In [2]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as mlt
%matplotlib inline
import seaborn as sns
```

In [3]:
```python
sf = pd.read_csv('AmazonSale.csv' ,encoding='unicode_escape')
```

```
C:\Users\Windows\AppData\Local\Temp\ipykernel_9356\3801401524.py:1: DtypeWarning: Columns (23) have mixed types. Specify dtype option on import or set low_memory=False.
  sf = pd.read_csv('AmazonSale.csv' ,encoding='unicode_escape')
```

In [5]:
```python
sf.shape
```

Out[5]:
```
(128975, 24)
```

In [7]:
```python
sf.columns
```

Out[7]:
```
Index(['index', 'Order ID', 'Date', 'Status', 'Fulfilment', 'Sales Channel ',
       'ship-service-level', 'Style', 'SKU', 'Category', 'Size', 'ASIN',
       'Courier Status', 'Qty', 'currency', 'Amount', 'ship-city',
       'ship-state', 'ship-postal-code', 'ship-country', 'promotion-ids',
       'B2B', 'fulfilled-by', 'Unnamed: 22'],
      dtype='object')
```

In [10]:
```python
sf.head()
```

Out[10]:

	index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Style	SKU	Category	...	currency	Amount	ship-city	ship-state	ship-postal-code
0	0	405-8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	SET389	SET389-KR-NP-S	Set	...	INR	647.62	MUMBAI	MAHARASHTRA	400081.0
1	1	171-9198151-1101146	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3781	JNE3781-KR-XXXL	kurta	...	INR	406.00	BENGALURU	KARNATAKA	560085.0
2	2	404-0687676-7273146	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3371	JNE3371-KR-XL	kurta	...	INR	329.00	NAVI MUMBAI	MAHARASHTRA	410210.0
3	3	403-9615377-8133951	04-30-22	Cancelled	Merchant	Amazon.in	Standard	J0341	J0341-DR-L	Western Dress	...	INR	753.33	PUDUCHERRY	PUDUCHERRY	605008.0
4	4	407-1069790-7240320	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3671	JNE3671-TU-XXXL	Top	...	INR	574.00	CHENNAI	TAMIL NADU	600073.0

5 rows × 24 columns

In [17]:
```python
sf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 128975 entries, 0 to 128974
Data columns (total 24 columns):
 #   Column              Non-Null Count   Dtype  
---  ------              --------------   -----  
 0   index               128975 non-null  int64  
 1   Order ID            128975 non-null  object 
 2   Date                128975 non-null  object 
 3   Status              128975 non-null  object 
 4   Fulfilment          128975 non-null  object 
 5   Sales Channel       128975 non-null  object 
 6   ship-service-level  128975 non-null  object 
 7   Style               128975 non-null  object 
 8   SKU                 128975 non-null  object 
 9   Category            128975 non-null  object 
 10  Size                128975 non-null  object 
 11  ASIN                128975 non-null  object 
 12  Courier Status      122108 non-null  object 
 13  Qty                 128975 non-null  int64  
 14  currency            121180 non-null  object 
 15  Amount              121180 non-null  float64
 16  ship-city           128942 non-null  object 
 17  ship-state          128942 non-null  object 
 18  ship-postal-code    128942 non-null  float64
 19  ship-country        128942 non-null  object 
 20  promotion-ids       79822 non-null   object 
 21  B2B                 128975 non-null  bool   
 22  fulfilled-by        39277 non-null   object 
 23  Unnamed: 22         79925 non-null   object 
dtypes: bool(1), float64(2), int64(2), object(19)
memory usage: 22.8+ MB
```

In [40]:
```python
sf.isnull().sum()
```

Out[40]:
```
index                     0
Order ID                  0
Date                      0
Status                    0
Fulfilment                0
Sales Channel             0
ship-service-level        0
Category                  0
Size                      0
ASIN                      0
Courier Status         6867
Qty                       0
currency               7795
Amount                 7795
ship-city                33
ship-state               33
ship-postal-code         33
ship-country             33
promotion-ids         49153
B2B                       0
fulfilled-by          89698
Unnamed: 22           49050
dtype: int64
```

In [9]:
```python
sf.dropna(inplace=True)
```

In [42]:
```python
sf.isnull().sum()
```

Out[42]:
```
index                 0
Order ID              0
Date                  0
Status                0
Fulfilment            0
Sales Channel         0
ship-service-level    0
Category              0
Size                  0
ASIN                  0
Courier Status        0
Qty                   0
currency              0
Amount                0
ship-city             0
ship-state            0
ship-postal-code      0
ship-country          0
promotion-ids         0
B2B                   0
fulfilled-by          0
Unnamed: 22           0
dtype: int64
```

In [7]:
```python
sf[sf.duplicated()]
```

Out[7]:

index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Style	SKU	Category	...	currency	Amount	ship-city	ship-state	ship-postal-code	ship-country	promotion-ids	B2B	fulfilled-by	Unnam

0 rows × 24 columns

In [8]:
```python
sf['Category'].value_counts()
```

Out[8]:
```
Category
Set              50284
kurta            49877
Western Dress    15500
Top              10622
Ethnic Dress      1159
Blouse             926
Bottom             440
Saree              164
Dupatta              3
Name: count, dtype: int64
```

In [43]:
```python
sf.describe(include=object)
```

Out[43]:

	index	Qty	Amount	ship-postal-code
count	19379.000000	19379.000000	19379.000000	19379.000000
mean	86650.405542	1.004799	673.245162	464618.933743
std	22204.535858	0.082081	287.866514	191536.937504
min	49051.000000	1.000000	0.000000	110001.000000
25%	67813.000000	1.000000	471.000000	380058.000000
50%	83317.000000	1.000000	678.000000	500032.000000
75%	105153.500000	1.000000	786.000000	600039.000000
max	128891.000000	5.000000	2796.000000	855107.000000

In [12]:
```python
sum(sf.Amount.isna())
```

Out[12]:
```
7795
```

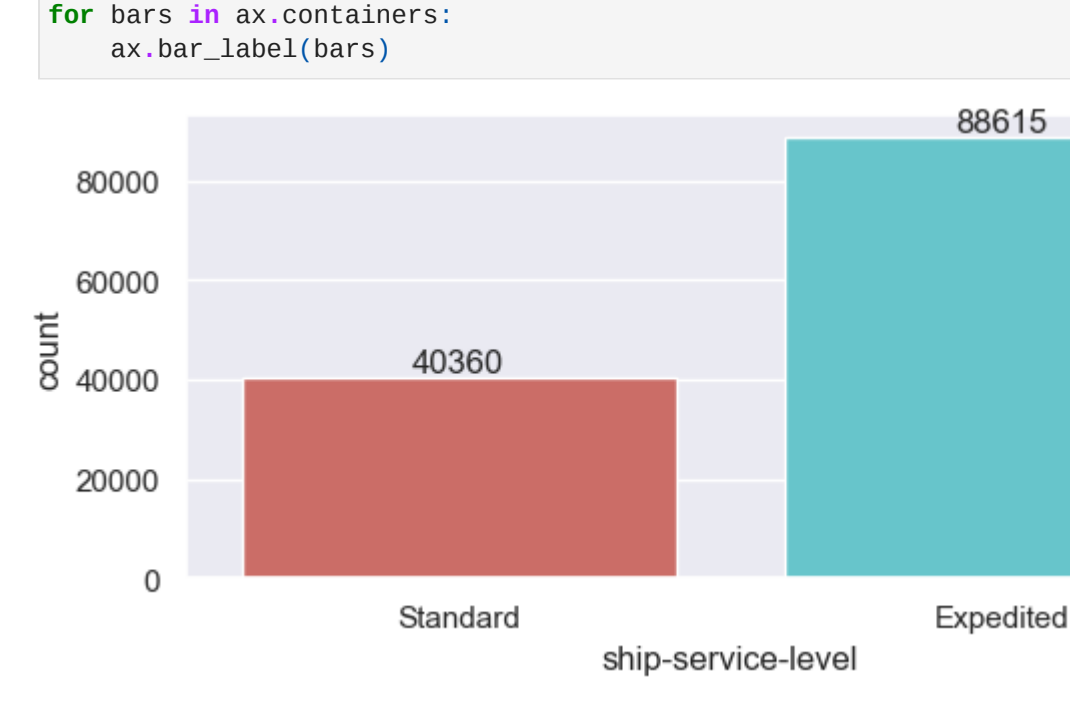
In [13]:
```python
sf.groupby('ship-state')['Qty'].sum()
```

Out[13]:
```
ship-state
ANDAMAN & NICOBAR      226
ANDHRA PRADESH        4819
APO                      0
AR                       1
ARUNACHAL PRADESH      130
                      ... 
goa                      5
orissa                   1
punjab                  13
rajasthan                6
rajsthan                 1
Name: Qty, Length: 69, dtype: int64
```

Quantity of orders product among states

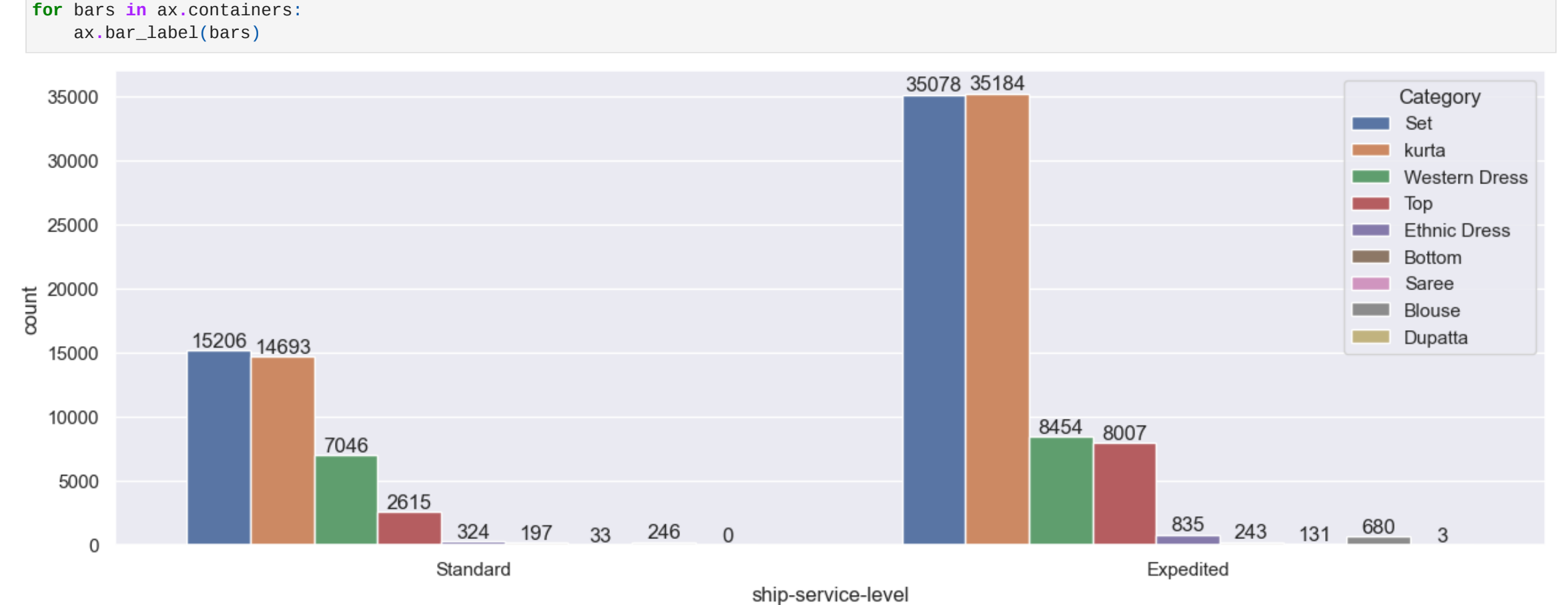
Exploratory Data Analysis

In [9]:
```python
ax = sns.countplot(x='ship-service-level', data=sf ,palette = 'hls')
sns.set(rc={'figure.figsize':(7,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```

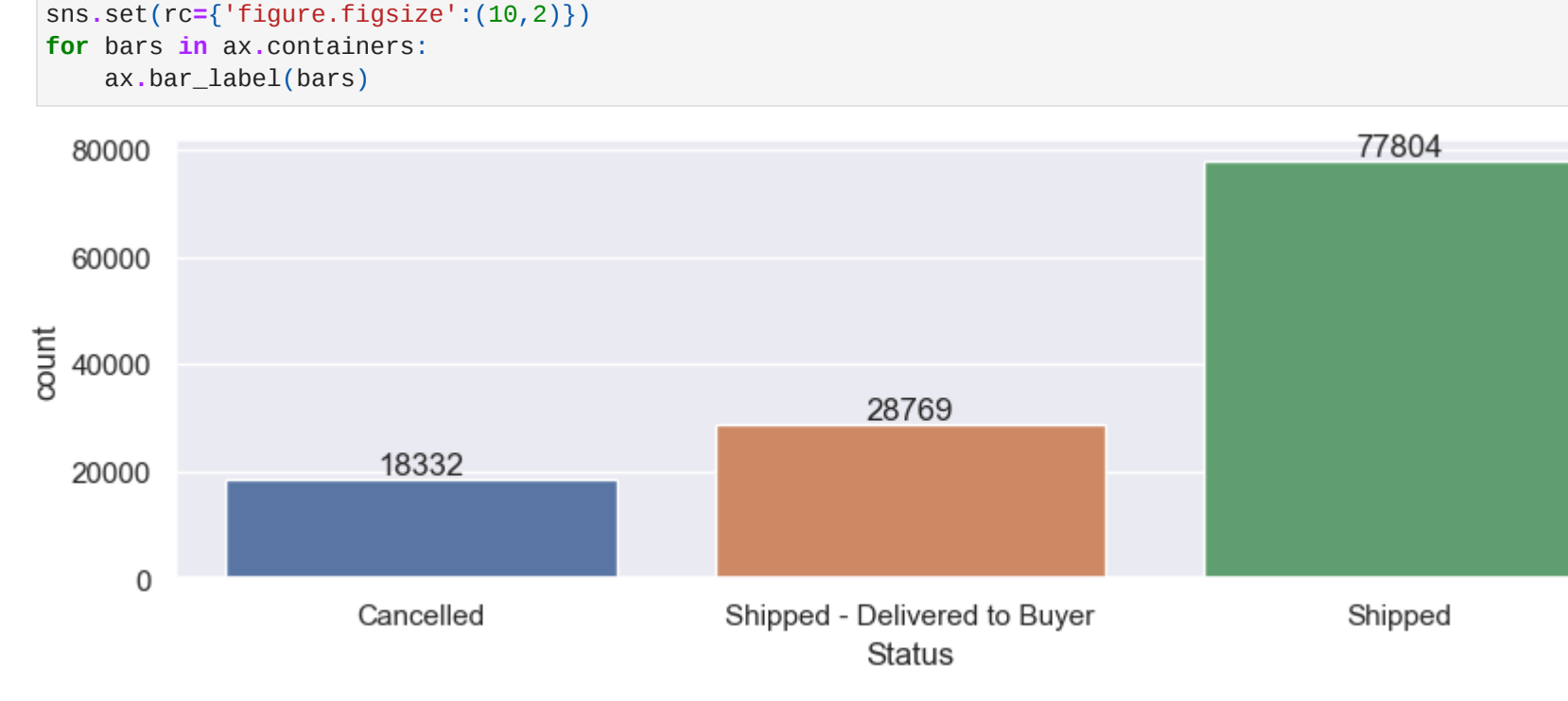


People order more by expedited ship method

In [24]:
```python
ax = sns.countplot(x='ship-service-level', data=sf ,hue='Category' )
sns.set(rc={'figure.figsize':(15,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```

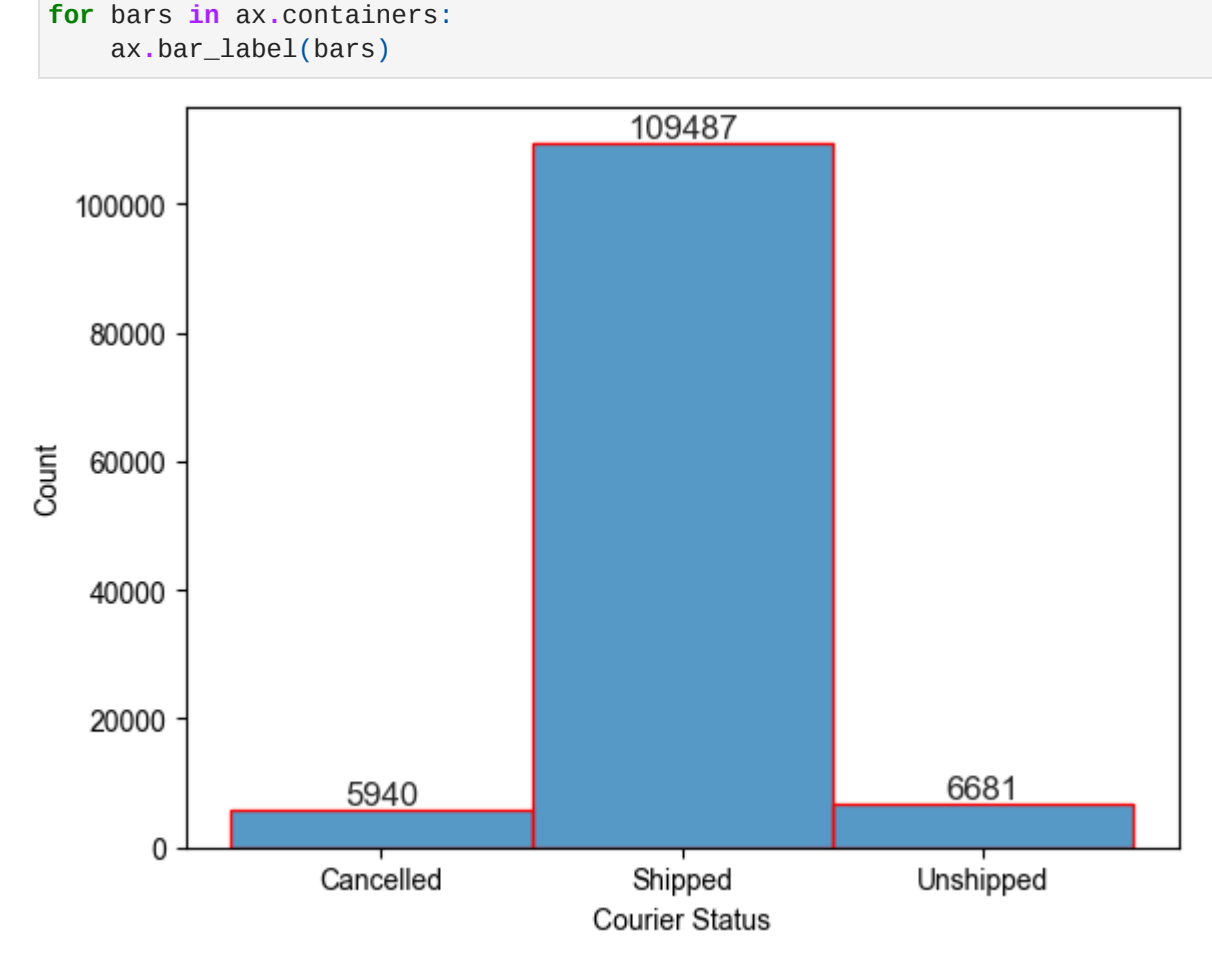


In [22]:
```python
ax = sns.countplot(x='Status', data=sf,order=['Cancelled','Shipped - Delivered to Buyer','Shipped'] )
sns.set(rc={'figure.figsize':(10,2)})
for bars in ax.containers:
    ax.bar_label(bars)
```

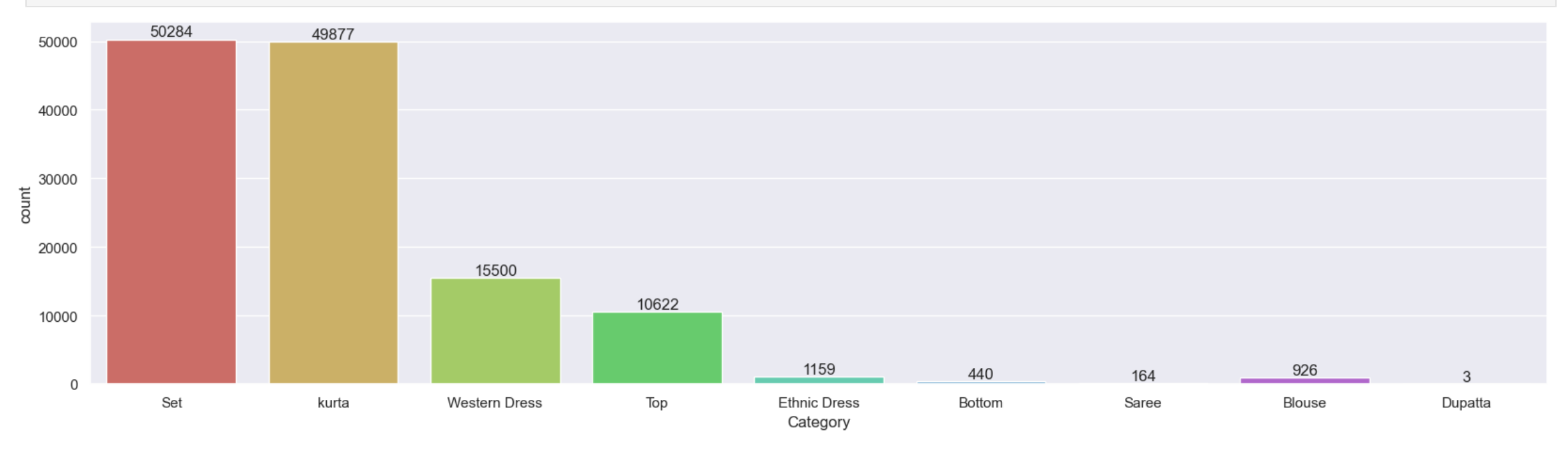


from above graph we can see that the canceeled orders are in more numbers and also delivery procedure has to be fast because more numbers of orders are in shipped state

In [11]:
```python
ax = sns.histplot(x='Courier Status', data=sf ,edgecolor='red')
sns.set(rc={'figure.figsize':(7,8)})
for bars in ax.containers:
    ax.bar_label(bars)
```

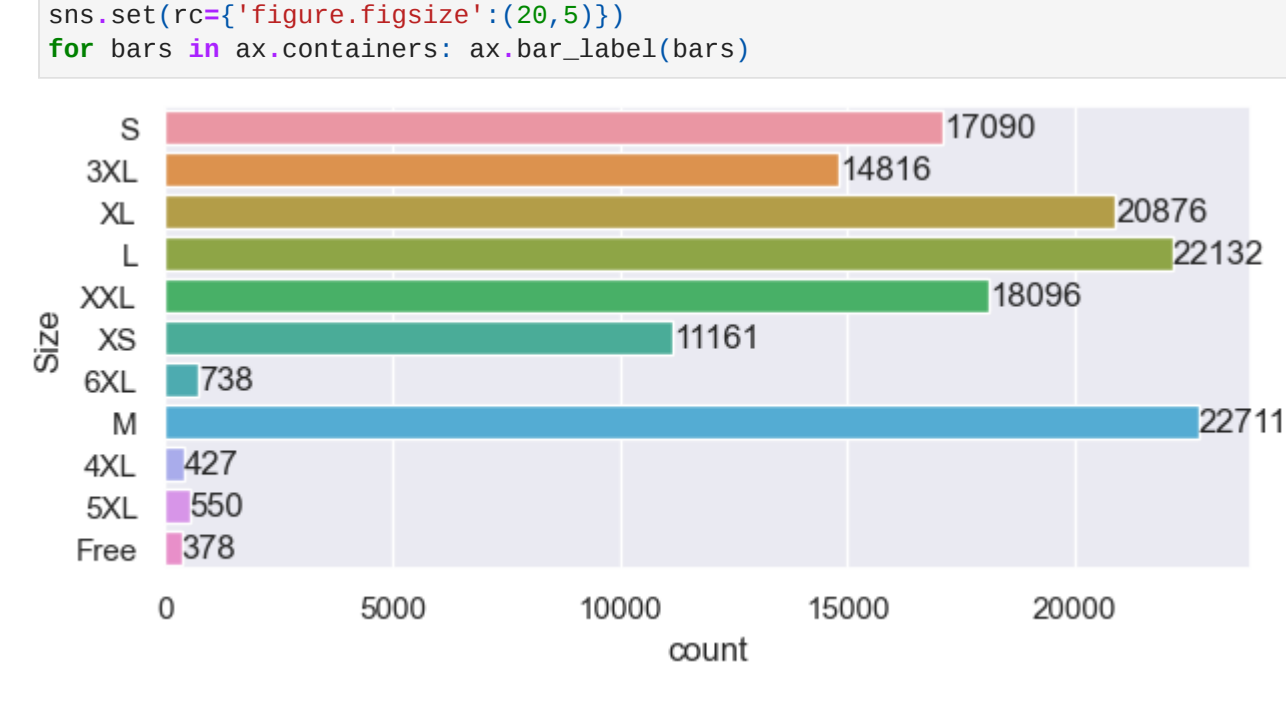


In [7]:
```python
ax = sns.countplot(x='Category', data=sf,color='violet',palette = 'hls')
sns.set(rc={'figure.figsize':(20,5)})
for bars in ax.containers: ax.bar_label(bars)
```



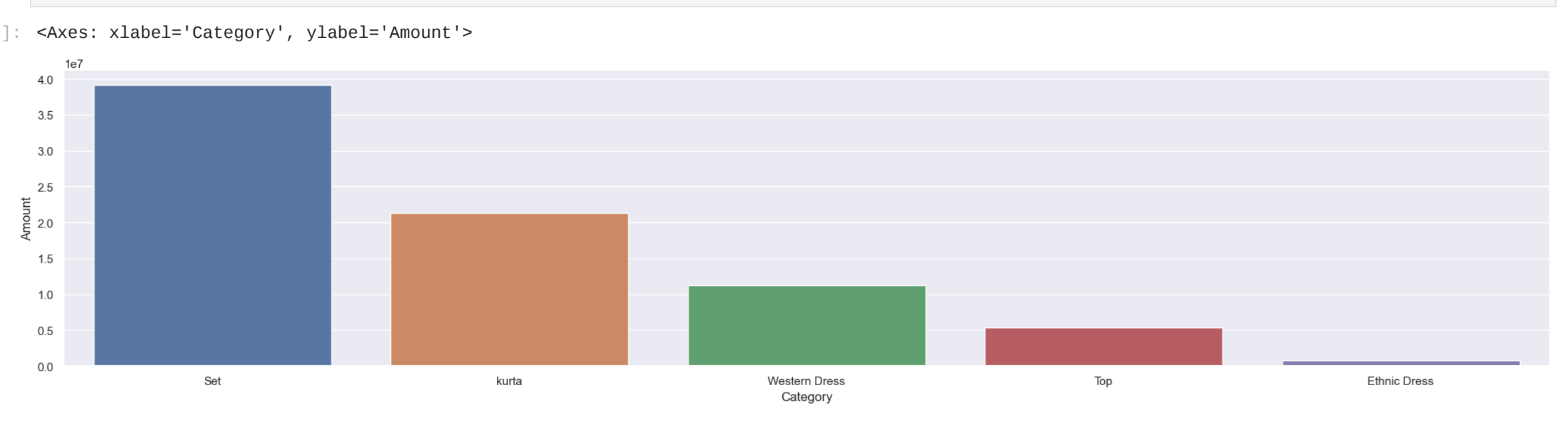
From above graph we can see that Set , Kurta , western Dresses and top are the topmost orders categories

In [27]:
```python
ax = sns.countplot(y='Size', data=sf)
sns.set(rc={'figure.figsize':(20,5)})
for bars in ax.containers: ax.bar_label(bars)
```



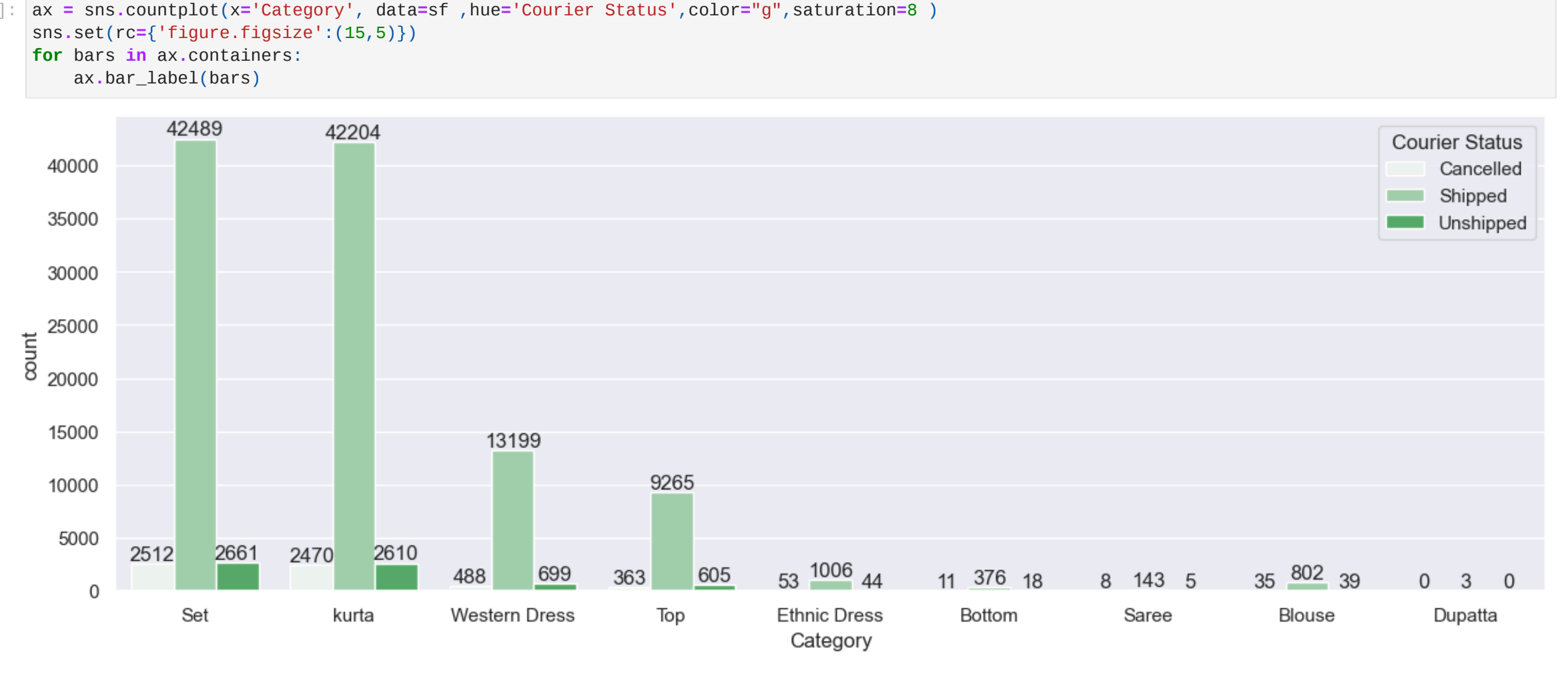
In [7]:
```python
sales_state = sf.groupby(['Category'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False).head(5)
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x = 'Category',y= 'Amount')
```

Out[7]:
```
<Axes: xlabel='Category', ylabel='Amount'>
```



SEt , Kurta ,western Dress , TOP are most selling products

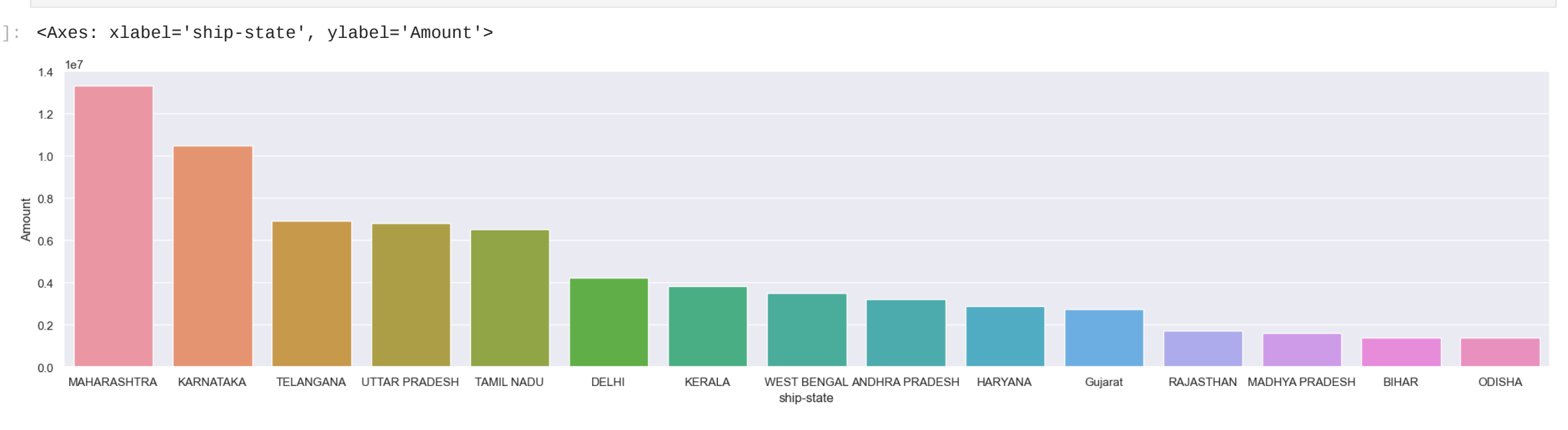
In [71]:
```python
ax = sns.countplot(x='Category', data=sf ,hue='Courier Status',color="g",saturation=8 )
sns.set(rc={'figure.figsize':(15,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



most cancelled and unshipped orders are in set , kurta and western Dress categories

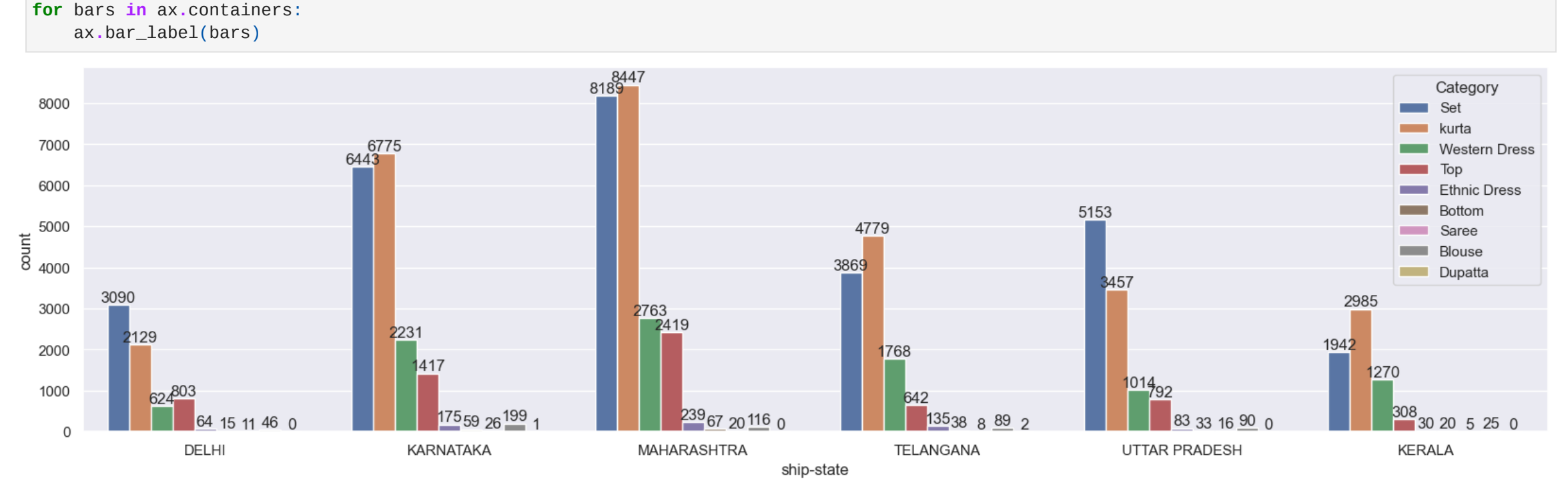
In [25]:
```python
sales_state = sf.groupby(['ship-state'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False).head(15)
sns.set(rc={'figure.figsize':(25,5)})
sns.barplot(data = sales_state, x = 'ship-state',y= 'Amount')
```

Out[25]:
```
<Axes: xlabel='ship-state', ylabel='Amount'>
```



From above graph we see the sale according to top selling states

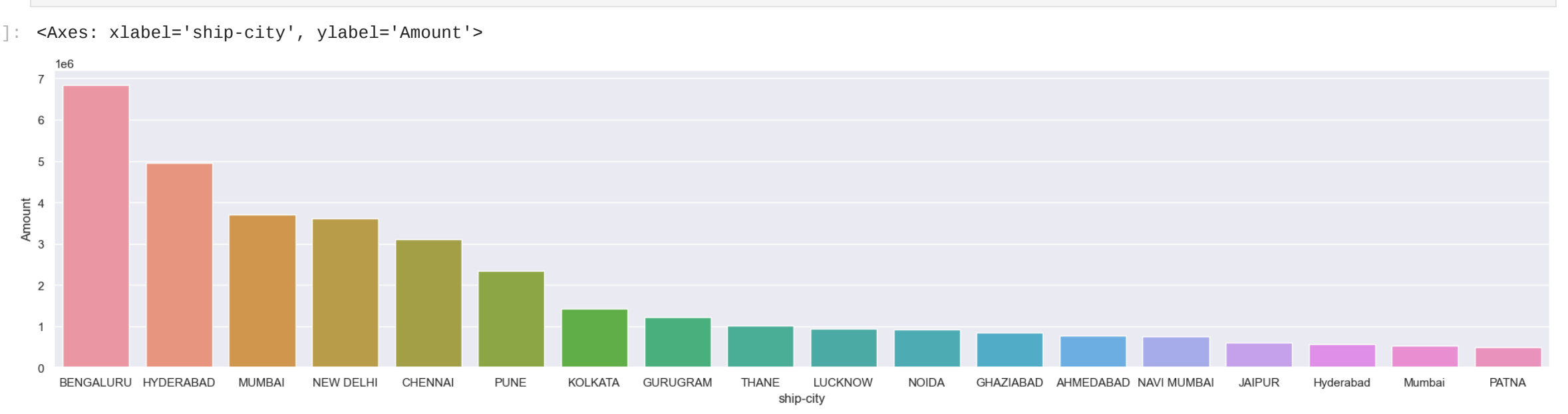
In [49]:
```python
ax = sns.countplot(x='ship-state', data=sf ,hue='Category' ,order =['DELHI','KARNATAKA','MAHARASHTRA','TELANGANA','UTTAR PRADESH','KERALA'])
sns.set(rc={'figure.figsize':(20,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



from above graph we can say that Maharastra, Karnatka , Telangana , Uttar pradesh , Tamil Naidu and Delhi are topmost purchasing states

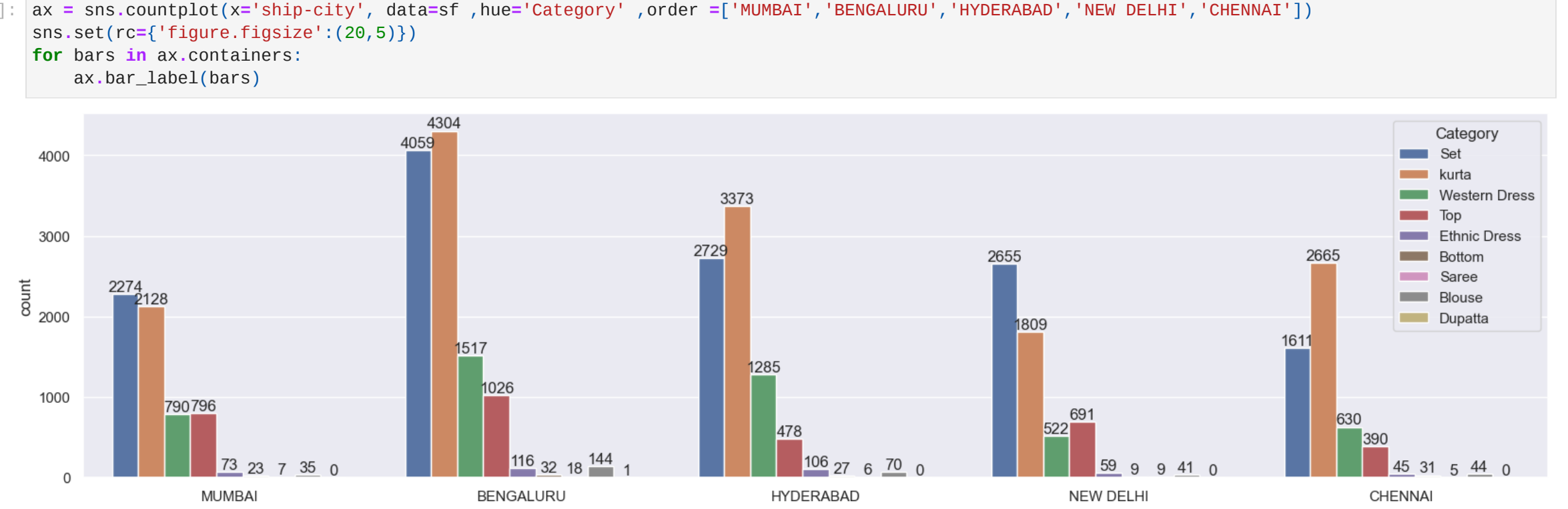
In [8]:
```python
sales_state = sf.groupby(['ship-city'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False).head(18)
sns.set(rc={'figure.figsize':(25,5)})
sns.barplot(data = sales_state, x = 'ship-city',y= 'Amount')
```

Out[8]:
```
<Axes: xlabel='ship-city', ylabel='Amount'>
```



Cities like Bengaluru , Hyderabad , mumbai ,New delhi , chennai , pune , kolkata are topmost purchasing cities

In [16]:
```python
ax = sns.countplot(x='ship-city', data=sf ,hue='Category' ,order =['MUMBAI','BENGALURU','HYDERABAD','NEW DELHI','CHENNAI'])
sns.set(rc={'figure.figsize':(20,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



sales of products according to top seeling cities among states

In [ ]: