Jupyter Diwali_Sales_Analysis Last Checkpoint: 3 hours ago

File Edit View Run Kernel Settings Help Trusted

Markdown JupyterLab Python 3 (ipykernel)

Diwali Sales Analysis

Importing the liberaries

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

Reading the csv datasets

```
[2]: df = pd.read_csv(r"C:\Users\ANKITA UPADHAYAY\Documents\Diwali Sales Data.csv",encoding= 'unicode_escape')
```

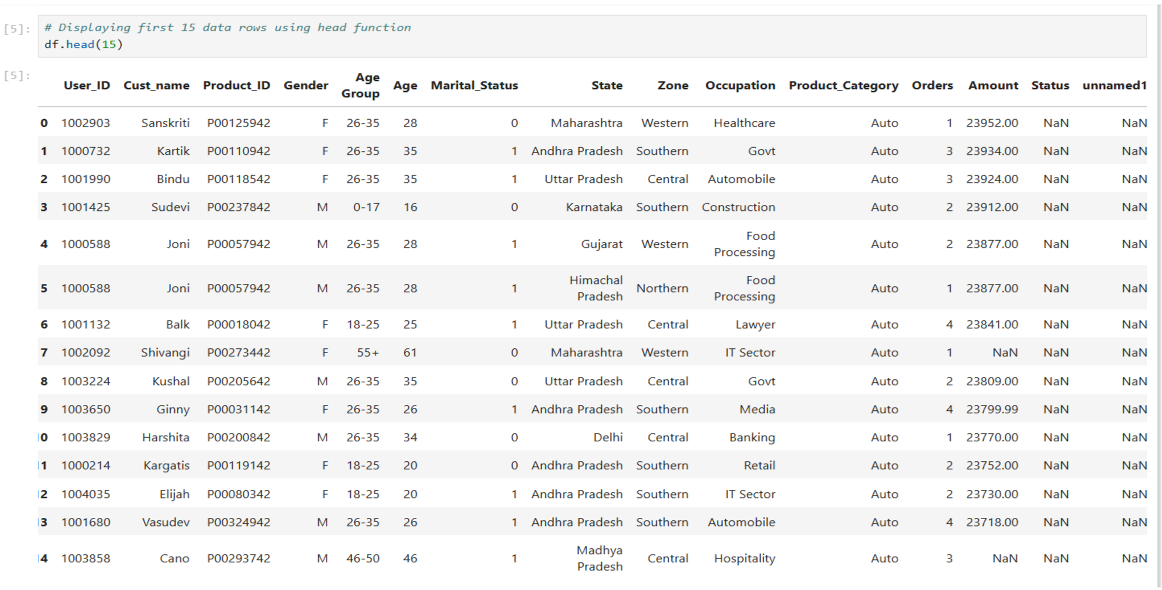
```
[3]: # Finding the shape of the datasets
     df.shape
```

```
[3]: (11251, 15)
```

```
[4]: # Displaying first 4 data rows using head function
     df.head()
```

[4]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount	Status	unnamed1
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952.0	NaN	NaN
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934.0	NaN	NaN
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924.0	NaN	NaN
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912.0	NaN	NaN
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877.0	NaN	NaN

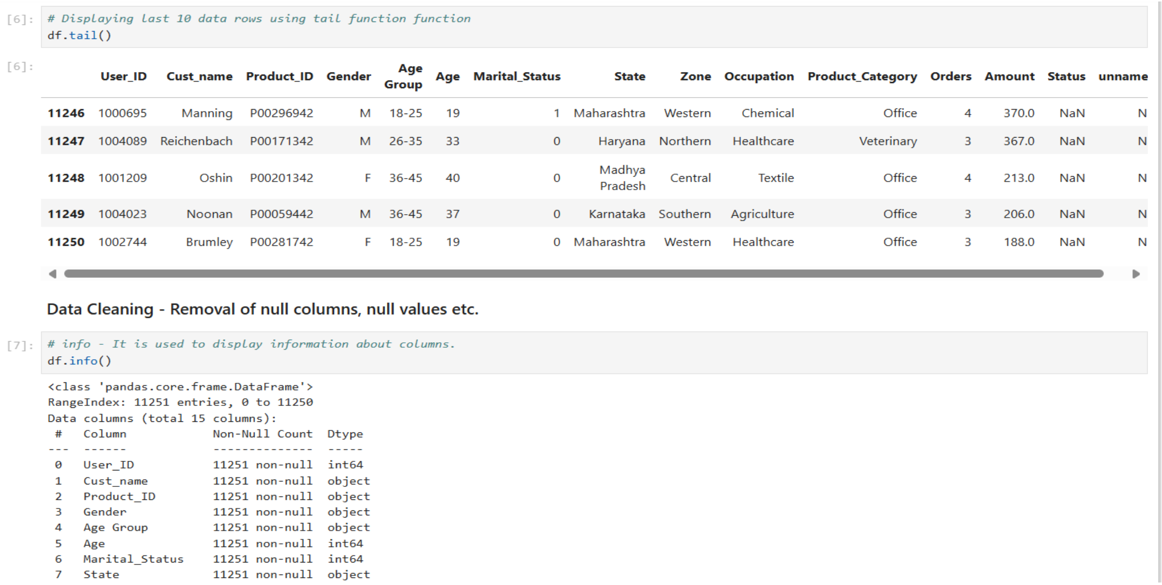


[5]:

```
# Displaying first 15 data rows using head function
df.head(15)
```

[5]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount	Status	unnamed1
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952.00	NaN	NaN
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934.00	NaN	NaN
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924.00	NaN	NaN
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912.00	NaN	NaN
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877.00	NaN	NaN
5	1000588	Joni	P00057942	M	26-35	28	1	Himachal Pradesh	Northern	Food Processing	Auto	1	23877.00	NaN	NaN
6	1001132	Balk	P00018042	F	18-25	25	1	Uttar Pradesh	Central	Lawyer	Auto	4	23841.00	NaN	NaN
7	1002092	Shivangi	P00273442	F	55+	61	0	Maharashtra	Western	IT Sector	Auto	1	NaN	NaN	NaN
8	1003224	Kushal	P00205642	M	26-35	35	0	Uttar Pradesh	Central	Govt	Auto	2	23809.00	NaN	NaN
9	1003650	Ginny	P00031142	F	26-35	26	1	Andhra Pradesh	Southern	Media	Auto	4	23799.99	NaN	NaN
10	1003829	Harshita	P00200842	M	26-35	34	0	Delhi	Central	Banking	Auto	1	23770.00	NaN	NaN
11	1000214	Kargatis	P00119142	F	18-25	20	0	Andhra Pradesh	Southern	Retail	Auto	2	23752.00	NaN	NaN
12	1004035	Elijah	P00080342	F	18-25	20	1	Andhra Pradesh	Southern	IT Sector	Auto	2	23730.00	NaN	NaN
13	1001680	Vasudev	P00324942	M	26-35	26	1	Andhra Pradesh	Southern	Automobile	Auto	4	23718.00	NaN	NaN
14	1003858	Cano	P00293742	M	46-50	46	1	Madhya Pradesh	Central	Hospitality	Auto	3	NaN	NaN	NaN



[6]:

```
# Displaying last 10 data rows using tail function function
df.tail()
```

[6]:

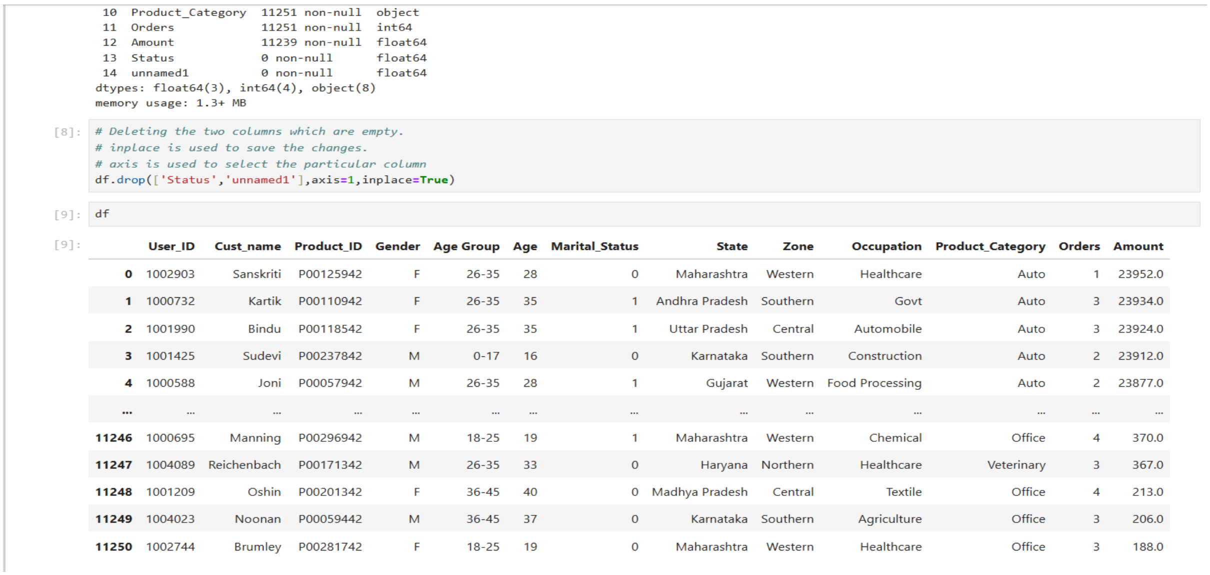
	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount	Status	unname
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemical	Office	4	370.0	NaN	N
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthcare	Veterinary	3	367.0	NaN	N
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	4	213.0	NaN	N
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	3	206.0	NaN	N
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healthcare	Office	3	188.0	NaN	N

Data Cleaning - Removal of null columns, null values etc.

[7]:

```
# info - It is used to display information about columns.
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11251 entries, 0 to 11250
Data columns (total 15 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   User_ID           11251 non-null  int64
 1   Cust_name         11251 non-null  object
 2   Product_ID        11251 non-null  object
 3   Gender            11251 non-null  object
 4   Age Group         11251 non-null  object
 5   Age               11251 non-null  int64
 6   Marital_Status    11251 non-null  int64
 7   State             11251 non-null  object
```



```
 10  Product_Category  11251 non-null  object
 11  Orders            11251 non-null  int64
 12  Amount            11239 non-null  float64
 13  Status            0 non-null      float64
 14  unnamed1          0 non-null      float64
dtypes: float64(3), int64(4), object(8)
memory usage: 1.3+ MB
```

[8]:

```
# Deleting the two columns which are empty.
# inplace is used to save the changes.
# axis is used to select the particular column
df.drop(['Status','unnamed1'],axis=1,inplace=True)
```

[9]:

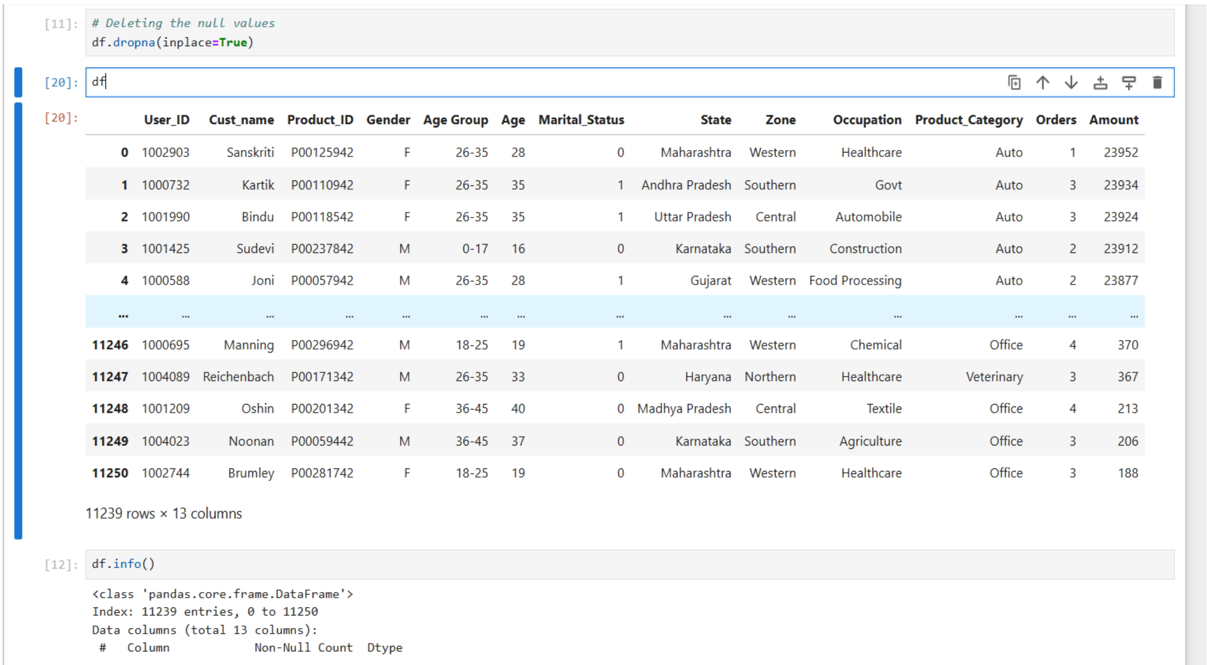
```
df
```

[9]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952.0
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934.0
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924.0
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912.0
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemical	Office	4	370.0
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthcare	Veterinary	3	367.0
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	4	213.0
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	3	206.0
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healthcare	Office	3	188.0

[10]:
```
# Displaying the null values of each column
pd.isnull(df).sum()
```

[10]:
```
User_ID              0
Cust_name            0
Product_ID           0
Gender               0
Age Group            0
Age                  0
Marital_Status       0
State                0
Zone                 0
Occupation           0
Product_Category     0
Orders               0
Amount              12
dtype: int64
```



[11]:
```
# Deleting the null values
df.dropna(inplace=True)
```

[20]:
```
df
```

[20]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877
...	...	...	...	...	...	...	...	...	...	...	...	...	...
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemical	Office	4	370
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthcare	Veterinary	3	367
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	4	213
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	3	206
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healthcare	Office	3	188

11239 rows × 13 columns

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

```
<class 'pandas.core.frame.DataFrame'>
Index: 11239 entries, 0 to 11250
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
```

[13]:
```
# Changing the Amount column data type from float to int using 'astype' function.
df['Amount'] = df['Amount'].astype('int')
```

[14]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 11239 entries, 0 to 11250
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   User_ID           11239 non-null  int64 
 1   Cust_name         11239 non-null  object
 2   Product_ID        11239 non-null  object
 3   Gender            11239 non-null  object
 4   Age Group         11239 non-null  object
 5   Age               11239 non-null  int64 
 6   Marital_Status    11239 non-null  int64 
 7   State             11239 non-null  object
 8   Zone              11239 non-null  object
 9   Occupation        11239 non-null  object
 10  Product_Category  11239 non-null  object
 11  Orders            11239 non-null  int64 
 12  Amount            11239 non-null  int64 
dtypes: int64(5), object(8)
memory usage: 1.2+ MB
```

[15]:
```
df['Amount'].dtypes
```

[15]:
```
dtype('int64')
```

[16]:
```
df.columns
```

[16]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

[17]:

```
# changing column name
df.rename(columns={'Cust_name':'Customer_name'})
```

[17]:

	User_ID	Customer_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877
...	...	...	...	...	...	...	...	...	...	...	...	...	...
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemical	Office	4	370
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthcare	Veterinary	3	367
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	4	213
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	3	206
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healthcare	Office	3	188

11239 rows × 13 columns

[19]:

```
# describe() - It is used to describe the statistical details of the datasets.
df.describe()
```

[19]:

	User_ID	Age	Marital_Status	Orders	Amount
count	1.123900e+04	11239.000000	11239.000000	11239.000000	11239.000000
mean	1.003004e+06	35.410357	0.420055	2.489634	9453.610553
std	1.716039e+03	12.753866	0.493589	1.114967	5222.355168
min	1.000001e+06	12.000000	0.000000	1.000000	188.000000
25%	1.001492e+06	27.000000	0.000000	2.000000	5443.000000
50%	1.003064e+06	33.000000	0.000000	2.000000	8109.000000
75%	1.004426e+06	43.000000	1.000000	3.000000	12675.000000
max	1.006040e+06	92.000000	1.000000	4.000000	23952.000000

Exploratary Data Analysis

[61]:

```
df.columns
```

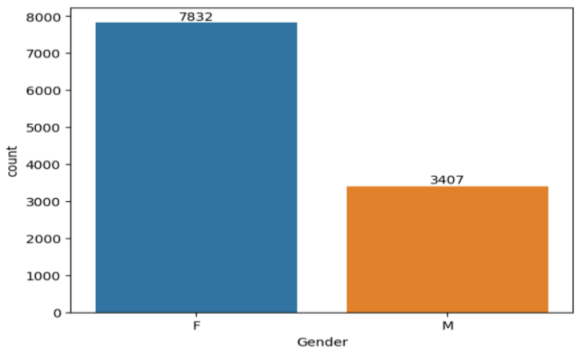
[61]:

```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

[142]:

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

# Specify colors for the bars
#palette = {"M": "skyblue", "F": "orange"}

# Create the count plot with the correct palette parameter
ax = sns.countplot(x='Gender', data=df , palette="tab10")

# Add labels to the bars
for bars in ax.containers:
    ax.bar_label(bars)

# Show the plot
plt.show()
```

```
C:\Users\ANKITA UPADHAYAY\AppData\Local\Temp\ipykernel_2000\2407492814.py:8: 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 s
ame effect.

  ax = sns.countplot(x='Gender', data=df , palette="tab10")
```



```
[143]: # Grouping by gender and finding the sum of amount and then sorting
       Sales_gen=df.groupby(['Gender'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)
```

```
[144]: Sales_gen
```

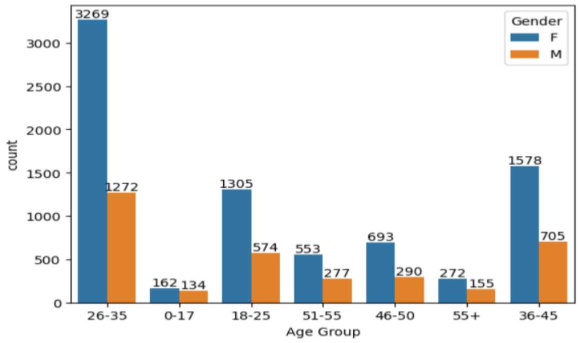
[144]:

	Gender	Amount
0	F	74335853
1	M	31913276

The graph shows that most purchases are done by female as compared to male.

Age

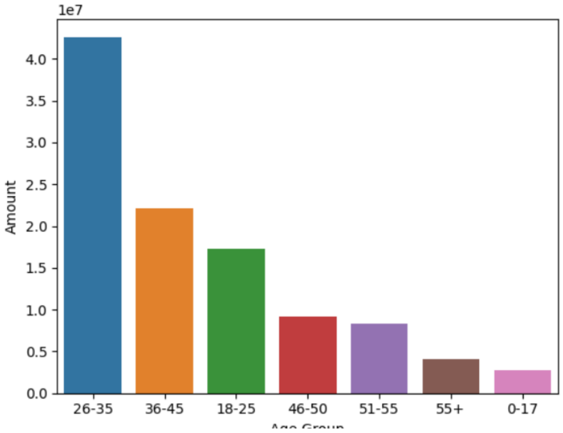
```
[146]: ax = sns.countplot(data = df, x = 'Age Group', hue = 'Gender',palette="tab10")

       for bars in ax.containers:
           ax.bar_label(bars)
```



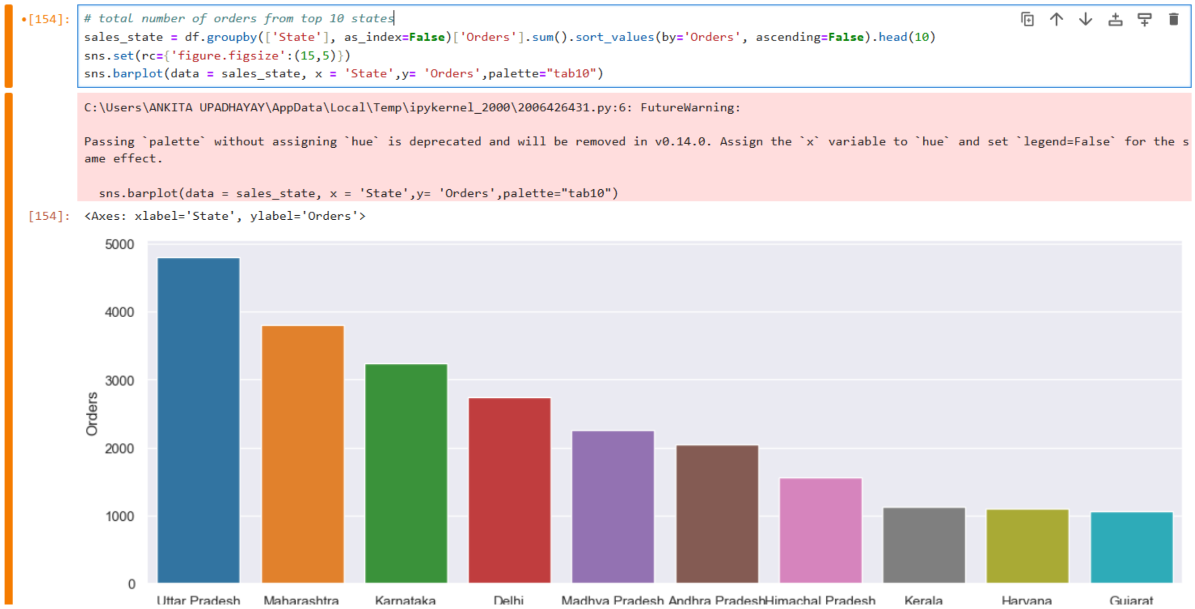
```
[5]: #grouping the column age wise
     Sales_age = df.groupby(['Age Group'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)
     sns.barplot(x = 'Age Group',y= 'Amount' ,data = Sales_age,palette="tab10")
```

```
C:\Users\ANKITA UPADHAYAY\AppData\Local\Temp\ipykernel_21268\2603042376.py:3: 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 s
ame effect.

  sns.barplot(x = 'Age Group',y= 'Amount' ,data = Sales_age,palette="tab10")
```

```
[5]: <Axes: xlabel='Age Group', ylabel='Amount'>
```



State

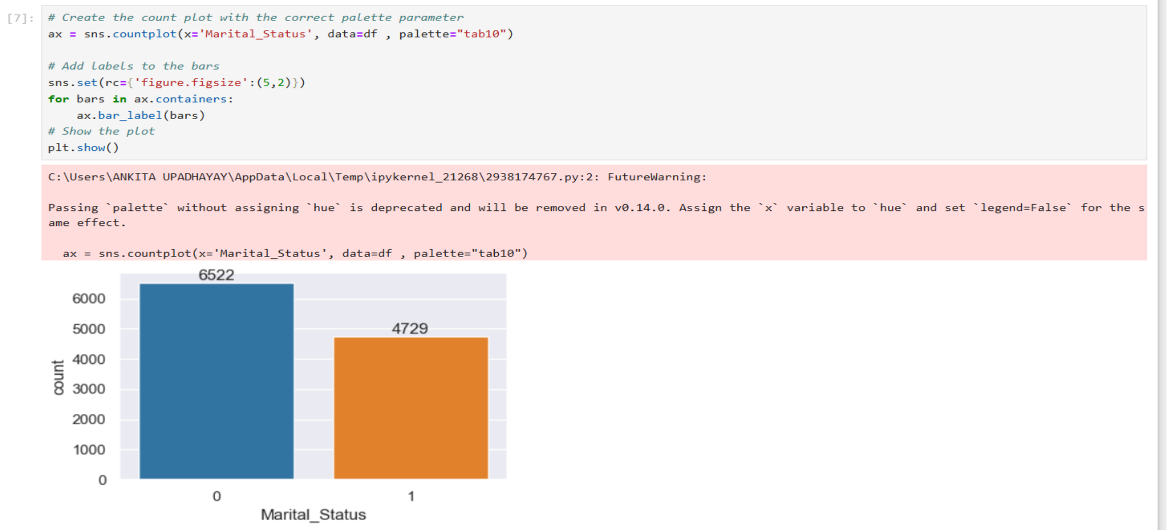


```
•[154]: # total number of orders from top 10 states
        sales_state = df.groupby(['State'], as_index=False)['Orders'].sum().sort_values(by='Orders', ascending=False).head(10)
        sns.set(rc={'figure.figsize':(15,5)})
        sns.barplot(data = sales_state, x = 'State',y= 'Orders',palette="tab10")
```

```
C:\Users\ANKITA UPADHAYAY\AppData\Local\Temp\ipykernel_2000\2006426431.py:6: 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 s
ame effect.

  sns.barplot(data = sales_state, x = 'State',y= 'Orders',palette="tab10")
```

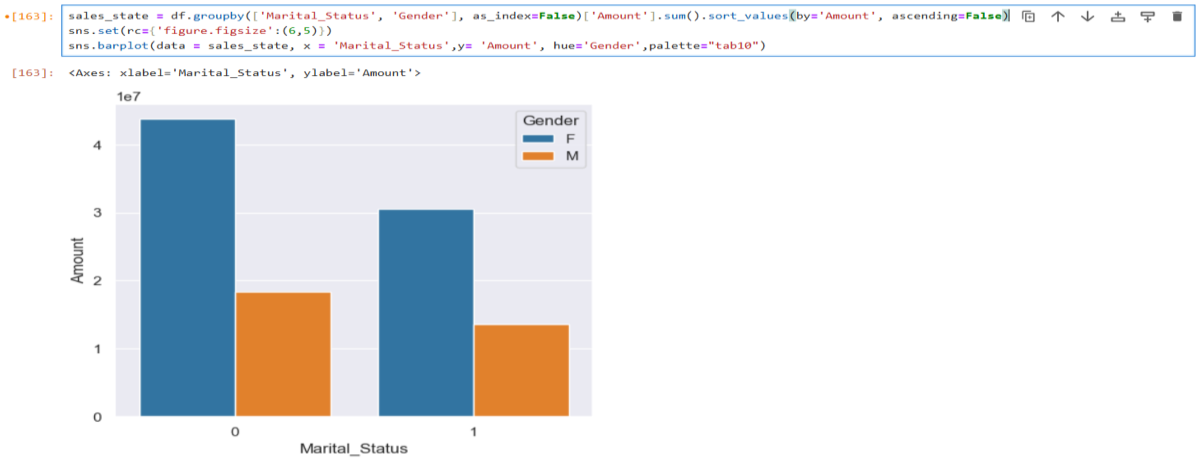
```
[154]: <Axes: xlabel='State', ylabel='Orders'>
```

Marital Status



```
[7]: # Create the count plot with the correct palette parameter
     ax = sns.countplot(x='Marital_Status', data=df , palette="tab10")

     # Add labels to the bars
     sns.set(rc={'figure.figsize':(5,2)})
     for bars in ax.containers:
         ax.bar_label(bars)
     # Show the plot
     plt.show()
```

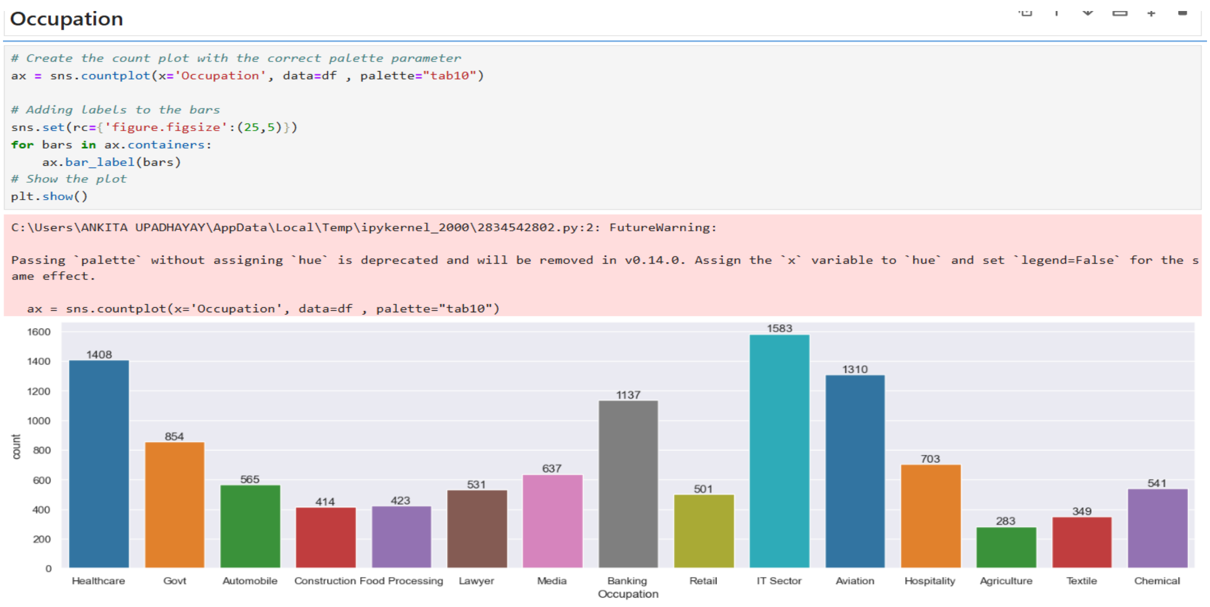
```
C:\Users\ANKITA UPADHAYAY\AppData\Local\Temp\ipykernel_21268\2938174767.py:2: FutureWarning:

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 s
ame effect.

  ax = sns.countplot(x='Marital_Status', data=df , palette="tab10")
```



```
•[163]: sales_state = df.groupby(['Marital_Status', 'Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)
        sns.set(rc={'figure.figsize':(6,5)})
        sns.barplot(data = sales_state, x = 'Marital_Status',y= 'Amount', hue='Gender',palette="tab10")
```

```
[163]: <Axes: xlabel='Marital_Status', ylabel='Amount'>
```

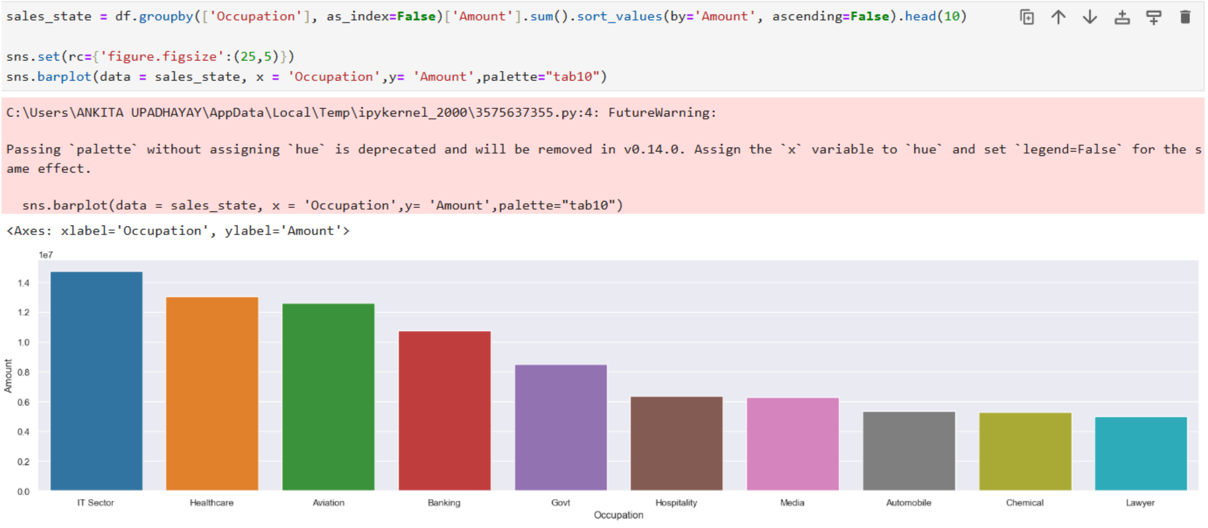
From above graph we can understand that most of the purchaser are married women.



Occupation

```
# Create the count plot with the correct palette parameter
ax = sns.countplot(x='Occupation', data=df , palette="tab10")

# Adding labels to the bars
sns.set(rc={'figure.figsize':(25,5)})
for bars in ax.containers:
    ax.bar_label(bars)
# Show the plot
plt.show()
```

```
C:\Users\ANKITA UPADHAYAY\AppData\Local\Temp\ipykernel_2000\2834542802.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 s
ame effect.

  ax = sns.countplot(x='Occupation', data=df , palette="tab10")
```

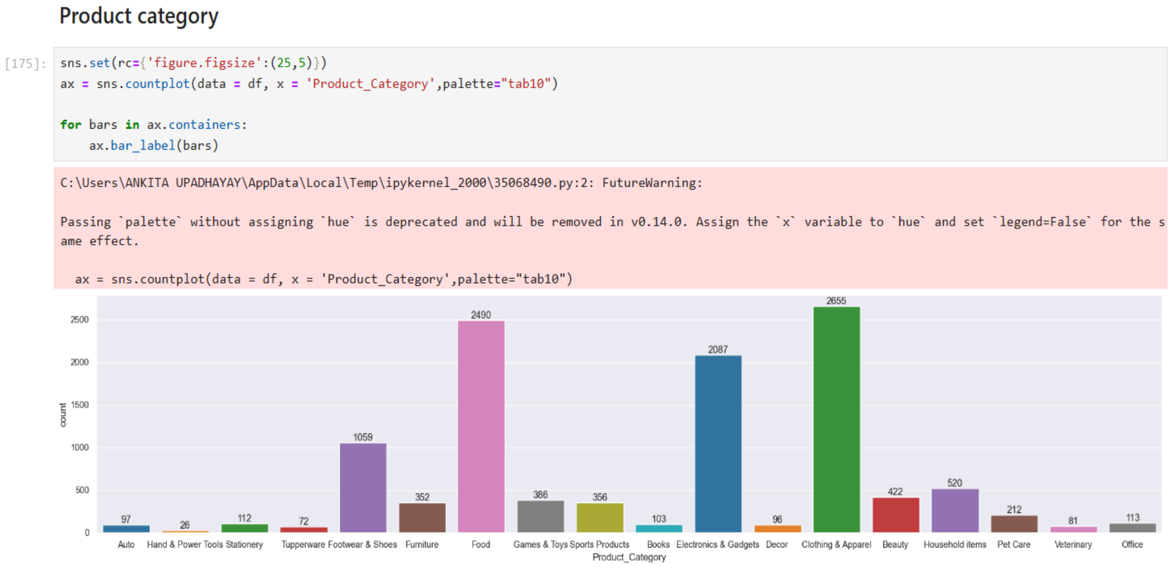


```
sales_state = df.groupby(['Occupation'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False).head(10)

sns.set(rc={'figure.figsize':(25,5)})
sns.barplot(data = sales_state, x = 'Occupation',y= 'Amount',palette="tab10")
```

```
C:\Users\ANKITA UPADHAYAY\AppData\Local\Temp\ipykernel_2000\3575637355.py:4: 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 s
ame effect.

  sns.barplot(data = sales_state, x = 'Occupation',y= 'Amount',palette="tab10")
```

```
<Axes: xlabel='Occupation', ylabel='Amount'>
```

From above graph , we can see that mostly purchaser arw from IT sector, Healtcare,Aviation ,Banking and followed by other occupation.



Product category

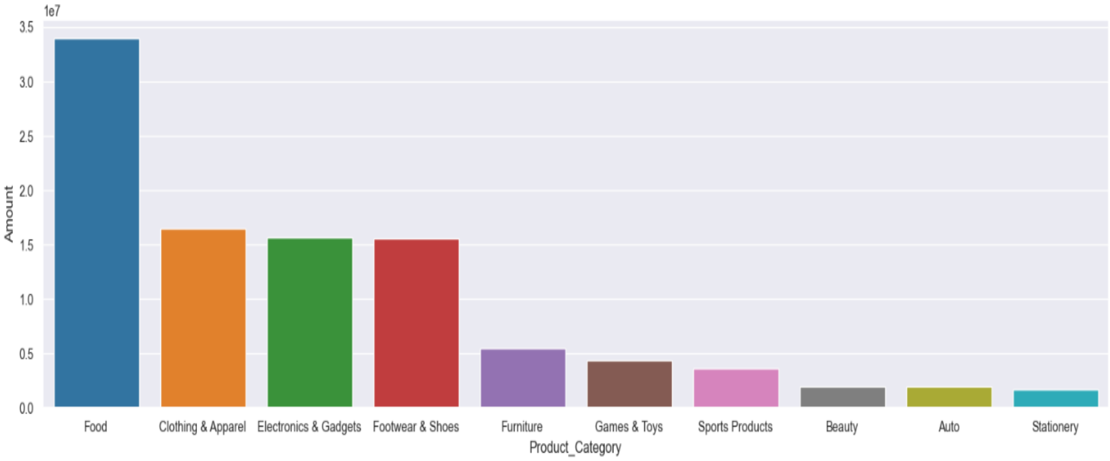
```
[175]: sns.set(rc={'figure.figsize':(25,5)})
       ax = sns.countplot(data = df, x = 'Product_Category',palette="tab10")

       for bars in ax.containers:
           ax.bar_label(bars)
```

```
C:\Users\ANKITA UPADHAYAY\AppData\Local\Temp\ipykernel_2000\35068490.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 s
ame effect.

  ax = sns.countplot(data = df, x = 'Product_Category',palette="tab10")
```

•[177]:
```
sales_state = df.groupby(['Product_Category'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False).head(10)
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x = 'Product_Category',y= 'Amount',palette="tab10")
```

```
C:\Users\ANKITA UPADHAYAY\AppData\Local\Temp\ipykernel_2000\3099995965.py:4: 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 s
ame effect.

  sns.barplot(data = sales_state, x = 'Product_Category',y= 'Amount',palette="tab10")
```

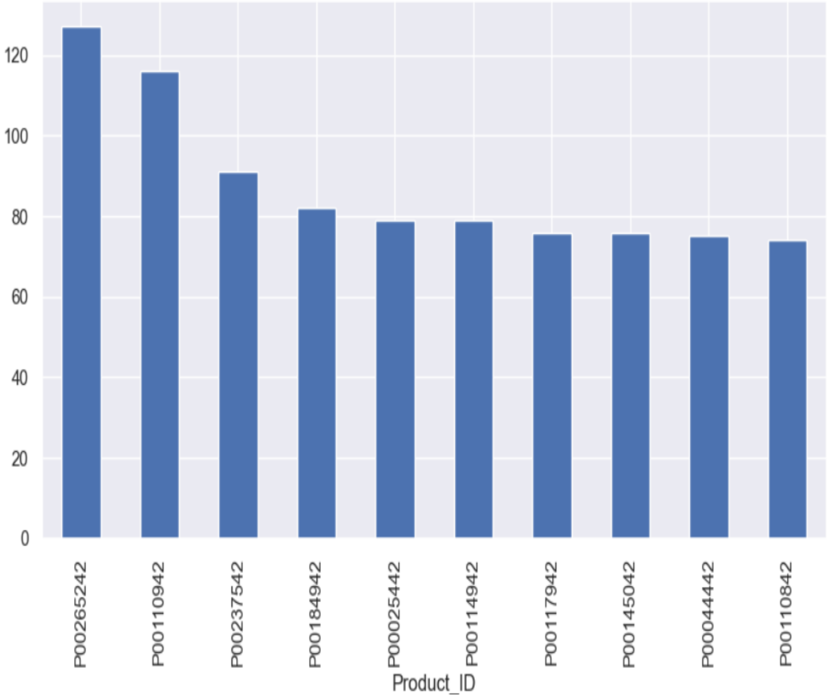
[177]:
```
<Axes: xlabel='Product_Category', ylabel='Amount'>
```



From above graph the product food is mostly purchased by the customers then clothing and other products.

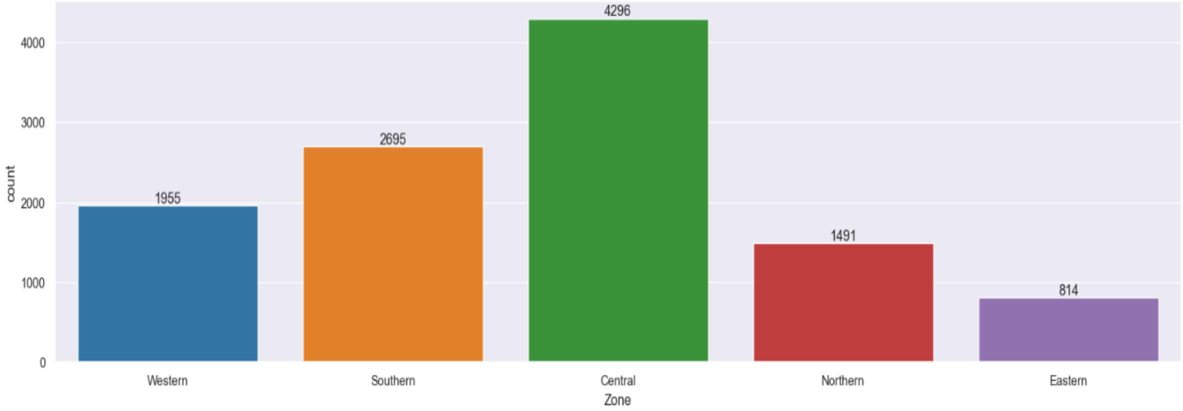
[22]:
```
# top 10 most sold products (same thing as above)

fig1, ax1 = plt.subplots(figsize=(10,5))
df.groupby('Product_ID')['Orders'].sum().nlargest(10).sort_values(ascending=False).plot(kind='bar')
```

[22]:
```
<Axes: xlabel='Product_ID'>
```



```
# Create the count plot with the correct palette parameter
ax = sns.countplot(x='Zone', data=df , palette="tab10")
# Add labels to the bars
sns.set(rc={'figure.figsize':(20,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```

```
C:\Users\ANKITA UPADHAYAY\AppData\Local\Temp\ipykernel_21088\1302372973.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 s
ame effect.

  ax = sns.countplot(x='Zone', data=df , palette="tab10")
```



Mostly customers are from central zone

[ ]:

Conclusion

1. The majority of customers are female.

2. The predominant age range for female customers is 26-35 years.

3. Female customers are predominantly married and primarily reside in Uttar Pradesh, Karnataka, and Maharashtra.

4. The primary occupations of these female customers are in the healthcare, IT, and aviation sectors.

5.The most frequently purchased items by these female customers include food, clothing, and electronics.

[ ]: