: sns.heatmap(pd.crosstab(df['Product_Size'], df['Packaging_Type']))
```

```
Out[18]: <Axes: xlabel='Packaging_Type', ylabel='Product_Size'>
```



which type 5 product used by male and there packing type and what are the main ingredients in that?

```
In [196]: df['Main_Ingredient'].value_counts()
```

Out[196]: Main_Ingredient

Retinol	2180
Glycerin	2174
Vitamin C	2168
Salicylic Acid	2149
Shea Butter	2148
Aloe Vera	2091
Hyaluronic Acid	2090
Name: count, dtype: int64	

```
In [204]: df[(df['Product_Name']=='Super Setting Spray') & (df['Gender_Target']=='Male') & (df['Category']=='Blush')]
```

	Product_Name	Brand	Category	Usage_Frequency	Price_USD	Rating	Number_of_Review
1388	Super Setting Spray	Huda Beauty	Blush	Weekly	100.46	4.3	9021
11656	Super Setting Spray	Bite Beauty	Concealer	Weekly	63.62	2.2	3271

```
In [19]: df[(df['Product_Name']=='Super Setting Spray') & (df['Gender_Target']=='Male') & (df['Category']=='Concealer')]
```

Out[19]:

	Product_Name	Brand	Category	Usage_Frequency	Price_USD	Rating	Number_of_Rev
1649	Super Setting Spray	Anastasia Beverly Hills	Bronzer	Daily	109.73	3.9	
1919	Super Setting Spray	Kiehl's	BB Cream	Monthly	62.72	2.8	
3250	Super Setting Spray	Shiseido	Concealer	Weekly	127.77	2.0	
6979	Super Setting Spray	NARS	Highlighter	Monthly	95.39	1.4	
10319	Super Setting Spray	Shiseido	Moisturizer	Monthly	118.49	4.8	
11037	Super Setting Spray	Ilia Beauty	Eye Shadow	Weekly	139.33	1.8	
11887	Super Setting Spray	Milk Makeup	Concealer	Occasional	89.38	4.5	
14687	Super Setting Spray	Becca	Setting Spray	Weekly	63.32	3.9	

In [208...]

```
df[(df['Product_Name']=='Super Setting Spray') & (df['Gender_Target']=='Male') & (df['Country_of_Origin']=='Italy')]
```

Out[208]:

(7, 14)

which country use top 5 brand most and what are there numbers of ratings?

In [210...]

```
df['Country_of-Origin'].value_counts()
```

Out[210]:

Country_of-Origin	count
Italy	1942
USA	1931
Australia	1873
France	1863
Japan	1861
South Korea	1851
UK	1846
Germany	1833

Name: count, dtype: int64

In [231...]

```
df[(df['Brand']=='Milk Makeup') & (df['Country_of-Origin']=='Italy') & (df['Rating']>=4.5)]
```

Out[231]:

	Product_Name	Brand	Category	Usage_Frequency	Price_USD	Rating	Number_of_Reviews
326	Divine Concealer	Milk Makeup	Foundation	Occasional	132.63	4.7	6
477	Super Exfoliator	Milk Makeup	Powder	Monthly	29.58	4.9	5
4542	Divine Blush	Milk Makeup	Primer	Occasional	35.79	4.4	
4659	Perfect Contour	Milk Makeup	Bronzer	Daily	65.18	4.5	7
5260	Perfect Cleanser	Milk Makeup	Concealer	Weekly	87.57	4.2	4
5866	Divine Foundation	Milk Makeup	Face Oil	Weekly	14.75	4.1	4
8037	Super Mascara	Milk Makeup	Eyeliner	Weekly	94.70	4.7	5
8712	Super Mascara	Milk Makeup	Eye Shadow	Daily	61.35	4.7	7
8891	Perfect Exfoliator	Milk Makeup	Moisturizer	Daily	43.56	4.5	3
9133	Divine Eye Shadow	Milk Makeup	BB Cream	Weekly	66.92	4.8	9
10265	Super Face Oil	Milk Makeup	Makeup Remover	Daily	108.45	4.3	2
11202	Ultra Cleanser	Milk Makeup	Face Oil	Monthly	10.96	4.8	5
13969	Magic Highlighter	Milk Makeup	Lipstick	Monthly	133.46	4.8	2
14213	Divine Blush	Milk Makeup	Moisturizer	Occasional	85.02	4.7	3

In [215...]

```
df[(df['Brand']=='Milk Makeup') & (df['Country_of-Origin']=='Australia') & (df['Rating']>=4.5)]
```

Out[215]: (24, 14)

In [216...]

```
df[(df['Brand']=='Milk Makeup') & (df['Country_of-Origin']=='USA') & (df['Rating']>=4.5)]
```

Out[216]: (13, 14)

In [227...]

```
df[(df['Brand']=='Make Up For Ever') & (df['Country_of-Origin']=='Italy') & (df['Rating']>=4.5)]
```

Out[227]: (9, 14)

In [228...]

```
df[(df['Brand']=='Make Up For Ever') & (df['Country_of-Origin']=='Australia') & (df['Rating']>=4.5)]
```

Out[228]: (8, 14)

In [229...]

```
df[(df['Brand']=='Make Up For Ever') & (df['Country_of-Origin']=='USA') & (df['Rating']>=4.5)]
```

Out[229]: (12, 14)

```
In [230... df[(df['Brand']=='Make Up For Ever') & (df['Country_of_Origin']=='Germany') & (df['
```

```
Out[230]: (15, 14)
```

Top 5 main ingredients and they are use in which brand and there price?

```
In [234... df[(df['Main_Ingredient']=='Retinol') & (df['Brand']=='ColourPop')]]
```

Out[234]:

	Product_Name	Brand	Category	Usage_Frequency	Price_USD	Rating	Number_of_Reviews
138	Divine Primer	ColourPop	CC Cream	Daily	67.61	2.4	
224	Divine Eye Shadow	ColourPop	Lip Gloss	Occasional	116.47	3.6	
310	Perfect Eyeliner	ColourPop	Concealer	Daily	129.74	2.0	
371	Magic Powder	ColourPop	Lip Gloss	Weekly	33.02	1.3	
514	Super Contour	ColourPop	Face Mask	Daily	138.00	3.3	
568	Divine Contour	ColourPop	Face Mask	Weekly	135.18	1.6	
1060	Perfect Powder	ColourPop	Eye Shadow	Monthly	12.50	2.3	
1304	Super Mascara	ColourPop	Makeup Remover	Weekly	111.67	3.8	
1772	Ultra Mascara	ColourPop	Exfoliator	Weekly	133.02	3.7	
1857	Magic Cleanser	ColourPop	Eye Shadow	Monthly	71.11	4.5	
2017	Magic Serum	ColourPop	Eyeliner	Monthly	24.41	2.2	
3338	Ultra Lip Gloss	ColourPop	Face Oil	Occasional	71.42	2.9	
3409	Super Lipstick	ColourPop	Face Mask	Occasional	85.60	4.2	
3540	Ultra Foundation	ColourPop	Makeup Remover	Daily	21.23	4.2	
3742	Perfect Lip Gloss	ColourPop	Bronzer	Occasional	102.24	2.8	
4187	Divine CC Cream	ColourPop	Lip Liner	Monthly	113.31	1.9	
4514	Perfect Serum	ColourPop	Eyeliner	Weekly	124.43	3.1	
4645	Magic Highlighter	ColourPop	Makeup Remover	Monthly	70.52	4.9	
5489	Divine Primer	ColourPop	Makeup Remover	Occasional	19.01	4.5	
5994	Ultra Contour	ColourPop	Contour	Weekly	103.66	1.5	
6521	Magic Mascara	ColourPop	Foundation	Occasional	59.40	3.6	
6638	Divine BB Cream	ColourPop	Serum	Monthly	92.26	1.6	
6662	Perfect Serum	ColourPop	Exfoliator	Daily	20.93	2.0	
6803	Super CC Cream	ColourPop	Blush	Occasional	88.46	4.1	
7002	Perfect Primer	ColourPop	Powder	Monthly	101.37	4.8	
7117	Ultra Face Mask	ColourPop	Makeup Remover	Occasional	99.83	3.7	
7119	Ultra Setting Spray	ColourPop	BB Cream	Monthly	117.18	3.8	
7299	Perfect Setting Spray	ColourPop	Exfoliator	Weekly	105.33	2.0	

	Product_Name	Brand	Category	Usage_Frequency	Price_USD	Rating	Number_of_Reviews
7567	Ultra Highlighter	ColourPop	Blush	Occasional	74.46	1.5	
7603	Perfect Eye Shadow	ColourPop	BB Cream	Occasional	133.46	4.4	
8118	Super Lipstick	ColourPop	Lipstick	Daily	45.60	3.2	
8208	Perfect Blush	ColourPop	Concealer	Occasional	139.02	2.6	
8241	Divine Cleanser	ColourPop	Lip Liner	Occasional	91.05	3.2	
8358	Super Powder	ColourPop	Lip Liner	Weekly	109.80	4.4	
8508	Ultra Eye Shadow	ColourPop	Blush	Monthly	118.98	3.5	
8957	Perfect Foundation	ColourPop	Serum	Occasional	72.78	3.3	
9054	Ultra CC Cream	ColourPop	CC Cream	Occasional	67.60	4.1	
9818	Magic Lipstick	ColourPop	Highlighter	Weekly	119.64	1.2	
10027	Magic Eye Shadow	ColourPop	Eyeliner	Monthly	11.78	1.8	
10201	Perfect Moisturizer	ColourPop	Face Oil	Occasional	57.55	1.4	
10435	Divine Lip Liner	ColourPop	Lipstick	Weekly	101.57	4.4	
11109	Super Bronzer	ColourPop	Lip Gloss	Occasional	129.70	3.9	
11330	Ultra Lip Liner	ColourPop	CC Cream	Daily	18.28	2.2	
11494	Magic Bronzer	ColourPop	Serum	Daily	130.13	2.2	
12024	Magic Lip Liner	ColourPop	BB Cream	Monthly	28.42	2.4	
12436	Ultra Blush	ColourPop	Eye Shadow	Weekly	34.60	1.3	
13200	Super Primer	ColourPop	Highlighter	Weekly	45.66	3.8	
13298	Magic Blush	ColourPop	Exfoliator	Weekly	141.71	4.8	
13932	Magic Mascara	ColourPop	Eyeliner	Weekly	100.63	1.9	
14744	Ultra Exfoliator	ColourPop	Setting Spray	Monthly	86.79	1.1	
14763	Magic BB Cream	ColourPop	CC Cream	Daily	59.97	2.2	
14847	Magic Eye Shadow	ColourPop	Powder	Occasional	71.02	1.6	

```
In [11]: df[(df['Main_Ingredient']=='Retinol') & (df['Brand']=='ColourPop')].numeric_only=True
```

which top 5 products are use by female and they suit on which skin type and there rating?

In [13]: `df[(df['Product_Name']=='Super Setting Spray') & (df['Gender_Target']=='Female') &`

	Product_Name	Brand	Category	Usage_Frequency	Price_USD	Rating	Number_of_Rev
2005	Super Setting Spray	Kylie Cosmetics	Eyeliner	Weekly	46.36	1.8	5
2483	Super Setting Spray	Drunk Elephant	Serum	Weekly	115.53	3.8	5
2634	Super Setting Spray	E.l.f.	Powder	Occasional	143.63	2.5	3
3292	Super Setting Spray	Becca	Mascara	Daily	72.31	3.4	9
9432	Super Setting Spray	Charlotte Tilbury	Serum	Daily	24.05	3.6	1
9478	Super Setting Spray	Hourglass	Setting Spray	Occasional	51.50	3.9	5
10260	Super Setting Spray	Bobby Brown	Primer	Occasional	56.86	4.4	6
11927	Super Setting Spray	Shiseido	Primer	Monthly	105.88	3.0	7
12447	Super Setting Spray	Too Faced	Primer	Daily	45.94	1.4	5
13088	Super Setting Spray	Juvia's Place	Face Mask	Occasional	148.19	4.5	8

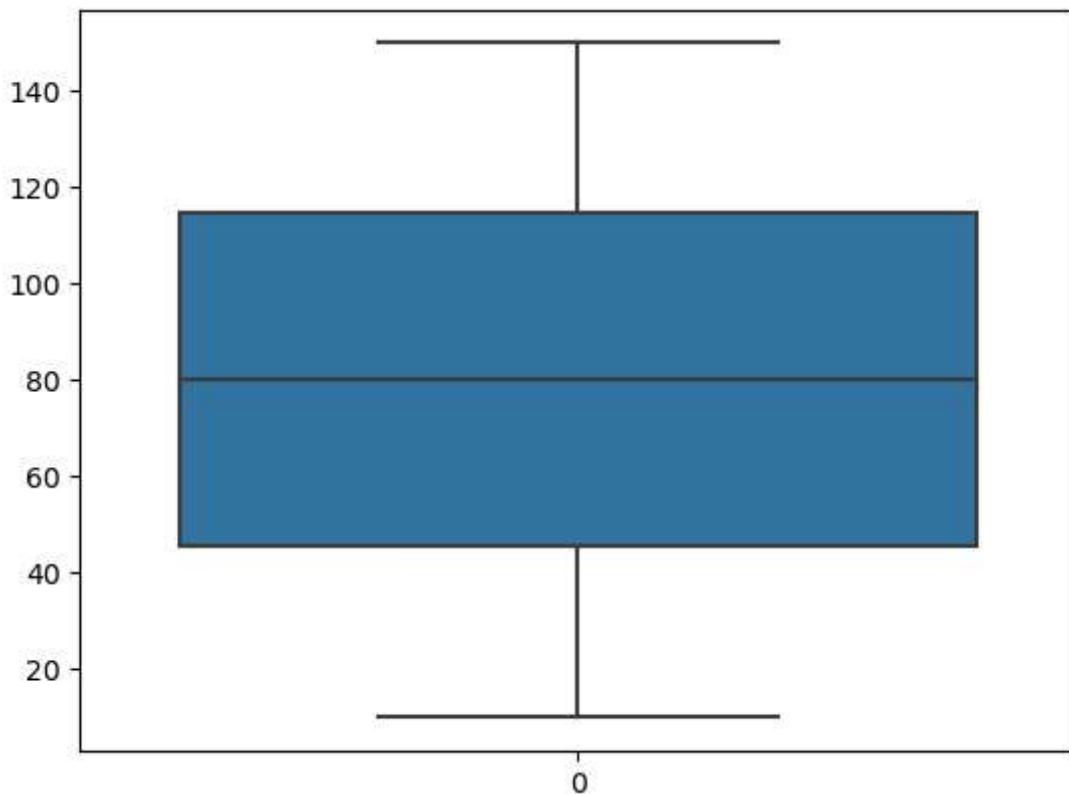


what is dist of price w.r.to products (Costly,medium,min)?

In [72]: `import seaborn as sns`

In [73]: `sns.boxplot(df['Price_USD'])`

Out[73]: <Axes: >



```
In [25]: df['Price_USD'].min()
```

```
Out[25]: 10.0
```

```
In [27]: df['Price_USD'].describe()
```

```
Out[27]: count    15000.000000
mean      80.134108
std       40.402983
min      10.000000
25%      45.480000
50%      80.040000
75%      114.760000
max     149.990000
Name: Price_USD, dtype: float64
```

```
In [44]: costly_price = df[(df['Price_USD'].mean()) &(df['Product_Name']=='Super Setting S
```

```

-----
TypeError                                         Traceback (most recent call last)
File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\ops\array_ops.py:311,
in na_logical_op(x, y, op)
    302     try:
    303         # For exposition, write:
    304         # yarr = isinstance(y, np.ndarray)
(...):
    309         # Then Cases where this goes through without raising include:
    310         # (xint or xbool) and (yint or bool)
--> 311     result = op(x, y)
    312 except TypeError:

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\roperator.py:54, in ra
nd_(left, right)
    53 def rand_(left, right):
---> 54     return operator.and_(right, left)


```

TypeError: ufunc 'bitwise_and' not supported for the input types, and the inputs could not be safely coerced to any supported types according to the casting rule ''safe''

During handling of the above exception, another exception occurred:

```

ValueError                                         Traceback (most recent call last)
File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\ops\array_ops.py:325,
in na_logical_op(x, y, op)
    324     try:
--> 325         result = libops.scalar_binop(x, y, op)
    326     except (
    327         TypeError,
    328         ValueError,
(...):
    331         NotImplementedError,
    332     ) as err:

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\_libs\ops.pyx:180, in panda
s._libs.ops.scalar_binop()


```

ValueError: Buffer dtype mismatch, expected 'Python object' but got 'bool'

The above exception was the direct cause of the following exception:

```

TypeError                                         Traceback (most recent call last)
Cell In[44], line 1
----> 1 costly_price = df[(df['Price_USD'].mean()) &(df['Product_Name']=='Super S
etting Spray')]

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\generic.py:2016, in ND
Frame.__array_ufunc__(self, ufunc, method, *inputs, **kwargs)
    2012 @final
    2013 def __array_ufunc__(
    2014     self, ufunc: np.ufunc, method: str, *inputs: Any, **kwargs: Any
    2015 ):
-> 2016     return arraylike.array_ufunc(self, ufunc, method, *inputs, **kwargs)

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\arraylike.py:273, in a
rray_ufunc(self, ufunc, method, *inputs, **kwargs)
    270 kwargs = _standardize_out_kwarg(**kwargs)
    271 # for binary ops, use our custom dunder methods
--> 273 result = maybe_dispatch_ufunc_to_dunder_op(self, ufunc, method, *inputs, *
kwargs)
    274 if result is not NotImplemented:
    275     return result


```

```

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\_libs\ops_dispatch.pyx:113, in pandas._libs.ops_dispatch.maybe_dispatch_ufunc_to_dunder_op()

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\ops\common.py:81, in _unpack_zerodim_and_defer.<locals>.new_method(self, other)
    77         return NotImplemented
    79     other = item_from_zerodim(other)
--> 81     return method(self, other)

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\arraylike.py:74, in OpsMixin.__rand__(self, other)
    72     @unpack_zerodim_and_defer("__rand__")
    73     def __rand__(self, other):
--> 74         return self._logical_method(other, roperator.rand_)

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\series.py:6107, in Series._logical_method(self, other, op)
    6104     lvalues = self._values
    6105     rvalues = extract_array(other, extract_numpy=True, extract_range=True)
-> 6107     res_values = ops.logical_op(lvalues, rvalues, op)
    6108     return self._construct_result(res_values, name=res_name)

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\ops\array_ops.py:401, in logical_op(left, right, op)
    397     # For int vs int `^`, `|`, `&` are bitwise operators and return
    398     # integer dtypes. Otherwise these are boolean ops
    399     filler = fill_int if is_self_int_dtype and is_other_int_dtype else fill_bool
ol
--> 401     res_values = na_logical_op(lvalues, rvalues, op)
    402     # error: Cannot call function of unknown type
    403     res_values = filler(res_values) # type: ignore[operator]

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\ops\array_ops.py:334,
in na_logical_op(x, y, op)
    326         except (
    327             TypeError,
    328             ValueError,
(...),
    331             NotImplementedError,
    332         ) as err:
    333             typ = type(y).__name__
--> 334             raise TypeError(
    335                 f"Cannot perform '{op.__name__}' with a dtypes [{x.dtype}]"
array "
    336                 f"and scalar of type [{typ}]"
    337             ) from err
    339     return result.reshape(x.shape)

TypeError: Cannot perform 'rand_' with a dtypes [bool] array and scalar of type [bool]

```

In []: costly_price['Price_USD'].mean()

In [70]: data=df[(df['Price_USD']==149)]
data['Product_Name'].value_counts()

Out[70]: Product_Name
Perfect Serum 1
Name: count, dtype: int64

In [43]: costly=df[(df['Price_USD'])]
costly['Product_Name'].value_counts()[0:6]

```

-----
KeyError Traceback (most recent call last)
Cell In[43], line 1
----> 1 costly=df[(df['Price_USD'])]
      2 costly['Product_Name'].value_counts()[0:6]

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\frame.py:3767, in DataFrame.__getitem__(self, key)
    3765     if is_iterator(key):
    3766         key = list(key)
-> 3767     indexer = self.columns._get_indexer_strict(key, "columns")[1]
    3769 # take() does not accept boolean indexers
    3770 if getattr(indexer, "dtype", None) == bool:

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:5877,
in Index._get_indexer_strict(self, key, axis_name)
    5874 else:
    5875     keyarr, indexer, new_indexer = self._reindex_non_unique(keyarr)
-> 5877 self._raise_if_missing(keyarr, indexer, axis_name)
    5879 keyarr = self.take(indexer)
    5880 if isinstance(key, Index):
    5881     # GH 42790 - Preserve name from an Index

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:5938,
in Index._raise_if_missing(self, key, indexer, axis_name)
    5936     if use_interval_msg:
    5937         key = list(key)
-> 5938     raise KeyError(f"None of [{key}] are in the [{axis_name}]")
    5940 not_found = list(ensure_index(key)[missing_mask.nonzero()[0]].unique())
    5941 raise KeyError(f"{not_found} not in index")

KeyError: "None of [Index([ 67.85, 116.43,  90.84,  55.17, 140.56, 135.82, 148.99,
21.43, 113.09, \n          133.57, \n          ... \n          97.2, 142.78, 131.51, 107.48,
14.89, 131.64, 76.35, 65.37, 59.93, \n          36.96], \n          dtype='float64', le
ngth=15000)] are in the [columns]"

```

In [34]: `costly_Products = df[(df['Price_USD'] > 80)]`

In [38]: `costly_Products['Product_Name'].value_counts()`

Out[38]:

Product_Name	count
Magic Face Oil	82
Magic Lip Liner	80
Divine Exfoliator	77
Super Setting Spray	77
Perfect Lip Liner	77
..	
Magic Moisturizer	49
Ultra Cleanser	48
Super Makeup Remover	45
Super Bronzer	45
Ultra Makeup Remover	45
Name: count, Length: 120, dtype: int64	

In [40]: `cheaper_Products = df[(df['Price_USD'] < 80)]`

In [41]: `cheaper_Products['Product_Name'].value_counts()`

```
Out[41]: Product_Name
          Super Exfoliator    78
          Ultra Eye Shadow   78
          Super Setting Spray 77
          Perfect Serum       77
          Super Lip Gloss      77
          ..
          Ultra Primer         48
          Perfect Bronzer      48
          Super Moisturizer     47
          Magic Moisturizer     44
          Magic Exfoliator      37
          Name: count, Length: 120, dtype: int64
```

```
In [48]: df['Main_Ingredient'].unique()
```

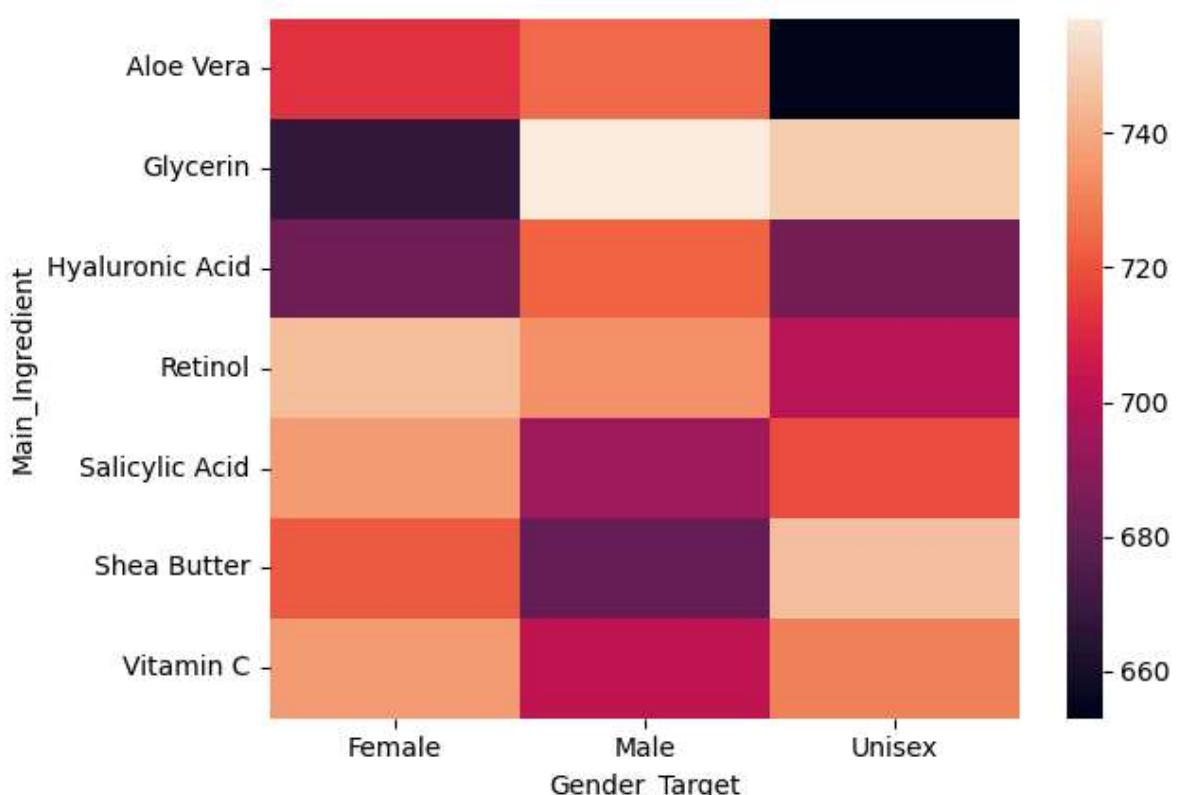
```
Out[48]: array(['Retinol', 'Shea Butter', 'Aloe Vera', 'Glycerin',
   'Hyaluronic Acid', 'Vitamin C', 'Salicylic Acid'], dtype=object)
```

```
In [49]: df['Main_Ingredient'].nunique()
```

```
Out[49]: 7
```

```
In [53]: sns.heatmap(pd.crosstab(df['Main_Ingredient'],df['Gender_Target']))
```

```
Out[53]: <Axes: xlabel='Gender_Target', ylabel='Main_Ingredient'>
```



```
In [52]: pd.crosstab(df['Main_Ingredient'],df['Gender_Target'])
```

Out[52]: Gender_Target Female Male Unisex

Main_Ingredient	Female	Male	Unisex
Aloe Vera	713	725	653
Glycerin	668	757	749
Hyaluronic Acid	683	723	684
Retinol	745	734	701
Salicylic Acid	736	695	718
Shea Butter	721	681	746
Vitamin C	736	702	730

In [22]: pd.crosstab(df['Product_Name'], df['Main_Ingredient'])

Main_Ingredient	Aloe Vera	Glycerin	Hyaluronic Acid	Retinol	Salicylic Acid	Shea Butter	Vitamin C
Product_Name							
Divine BB Cream	13	17	16	25	18	19	16
Divine Blush	15	23	22	18	20	17	17
Divine Bronzer	12	26	18	21	16	14	18
Divine CC Cream	17	21	21	18	13	15	20
Divine Cleanser	15	22	23	16	19	14	17
...
Ultra Moisturizer	17	25	13	11	18	14	18
Ultra Powder	18	14	17	15	26	12	18
Ultra Primer	13	10	17	15	20	17	9
Ultra Serum	22	15	20	16	11	21	14
Ultra Setting Spray	17	21	13	19	17	17	23

120 rows × 7 columns

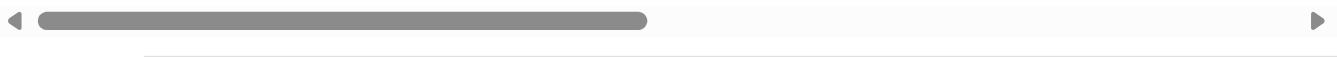
In [75]: male_products=df[(df['Main_Ingredient']=='Glycerin') & (df['Gender_Target']=='Male')}

In [76]: male_products

Out[76]:

	Product_Name	Brand	Category	Usage_Frequency	Price_USD	Rating	Number_of_Reviews
3	Divine Serum	Ilia Beauty	Face Mask	Occasional	55.17	3.2	
31	Magic BB Cream	Juvia's Place	Highlighter	Occasional	46.66	4.6	
36	Divine Eyeliner	Fenty Beauty	Eye Shadow	Occasional	74.51	4.6	
55	Perfect Serum	E.l.f.	Concealer	Occasional	133.56	1.3	
69	Perfect Concealer	Becca	Lipstick	Daily	38.70	3.5	
...
14913	Perfect CC Cream	Huda Beauty	Lip Liner	Monthly	98.70	3.8	
14937	Super Blush	Kylie Cosmetics	Cleanser	Daily	19.46	2.3	
14938	Magic Bronzer	Morphe	Concealer	Daily	102.53	4.5	
14939	Divine Cleanser	Kiehl's	Face Oil	Monthly	149.36	3.4	
14951	Ultra Moisturizer	Yves Saint Laurent	Highlighter	Monthly	51.50	1.6	

757 rows × 14 columns


In [77]: `male_products['Product_Name'].value_counts()`

Out[77]:

Product_Name	
Divine Blush	11
Ultra Bronzer	11
Magic Makeup Remover	10
Divine Setting Spray	10
Divine Mascara	10
	..
Ultra Makeup Remover	2
Magic Highlighter	2
Ultra Lipstick	2
Super Exfoliator	2
Magic Powder	1

Name: count, Length: 119, dtype: int64

In [78]: `male_products=df[(df['Main_Ingredient']=='Aloe Vera') & (df['Gender_Target']=='Male')]`In [80]: `male_products['Product_Name'].value_counts()`

```
Out[80]: Product_Name
Super Contour      13
Super Lip Gloss   12
Divine Eyeliner   12
Magic Concealer   12
Magic Lipstick    11
...
Divine Contour    2
Perfect Lip Gloss 2
Ultra Eyeliner   2
Ultra Primer      2
Magic Eye Shadow  2
Name: count, Length: 120, dtype: int64
```

```
In [81]: male_products=df[(df['Main_Ingredient']=='Hyaluronic Acid') & (df['Gender_Target']=='Male')]
```

```
In [98]: male_products['Category'].value_counts()
```

```
Out[98]: Category
Highlighter      44
Face Mask        43
Setting Spray    40
Face Oil          34
Cleanser          33
BB Cream          33
Lip Gloss          32
Eyeliner          31
Blush              31
Serum              30
Concealer          30
Lip Liner          29
Foundation         29
Powder              29
Mascara             29
Bronzer             29
Lipstick             29
Moisturizer         28
Primer              25
CC Cream             25
Eye Shadow           24
Makeup Remover       24
Contour              22
Exfoliator            20
Name: count, dtype: int64
```

```
In [99]: male_products['Product_Name'].value_counts()
```

```
Out[99]: Product_Name
Perfect Eye Shadow 16
Magic Bronzer       14
Ultra Eye Shadow   13
Super Bronzer       12
Ultra Exfoliator   12
...
Divine Face Mask    2
Perfect Lipstick    2
Super Eyeliner      2
Ultra Cleanser      2
Ultra Foundation    2
Name: count, Length: 118, dtype: int64
```

```
In [100...]: female_products=df[(df['Main_Ingredient']=='Retinol') & (df['Gender_Target']=='Female')]
```

```
In [101]: female_products['Product_Name'].value_counts()
```

```
Out[101]: Product_Name
Perfect BB Cream    13
Ultra Eye Shadow   11
Divine BB Cream    11
Perfect Contour    11
Magic Highlighter  11
..
Super Face Mask    2
Super Moisturizer  2
Magic Lipstick     2
Ultra Contour      2
Magic CC Cream     1
Name: count, Length: 119, dtype: int64
```

```
In [88]: female_products=df[(df['Main_Ingredient']=='Salicylic Acid') & (df['Gender_Target'])]
```

```
In [97]: female_products['Product_Name'].value_counts()
```

```
Out[97]: Product_Name
Perfect Highlighter 12
Perfect Lipstick    11
Perfect Setting Spray 11
Magic Serum          10
Divine Face Mask    10
..
Super BB Cream       2
Magic Eye Shadow    2
Perfect Primer       2
Perfect Concealer    2
Divine Lipstick      1
Name: count, Length: 120, dtype: int64
```

```
In [96]: female_products['Category'].value_counts()
```

```
Out[96]: Category
Bronzer            42
Eyeliner           40
Contour            38
Concealer          36
Setting Spray      35
Highlighter        34
Blush              33
Face Mask          33
Lipstick            33
BB Cream            32
Eye Shadow          32
Primer              32
Exfoliator          30
CC Cream            30
Foundation          29
Powder              29
Face Oil             26
Lip Liner            26
Moisturizer          26
Mascara              25
Serum                25
Cleanser              24
Makeup Remover      23
Lip Gloss             23
Name: count, dtype: int64
```

```
In [33]: df['Category'].nunique()
```

```
Out[33]: 24
```

```
In [105... unisex_products=df[(df['Main_Ingredient']=='Glycerin') & (df['Gender_Target']=='Unis...
```

```
In [106... unisex_products['Product_Name']
```

```
Out[106]: 42          Perfect Face Mask
44          Divine CC Cream
95          Perfect Cleanser
131         Divine Primer
156         Perfect Setting Spray
...
14843        Perfect Concealer
14855        Magic Foundation
14895        Magic Powder
14953        Super Mascara
14954        Divine Blush
Name: Product_Name, Length: 749, dtype: object
```

```
In [107... unisex_products['Category']
```

```
Out[107]: 42          Blush
44          Mascara
95          Bronzer
131         Eyeliner
156         Face Mask
...
14843        Cleanser
14855        Face Mask
14895        Primer
14953        Mascara
14954        Eyeliner
Name: Category, Length: 749, dtype: object
```

How does the price of products impact their rating for different gender-targeted products?

```
In [121... def Price_Rating_Gender (Price_USD,Rating,Gender_Target):
    data = f"price {Price_USD} rating {Rating} Gender {Gender_Target}"
    return data;
```

```
In [60]: filter_data=data[(df['Price_USD'] > 80) & (df['Rating'] > 3) & (df['Gender_Target'] == 'Male')]
print(filter_data)
```

Empty DataFrame

Columns: [Product_Name, Brand, Category, Usage_Frequency, Price_USD, Rating, Number_of_Reviews, Product_Size, Skin_Type, Gender_Target, Packaging_Type, Main_Ingredient, Cruelty_Free, Country_of_Origin]

Index: []

C:\Users\Rutuja\AppData\Local\Temp\ipykernel_22340\692443535.py:1: UserWarning: Boolean Series key will be reindexed to match DataFrame index.

```
filter_data=data[(df['Price_USD'] > 80) & (df['Rating'] > 3) & (df['Gender_Target'] == 'Male')]
```

```
In [61]: pd.crosstab(df['Product_Name'], df['Number_of_Reviews'])
```

```
Out[61]: Number_of_Reviews 52 53 55 59 60 61 62 63 64 65 ... 9991 9992 9993 9994 9995
```

Product_Name	52	53	55	59	60	61	62	63	64	65	...	9991	9992	9993	9994	9995
Divine BB Cream	1	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0
Divine Blush	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
Divine Bronzer	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
Divine CC Cream	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
Divine Cleanser	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
...
Ultra Moisturizer	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
Ultra Powder	0	0	0	0	0	0	0	0	0	0	...	0	0	1	0	0
Ultra Primer	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
Ultra Serum	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
Ultra Setting Spray	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0

120 rows × 7734 columns

```
In [44]: df['Brand'].value_counts()
```

```
Out[44]:
```

Brand	count
Milk Makeup	426
Make Up For Ever	414
Kiehl's	411
NARS	400
E.l.f.	399
Farsali	399
Yves Saint Laurent	394
Morphe	392
Bite Beauty	392
Sisley	392
KVD Beauty	391
Glossier	388
Becca	388
Huda Beauty	384
RMS Beauty	382
Bobby Brown	380
Ilia Beauty	379
Clinique	377
Rare Beauty	375
Tatcha	374
Too Faced	371
Kylie Cosmetics	370
Shiseido	370
Anastasia Beverly Hills	369
Juvia's Place	368
Drunk Elephant	368
Hourglass	366
Natasha Denona	366
Fenty Beauty	366
Charlotte Tilbury	364
Tarte	361
Danessa Myricks	358
Bourjois	357
Urban Decay	356
Laura Mercier	355
Perricone MD	349
IT Cosmetics	346
Patrick Ta	345
Pat McGrath Labs	330
ColourPop	328

Name: count, dtype: int64

```
In [47]: df['Brand'].value_counts()[-3:]
```

```
Out[47]:
```

Brand	count
Patrick Ta	345
Pat McGrath Labs	330
ColourPop	328

Name: count, dtype: int64

Recommendation Analysis

```
In [30]: def Customer_details(Name):
    skin=input('Enter Your skin:')
    gender=input('Enter your Gender:')
    cat=input('Enter your Categorie:')
    Rating=float(input('Enter your Rating'))
    data=df[(df['Skin_Type']==skin) & (df['Gender_Target']==gender) & (df['Category']
    return data['Product_Name']);
```

```
In [31]: Customer_details('Rutuja')
```

```
Enter Your skin:Dry  
Enter your Gender:Female  
Enter your Categorie:Face Mask  
Enter your Rating4.1  
Out[31]:  
9          Perfect Bronzer  
5210       Ultra Exfoliator  
6221       Magic Eyeliner  
8825       Divine Makeup Remover  
10248      Magic Setting Spray  
10283      Ultra CC Cream  
13593      Magic Setting Spray  
14752      Magic Cleanser  
Name: Product_Name, dtype: object
```

```
In [65]: data=df[(df['Skin_Type']=='Dry') & (df['Gender_Target']=='Female') & (df['Category']
```

```
In [66]: data['Product_Name'].shape
```

```
Out[66]: (32,)
```

```
In [64]: df['Rating'].value_counts()
```

```
Out[64]: Rating
1.5    420
4.8    411
2.6    408
4.5    404
4.6    398
4.7    396
3.5    394
4.9    394
1.8    391
1.7    389
2.1    386
4.2    386
2.2    384
2.9    382
4.3    380
4.0    379
1.3    377
1.2    377
4.1    377
1.1    374
2.0    373
1.6    372
1.9    371
2.8    371
3.4    370
2.3    369
3.3    368
2.7    367
3.1    367
3.0    362
1.4    361
4.4    360
3.9    359
3.2    352
2.5    351
3.8    348
2.4    333
3.6    332
3.7    324
1.0    198
5.0    185
Name: count, dtype: int64
```

```
In [ ]: df['Product_Name']
```