DRESS DATASET

In [1]:

```
#Import the required Libraries.
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Data Cleaning

Data Reading & Data Types

In [2]:

```
data0 = pd.read_csv("../downloads/dress-attribute-sales-dataset/Attribute dataset.csv"
data1 = pd.read_csv("../downloads/dress-attribute-sales-dataset/DressSales.csv")
```

In [3]:

```
data0.head()
```

Out[3]:

	Dress_ID	Style	Price	Rating	Size	Season	NeckLine	SleeveLength	Material	FabricTy
0	1006032852	Sexy	Low	4.6	M	Summer	o-neck	sleevless	NaN	chif
1	1212192089	Casual	Low	0.0	L	Summer	o-neck	Petal	microfiber	N
2	1190380701	vintage	High	0.0	L	Automn	o-neck	full	polyster	N
3	966005983	Brief	Average	4.6	L	Spring	o-neck	full	silk	chif
4	876339541	cute	Low	4.5	M	Summer	o-neck	butterfly	chiffonfabric	chif

In [4]:

```
data1.head()
```

Out[4]:

	Dress_ID	29-08-2013	31-08-2013	09-02-2013	09-04-2013	09-06-2013	09-08-2013	09-10-2013	09-12-2013	14-09-2013	...	24-09-2013	26-09-2013	28-09-2013	2
0	1006032852	2114	2274	2491	2660	2727	2887	2930	3119	3204	...	3554	3624.0	3706	37
1	1212192089	151	275	570	750	813	1066	1164	1558	1756	...	2710	2942.0	3258	33
2	1190380701	6	7	7	7	8	8	9	10	10	...	11	11.0	11	
3	966005983	1005	1128	1326	1455	1507	1621	1637	1723	1746	...	1878	1892.0	1914	19
4	876339541	996	1175	1304	1396	1432	1559	1570	1638	1655	...	2032	2156.0	2252	23

5 rows × 24 columns

In [5]:

```
data0.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 479 entries, 0 to 478
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Dress_ID        479 non-null    int64  
 1   Style           479 non-null    object 
 2   Price           477 non-null    object 
 3   Rating          479 non-null    float64
 4   Size            479 non-null    object 
 5   Season          477 non-null    object 
 6   NeckLine        476 non-null    object 
 7   SleeveLength    477 non-null    object 
 8   Material        360 non-null    object 
 9   FabricType      223 non-null    object 
 10  Decoration      255 non-null    object 
 11  Pattern Type    377 non-null    object 
 12  Recommendation  479 non-null    int64  
dtypes: float64(1), int64(2), object(10)
memory usage: 48.8+ KB
```

In [6]:

```
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 479 entries, 0 to 478
Data columns (total 24 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Dress_ID    479 non-null    int64  
 1   29-08-2013  479 non-null    int64  
 2   31-08-2013  479 non-null    int64  
 3   09-02-2013  479 non-null    int64  
 4   09-04-2013  479 non-null    int64  
 5   09-06-2013  479 non-null    int64  
 6   09-08-2013  479 non-null    int64  
 7   09-10-2013  479 non-null    int64  
 8   09-12-2013  479 non-null    object 
 9   14-09-2013  479 non-null    object 
 10  16-09-2013  479 non-null    object 
 11  18-09-2013  479 non-null    object 
 12  20-09-2013  479 non-null    object 
 13  22-09-2013  479 non-null    object 
 14  24-09-2013  479 non-null    int64  
 15  26-09-2013  257 non-null    float64
 16  28-09-2013  479 non-null    int64  
 17  30-09-2013  222 non-null    float64
 18  10-02-2013  220 non-null    float64
 19  10-04-2013  221 non-null    float64
 20  10-06-2013  479 non-null    int64  
 21  10-08-2013  224 non-null    float64
 22  10-10-2013  224 non-null    float64
 23  10-12-2013  479 non-null    int64  
dtypes: float64(6), int64(12), object(6)
memory usage: 89.9+ KB
```

In [7]:

```
data0.Size.value_counts()
```

Out[7]:

```
M        171
free     165
L         93
S         34
XL        14
small      1
s          1
Name: Size, dtype: int64
```

In [8]:

```
data0.Size.replace({'M':"Medium", 'L':'Large', 'XL':'Extra Large', 'free':'Free', 'S':
```

In [9]:

```
data0.Size.value_counts()
```

Out[9]:

```
Medium         171
Free           165
Large           93
Small           36
Extra Large     14
Name: Size, dtype: int64
```

In [10]:

```
data0.Size.isnull().sum()
```

Out[10]:

```
0
```

In [11]:

```
# Print the value counts of each category in "Size" column.
data0.Size.value_counts()/data0.Size.shape[0] *100.0
```

Out[11]:

```
Medium         35.699374
Free           34.446764
Large          19.415449
Small           7.515658
Extra Large     2.922756
Name: Size, dtype: float64
```

In [12]:

```
#OR

data0.Size.value_counts(normalize = True)*100
```

Out[12]:

```
Medium         35.699374
Free           34.446764
Large          19.415449
Small           7.515658
Extra Large     2.922756
Name: Size, dtype: float64
```

Impute/Remove Missing values

In [13]:

```
# Print the null count of each variables of data0 and data1.
print(data0.info())
print('\n')
print(data1.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 479 entries, 0 to 478
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Dress_ID        479 non-null    int64  
 1   Style           479 non-null    object 
 2   Price           477 non-null    object 
 3   Rating          479 non-null    float64
 4   Size            479 non-null    object 
 5   Season          477 non-null    object 
 6   NeckLine        476 non-null    object 
 7   SleeveLength    477 non-null    object 
 8   Material        360 non-null    object 
 9   FabricType      223 non-null    object 
 10  Decoration      255 non-null    object 
 11  Pattern Type    377 non-null    object 
 12  Recommendation  479 non-null    int64  
dtypes: float64(1), int64(2), object(10)
memory usage: 48.8+ KB
None


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 479 entries, 0 to 478
Data columns (total 24 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Dress_ID    479 non-null    int64  
 1   29-08-2013  479 non-null    int64  
 2   31-08-2013  479 non-null    int64  
 3   09-02-2013  479 non-null    int64  
 4   09-04-2013  479 non-null    int64  
 5   09-06-2013  479 non-null    int64  
 6   09-08-2013  479 non-null    int64  
 7   09-10-2013  479 non-null    int64  
 8   09-12-2013  479 non-null    object 
 9   14-09-2013  479 non-null    object 
 10  16-09-2013  479 non-null    object 
 11  18-09-2013  479 non-null    object 
 12  20-09-2013  479 non-null    object 
 13  22-09-2013  479 non-null    object 
 14  24-09-2013  479 non-null    int64  
 15  26-09-2013  257 non-null    float64
 16  28-09-2013  479 non-null    int64  
 17  30-09-2013  222 non-null    float64
 18  10-02-2013  220 non-null    float64
 19  10-04-2013  221 non-null    float64
 20  10-06-2013  479 non-null    int64  
 21  10-08-2013  224 non-null    float64
 22  10-10-2013  224 non-null    float64
 23  10-12-2013  479 non-null    int64  
dtypes: float64(6), int64(12), object(6)
memory usage: 89.9+ KB
None
```

In [14]:

```
data1['18-09-2013'][data1['18-09-2013']=='removed']
# for column 18-09-2013 there is a value in 263th row as 'removed'
```

Out[14]:

```
263    removed
Name: 18-09-2013, dtype: object
```

In [15]:

```python
# Do the required changes in the "Dress Sales" data set to get null values on string v
data1.loc[data1['09-12-2013']== 'Removed',"09-12-2013"] = np.NaN

data1.loc[data1['14-09-2013']== 'removed',"14-09-2013"] = np.NaN

data1.loc[data1['16-09-2013']== 'removed',"16-09-2013"] = np.NaN

data1.loc[data1['18-09-2013']== 'removed',"18-09-2013"] = np.NaN

data1.loc[data1['20-09-2013']== 'removed',"20-09-2013"] = np.NaN

data1.loc[data1['22-09-2013']== 'Orders',"22-09-2013"] = np.NaN
```

In [16]:

```python
# Convert the object type columns in "Dress Sales" into float type of data type.

data1['18-09-2013']=data1['18-09-2013'].astype(float)
data1['14-09-2013']=data1['14-09-2013'].astype(float)
data1['16-09-2013']=data1['16-09-2013'].astype(float)
data1['20-09-2013']=data1['20-09-2013'].astype(float)
data1['22-09-2013']=data1['22-09-2013'].astype(float)
data1['09-12-2013']=data1['09-12-2013'].astype(float)
```

In [17]:

```python
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 479 entries, 0 to 478
Data columns (total 24 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Dress_ID    479 non-null    int64  
 1   29-08-2013  479 non-null    int64  
 2   31-08-2013  479 non-null    int64  
 3   09-02-2013  479 non-null    int64  
 4   09-04-2013  479 non-null    int64  
 5   09-06-2013  479 non-null    int64  
 6   09-08-2013  479 non-null    int64  
 7   09-10-2013  479 non-null    int64  
 8   09-12-2013  478 non-null    float64
 9   14-09-2013  478 non-null    float64
 10  16-09-2013  478 non-null    float64
 11  18-09-2013  478 non-null    float64
 12  20-09-2013  478 non-null    float64
 13  22-09-2013  478 non-null    float64
 14  24-09-2013  479 non-null    int64  
 15  26-09-2013  257 non-null    float64
 16  28-09-2013  479 non-null    int64  
 17  30-09-2013  222 non-null    float64
 18  10-02-2013  220 non-null    float64
 19  10-04-2013  221 non-null    float64
 20  10-06-2013  479 non-null    int64  
 21  10-08-2013  224 non-null    float64
 22  10-10-2013  224 non-null    float64
 23  10-12-2013  479 non-null    int64  
dtypes: float64(12), int64(12)
memory usage: 89.9 KB
```

In [18]:
```
# Print the null percetange of each column of data1.
(data1.isnull().sum()[data1.isnull().sum()/data1.shape[0]*100.0 >40])[:]
```

Out[18]:
```
26-09-2013    222
30-09-2013    257
10-02-2013    259
10-04-2013    258
10-08-2013    255
10-10-2013    255
dtype: int64
```

In [19]:
```
# Drop the columns in "Dress Sales" which have more than 40% of missing values.
data1.drop(columns =['26-09-2013','30-09-2013','10-02-2013','10-04-2013','10-08-2013',
```

In [20]:
```
data1.isnull().sum()/data1.shape[0]*100.0
```

Out[20]:
```
Dress_ID      0.000000
29-08-2013    0.000000
31-08-2013    0.000000
09-02-2013    0.000000
09-04-2013    0.000000
09-06-2013    0.000000
09-08-2013    0.000000
09-10-2013    0.000000
09-12-2013    0.208768
14-09-2013    0.208768
16-09-2013    0.208768
18-09-2013    0.208768
20-09-2013    0.208768
22-09-2013    0.208768
24-09-2013    0.000000
28-09-2013    0.000000
10-06-2013    0.000000
10-12-2013    0.000000
dtype: float64
```

In [21]:
```
data0.info() #Attribute Dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 479 entries, 0 to 478
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Dress_ID        479 non-null    int64  
 1   Style           479 non-null    object 
 2   Price           477 non-null    object 
 3   Rating          479 non-null    float64
 4   Size            479 non-null    object 
 5   Season          477 non-null    object 
 6   NeckLine        476 non-null    object 
 7   SleeveLength    477 non-null    object 
 8   Material        360 non-null    object 
 9   FabricType      223 non-null    object 
 10  Decoration      255 non-null    object 
 11  Pattern Type    377 non-null    object 
 12  Recommendation  479 non-null    int64  
dtypes: float64(1), int64(2), object(10)
memory usage: 48.8+ KB
```

In [22]:

```
data0.Season.value_counts()
```

Out[22]:

```
Summer    154
Spring    116
Winter     94
Automn     59
winter     45
Autumn      7
spring      2
Name: Season, dtype: int64
```

In [23]:

```
data1.columns
```

Out[23]:

```
Index(['Dress_ID', '29-08-2013', '31-08-2013', '09-02-2013', '09-04-2013',
       '09-06-2013', '09-08-2013', '09-10-2013', '09-12-2013', '14-09-2013',
       '16-09-2013', '18-09-2013', '20-09-2013', '22-09-2013', '24-09-2013',
       '28-09-2013', '10-06-2013', '10-12-2013'],
      dtype='object')
```

In [24]:

```
# Create the four seasons columns in data1, according to the above criteria.
Summer = (data1[['29-08-2013','31-08-2013','09-06-2013','09-08-2013', '10-06-2013']].s

Autumn = (data1[['09-10-2013', '14-09-2013','16-09-2013', '18-09-2013', '20-09-2013',
       '28-09-2013']].sum()).sum()

Winter = (data1[['09-02-2013','10-12-2013']].sum()).sum()

Spring = (data1['09-04-2013'].sum()).sum()
```

In [25]:

```
print(Summer)
print(Autumn)
print(Winter)
print(Spring)
```

```
691907
1363288.0
314990
143600
```

In [26]:

```
data1['Spring'] = data1.apply(lambda x: x['09-04-2013'], axis=1)

data1['Summer'] = data1.apply(lambda x: x['29-08-2013'] + x['31-08-2013']+ x['09-06-20

data1['Winter'] = data1.apply(lambda x: x['09-02-2013'] + x['09-12-2013']+ x['10-12-20

data1['Autumn'] = data1.apply(lambda x: x['09-10-2013'] + x['14-09-2013']+ x['16-09-20
```

In [27]:

```
# calculate the sum of sales in each seasons in data1 i.e. "Dress Sales".
data1.sum()
```

Out[27]:

```
Dress_ID      4.321739e+11
29-08-2013    9.488300e+04
31-08-2013    1.004830e+05
09-02-2013    1.070810e+05
09-04-2013    1.436000e+05
09-06-2013    1.459730e+05
09-08-2013    1.516200e+05
09-10-2013    1.533280e+05
09-12-2013    1.596930e+05
14-09-2013    1.627920e+05
16-09-2013    1.653180e+05
18-09-2013    1.678330e+05
20-09-2013    1.723510e+05
22-09-2013    1.764140e+05
24-09-2013    1.786380e+05
28-09-2013    1.866140e+05
10-06-2013    1.989480e+05
10-12-2013    2.079090e+05
Spring        1.436000e+05
Summer        6.919070e+05
Winter        4.736400e+05
Autumn        1.353543e+06
dtype: float64
```

In [28]:

```
# Now let's merge data1 with data0 with left join manner, so that the information of a
data0 = pd.merge(left=data0,right=data1, how='left', left_on='Dress_ID', right_on='Dre
data0.head()
```

Out[28]:

	Dress_ID	Style	Price	Rating	Size	Season	NeckLine	SleeveLength	Material	Fab
0	1006032852	Sexy	Low	4.6	Medium	Summer	o-neck	sleevless	NaN	
1	1212192089	Casual	Low	0.0	Large	Summer	o-neck	Petal	microfiber	
2	1190380701	vintage	High	0.0	Large	Automn	o-neck	full	polyster	
3	966005983	Brief	Average	4.6	Large	Spring	o-neck	full	silk	
4	876339541	cute	Low	4.5	Medium	Summer	o-neck	butterfly	chiffonfabric	

5 rows × 34 columns

In [29]:

```
# Now Drop the Date columns from data0 as it is already combined into four seasons.
data0.drop(data0.loc[:,'29-08-2013':'10-12-2013'].columns, axis= 1, inplace= True)
```

In [30]:

```
data0.head()
```

Out[30]:

	Dress_ID	Style	Price	Rating	Size	Season	NeckLine	SleeveLength	Material	Fab
0	1006032852	Sexy	Low	4.6	Medium	Summer	o-neck	sleevless	NaN	
1	1212192089	Casual	Low	0.0	Large	Summer	o-neck	Petal	microfiber	
2	1190380701	vintage	High	0.0	Large	Automn	o-neck	full	polyster	
3	966005983	Brief	Average	4.6	Large	Spring	o-neck	full	silk	
4	876339541	cute	Low	4.5	Medium	Summer	o-neck	butterfly	chiffonfabric	

In [31]:

```
# Print the null count of each columns in data0 dataframe i.e. combined data frame of
data0.isnull().sum()
```

Out[31]:

```
Dress_ID            0
Style               0
Price               2
Rating              0
Size                0
Season              2
NeckLine            3
SleeveLength        2
Material          119
FabricType        256
Decoration        224
Pattern Type      102
Recommendation      0
Spring              0
Summer              0
Winter              1
Autumn              2
dtype: int64
```

In [32]:

```
#correcting the spellings.
data0.Season= data0.Season.replace('Automn', "Autumn")

data0.Season= data0.Season.replace('spring', "Spring")

data0.Season= data0.Season.replace('winter', "Winter")
```

In [33]:

```
data0.head()
```

Out[33]:

	Dress_ID	Style	Price	Rating	Size	Season	NeckLine	SleeveLength	Material	Fab
0	1006032852	Sexy	Low	4.6	Medium	Summer	o-neck	sleevless	NaN	
1	1212192089	Casual	Low	0.0	Large	Summer	o-neck	Petal	microfiber	
2	1190380701	vintage	High	0.0	Large	Autumn	o-neck	full	polyster	
3	966005983	Brief	Average	4.6	Large	Spring	o-neck	full	silk	
4	876339541	cute	Low	4.5	Medium	Summer	o-neck	butterfly	chiffonfabric	

Univariate Analysis

In [34]:

```
data0['TotalSales'] = data0.apply(lambda x: x['Summer'] + x['Winter']+ x['Autumn']+ x[
```

In [35]:

```
# Group "Style" categories into "Others" which have less than 50000 sales across all t
data0.groupby(data0.Style).sum()[(data0.groupby(data0.Style).sum()).TotalSales<50000]
```

Out[35]:

	Dress_ID	Rating	Recommendation	Spring	Summer	Winter	Autumn	TotalSales
Style								
Flare	2011575734	0.0	1	71.0	451.0	406.0	1161.0	2089.0
Novelty	6799573268	14.1	3	563.0	2866.0	2504.0	7353.0	13286.0
OL	1194626925	0.0	0	15.0	63.0	53.0	158.0	289.0
fashion	836788720	4.0	0	19.0	91.0	74.0	211.0	395.0

In [36]:

```
#Replace with 'Others'
data0.replace(data0.groupby(data0.Style).sum()[(data0.groupby(data0.Style).sum()).Tota
```

In [37]:

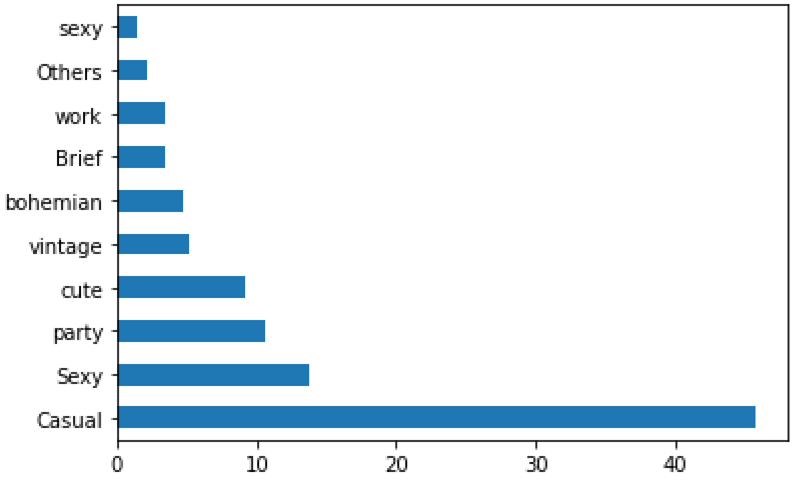
```
round(data0.Style.value_counts(normalize=True)*100.0,1)
```

Out[37]:

```
Casual      45.7
Sexy        13.8
party       10.6
cute         9.2
vintage      5.2
bohemian     4.8
Brief        3.5
work         3.5
Others       2.1
sexy         1.5
Name: Style, dtype: float64
```

In [38]:

```
# Calculate the percentage of each categories in the "Style" variable.
(round(data0.Style.value_counts(normalize=True)*100.0,1)).plot.barh()
plt.show()
```



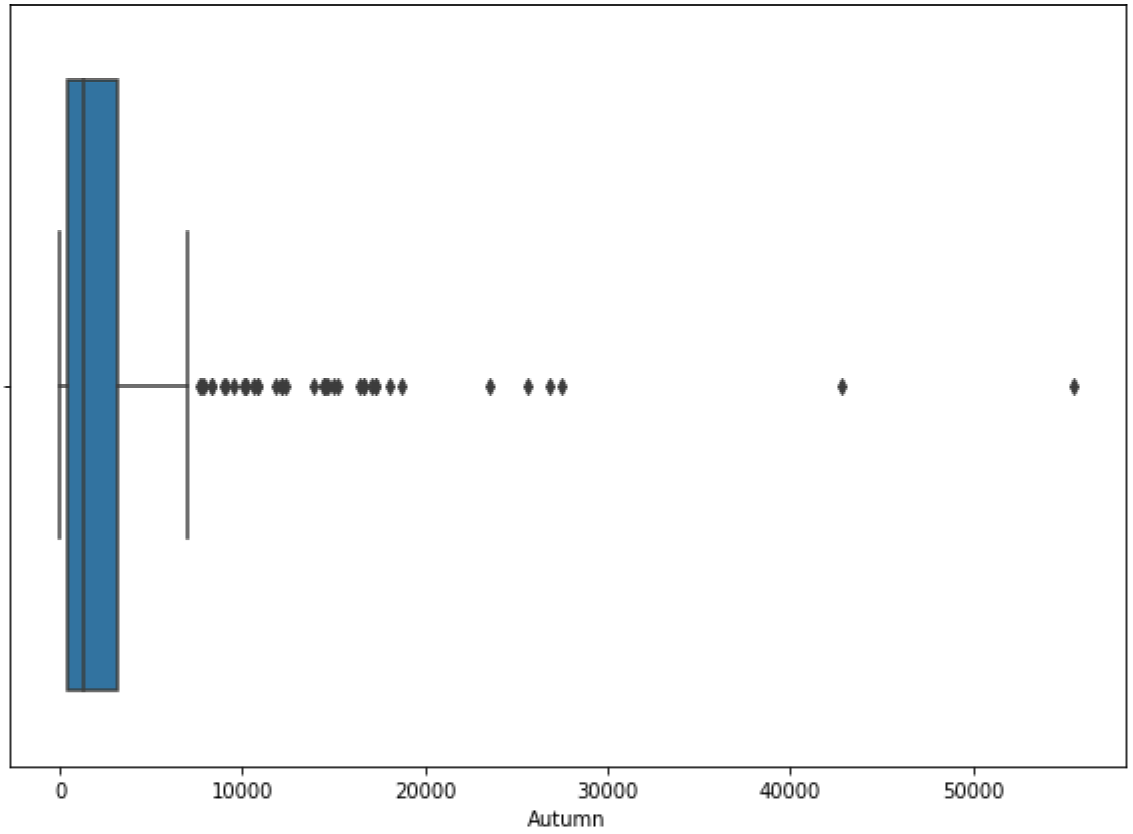
In [39]:
```
# Group "material" categories into "Others" which have less than 25000 sales across al
data0.replace(data0.groupby(data0.Material).sum()[(data0.groupby(data0.Material).sum()
```

In [40]:
```
# Group "fabric type" categories into "Others" which have less than 25000 sales across
data0.replace(data0.groupby(data0.FabricType).sum()[(data0.groupby(data0.FabricType).s
```

In [41]:
```
# Group "patern type" categories into "Others" which have less than 25000 sales across
data0.replace(data0.groupby(data0['Pattern Type']).sum()[(data0.groupby(data0['Patterr
```

In [42]:
```
# Group "decoration" categories into "Others" which have less than 25000 sales across
data0.replace(data0.groupby(data0.Decoration).sum()[(data0.groupby(data0.Decoration).s
```

In [43]:
```
# plot the boxplot of "Autumn" column.
import warnings
warnings.filterwarnings('ignore')
plt.figure(figsize = (10,7))
sns.boxplot(data0.Autumn)
plt.show()
```



In [44]:
```
# Find the maximum and 99th percentile of Winter season.
data0.Winter.max() - data0.Winter.quantile(.99)
```

Out[44]:
```
11469.969999999983
```

In [45]:
```
# Find the maximum and 99th percentile of Winter season.
data0.Summer.max() - data0.Summer.quantile(.99)
```

Out[45]:
```
21623.000000000007
```

In [46]:

```
# Find the maximum and 99th percentile of Spring season.
data0.Autumn.max() - data0.Autumn.quantile(.99)
```

Out[46]:

```
31528.19999999998
```

In [47]:

```
# Find the maximum and 99th percentile of Autumn season.
data0.Spring.max() - data0.Spring.quantile(.99)
```

Out[47]:

```
4717.12
```

Bivariate Analysis

In [48]:

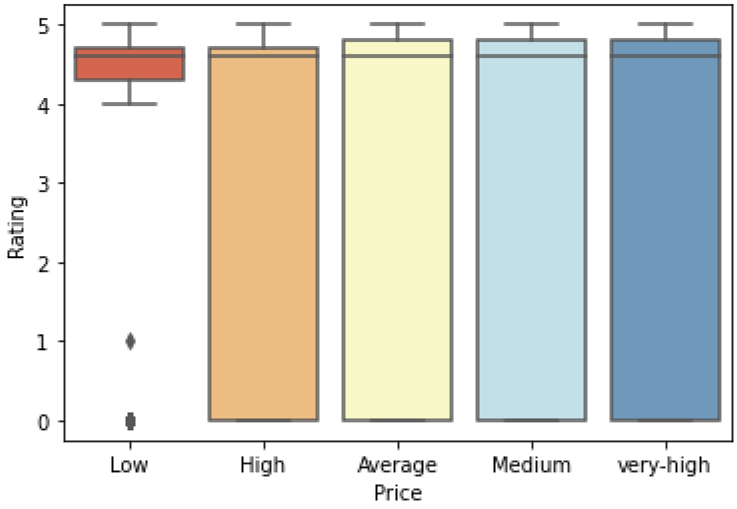
```
data0.groupby(by = 'Price')['Rating'].mean()
```

Out[48]:

```
Price
Average      3.464167
High         2.914286
Low          3.692121
Medium       3.156667
very-high    3.123810
Name: Rating, dtype: float64
```

In [49]:

```
sns.boxplot(data = data0, x = 'Price', y = 'Rating', palette= 'RdYlBu')
plt.show()
```



In [50]:

```
data0.groupby(by = 'Style').Rating.median()
```

Out[50]:

```
Style
Brief       4.60
Casual      4.60
Others      0.00
Sexy        4.55
bohemian    4.60
cute        4.55
party       4.70
sexy        4.50
vintage     4.60
work        4.70
Name: Rating, dtype: float64
```

In [51]:

```
data0.head()
```

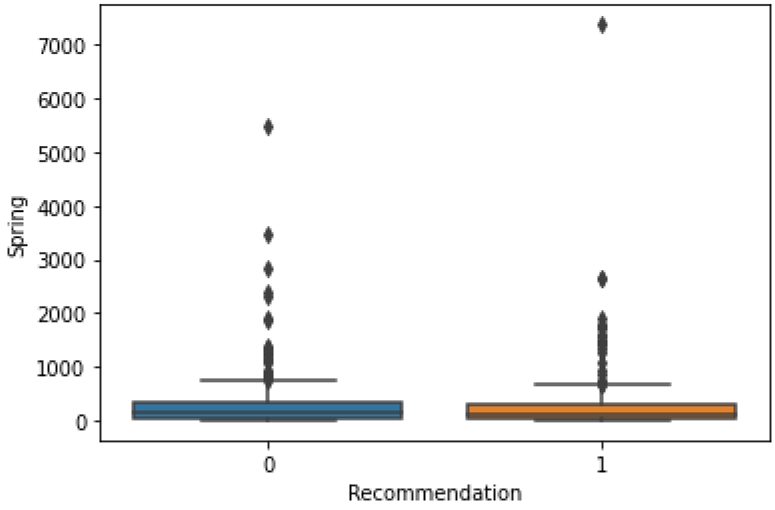
Out[51]:

	Dress_ID	Style	Price	Rating	Size	Season	NeckLine	SleeveLength	Material	Fab
0	1006032852	Sexy	Low	4.6	Medium	Summer	o-neck	sleevless	NaN	
1	1212192089	Casual	Low	0.0	Large	Summer	o-neck	Petal	microfiber	
2	1190380701	vintage	High	0.0	Large	Autumn	o-neck	full	polyster	
3	966005983	Brief	Average	4.6	Large	Spring	o-neck	full	silk	
4	876339541	cute	Low	4.5	Medium	Summer	o-neck	butterfly	chiffonfabric	

In [52]:

```
print(data0.groupby('Recommendation')['Spring'].mean())
sns.boxplot(data=data0, x="Recommendation",y="Spring")
plt.show()
```

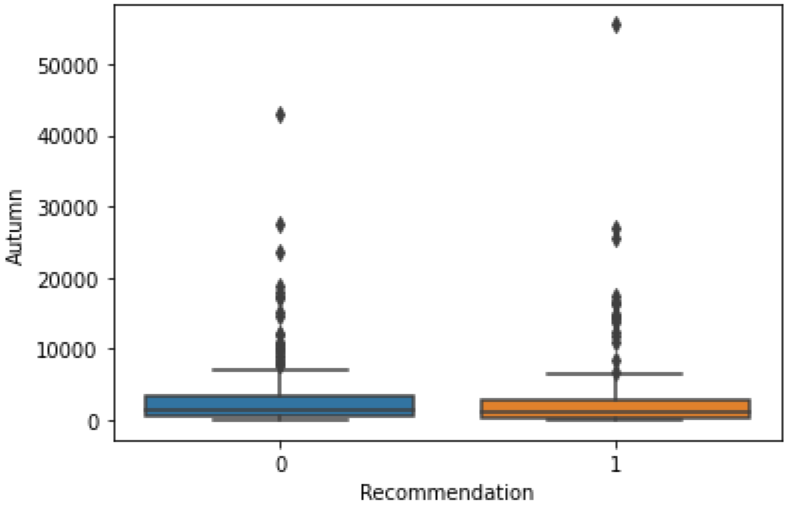
```
Recommendation
0    298.051095
1    302.117073
Name: Spring, dtype: float64
```



In [53]:

```
print(data0.groupby('Recommendation')['Autumn'].mean())
sns.boxplot(data=data0, x="Recommendation",y="Autumn")
plt.show()
```

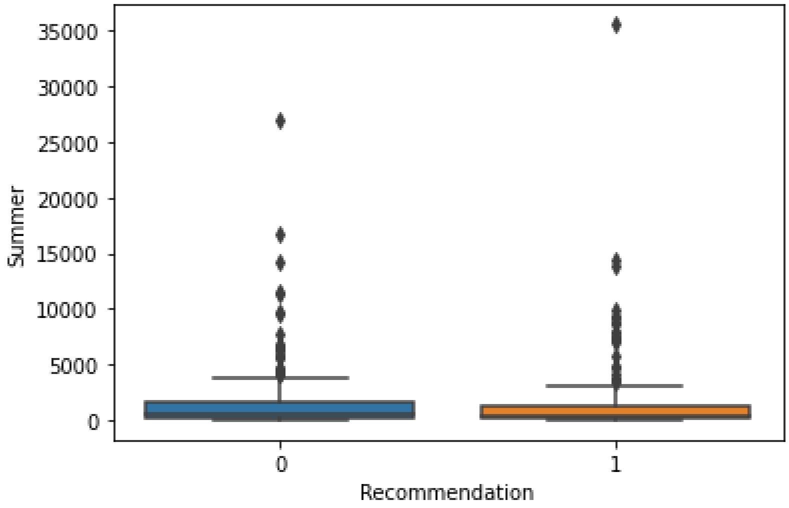
```
Recommendation
0    2840.193431
1    2834.137931
Name: Autumn, dtype: float64
```



In [54]:

```
print(data0.groupby('Recommendation')['Summer'].mean())
sns.boxplot(data=data0, x="Recommendation",y="Summer")
plt.show()
```

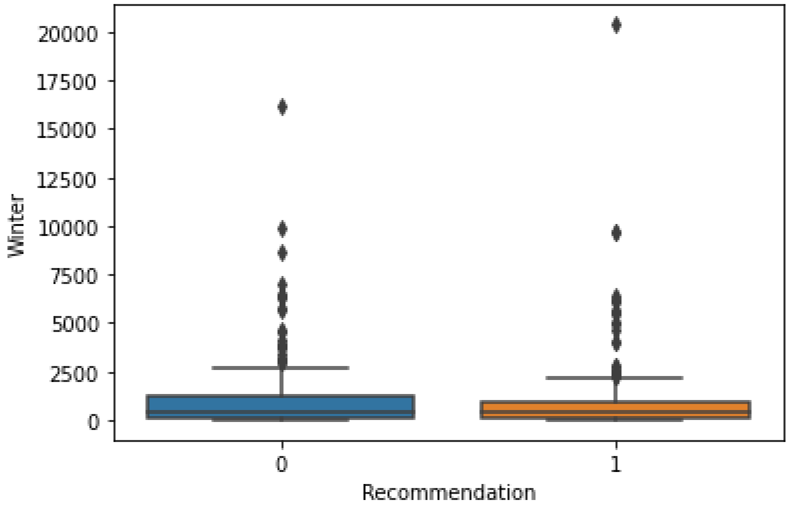
```
Recommendation
0    1430.149635
1    1463.639024
Name: Summer, dtype: float64
```



In [55]:

```
print(data0.groupby('Recommendation')['Winter'].mean())
sns.boxplot(data=data0, x="Recommendation",y="Winter")
plt.show()
```

```
Recommendation
0    985.660584
1    997.887255
Name: Winter, dtype: float64
```



In [56]:

```
res = data0.groupby('Recommendation')[['Winter','Summer','Autumn','Spring']].mean()
res
```

Out[56]:

	Winter	Summer	Autumn	Spring
Recommendation				
0	985.660584	1430.149635	2840.193431	298.051095
1	997.887255	1463.639024	2834.137931	302.117073

In [57]:

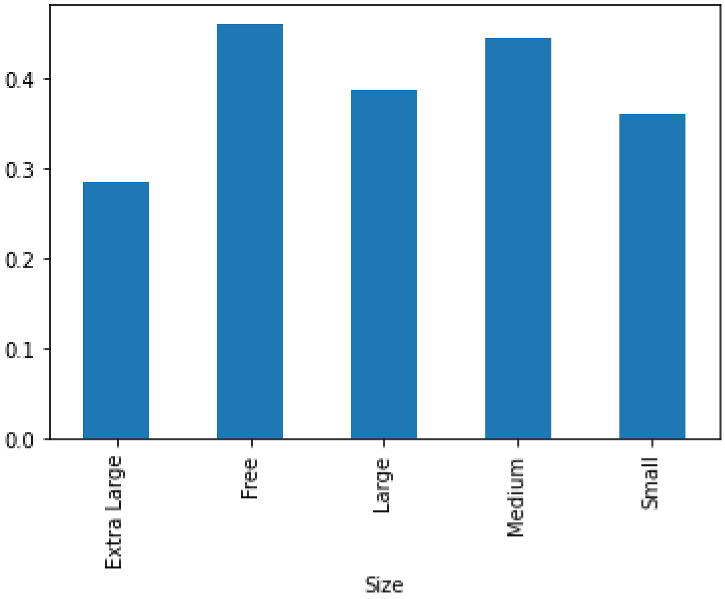
```
data0.groupby('Size')['Recommendation'].sum()
```

Out[57]:

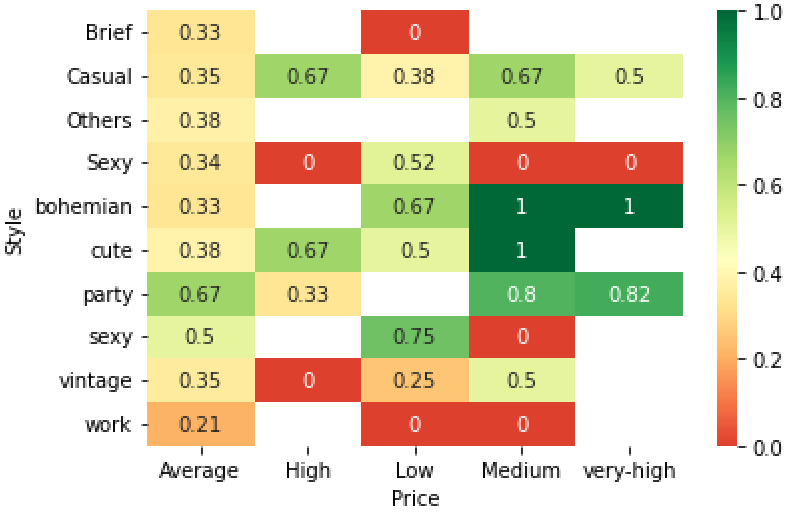
```
Size
Extra Large     4
Free           76
Large          36
Medium         76
Small          13
Name: Recommendation, dtype: int64
```

In [58]:

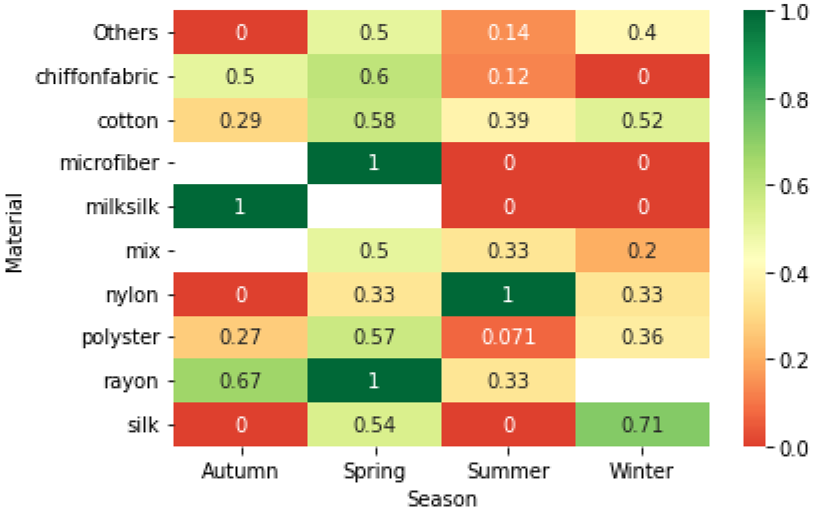
```
data0.groupby(['Size'])['Recommendation'].mean().plot.bar()
plt.show()
```



In [59]:
```
res = pd.pivot_table(data=data0, index="Style", columns="Price", values="Recommendatio
sns.heatmap(res, cmap="RdYlGn", annot=True, center=0.427)
plt.show()
```



In [60]:
```
res = pd.pivot_table(data=data0, index="Material", columns="Season", values="Recommend
sns.heatmap(res, cmap="RdYlGn", annot=True, center=0.427)
plt.show()
```


Others
chiffonfabric
cotton
microfiber
milksilk
mix
nylon
polyster
rayon
silk
Material
Autumn
Spring
Summer
Winter
Season
0 0.5 0.14 0.4
0.5 0.6 0.12 0
0.29 0.58 0.39 0.52
1 0 0
1 0 0
0.5 0.33 0.2
0 0.33 1 0.33
0.27 0.57 0.071 0.36
0.67 1 0.33
0 0.54 0 0.71
1.0
0.8
0.6
0.4
0.2
0.0

```
In [ ]:
```