In [1]:
```python
# import python libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt # visualizing data
%matplotlib inline
import seaborn as sns
```

In [2]:
```python
# import csv file
df = pd.read_csv(r'C:\Users\ASUS\Desktop\data sets\Vrinda Store Data Analysis.csv')
```

In [3]:
```python
df.shape
```

Out[3]:
```
(31047, 20)
```

In [4]:
```python
df.head()
```

Out[4]:

	index	Order ID	Cust ID	Gender	Age	Date	Status	Channel	SKU	Category	Size	Qty	currency	Amount	ship-city	ship-state	ship-postal-code	ship-country	B2B	Age Group
0	1	171-1029312-3038738	1029312	Women	44	12/4/2022	Delivered	Myntra	JNE1233-BLUE-KR-031-XXL	kurta	XXL	1	INR	376	MOHALI	PUNJAB	140301	IN	False	26-35
1	2	405-2183842-2225946	2183842	Women	29	12/4/2022	Delivered	Ajio	SET414-KR-NP-L	Set	L	1	INR	1449	GURUGRAM	HARYANA	122002	IN	False	26-35
2	3	171-1641533-8921966	1641533	Women	67	12/4/2022	Delivered	Myntra	SET261-KR-PP-S	Set	S	1	INR	453	KOLKATA	WEST BENGAL	700029	IN	False	26-35
3	4	404-7490807-6300351	7490807	Women	20	12/4/2022	Delivered	Amazon	SET110-KR-PP-M	Set	M	1	INR	729	THANJAVUR	TAMIL NADU	613007	IN	False	0-17
4	5	403-9293516-4577154	9293516	Women	62	12/4/2022	Delivered	Myntra	JNE2294-KR-A-XXL	kurta	XXL	1	INR	544	GURUGRAM	HARYANA	122001	IN	False	26-35

In [5]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31047 entries, 0 to 31046
Data columns (total 20 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   index             31047 non-null  int64 
 1   Order ID          31047 non-null  object
 2   Cust ID           31047 non-null  int64 
 3   Gender            31047 non-null  object
 4   Age               31047 non-null  int64 
 5   Date              31047 non-null  object
 6   Status            31047 non-null  object
 7   Channel           31047 non-null  object
 8   SKU               31047 non-null  object
 9   Category          31047 non-null  object
 10  Size              31047 non-null  object
 11  Qty               31047 non-null  int64 
 12  currency          31047 non-null  object
 13  Amount            31047 non-null  int64 
 14  ship-city         31047 non-null  object
 15  ship-state        31047 non-null  object
 16  ship-postal-code  31047 non-null  int64 
 17  ship-country      31047 non-null  object
 18  B2B               31047 non-null  bool  
 19  Age Group         11251 non-null  object
dtypes: bool(1), int64(6), object(13)
memory usage: 4.5+ MB
```

In [6]:
```python
#check for null values
pd.isnull(df).sum()
```

Out[6]:
```
index                   0
Order ID                0
Cust ID                 0
Gender                  0
Age                     0
Date                    0
Status                  0
Channel                 0
SKU                     0
Category                0
Size                    0
Qty                     0
currency                0
Amount                  0
ship-city               0
ship-state              0
ship-postal-code        0
ship-country            0
B2B                     0
Age Group           19796
dtype: int64
```

In [7]:
```python
# drop null values
df.dropna(inplace=True)
```

In [8]:
```python
df.columns
```

Out[8]:
```
Index(['index', 'Order ID', 'Cust ID', 'Gender', 'Age', 'Date', 'Status',
       'Channel ', 'SKU', 'Category', 'Size', 'Qty', 'currency', 'Amount',
       'ship-city', 'ship-state', 'ship-postal-code', 'ship-country', 'B2B',
       'Age Group'],
      dtype='object')
```

In [9]:
```python
# describe() method returns description of the data in the DataFrame (i.e. count, mean, std, etc)
df.describe()
```

Out[9]:

	index	Cust ID	Age	Qty	Amount	ship-postal-code
count	11251.000000	1.125100e+04	11251.000000	11251.000000	11251.000000	11251.000000
mean	5626.000000	4.910863e+06	39.521465	1.005866	685.820905	459062.136343
std	3248.028274	2.892268e+06	15.105877	0.092190	268.982986	198674.130690
min	1.000000	8.950000e+02	18.000000	1.000000	229.000000	110001.000000
25%	2813.500000	2.395110e+06	28.000000	1.000000	487.000000	302024.000000
50%	5626.000000	4.874074e+06	37.000000	1.000000	646.000000	500020.000000
75%	8438.500000	7.396564e+06	47.000000	1.000000	801.000000	600030.500000
max	11251.000000	9.995247e+06	78.000000	4.000000	3036.000000	855116.000000

In [10]:
```python
# use describe() for specific columns
df[['Age', 'Amount']].describe()
```

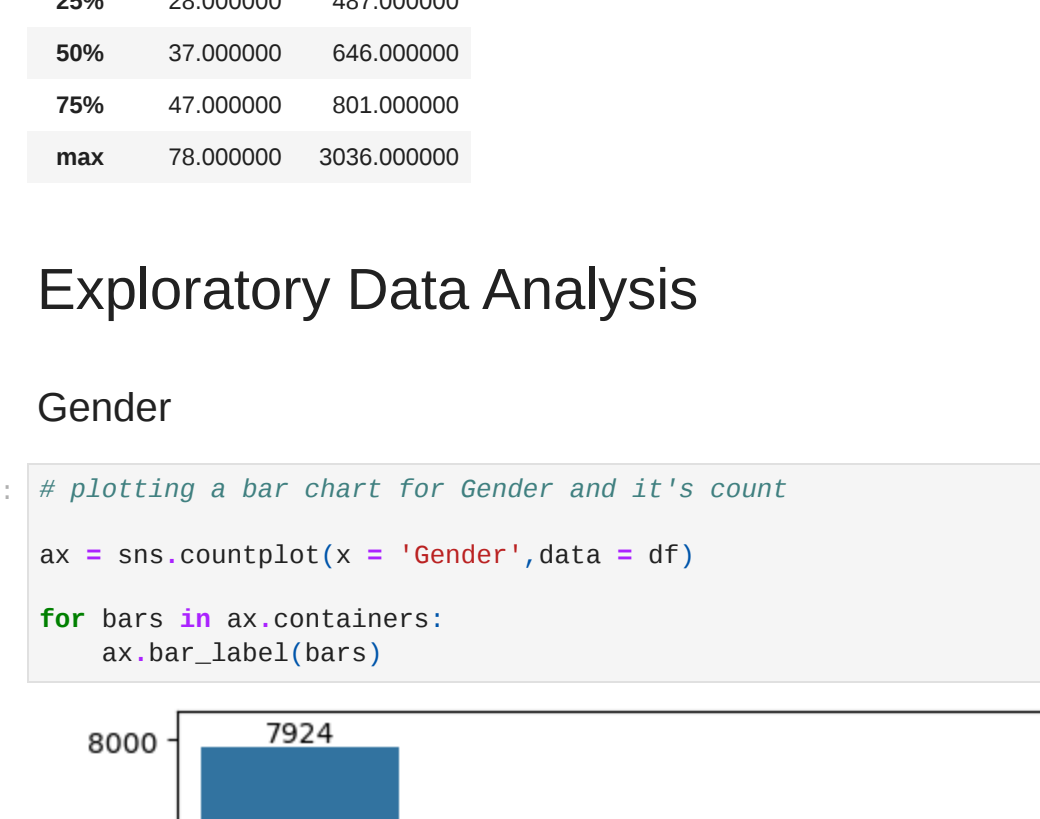
Out[10]:

	Age	Amount
count	11251.000000	11251.000000
mean	39.521465	685.820905
std	15.105877	268.982986
min	18.000000	229.000000
25%	28.000000	487.000000
50%	37.000000	646.000000
75%	47.000000	801.000000
max	78.000000	3036.000000

Exploratory Data Analysis

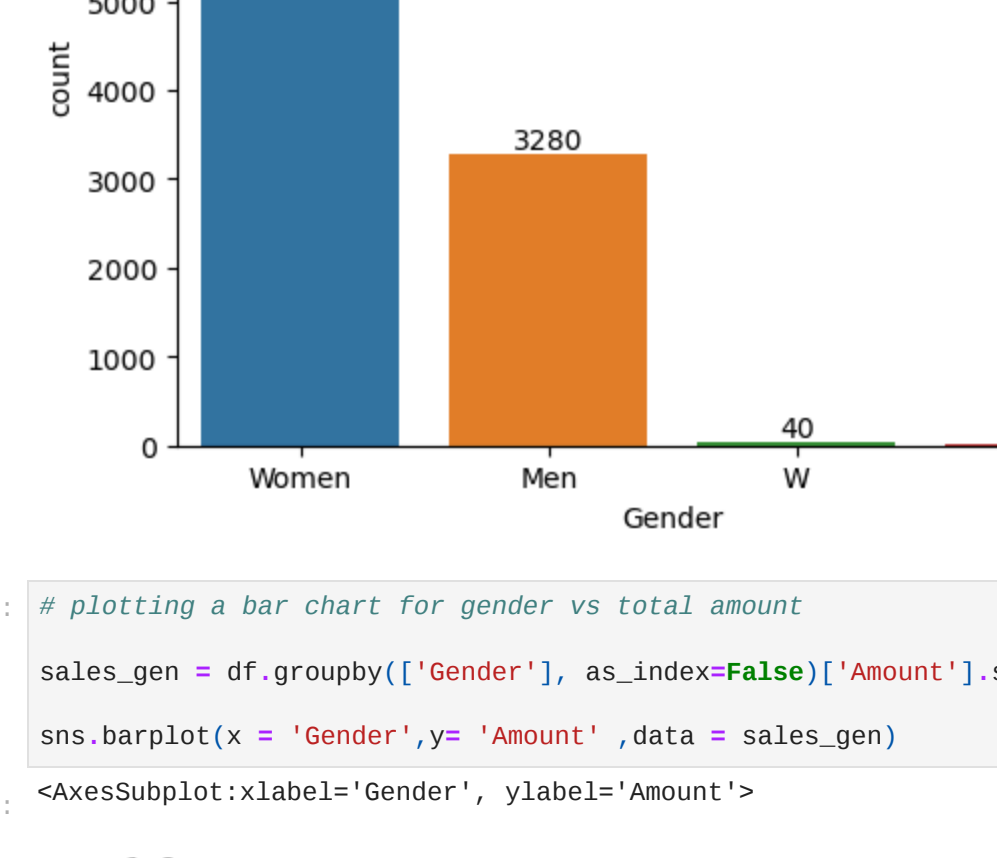
Gender

In [11]:
```python
# plotting a bar chart for Gender and it's count

ax = sns.countplot(x = 'Gender',data = df)

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



In [12]:
```python
# plotting a bar chart for gender vs total amount

sales_gen = df.groupby(['Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)

sns.barplot(x = 'Gender',y= 'Amount' ,data = sales_gen)
```

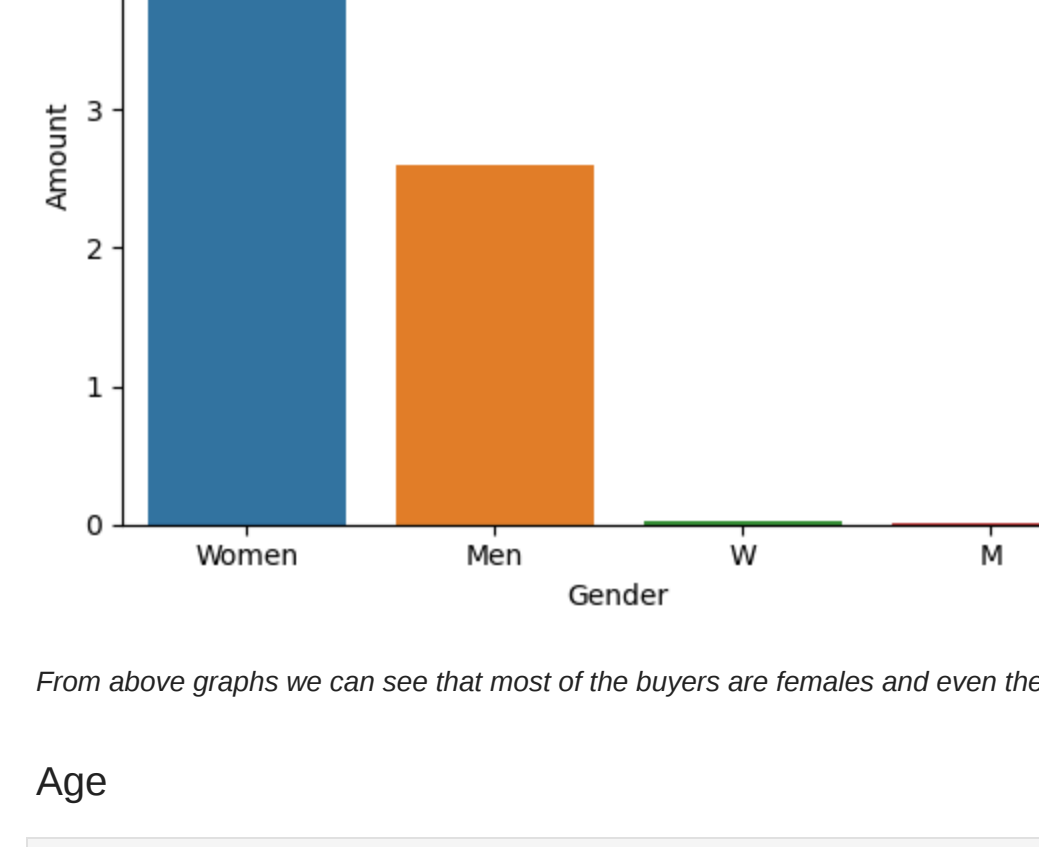
Out[12]:
```
<AxesSubplot:xlabel='Gender', ylabel='Amount'>
```



From above graphs we can see that most of the buyers are females and even the purchasing power of females are greater than men

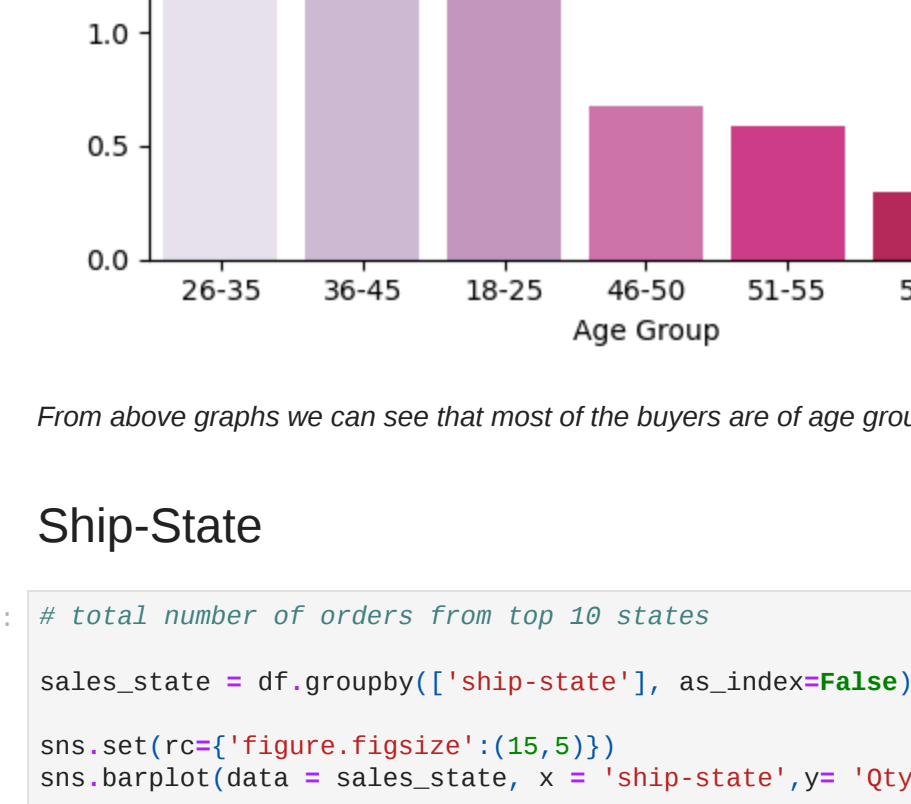
Age

In [13]:
```python
ax = sns.countplot(data = df, x = 'Age Group', hue = 'Gender', color = 'r')

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



In [14]:
```python
# Total Amount vs Age Group
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, color = 'GREEN', palette= 'PuRd')
```

Out[14]:
```
<AxesSubplot:xlabel='Age Group', ylabel='Amount'>
```

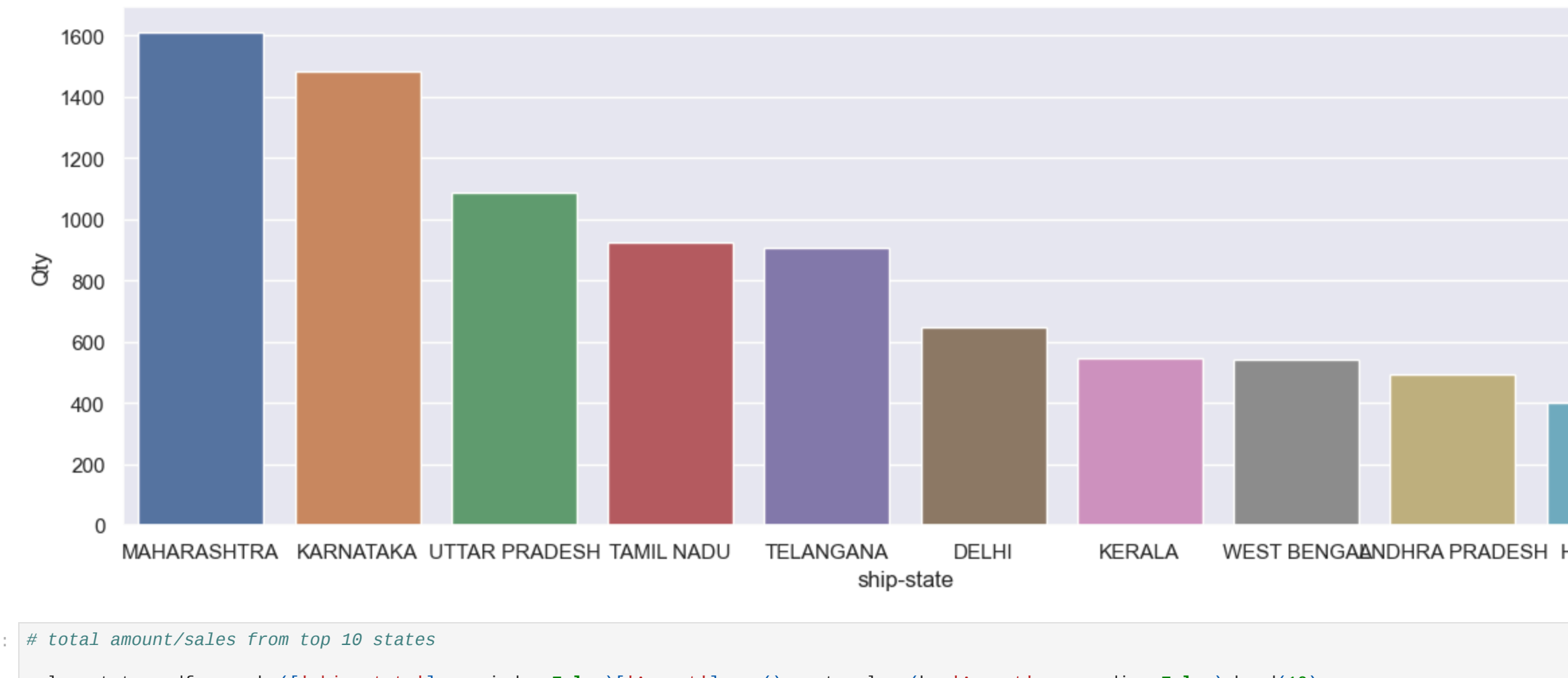


From above graphs we can see that most of the buyers are of age group between 26-35 yrs female

Ship-State

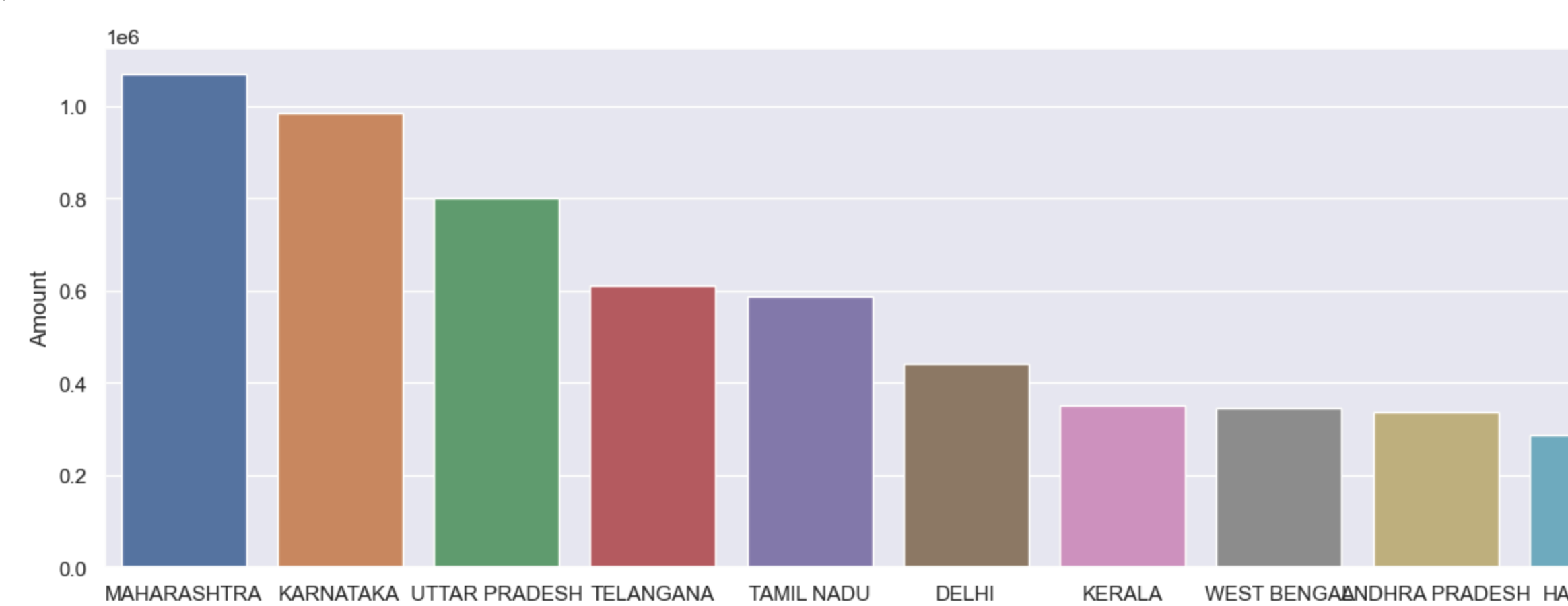
In [15]:
```python
# total number of orders from top 10 states

sales_state = df.groupby(['ship-state'], as_index=False)['Qty'].sum().sort_values(by='Qty', ascending=False).head(10)

sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(data = sales_state, x = 'ship-state',y= 'Qty')
```

Out[15]:
```
<AxesSubplot:xlabel='ship-state', ylabel='Qty'>
```



In [51]:
```python
# total amount/sales from top 10 states

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

sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(data = sales_state, x = 'ship-state',y= 'Amount')
```

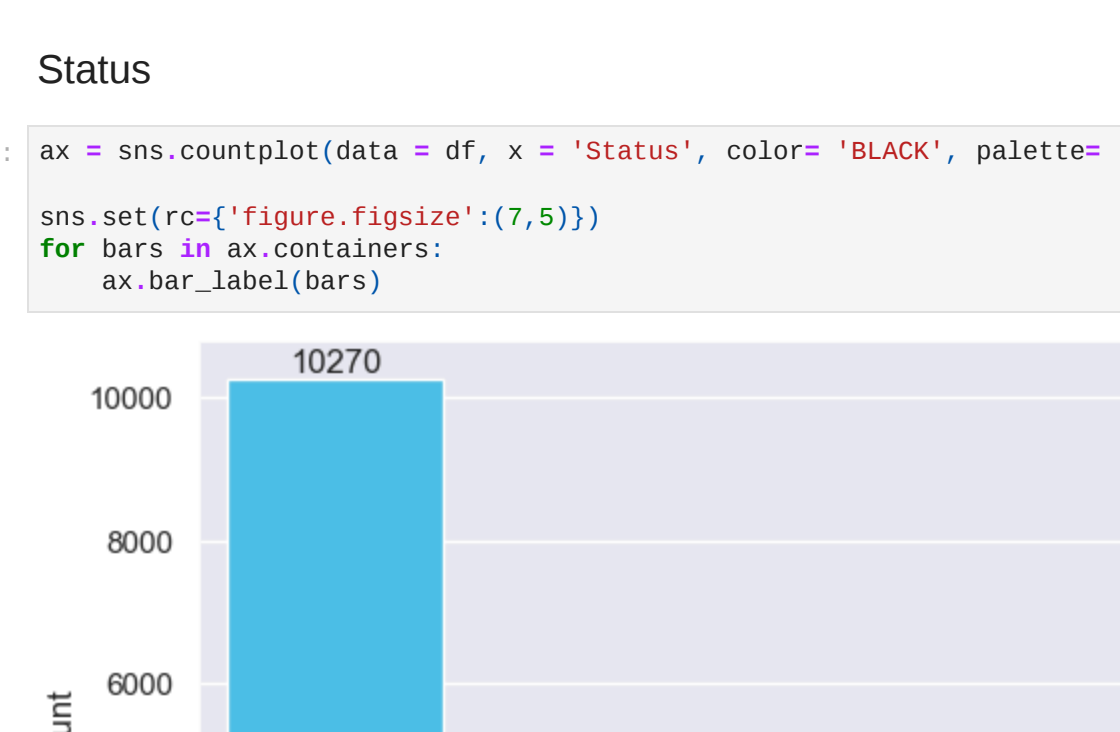
Out[51]:
```
<AxesSubplot:xlabel='ship-state', ylabel='Amount'>
```



From above graphs we can see that most of the shipments are from Maharashtra, Karnataka and Uttar Pradesh respectively

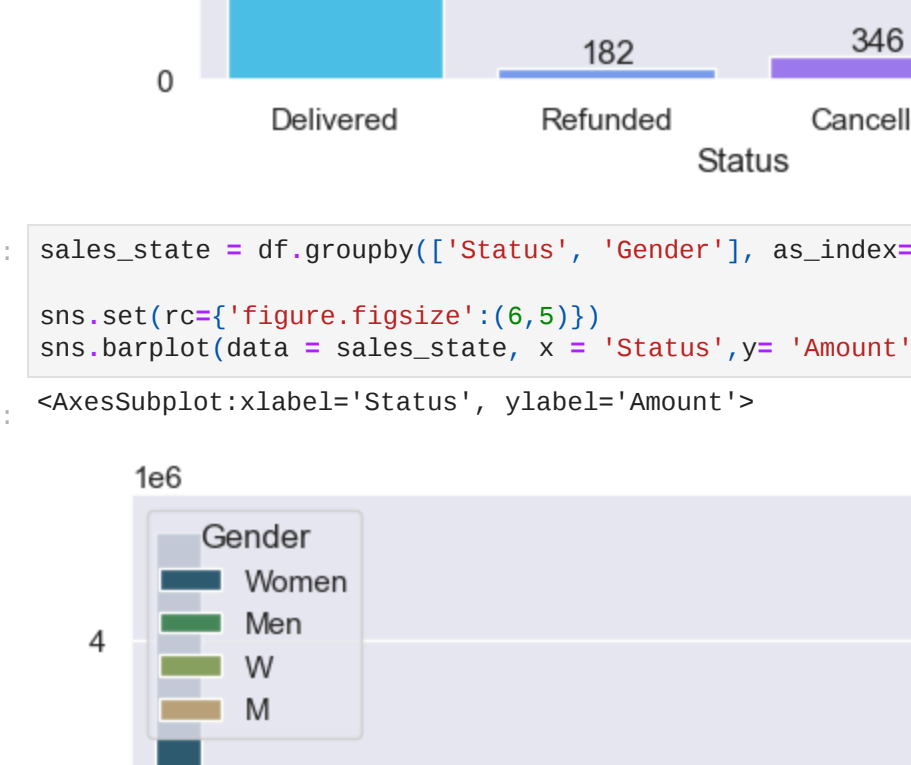
Status

In [22]:
```python
ax = sns.countplot(data = df, x = 'Status', color= 'BLACK', palette= 'cool')

sns.set(rc={'figure.figsize':(7,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



In [55]:
```python
sales_state = df.groupby(['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 = 'Status',y= 'Amount', hue='Gender', palette= 'gist_earth')
```

Out[55]:
```
<AxesSubplot:xlabel='Status', ylabel='Amount'>
```



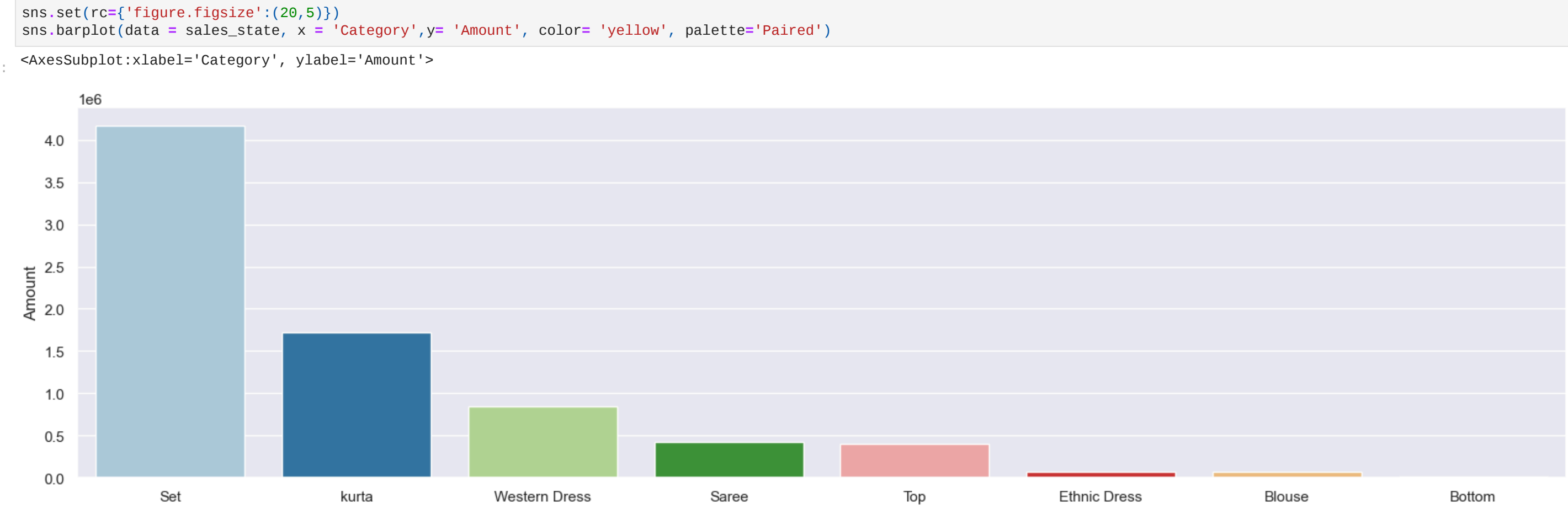
From above graphs we can see that most of the buyers which are womens kept their orders did not returned

Category

In [31]:
```python
sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data = df, x = 'Category')

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

In [36]:
```python
sales_state = df.groupby(['Category'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x = 'Category',y= 'Amount', color= 'yellow', palette='Paired')
```

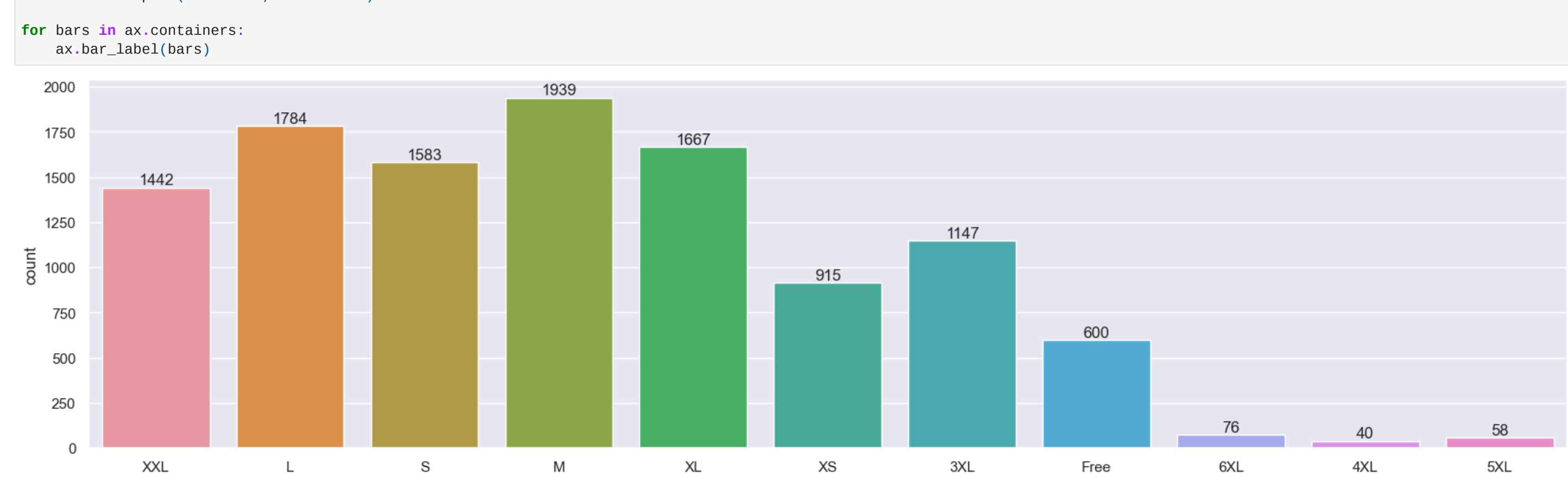
Out[36]:
```
<AxesSubplot:xlabel='Category', ylabel='Amount'>
```



From above graphs we can see that most of the purchased products are Set, kurta and Western Dress

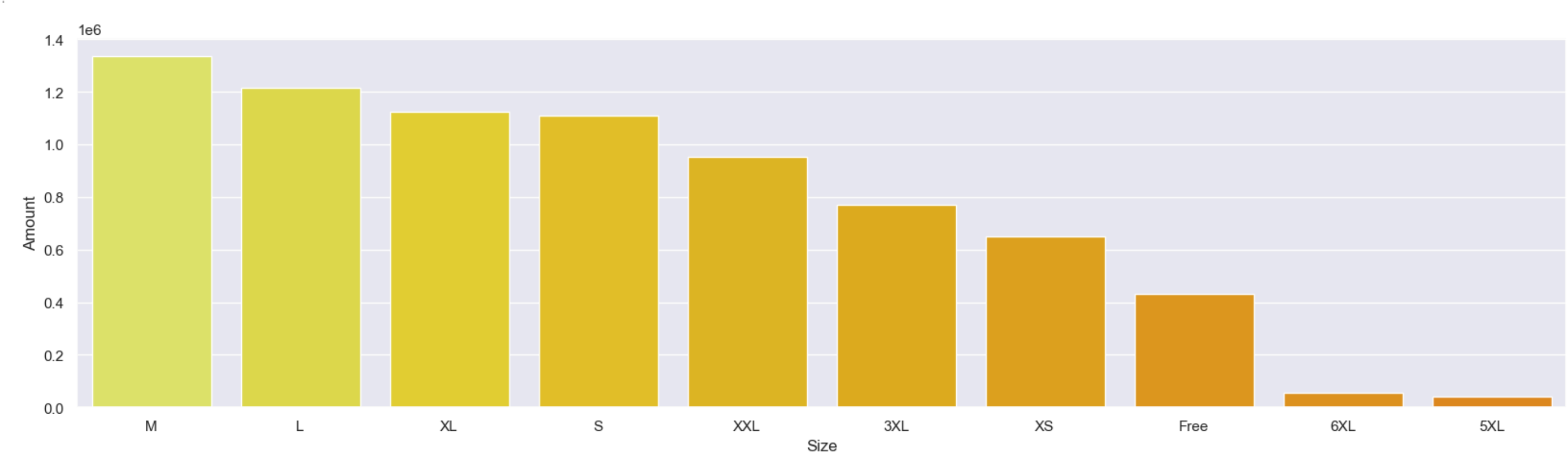
Size

In [37]:
```python
sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data = df, x = 'Size')

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



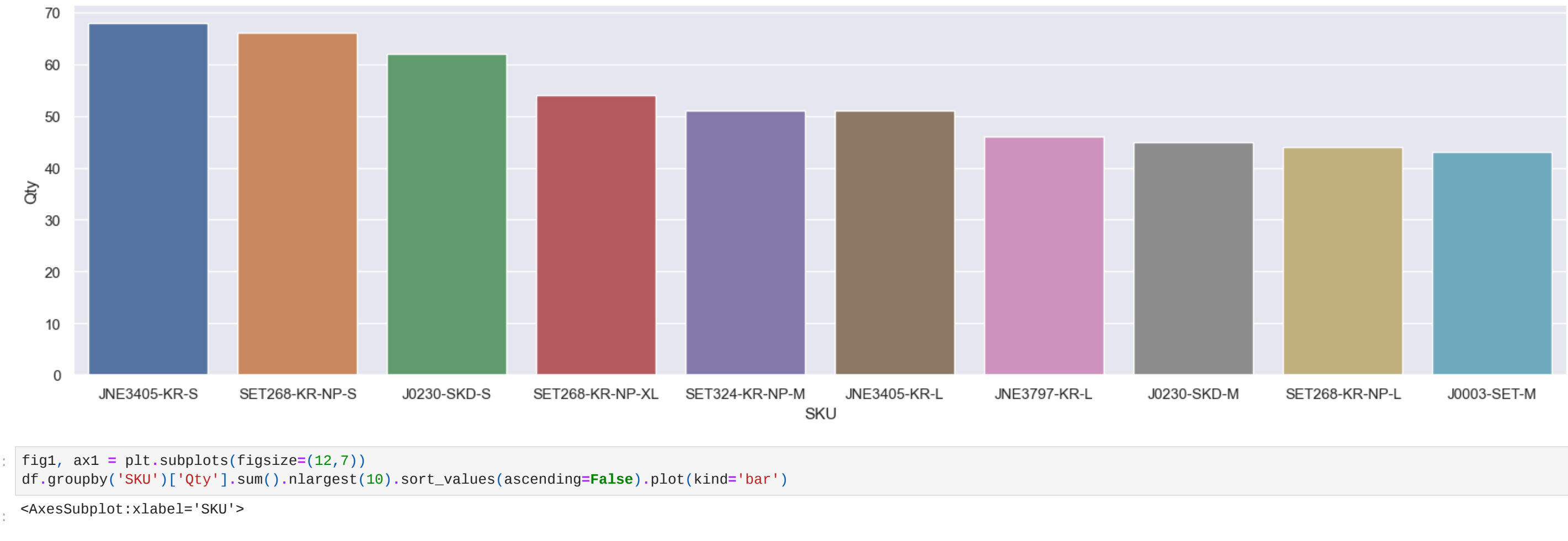
In [41]:
```python
sales_state = df.groupby(['Size'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False).head(10)

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x = 'Size',y= 'Amount', palette= 'Wistia')
```

Out[41]:
```
<AxesSubplot:xlabel='Size', ylabel='Amount'>
```



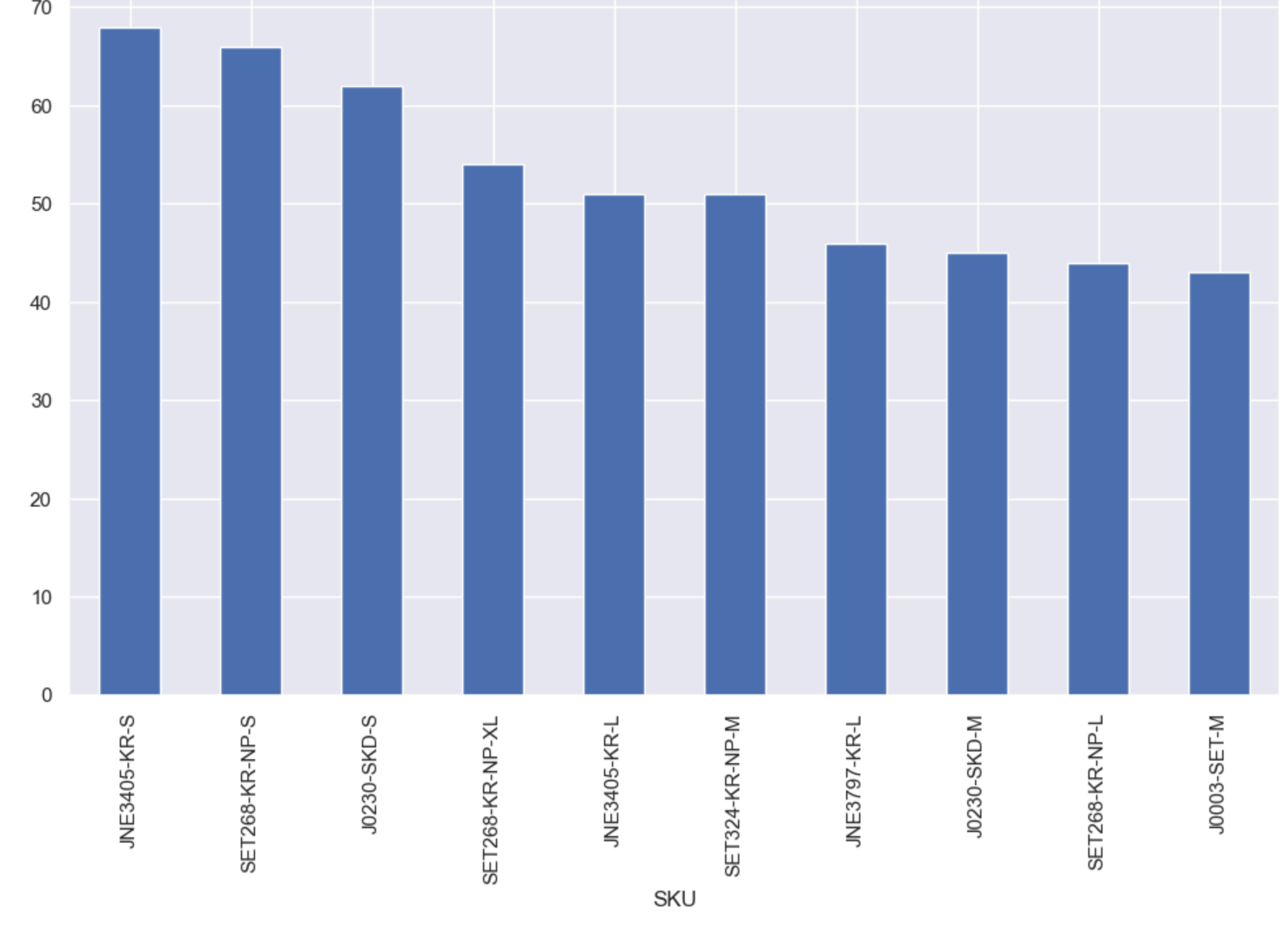
In [44]:
```python
sales_state = df.groupby(['SKU'], as_index=False)['Qty'].sum().sort_values(by='Qty', ascending=False).head(10)

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x = 'SKU',y= 'Qty')
```

Out[44]:
```
<AxesSubplot:xlabel='SKU', ylabel='Qty'>
```



In [48]:
```python
fig1, ax1 = plt.subplots(figsize=(12,7))
df.groupby('SKU')['Qty'].sum().nlargest(10).sort_values(ascending=False).plot(kind='bar')
```

Out[48]:
```
<AxesSubplot:xlabel='SKU'>
```



Conclusion:

Womens of age group 26-35 yrs from Maharastra, Karnataka and UP orders Set, Kurtas and western dress and order perfectly as per their size.

Thank you!