In [3]:
```python
#1
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [5]:
```python
#2
df = pd.read_csv(r"C:\Users\SWATHI\Desktop\Diwali Sales Data.csv", encoding = 'unicode_escape')
df.head(10)
```

Out[5]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Pro
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	
5	1000588	Joni	P00057942	M	26-35	28	1	Himachal Pradesh	Northern	Food Processing	
6	1001132	Balk	P00018042	F	18-25	25	1	Uttar Pradesh	Central	Lawyer	
7	1002092	Shivangi	P00273442	F	55+	61	0	Maharashtra	Western	IT Sector	
8	1003224	Kushal	P00205642	M	26-35	35	0	Uttar Pradesh	Central	Govt	
9	1003650	Ginny	P00031142	F	26-35	26	1	Andhra Pradesh	Southern	Media	

In [7]:
```python
#3
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 
 8   Zone              11251 non-null  object 
 9   Occupation        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
```

In [9]:
```python
#4
df = df.drop(columns = ["Status","unnamed1"])
df
```

Out[9]:

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

11251 rows × 13 columns

In [11]:
```python
#5
df.isnull().sum()
```

Out[11]:
```
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 [13]:
```python
#6
df = df.dropna()
df
```

Out[13]:

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

11239 rows × 13 columns

In [15]:
```python
#7
df1 = df.filter(items=['State','Orders','Amount'], axis=1)
df1
```

Out[15]:

	State	Orders	Amount
0	Maharashtra	1	23952.0
1	Andhra Pradesh	3	23934.0
2	Uttar Pradesh	3	23924.0
3	Karnataka	2	23912.0
4	Gujarat	2	23877.0
...	...	...	...
11246	Maharashtra	4	370.0
11247	Haryana	3	367.0
11248	Madhya Pradesh	4	213.0
11249	Karnataka	3	206.0
11250	Maharashtra	3	188.0

11239 rows × 3 columns

In [17]:
```python
#8
df2 = df1.sort_values(by=["Orders", "Amount"], ascending=[False, False])
df2
```

Out[17]:

	State	Orders	Amount
6	Uttar Pradesh	4	23841.00
9	Andhra Pradesh	4	23799.99
13	Andhra Pradesh	4	23718.00
20	Andhra Pradesh	4	23546.00
27	Andhra Pradesh	4	23451.00
...	...	...	...
11233	Maharashtra	1	563.00
11238	Karnataka	1	555.00
11239	Delhi	1	407.00
11240	Delhi	1	396.00
11243	Gujarat	1	382.00

11239 rows × 3 columns

In [19]:
```python
#9
df3 = df.groupby('Age Group').describe()
df3
```

Out[19]:

	User_ID									
	count	mean	std	min	25%	50%	75%	max	count	m
Age Group										
0-17	296.0	1.002652e+06	1799.720983	1000001.0	1001220.0	1002047.0	1004493.75	1006006.0	296.0	14.533
18-25	1879.0	1.002824e+06	1714.269054	1000018.0	1001396.0	1002898.0	1004110.50	1006028.0	1879.0	21.492
26-35	4541.0	1.003105e+06	1716.253427	1000003.0	1001611.0	1003224.0	1004508.00	1006040.0	4541.0	30.446
36-45	2283.0	1.002975e+06	1681.977053	1000007.0	1001449.0	1002962.0	1004365.00	1006011.0	2283.0	40.432
46-50	983.0	1.003043e+06	1808.049610	1000004.0	1001496.0	1003201.0	1004510.00	1006039.0	983.0	48.030
51-55	830.0	1.003024e+06	1673.324008	1000017.0	1001575.0	1003242.0	1004373.00	1006033.0	830.0	52.998
55+	427.0	1.002986e+06	1615.330546	1000002.0	1001896.0	1002676.0	1004171.00	1005986.0	427.0	73.819

7 rows × 40 columns

In [21]:
```python
#10
df4 = df.filter(items = ['Cust_name','Gender', 'Occupation'], axis = 1)
df5 = df.filter(items = ['Cust_name','Product_ID', 'Marital_Status'], axis = 1)
df6 = df4.merge(df5, how = 'inner', on = 'Cust_name')
df6
```

Out[21]:

	Cust_name	Gender	Occupation	Product_ID	Marital_Status
0	Sanskriti	F	Healthcare	P00125942	0
1	Sanskriti	F	Healthcare	P00182142	1
2	Sanskriti	F	Healthcare	P00030942	1
3	Sanskriti	F	Healthcare	P00165442	1
4	Sanskriti	F	Healthcare	P00221142	1
...	...	...	...	...	...
125974	Brumley	F	Healthcare	P00085242	0
125975	Brumley	F	Healthcare	P00251542	0
125976	Brumley	F	Healthcare	P00159442	0
125977	Brumley	F	Healthcare	P0095342	0
125978	Brumley	F	Healthcare	P00281742	0

125979 rows × 5 columns

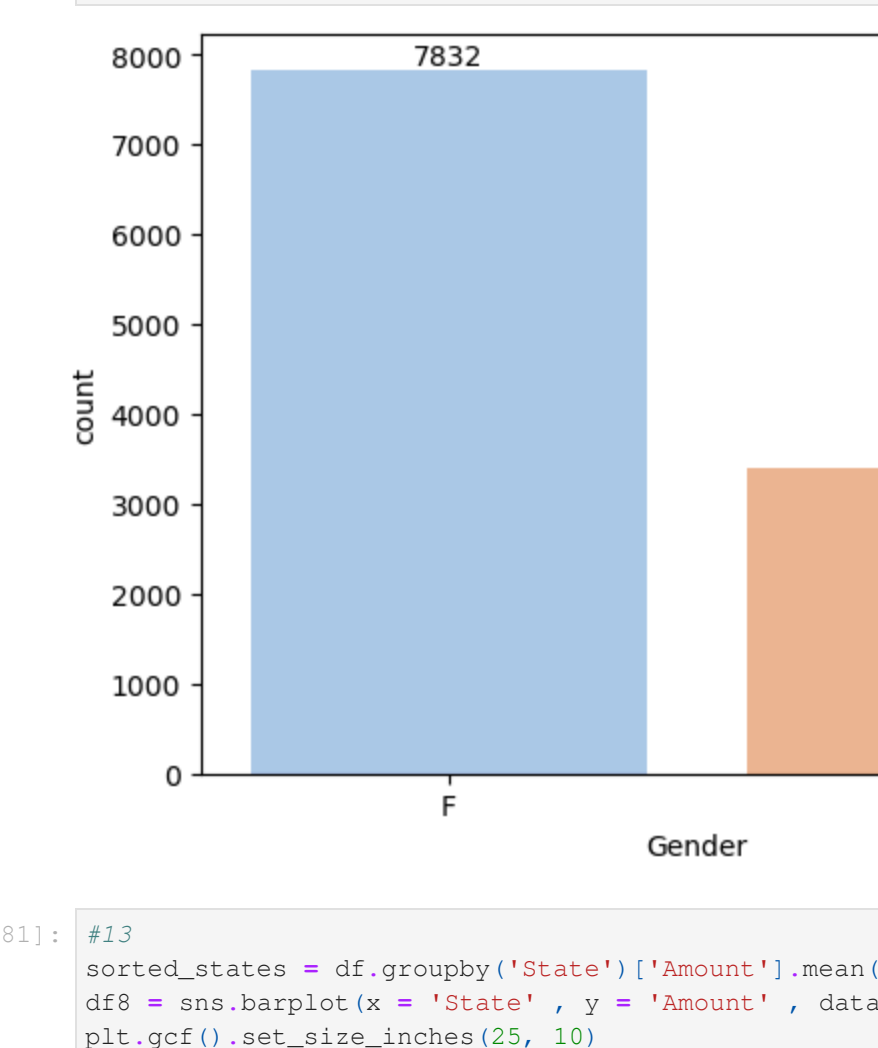
In [23]:
```python
#11
df = df.replace('Govt','Government')
df
```

Out[23]:

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

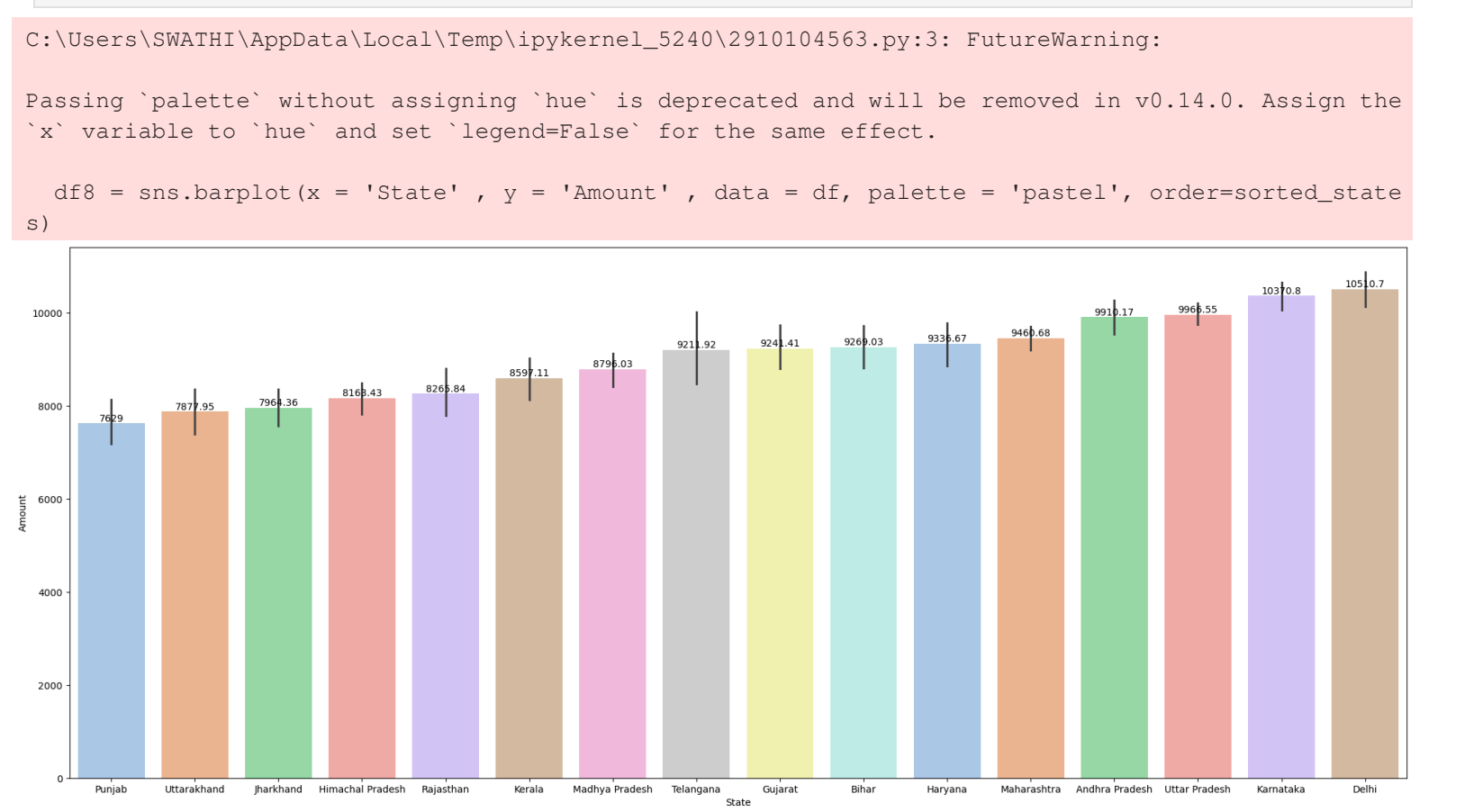
11239 rows × 13 columns

In [25]:
```python
#12
df7 = sns.countplot(x = 'Gender', data = df, hue = 'Gender', palette = 'pastel')
for bars in df7.containers :
    df7.bar_label(bars)
```



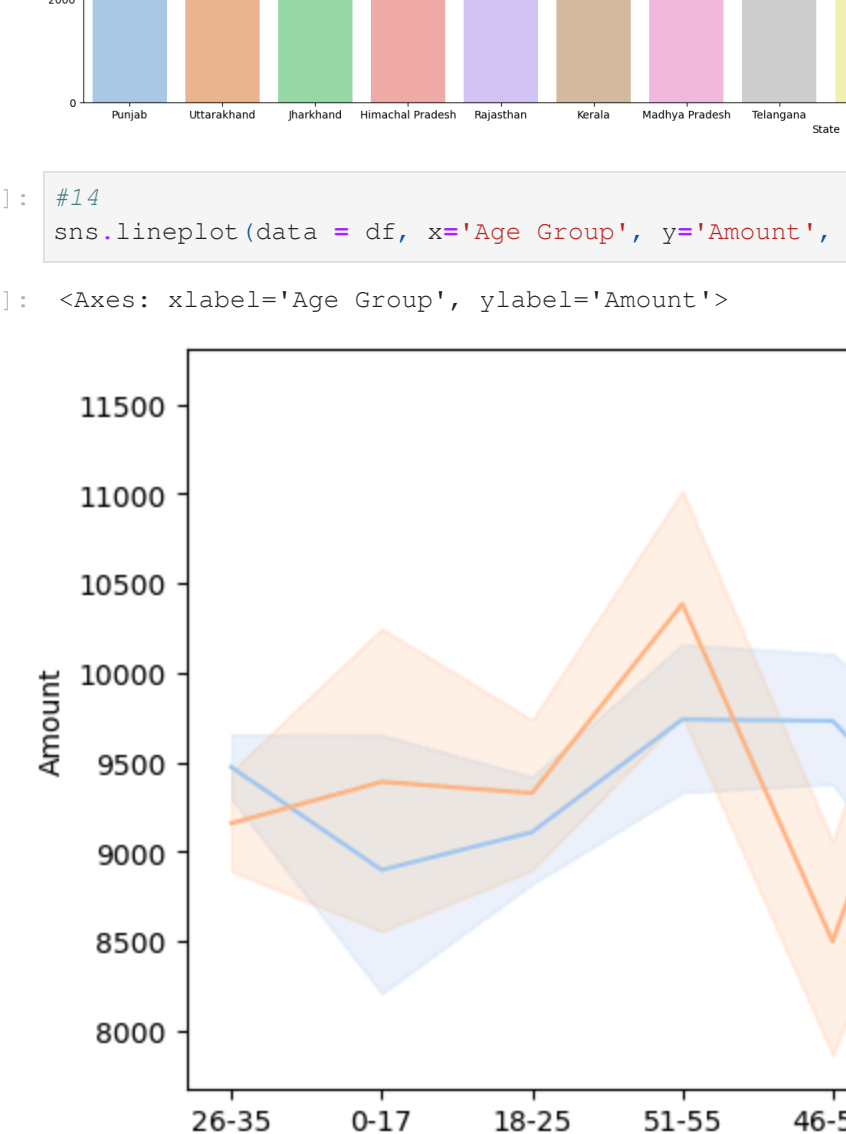
In [81]:
```python
#13
sorted_states = df.groupby('State')['Amount'].mean().sort_values(ascending=True).index
df8 = sns.barplot(x = 'State' , y = 'Amount' , data = df, palette = 'pastel', order=sorted_states)
plt.gcf().set_size_inches(25, 10)
for bars in df8.containers :
    df8.bar_label(bars)
```

```
C:\Users\SWATHI\AppData\Local\Temp\ipykernel_5240\2910104563.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 same effect.

  df8 = sns.barplot(x = 'State' , y = 'Amount' , data = df, palette = 'pastel', order=sorted_state
s)
```



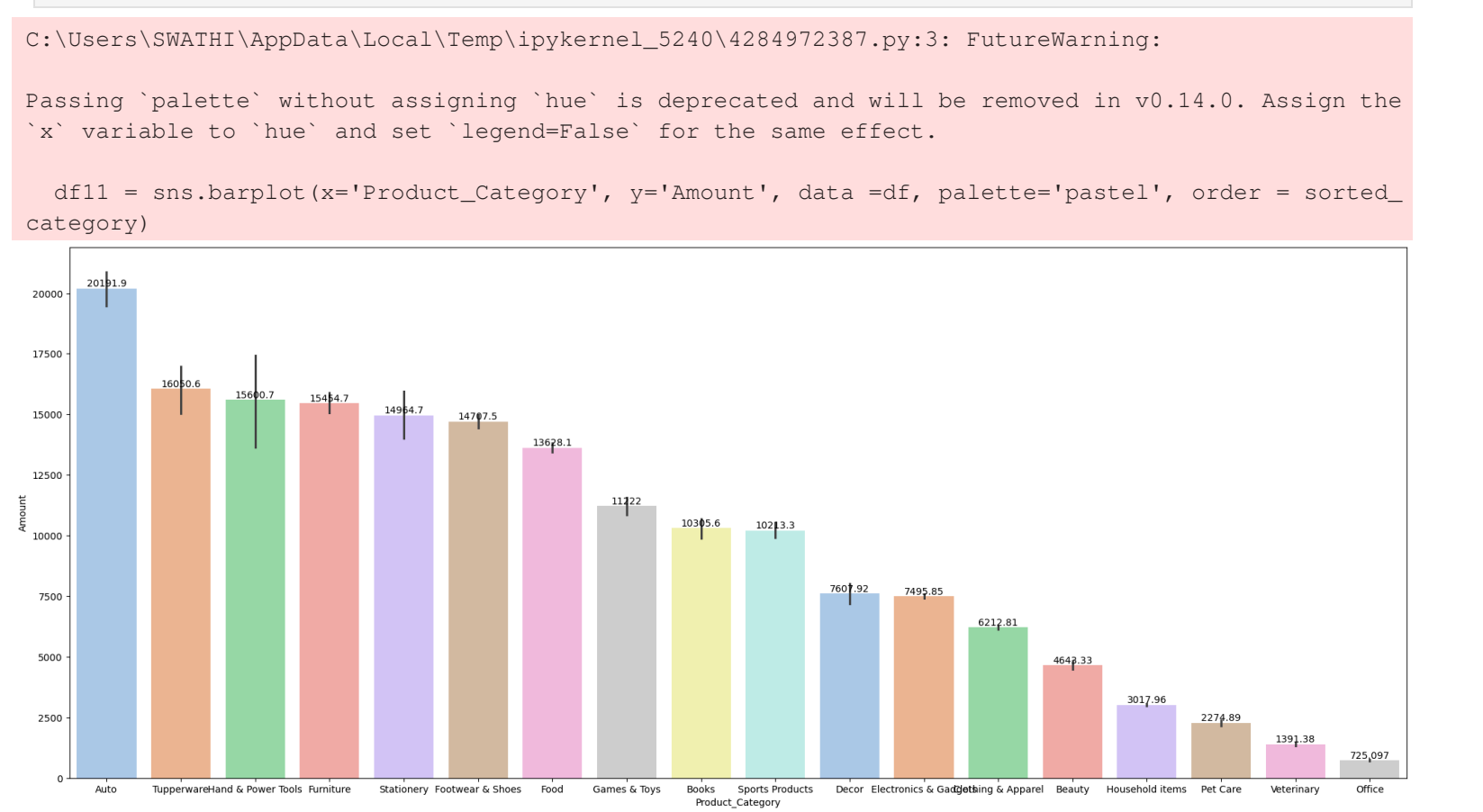
In [71]:
```python
#14
sns.lineplot(data = df, x='Age Group', y='Amount', hue='Gender', palette='pastel')
```

Out[71]: `<Axes: xlabel='Age Group', ylabel='Amount'>`



In [175…
```python
#15
sorted_category = df.groupby('Product_Category')['Amount'].mean().sort_values(ascending= False).i
df11 = sns.barplot(x='Product_Category', y='Amount', data =df, palette='pastel', order = sorted_ca
plt.gcf().set_size_inches(25, 10)
for bars in df11.containers :
    df11.bar_label(bars)
```

```
C:\Users\SWATHI\AppData\Local\Temp\ipykernel_5240\4284972387.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 same effect.

  df11 = sns.barplot(x='Product_Category', y='Amount', data =df, palette='pastel', order = sorted_
category)
```



In [173…
```python
#16
sns.countplot(x='Occupation', data=df, hue = 'Product_Category',palette='pastel')
plt.gcf().set_size_inches(25, 10)
plt.show()
```