This project analyzes Diwali sales data using Python libraries like NumPy, Pandas, Matplotlib, and Seaborn. The dataset is imported into a DataFrame to perform data cleaning, visualization, and gain insights into customer purchasing behavior.

In [35]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
data = pd.read_csv("Diwali_Sales_Data.csv", encoding='latin1')
df = pd.DataFrame(data)
df
```

Out[35]:

	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3766	1005261	Apoorva	P00057942	F	36-45	41	1	Delhi	Central	IT Sector	Footwear & Shoes	2	NaN	NaN	NaN
3767	1005265	Sakshi	P00296242	F	46-50	48	1	Delhi	Central	Banking	Footwear & Shoes	1	NaN	NaN	NaN
3768	1004528	Anurag	P00338442	F	26-35	33	1	Uttar Pradesh	Central	Automobile	Food	2	NaN	NaN	NaN
3769	1005261	Apoorva	P00057942	F	36-45	41	1	Delhi	Central	IT Sector	Footwear & Shoes	2	NaN	NaN	NaN
3770	1005265	Sakshi	P00296242	F	46-50	48	1	Delhi	Central	Banking	Footwear & Shoes	1	NaN	NaN	NaN

3771 rows × 15 columns

In [36]:
```
df.shape  # check how many Column and Rows
```

Out[36]: (3771, 15)

In [37]:
```
df.info()  #check data
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3771 entries, 0 to 3770
Data columns (total 15 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   User_ID           3771 non-null   int64
 1   Cust_name         3771 non-null   object
 2   Product_ID        3771 non-null   object
 3   Gender            3771 non-null   object
 4   Age Group         3771 non-null   object
 5   Age               3771 non-null   int64
 6   Marital_Status    3771 non-null   int64
 7   State             3771 non-null   object
 8   Zone              3771 non-null   object
 9   Occupation        3771 non-null   object
 10  Product_Category  3771 non-null   object
 11  Orders            3771 non-null   int64
 12  Amount            3759 non-null   float64
 13  Status            0 non-null      float64
 14  unnamed1          0 non-null      float64
dtypes: float64(3), int64(4), object(8)
memory usage: 442.0+ KB
```

In [38]:
```
# status column need Remove beacuse of Blank columns
# unnamed1 column need Remove beacuse of Blank columns
df.drop(["Status","unnamed1"], axis=1, inplace=True) # axis say select entire coloumn, # inplace say save the changes.
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3771 entries, 0 to 3770
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   User_ID           3771 non-null   int64
 1   Cust_name         3771 non-null   object
 2   Product_ID        3771 non-null   object
 3   Gender            3771 non-null   object
 4   Age Group         3771 non-null   object
 5   Age               3771 non-null   int64
 6   Marital_Status    3771 non-null   int64
 7   State             3771 non-null   object
 8   Zone              3771 non-null   object
 9   Occupation        3771 non-null   object
 10  Product_Category  3771 non-null   object
 11  Orders            3771 non-null   int64
 12  Amount            3759 non-null   float64
dtypes: float64(1), int64(4), object(8)
memory usage: 383.1+ KB
```

In [39]:
```
pd.isnull(df).sum()  #find null in each cell and sum of null by column
```

Out[39]:
```
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
```

In [40]:
```
# as you see there are 12 null in Amount coloumn that mean the 12 individual can buy anything that whay we remove them
df.dropna(inplace=True)  #delete NA value and inplace save them
pd.isnull(df).sum()
```

Out[40]:
```
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              0
dtype: int64
```

In [41]:
```
df.drop_duplicates(inplace=True)  # there is some duplicate vale are available in the data need remove.
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 3752 entries, 0 to 3751
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   User_ID           3752 non-null   int64
 1   Cust_name         3752 non-null   object
 2   Product_ID        3752 non-null   object
 3   Gender            3752 non-null   object
 4   Age Group         3752 non-null   object
 5   Age               3752 non-null   int64
 6   Marital_Status    3752 non-null   int64
 7   State             3752 non-null   object
 8   Zone              3752 non-null   object
 9   Occupation        3752 non-null   object
 10  Product_Category  3752 non-null   object
 11  Orders            3752 non-null   int64
 12  Amount            3752 non-null   float64
dtypes: float64(1), int64(4), object(8)
memory usage: 410.4+ KB
```

In [42]:
```
df["Amount"] = df["Amount"].astype(int) #in previously df.info we see Amount in float that's why i change float to int
df["Amount"].dtype
```

Out[42]: dtype('int64')

In [43]:
```
df.columns
```

Out[43]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [44]:
```
df.rename(columns={"Age Group" : "Age_Group"}, inplace=True)   #change Column name
df.columns
```

Out[44]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age_Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [45]:
```
df.describe()
```

Out[45]:

	User_ID	Age	Marital_Status	Orders	Amount
count	3.752000e+03	3752.000000	3752.000000	3752.000000	3752.000000
mean	1.003033e+06	36.117537	0.417377	2.466418	12614.667377
std	1.749201e+03	13.924207	0.493192	1.115052	5506.631335
min	1.000001e+06	12.000000	0.000000	1.000000	567.000000
25%	1.001501e+06	27.000000	0.000000	1.000000	8007.250000
50%	1.003056e+06	33.000000	0.000000	2.000000	12274.500000
75%	1.004527e+06	44.000000	1.000000	3.000000	16681.000000
max	1.006040e+06	92.000000	1.000000	4.000000	23952.000000

In [46]:
```
df[["Age","Orders","Amount"]].describe().astype(int)
```

Out[46]:

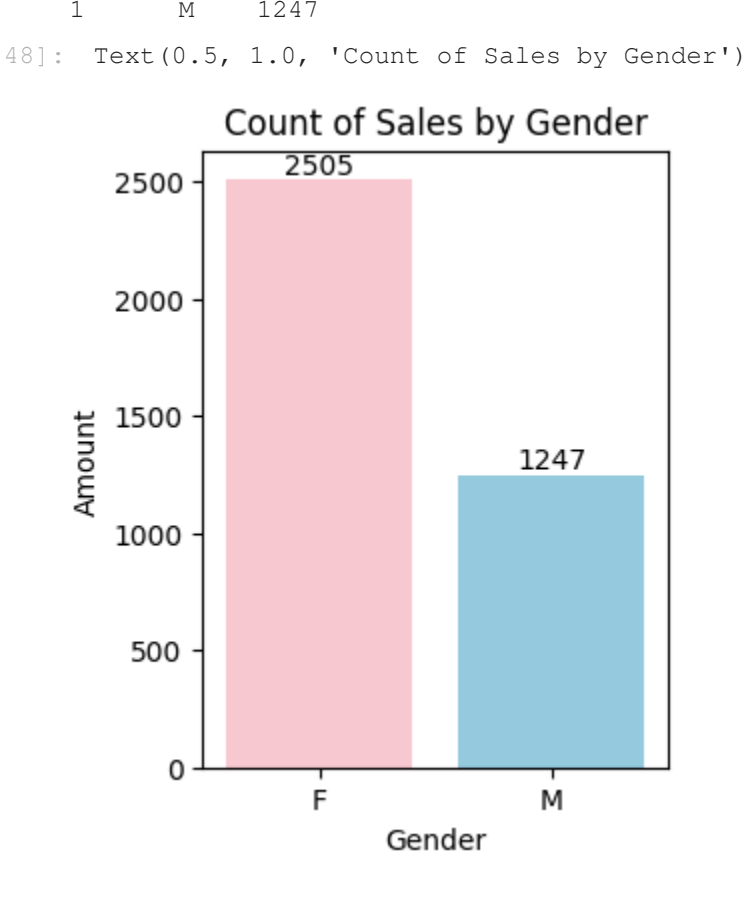
	Age	Orders	Amount
count	3752	3752	3752
mean	36	2	12614
std	13	1	5506
min	12	1	567
25%	27	1	8007
50%	33	2	12274
75%	44	3	16681
max	92	4	23952

In [47]:
```
# Exploratory Data Analysis
```

In [48]:
```
plt.figure(figsize=(3,4))
group_by = df.groupby("Gender")["Amount"].count().reset_index()
print(group_by)
sales_ = sns.barplot(data = group_by, x = "Gender", y="Amount", hue="Gender",palette={"M": "skyblue", "F": "pink"}, legend=False)
for container in sales_.containers:
    plt.bar_label(container, label_type="edge", fmt="%.0f")
plt.title("Count of Sales by Gender")
```

```
  Gender  Amount
0      F    2505
1      M    1247
```

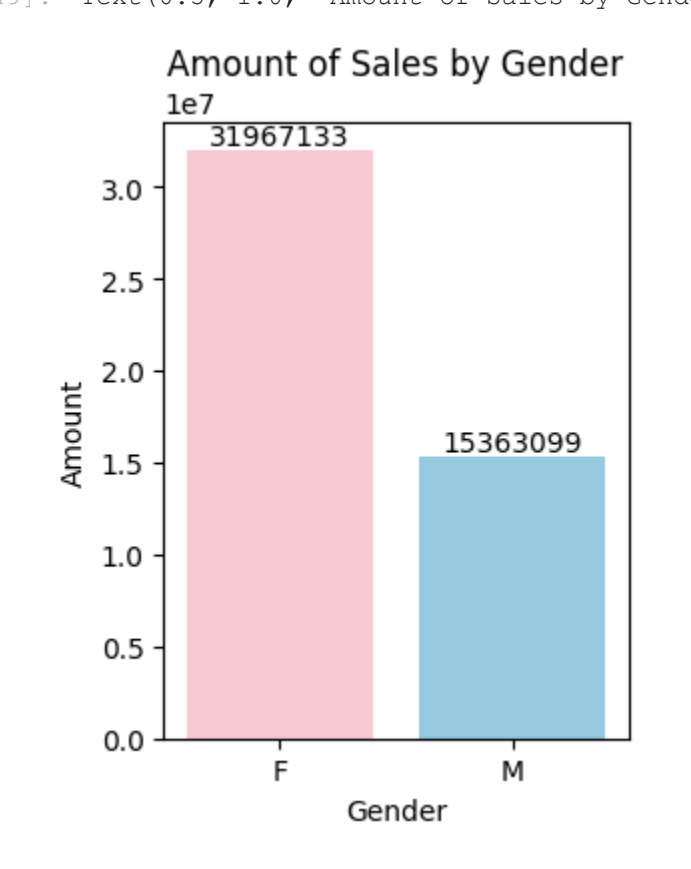
Out[48]: Text(0.5, 1.0, 'Count of Sales by Gender')



In [49]:
```
plt.figure(figsize=(3,4))
group_by = df.groupby("Gender")["Amount"].sum().reset_index()
print(group_by)
sales_ = sns.barplot(data = group_by, x = "Gender", y="Amount", hue="Gender",palette={"M": "skyblue", "F": "pink"}, legend=False)
for container in sales_.containers:
    plt.bar_label(container, label_type="edge", fmt="%.0f")
plt.title("Amount of Sales by Gender")
```

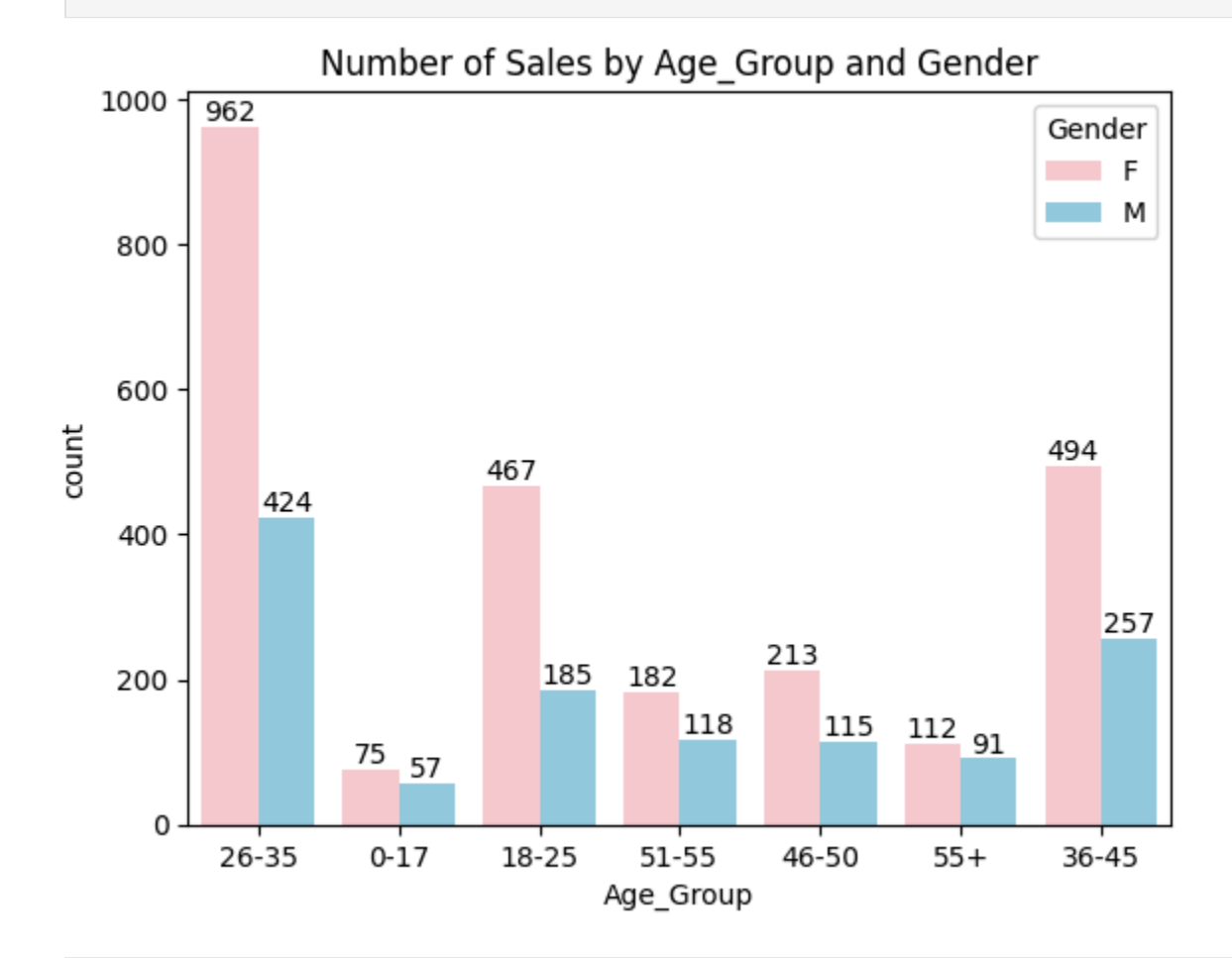
```
  Gender    Amount
0      F  31967133
1      M  15363099
```

Out[49]: Text(0.5, 1.0, 'Amount of Sales by Gender')

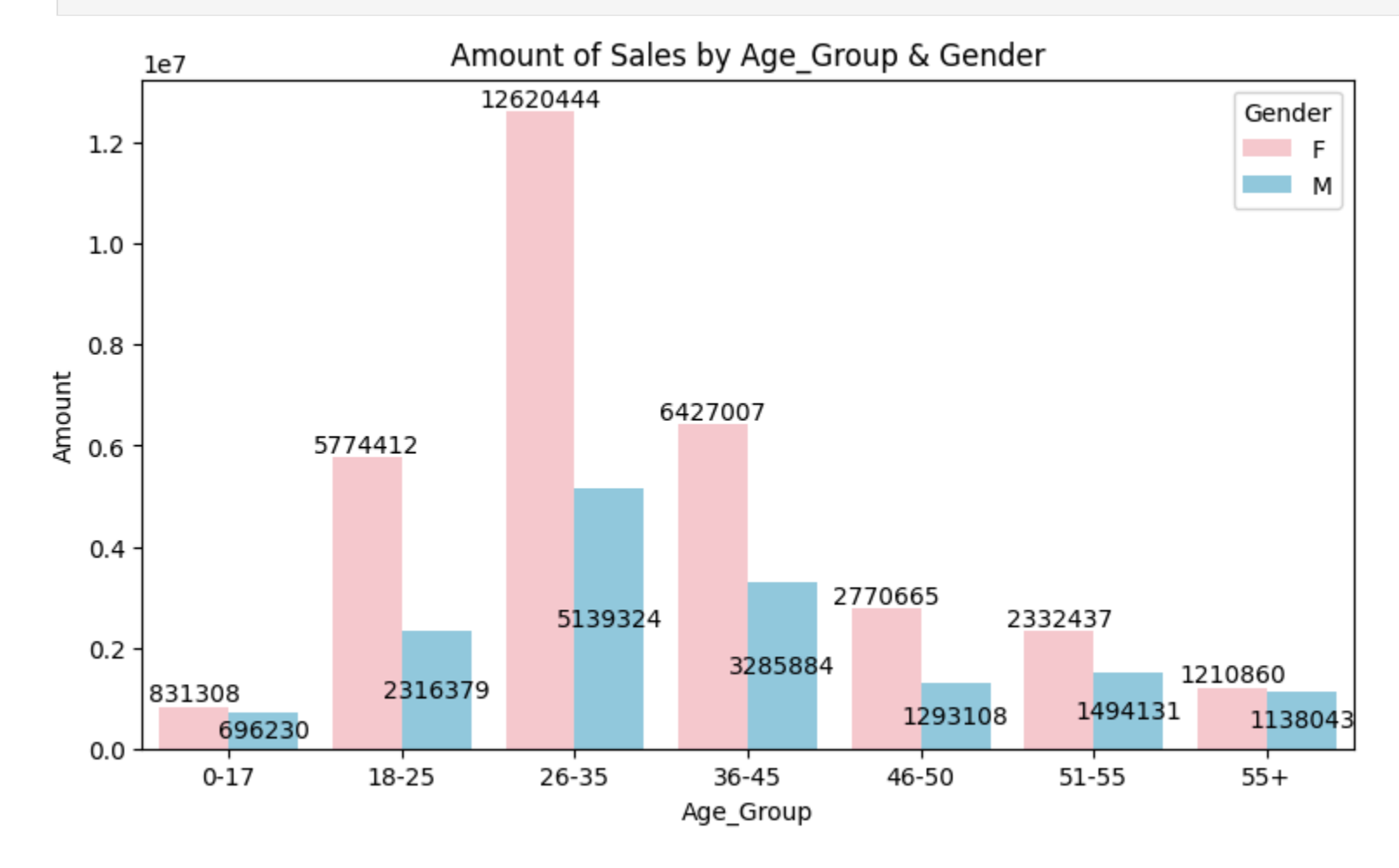


point 1. Count of Sales by Gender Amount of Sales by Gender Both say Females Spend more money than Males for Diwali_Sales

In [50]:
```
Age_Group_Gender_count = sns.countplot(data=df,x="Age_Group", hue="Gender",palette={"F":"pink","M":"skyblue"})
for container in Age_Group_Gender_count.containers:
    plt.bar_label(container, label_type="edge")
plt.title("Number of Sales by Age_Group and Gender")
plt.show()
```



In [51]:
```
plt.figure(figsize=(9,5))
group_bys = df.groupby(["Age_Group", "Gender"])["Amount"].sum().reset_index()
age_gender_amount = sns.barplot(data=group_bys, x="Age_Group", y="Amount", hue="Gender", palette={"F": "Pink", "M": "skyblue"})

for container, (gender,Age_gender_amount) in zip(age_gender_amount.containers, group_bys.groupby("Gender")):
    if gender == "M":
        plt.bar_label(container, label_type="center", fmt="%.0f")  # Center labels for Male
    else:
        plt.bar_label(container, label_type="edge", fmt="%.0f")  # Edge labels for Female

plt.title("Amount of Sales by Age_Group & Gender")
plt.show()
```

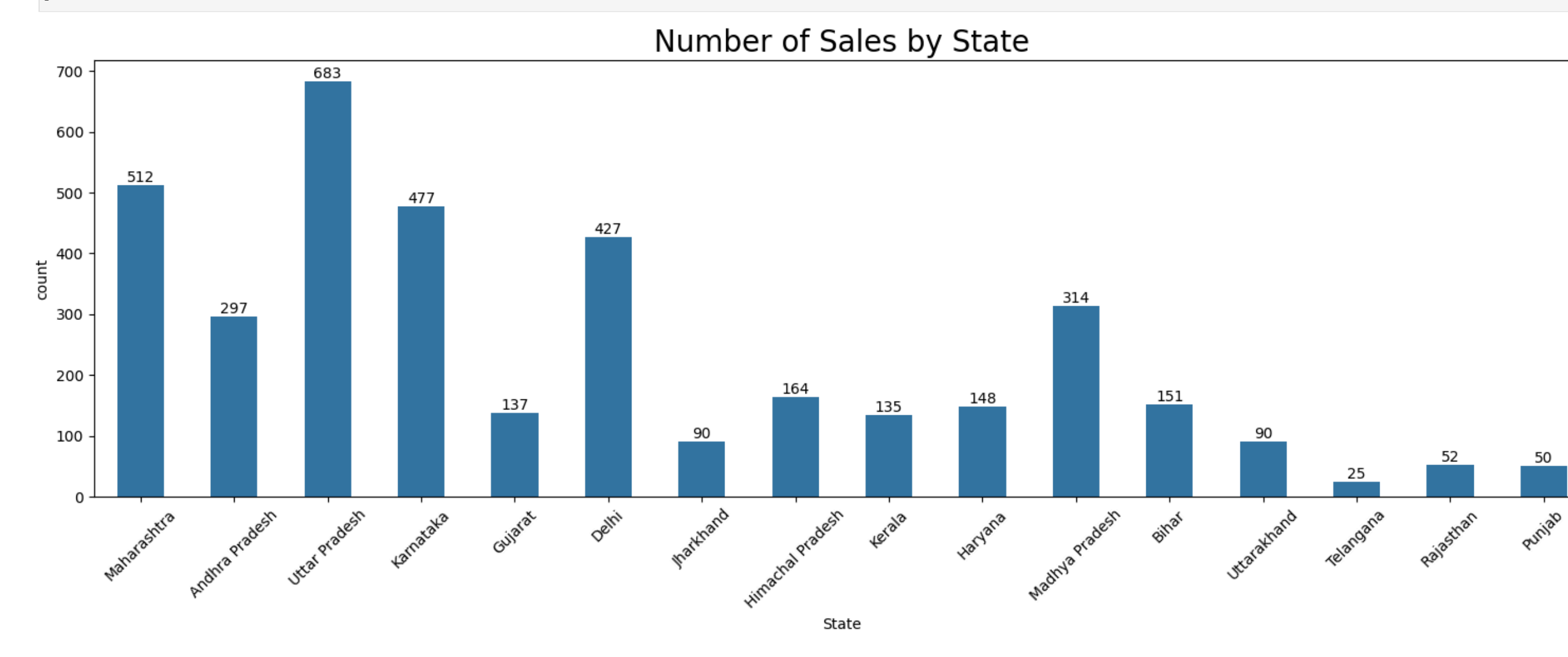


Point 2. Number of Sales by Age_Group and Gender Amount of Sales by Age_Group & Gender Both said Across all age groups, female are the primary drivers of Diwali sales, consistently outspending male during the festive season.

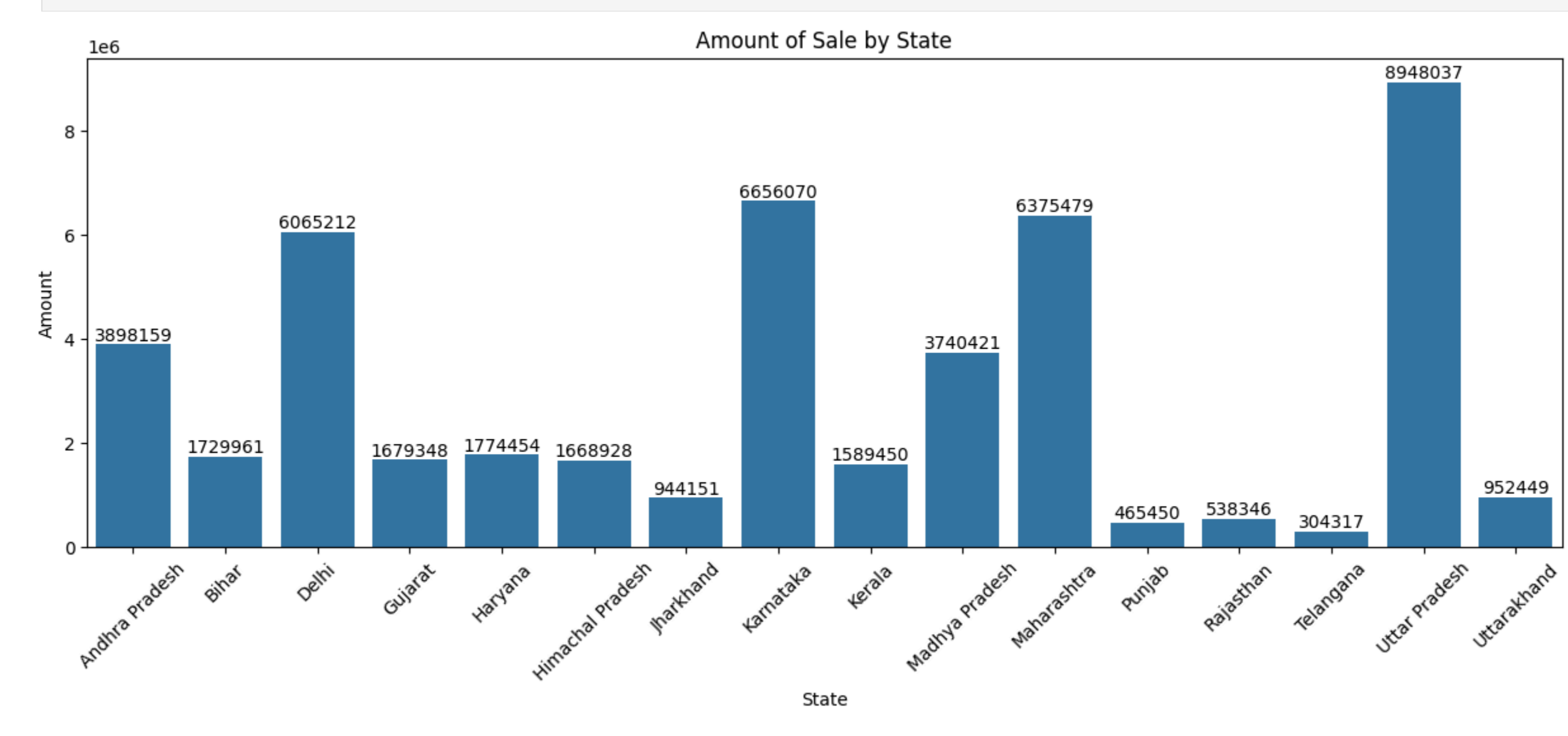
In [52]:
```
df.columns
```

Out[52]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age_Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [53]:
```
plt.figure(figsize=(15,6))
ox=sns.countplot(data=df, x="State", width=0.5)
for container in ox.containers:
    plt.bar_label(container, label_type="edge", fmt="%.0f")
plt.xlabel("State",fontsize=10)
plt.title("Number of Sales by State", fontsize=20, )
plt.xticks(rotation=45)  # Rotate x labels for better readability
plt.tight_layout()  # Adjust layout to prevent clipping of tick-labels
plt.show()
```

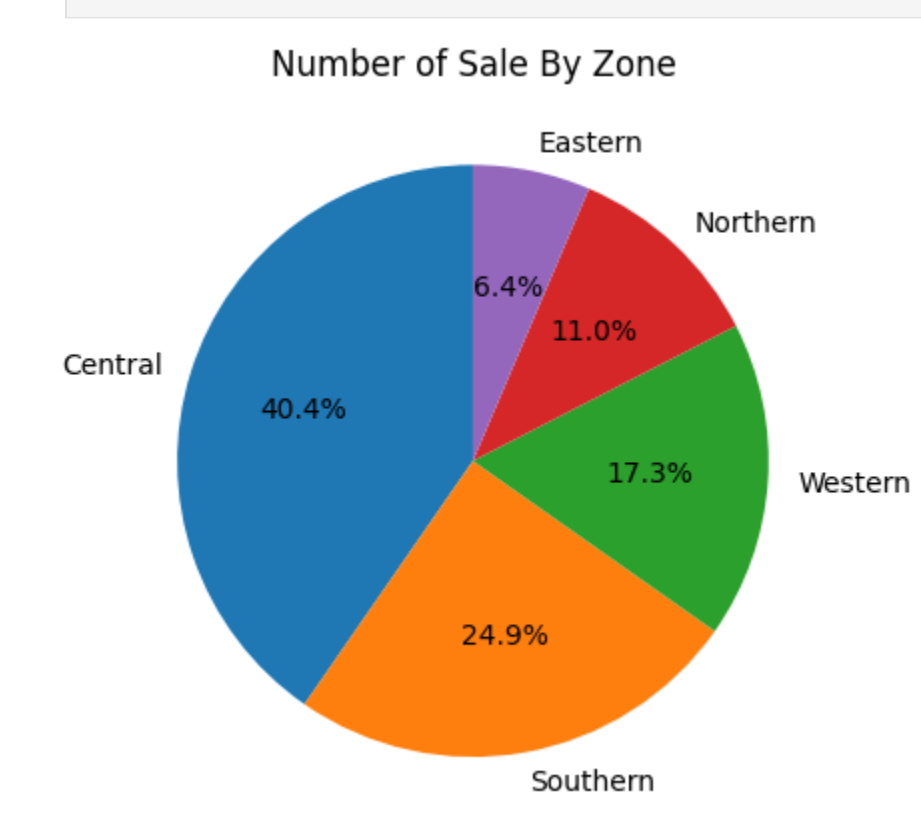


In [54]:
```
plt.figure(figsize=(15,5))
groupp=df.groupby("State")["Amount"].sum().reset_index()
dox=sns.barplot(data=groupp, x="State", y="Amount")
for container in dox.containers:
    plt.bar_label(container, label_type="edge", fmt="%.0f")
plt.title("Amount of Sale by State")
plt.xticks(rotation=45)
plt.show()
```

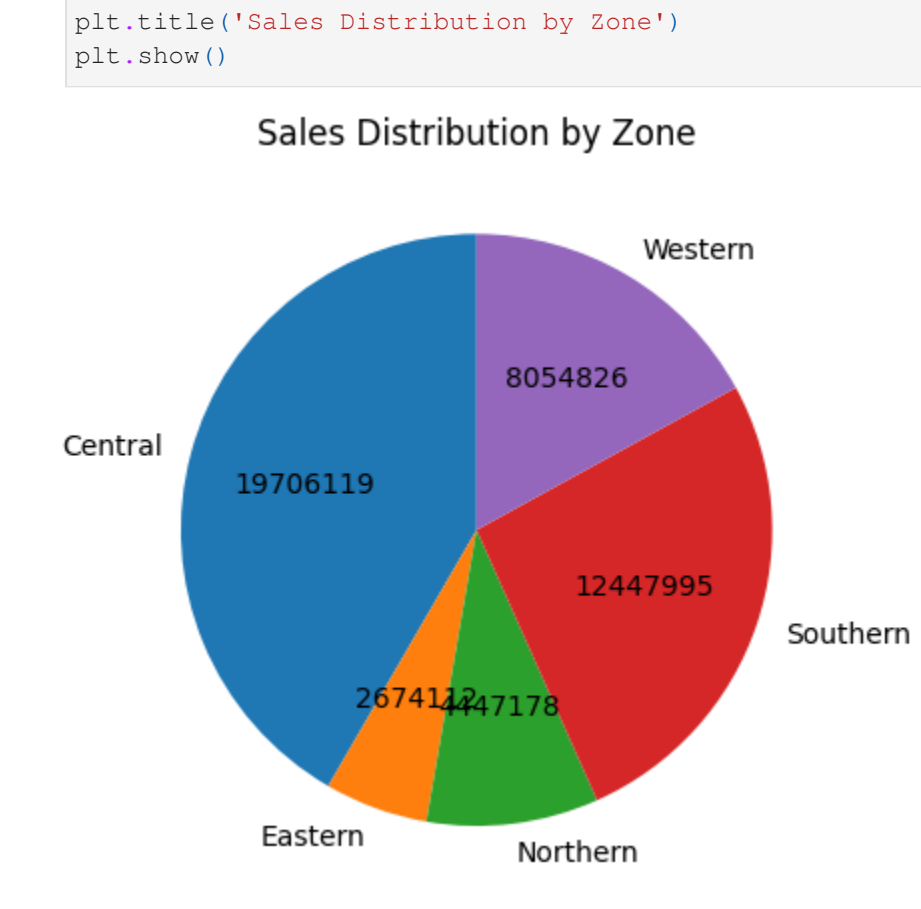


Point 3. Number of Sales by State Amount of Sale by State The both graph said Maharashtra dominates Diwali sales with over 19 million in revenue, significantly outpacing other states. Karnataka and Uttar Pradesh follow with strong sales figures exceeding 13 million each. Several states lag considerably, demonstrating a wide disparity in sales performance across regions.

In [55]:
```
zone=df["Zone"].value_counts()
plt.pie(zone,labels=zone.index, autopct='%1.1f%%',startangle=90,textprops={"color": "black"})
plt.title("Number of Sale By Zone")
plt.show()
```



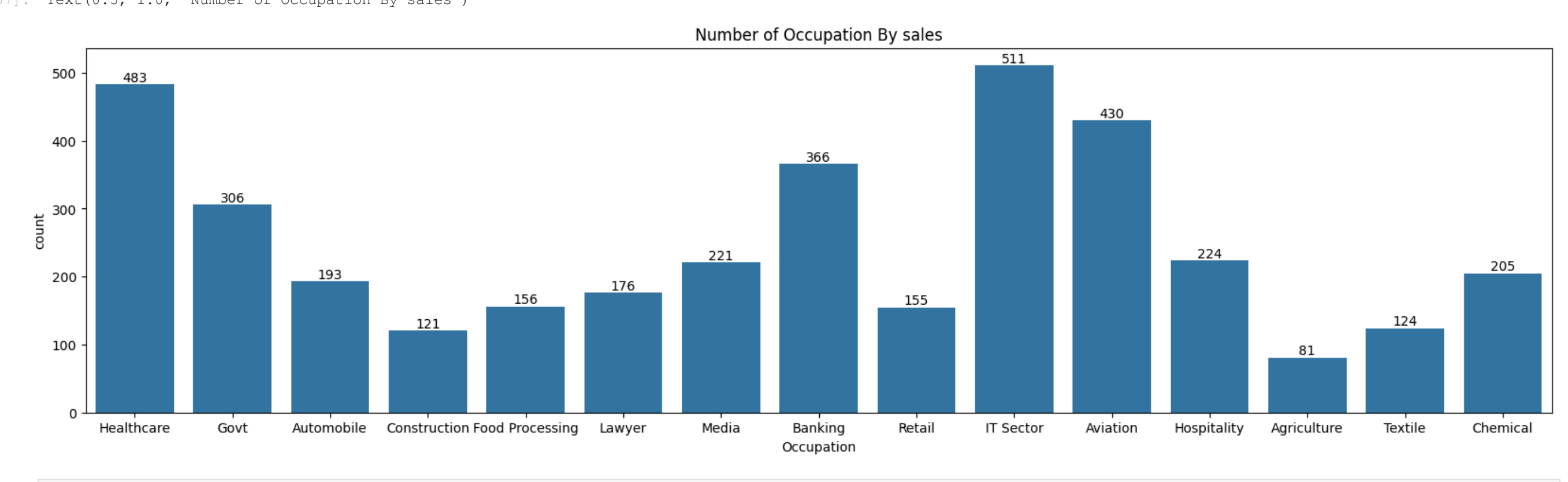
In [56]:
```
grouped = df.groupby("Zone")["Amount"].sum()

def absolute_value(val):
    total = grouped.sum()
    absolute = int(round((val / 100) * total))
    return f'{absolute}'

plt.pie(grouped, labels=grouped.index, autopct=absolute_value, startangle=90, textprops={"color": "black"})
plt.title('Sales Distribution by Zone')
plt.show()
```



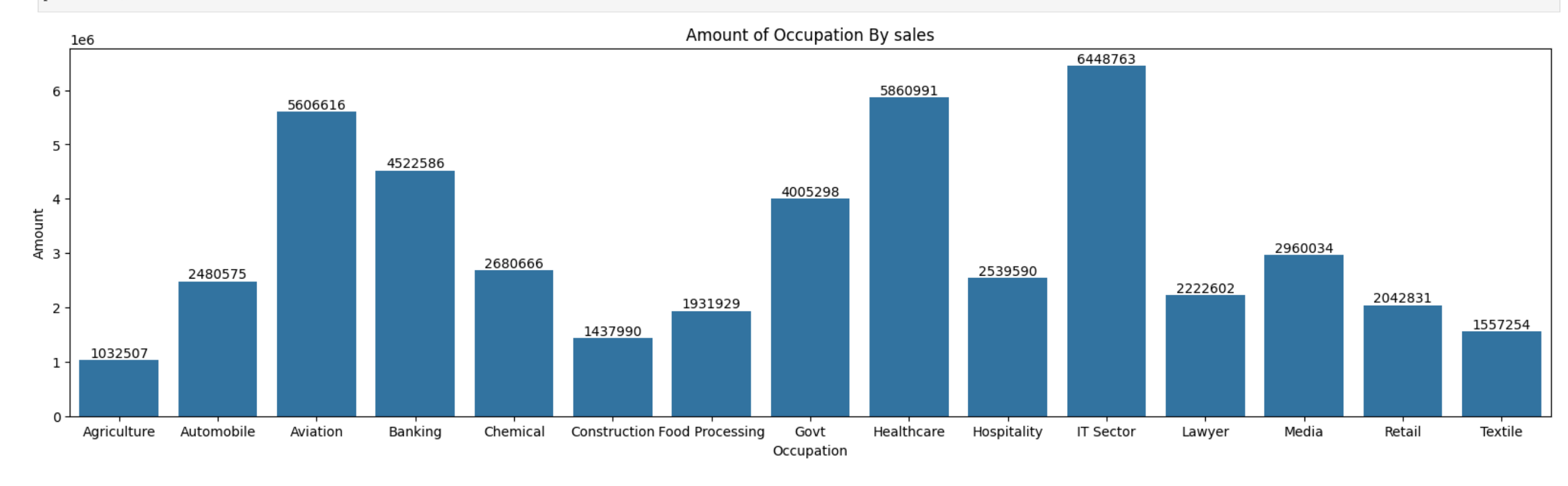
Point 4. Number of Sale By Zone Sales Distribution by Zone The Central zone sells way more than any other zone. Southern and Western zones sell a decent amount, but much less than Central. The Eastern and Northern zones sell the least.

In [57]:
```
plt.figure(figsize=(20,5))
occu = sns.countplot(data=df, x="Occupation")
for container in occu.containers:
    plt.bar_label(container,label_type="edge")
plt.title("Number of Occupation By sales")
```

Out[57]: Text(0.5, 1.0, 'Number of Occupation By sales')



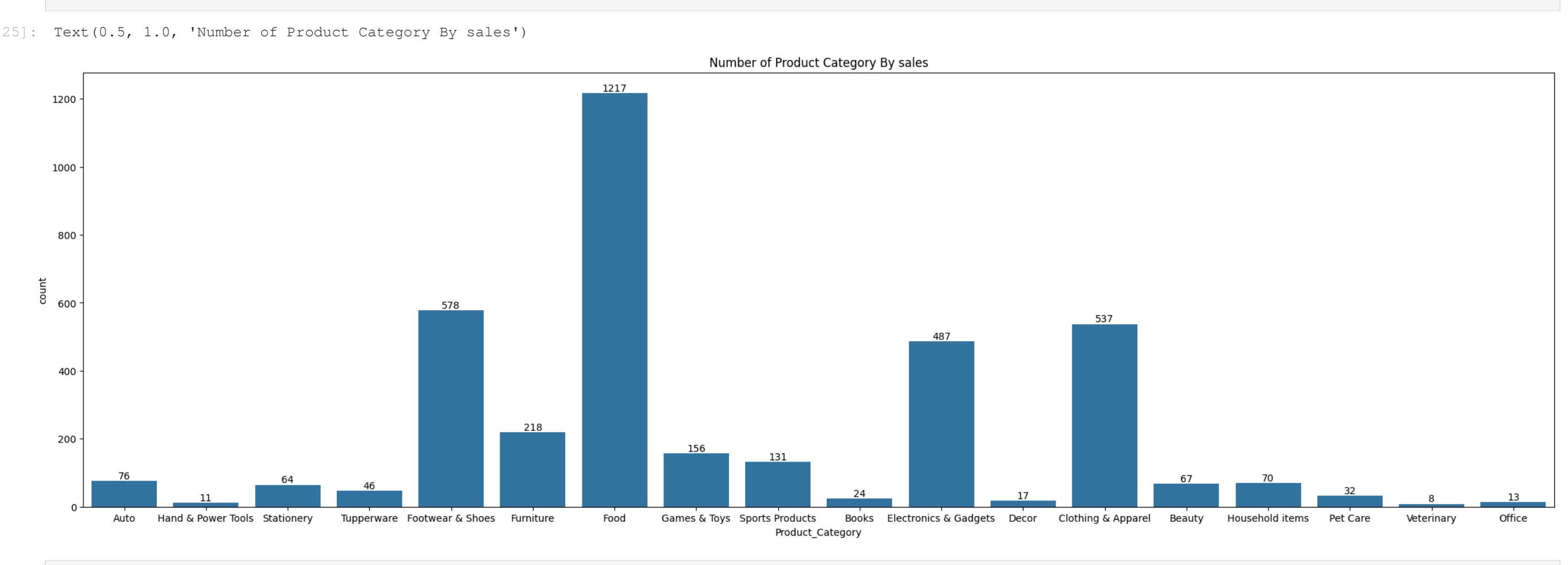
In [58]:
```
plt.figure(figsize=(20,5))
drope = df.groupby("Occupation")["Amount"].sum().reset_index()
occup=sns.barplot(data=drope,x="Occupation", y="Amount")
for container in occup.containers:
    plt.bar_label(container,label_type="edge",fmt="%.0f")
plt.title("Amount of Occupation By sales")
plt.show()
```



Point 5. Number of Occupation By sales Amount of Occupation By sales This bar chart displays the total sales amount generated by customers in different occupations. The IT Sector leads with the highest sales value, followed by Healthcare and Aviation. Several other professions contribute significantly, while occupations like Agriculture and Textile show considerably lower spending.

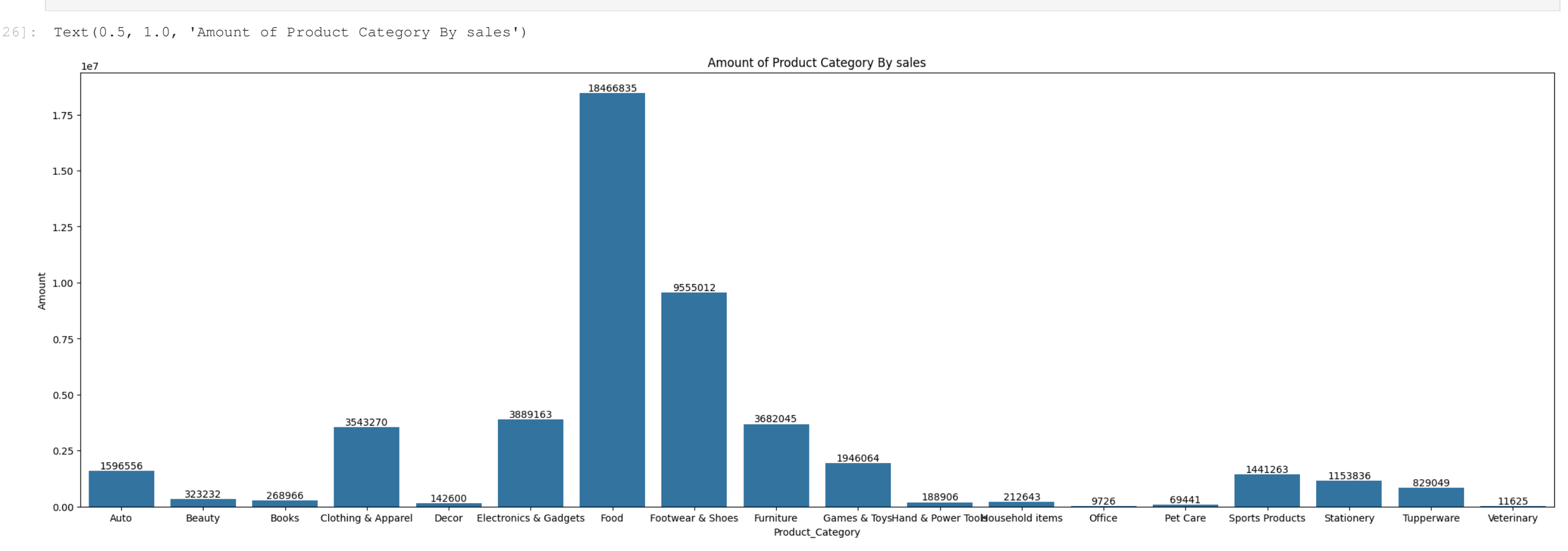
In [25]:
```
plt.figure(figsize=(27,8))
pro_ca=sns.countplot(data=df,x="Product_Category")
for container in pro_ca.containers:
    plt.bar_label(container,label_type="edge")
plt.title("Number of Product Category By sales")
```

Out[25]: Text(0.5, 1.0, 'Number of Product Category By sales')



In [26]:
```
plt.figure(figsize=(27,8))
groped=df.groupby("Product_Category")["Amount"].sum().reset_index()
prod_cat=sns.barplot(data=groped,x="Product_Category", y="Amount")
for container in prod_cat.containers:
    plt.bar_label(container,label_type="edge",fmt="%.0f")
plt.title("Amount of Product Category By sales")
```

Out[26]: Text(0.5, 1.0, 'Amount of Product Category By sales')



Part 6. Number of Product Category By sales Amount of Product Category By sales Food is the top-selling category by revenue, significantly exceeding all others. Footwear & Shoes and Clothing & Apparel are strong secondary revenue drivers. Several categories, including Auto, Beauty, Books, Office, Pet Care, and Veterinary, generate comparatively low sales revenue.

conclusion

Diwali sales are led by food, generating 34 million, followed by footwear and clothing with around 15 million each. Female shoppers drive most of these sales, making them the key demographic. Future strategies should focus on their preferences while boosting sales in weaker categories like auto, beauty, and books.