Working on Real Project with Python
(A part of Big Data Analysis)

Flipkart DataSet

We will analyze this data using the Pandas Library, Numpy, Matplot, Seaborn library....

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
bb = pd.read_csv("flipkart_bbd_transactions.csv")
```

[3]:
```
bb.head()
# bb.tail()
```

[3]:

	Transaction_ID	Year	Season	Category	City	State	Customer_ID	Product_ID	Payment_Mode	MRP	Discount_%	Quantity	Return_Flag	Out_of_Stock
0	T1	2022	Festive	Beauty	Pune	Tamil Nadu	C215	P439	Wallet	36870.02	37	2	0	0
1	T2	2023	Monsoon	Mobiles	Bangalore	West Bengal	C1296	P488	COD	32159.25	39	1	0	0
2	T3	2021	Summer	Electronics	Hyderabad	West Bengal	C1288	P129	Wallet	37994.27	58	1	0	0
3	T4	2023	Winter	Beauty	Pune	Telangana	C1621	P194	Wallet	41038.08	60	4	0	0
4	T5	2023	Summer	Home	Bangalore	Delhi	C383	P261	Wallet	46164.46	19	3	0	0

[4]:
```
bb.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6000 entries, 0 to 5999
Data columns (total 18 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Transaction_ID  6000 non-null   object
 1   Year            6000 non-null   int64
 2   Season          6000 non-null   object
 3   Category        6000 non-null   object
 4   City            6000 non-null   object
 5   State           6000 non-null   object
 6   Customer_ID     6000 non-null   object
 7   Product_ID      6000 non-null   object
 8   Payment_Mode    6000 non-null   object
 9   MRP             6000 non-null   float64
 10  Discount_%      6000 non-null   int64
 11  Quantity        6000 non-null   int64
 12  Return_Flag     6000 non-null   int64
 13  Out_of_Stock    6000 non-null   int64
 14  Hour_of_Day     6000 non-null   int64
 15  Selling_Price   6000 non-null   float64
 16  Revenue         6000 non-null   float64
 17  Margin          6000 non-null   float64
dtypes: float64(4), int64(6), object(8)
memory usage: 843.9+ KB
```

[5]:
```
bb.dtypes
```

[5]:
```
Transaction_ID     object
Year                int64
Season             object
Category           object
City               object
State              object
Customer_ID        object
Product_ID         object
Payment_Mode       object
MRP               float64
Discount_%          int64
Quantity            int64
Return_Flag         int64
Out_of_Stock        int64
Hour_of_Day         int64
Selling_Price     float64
Revenue           float64
Margin            float64
dtype: object
```

[6]:
```
# bb.columns
bb.columns.tolist()
```

[6]:
```
['Transaction_ID',
 'Year',
 'Season',
 'Category',
 'City',
 'State',
```

```
'Customer_ID',
'Product_ID',
'Payment_Mode',
'MRP',
'Discount_%',
'Quantity',
'Return_Flag',
'Out_of_Stock',
'Hour_of_Day',
'Selling_Price',
'Revenue',
'Margin']
```

[7]:
```
bb.shape
```

[7]: (6000, 18)

[8]:
```
# bb.describe()
bb.describe().transpose()
```

[8]:

	count	mean	std	min	25%	50%	75%	max
Year	6000.0	2021.012500	1.420451	2019.00	2020.0000	2021.000	2022.0000	2023.00
MRP	6000.0	24725.430213	14195.993146	205.49	12425.0225	24693.015	36576.1575	49984.84
Discount_%	6000.0	37.076833	18.930654	5.00	21.0000	37.000	54.0000	69.00
Quantity	6000.0	2.503333	1.116332	1.00	2.0000	2.000	4.0000	4.00
Return_Flag	6000.0	0.101500	0.302015	0.00	0.0000	0.000	0.0000	1.00
Out_of_Stock	6000.0	0.046167	0.209863	0.00	0.0000	0.000	0.0000	1.00
Hour_of_Day	6000.0	11.590833	6.928161	0.00	5.0000	12.000	18.0000	23.00
Selling_Price	6000.0	15521.408242	10355.982621	73.98	7134.5300	14025.030	22301.2675	46359.17
Revenue	6000.0	39013.120618	33803.025878	76.00	13843.8450	28740.180	55984.9150	185094.20
Margin	6000.0	22964.929358	22870.283472	39.89	6092.8700	15339.865	32714.8975	133730.76

```
Here, we can see Filkart sells Highest in 2023
```

[9]:
```
bb.isnull().count()
```

[9]:
```
Transaction_ID    6000
Year              6000
Season            6000
Category          6000
City              6000
State             6000
Customer_ID       6000
Product_ID        6000
Payment_Mode      6000
MRP               6000
Discount_%        6000
Quantity          6000
Return_Flag       6000
Out_of_Stock      6000
Hour_of_Day       6000
Selling_Price     6000
Revenue           6000
Margin            6000
dtype: int64
```

[10]:
```
bb.duplicated().sum()
```

[10]: np.int64(0)

[60]:
```
bb.MRP.mean().tolist()
```

[60]: 24725.430213333337

[64]:
```
print( "Most Qty",bb.Quantity.max())
```

```
Most Qty 4
```

[67]:
```
bb["Discount_%"].max()
```

[67]: 69

[73]:
```
bb.Margin.max()
```

[73]: 133730.76

[11]:
```
bb.tail(2)
```

[11]:

	City	State	Customer_ID	Product_ID	Payment_Mode	MRP	Discount_%	Quantity	Return_Flag	Out_of_Stock	Hour_of_Day	Selling_Price	Revenue	Margin
	Delhi	Tamil Nadu	C1920	P417	Debit Card	39251.70	55	4	0	0	11	17663.26	70653.04	86353.76
	Delhi	Maharashtra	C1631	P425	Debit Card	43002.96	23	3	0	0	16	33112.28	99336.84	29672.04

[12]:
```
bb.head(2)
```

[12]:

	Transaction_ID	Year	Season	Category	City	State	Customer_ID	Product_ID	Payment_Mode	MRP	Discount_%	Quantity	Return_Flag	Out_of_Stock	Ho
0	T1	2022	Festive	Beauty	Pune	Tamil Nadu	C215	P439	Wallet	36870.02	37	2	0	0	
1	T2	2023	Monsoon	Mobiles	Bangalore	West Bengal	C1296	P488	COD	32159.25	39	1	0	0	

1. What are all different Category ?

[13]:
```
bb.Category.unique().tolist()
```

[13]:
```
['Beauty', 'Mobiles', 'Electronics', 'Home', 'Grocery', 'Fashion']
```

[14]:
```
bb.head(1)
```

[14]:

	Transaction_ID	Year	Season	Category	City	State	Customer_ID	Product_ID	Payment_Mode	MRP	Discount_%	Quantity	Return_Flag	Out_of_Stock	Hour_of_D
0	T1	2022	Festive	Beauty	Pune	Tamil Nadu	C215	P439	Wallet	36870.02	37	2	0	0	

2. Average Revenue?

[15]:
```
bb.Revenue.mean().tolist()
```

[15]:
```
39013.120618333334
```

3. Most Payment Mode?

[16]:
```
bb.Payment_Mode.unique().tolist()
```

[16]:
```
['Wallet', 'COD', 'Credit Card', 'UPI', 'Debit Card']
```

[17]:
```
bb.Payment_Mode.max()
```

[17]:
```
'Wallet'
```

4. Highest Payment Mode by MRP?

[18]:
```
print("Highest MRP is", bb.MRP.max() )
```

```
Highest MRP is 49984.84
```

[19]:
```
bb.groupby("Payment_Mode")["MRP"].max()
```

[19]:
```
Payment_Mode
COD            49975.54
Credit Card    49972.64
Debit Card     49980.00
UPI            49963.97
Wallet         49984.84
Name: MRP, dtype: float64
```

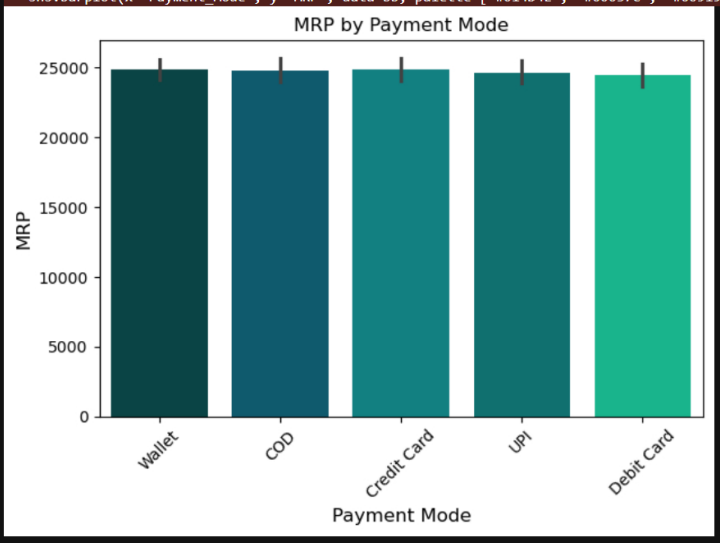
[43]:
```
sns.barplot(x='Payment_Mode', y='MRP', data=bb, palette=['#014D4E', '#00637C', '#009193', '#008080', '#00CC99'])

plt.title('MRP by Payment Mode')
plt.xlabel("Payment Mode", fontsize = 12)
plt.ylabel("MRP", fontsize = 12)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
C:\Users\ACER\AppData\Local\Temp\ipykernel_18940\1681342419.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the sam
e effect.

  sns.barplot(x='Payment_Mode', y='MRP', data=bb, palette=['#014D4E', '#00637C', '#009193', '#008080', '#00CC99'])
```



If you look this then you see there is not a big changes and difference between all modes.

5. Most Payment by Year?

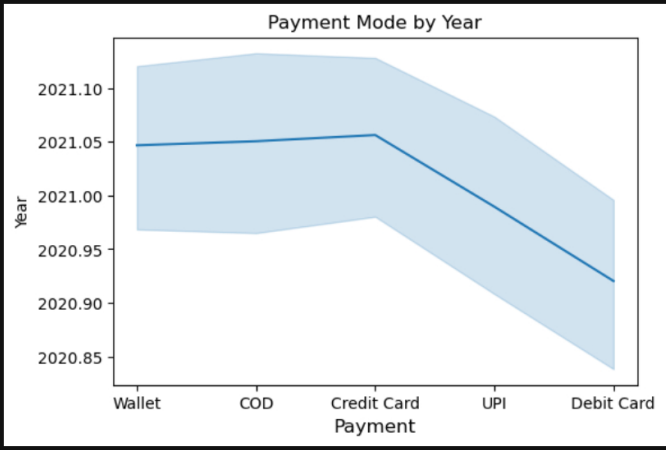
[21]:
```
bb.Year.max()
```

[21]:
```
2023
```

[22]:
```
bb.groupby("Payment_Mode")["Year"].max()
```

[22]:
```
Payment_Mode
COD            2023
Credit Card    2023
Debit Card     2023
UPI            2023
Wallet         2023
Name: Year, dtype: int64
```

[23]:
```
plt.figure(figsize=(6,4))
sns.lineplot(x='Payment_Mode', y='Year', data=bb)
plt.title('Payment Mode by Year')
plt.xlabel("Payment", fontsize = 12)
plt.show()
```



[24]:
```
bb.head(2)
```

[24]:

	Transaction_ID	Year	Season	Category	City	State	Customer_ID	Product_ID	Payment_Mode	MRP	Discount_%	Quantity	Return_Flag	Out_of_Stock	Ho
0	T1	2022	Festive	Beauty	Pune	Tamil Nadu	C215	P439	Wallet	36870.02	37	2	0	0	
1	T2	2023	Monsoon	Mobiles	Bangalore	West Bengal	C1296	P488	COD	32159.25	39	1	0	0	

6. Highest Revenue by State?

[25]:
```
bb.State.max()
```

[25]:
```
'West Bengal'
```

[26]:
```
bb.groupby("State")["Revenue"].max()
```

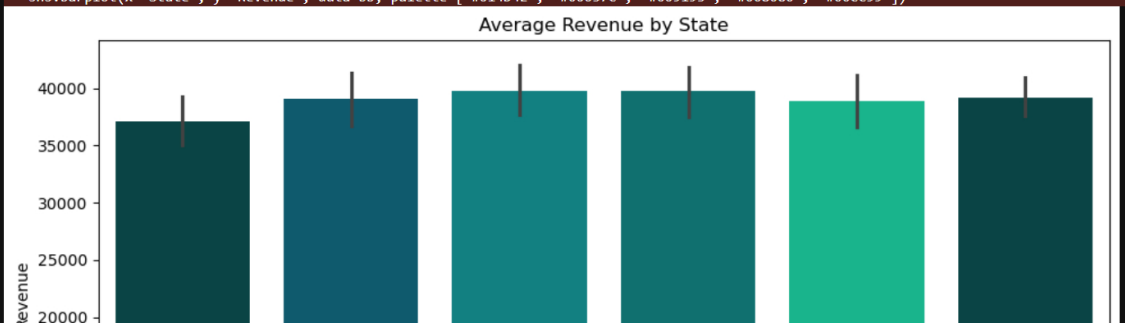
[26]:
```
State
Delhi          176998.32
Karnataka      169834.28
Maharashtra    182947.04
Tamil Nadu     176625.40
Telangana      181862.40
West Bengal    185094.20
Name: Revenue, dtype: float64
```

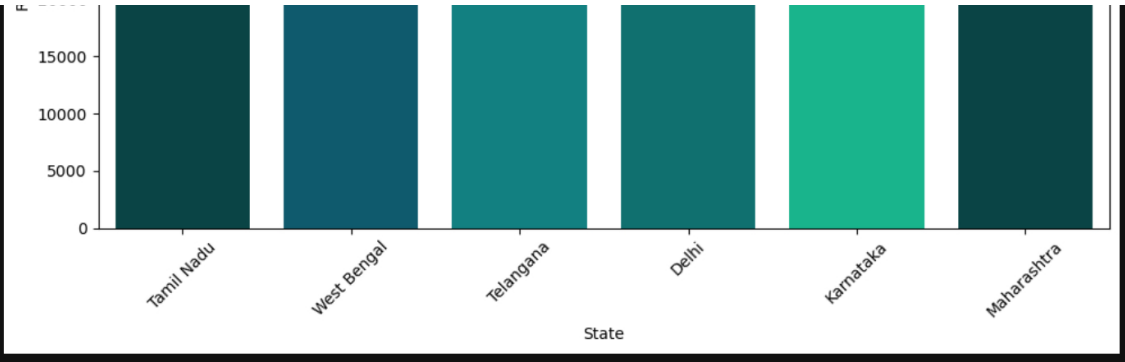
[47]:
```
plt.figure(figsize=(10, 6))
sns.barplot(x="State", y="Revenue", data=bb, palette=['#014D4E', '#00637C', '#009193', '#008080', '#00CC99'])
plt.title("Average Revenue by State")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
C:\Users\ACER\AppData\Local\Temp\ipykernel_18940\1694222907.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the sam
e effect.

  sns.barplot(x="State", y="Revenue", data=bb, palette=['#014D4E', '#00637C', '#009193', '#008080', '#00CC99'])
C:\Users\ACER\AppData\Local\Temp\ipykernel_18940\1694222907.py:2: UserWarning:
The palette list has fewer values (5) than needed (6) and will cycle, which may produce an uninterpretable plot.
  sns.barplot(x="State", y="Revenue", data=bb, palette=['#014D4E', '#00637C', '#009193', '#008080', '#00CC99'])
```



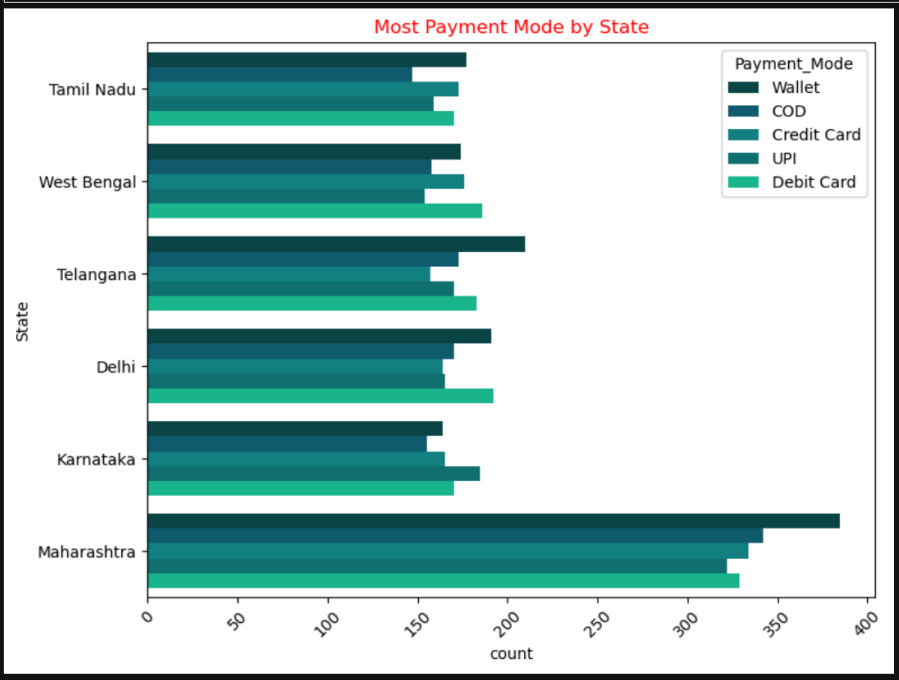


7. Highest Payment Mode by State?

[28]:
```
bb.groupby("State")["Payment_Mode"].max()
```

[28]:
```
State
Delhi          Wallet
Karnataka      Wallet
Maharashtra    Wallet
Tamil Nadu     Wallet
Telangana      Wallet
West Bengal    Wallet
Name: Payment_Mode, dtype: object
```

[29]:
```
plt.figure(figsize=(8, 6))
sns.countplot(y="State", hue="Payment_Mode", data=bb, palette=['#014D4E', '#00637C', '#009193', '#008080', '#00CC99'])  # or use a custom list of HEX codes
plt.title("Most Payment Mode by State", color = "Red")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



8. Most Payment Mode by Category?

[30]:
```
bb.groupby("Category")["Payment_Mode"].max()
```

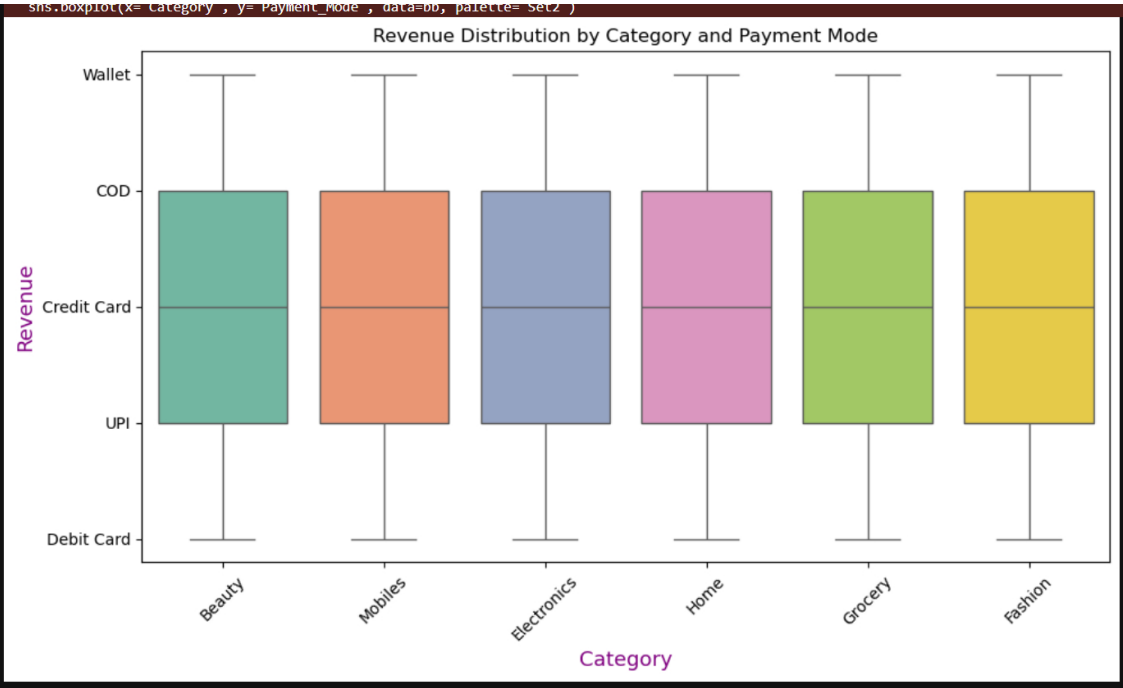
[30]:
```
Category
Beauty         Wallet
Electronics    Wallet
Fashion        Wallet
Grocery        Wallet
Home           Wallet
Mobiles        Wallet
Name: Payment_Mode, dtype: object
```

[54]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(x="Category", y="Payment_Mode", data=bb, palette="Set2")
plt.title("Revenue Distribution by Category and Payment Mode")
plt.xticks(rotation=45)
plt.xlabel("Category", fontsize=13, color="purple")
plt.ylabel("Revenue", fontsize=13, color="purple")
plt.tight_layout()
plt.show()
```

```
C:\Users\ACER\AppData\Local\Temp\ipykernel_18940\2385883387.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the sam
e effect.
```

```
sns.boxplot(x= "Category" , y= "Payment_Mode" , data=bb, palette= "Set2" )
```

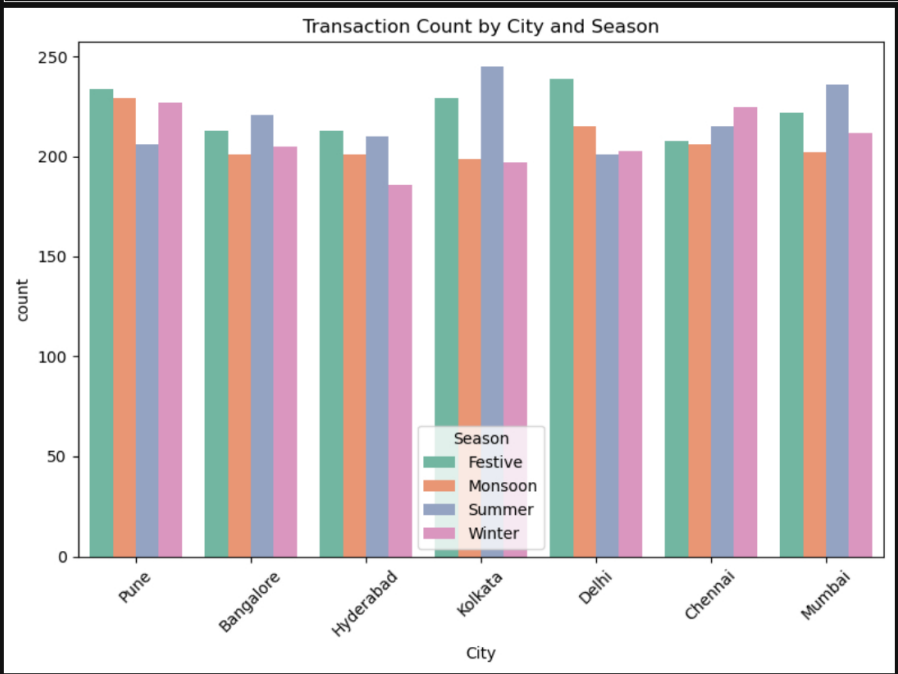


9. Highest Season sell in which City?

[32]:
```
bb.groupby("Season")["City"].max()
```

[32]:
```
Season
Festive    Pune
Monsoon    Pune
Summer     Pune
Winter     Pune
Name: City, dtype: object
```

[48]:
```
plt.figure(figsize=(8, 6))
sns.countplot(x="City", hue="Season", data=bb, palette="Set2")
plt.title("Transaction Count by City and Season")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



[34]:
```
bb.head(3)
```

[34]:

	Transaction_ID	Year	Season	Category	City	State	Customer_ID	Product_ID	Payment_Mode	MRP	Discount_%	Quantity	Return_Flag	Out_of_Stock	H
0	T1	2022	Festive	Beauty	Pune	Tamil Nadu	C215	P439	Wallet	36870.02	37	2	0	0	
1	T2	2023	Monsoon	Mobiles	Bangalore	West Bengal	C1296	P488	COD	32159.25	39	1	0	0	
2	T3	2021	Summer	Electronics	Hyderabad	West Bengal	C1288	P129	Wallet	37994.27	58	1	0	0	

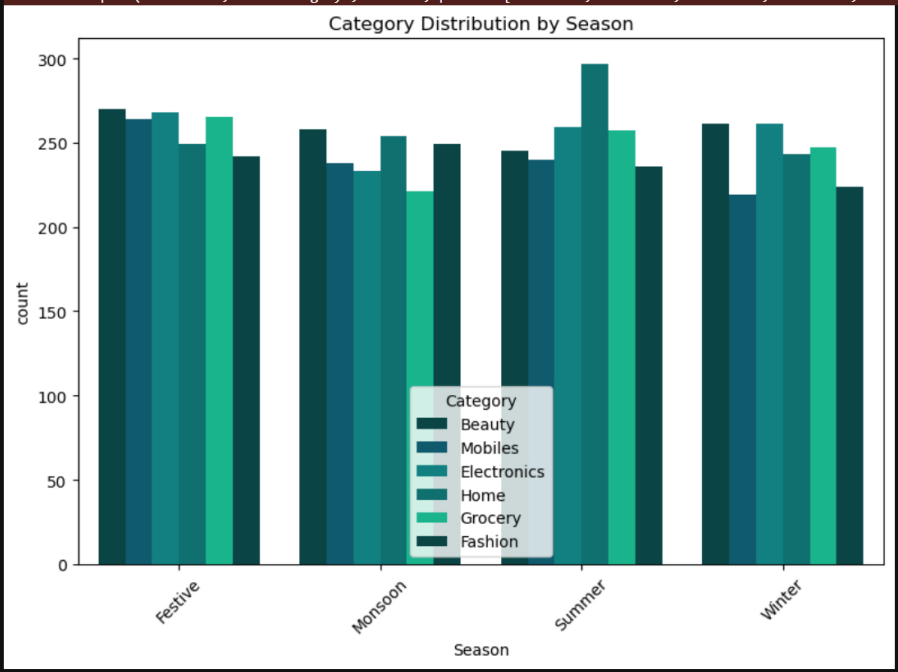
10. Highest category sell by Season?

[35]:
```
bb.groupby("Season")["Category"].max()
```

[35]:
```
Season
Festive    Mobiles
Monsoon    Mobiles
Summer     Mobiles
Winter     Mobiles
Name: Category, dtype: object
```

[36]:
```
plt.figure(figsize=(8, 6))
sns.countplot(x="Season", hue="Category", data=bb, palette=['#014D4E', '#00637C', '#009193', '#008080', '#00CC99'])
plt.title("Category Distribution by Season")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

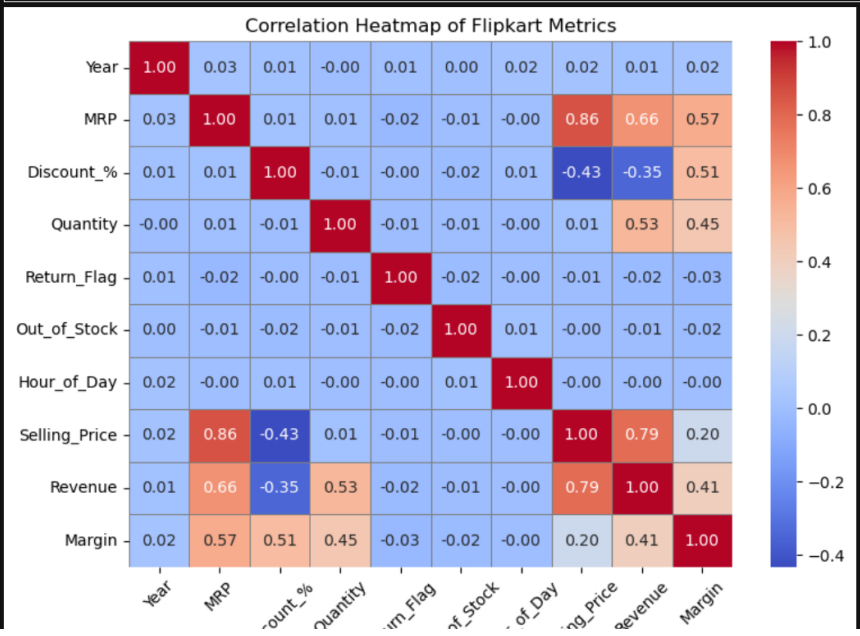
```
C:\Users\ACER\AppData\Local\Temp\ipykernel_18940\3380567972.py:2: UserWarning:
The palette list has fewer values (5) than needed (6) and will cycle, which may produce an uninterpretable plot.
  sns.countplot(x="Season", hue="Category", data=bb, palette=['#014D4E', '#00637C', '#009193', '#008080', '#00CC99'])
```



Heatplot of Numirical Values---

[57]:
```
plt.figure(figsize=(8,6))
corr = bb.corr(numeric_only=True)  # I get so many error becouse I didn't us this!!
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5, linecolor='gray')

plt.title('Correlation Heatmap of Flipkart Metrics')
plt.xticks(rotation=45)
plt.yticks(rotation=0)
plt.tight_layout()
plt.show()
```




Disc
Retu
Out_t
Hour
Sellin

By -- Vicky Raj.