In [ ]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:

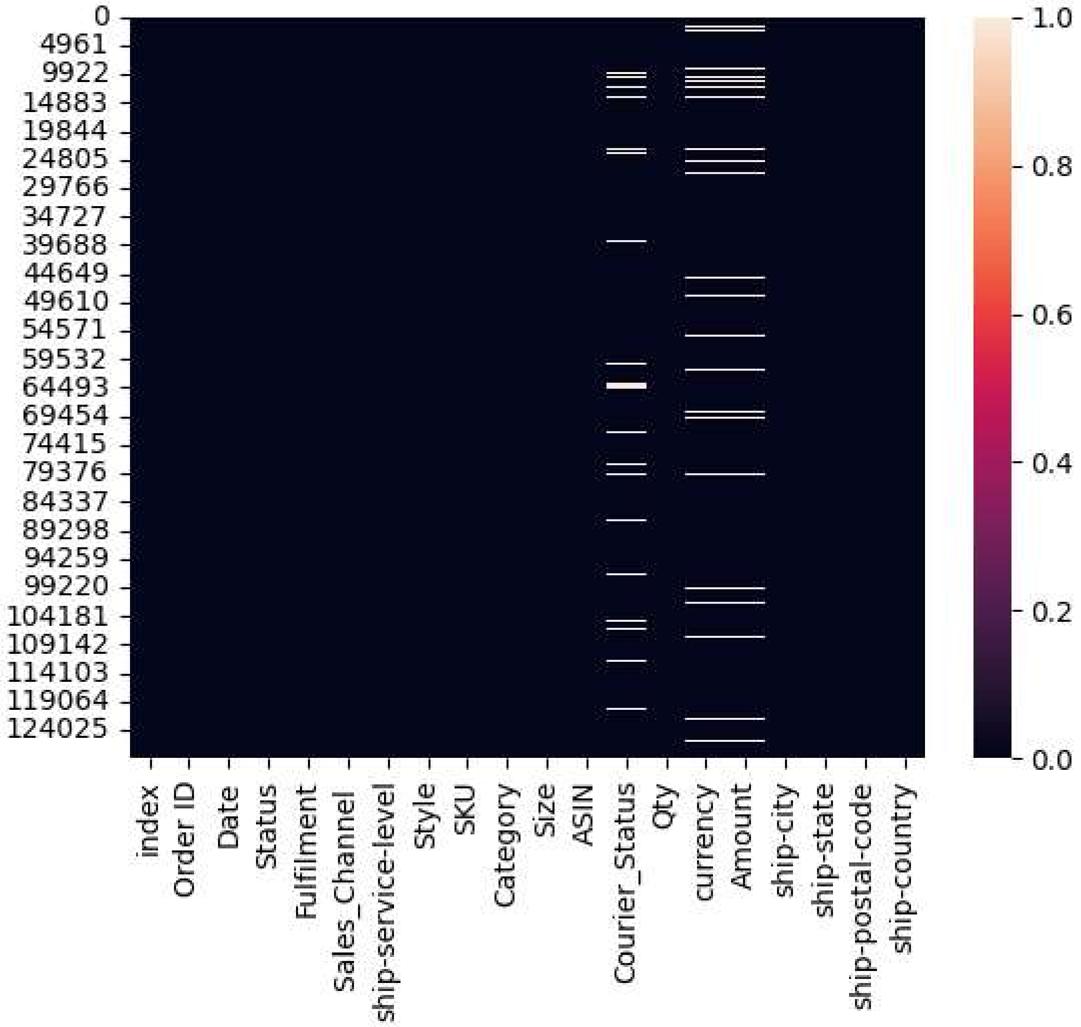
```
df=pd.read_csv("C:\\Users\\jansi pc\\Desktop\\New folder\\Amazon Sale Report.csv")
df1=df.copy()
```

In [ ]:

```
sns.heatmap(df.isnull())
```

Out[ ]:

```
<Axes: >
```



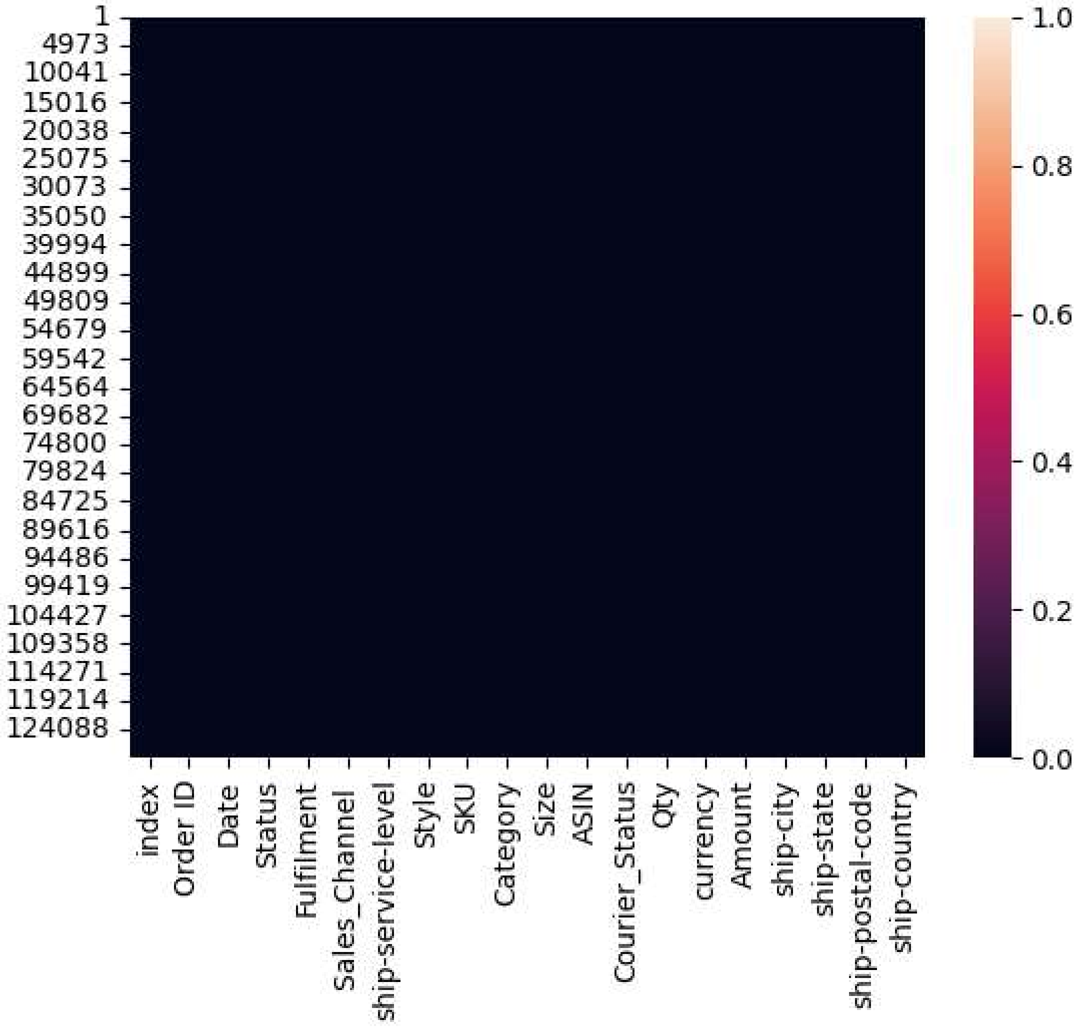
In [ ]:

```
df.dropna(subset=["Courier_Status"],axis=0,inplace=True)
mean1=df["Qty"].mean()
df["Qty"].replace(np.nan,mean1,inplace=True)
mean2=df["Amount"].mean()
df["Amount"].replace(np.nan,mean2,inplace=True)
df["currency"].fillna("INR",inplace=True)
```

In [ ]:

```
sns.heatmap(df.isnull())
```

```
Out[ ]: <Axes: >
```



```
In [ ]: df["Courier_Status"].unique()
```

```
Out[ ]: array(['Shipped', 'Cancelled', 'Unshipped'], dtype=object)
```

```
In [ ]: df=df[df["Courier_Status"]=="Shipped"]
```

In [ ]:

```
df["year"]=pd.DatetimeIndex(df.Date).year
df["month"]=pd.DatetimeIndex(df.Date).month
df["Day"]=pd.DatetimeIndex(df.Date).day
df=df[df["month"]!=3]
```

In [ ]:

```
df.sample(10)
```

Out[ ]:

	index	Order ID	Date	Status	Fulfilment	Sales_Channel	ship-service-level	Style	SKU	Category	...	Qty	currency	Amo
63544	63544	406-5291558-2295546	5/20/2022	Shipped	Amazon	Amazon.in	Expedited	SET224	SET224-KR-NP-XXXL	Set	...	1	INR	11
707	707	171-6640050-9516349	4/30/2022	Shipped	Amazon	Amazon.in	Expedited	JNE3534	JNE3534-KR-XS	kurta	...	1	INR	3
34273	34273	405-0396225-0625957	4/10/2022	Shipped	Amazon	Amazon.in	Expedited	J0338	J0338-DR-L	Western Dress	...	1	INR	7
51623	51623	406-7139818-8082728	5/29/2022	Shipped	Amazon	Amazon.in	Expedited	JNE3634	JNE3634-KR-XL	kurta	...	1	INR	5
48355	48355	407-6024343-9925121	4/1/2022	Shipped	Amazon	Amazon.in	Expedited	J0230	J0230-SKD-L	Set	...	1	INR	9
74813	74813	408-7714562-3952335	5/10/2022	Shipped	Amazon	Amazon.in	Expedited	JNE3601	JNE3601-KR-XS	kurta	...	1	INR	3
61963	61963	404-6520459-5200332	5/21/2022	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	J0041	J0041-SET-M	Set	...	1	INR	6
94882	94882	403-6951804-6688300	6/26/2022	Shipped - Picked Up	Merchant	Amazon.in	Standard	JNE2014	JNE2014-KR-178-M	kurta	...	1	INR	3
96638	96638	408-8721426-3225144	6/24/2022	Shipped	Amazon	Amazon.in	Expedited	SET357	SET357-KR-NP-M	Set	...	1	INR	7

	index	Order ID	Date	Status	Fulfilment	Sales_Channel	ship-service-level	Style	SKU	Category	...	Qty	currency	Amo
36006	36006	405-9694068-8223534	4/9/2022	Shipped	Amazon	Amazon.in	Expedited	J0081	J0081-DR-XXL	Western Dress	...	1	INR	6

10 rows × 23 columns

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

```
Out[ ]: 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', 'year', 'month',
               'Day'],
              dtype='object')
```

```
In [ ]: df.dtypes
```

```
Out[ ]: index                  int64
        Order ID              object
        Date                  object
        Status                object
        Fulfilment            object
        Sales_Channel         object
        ship-service-level    object
        Style                 object
        SKU                   object
        Category              object
        Size                  object
        ASIN                  object
        Courier_Status        object
        Qty                    int64
        currency              object
        Amount               float64
        ship-city             object
        ship-state            object
        ship-postal-code     float64
        ship-country          object
        year                   int32
        month                  int32
        Day                    int32
        dtype: object
```

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

```
Out[ ]: (109335, 23)
```

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

Out[ ]:

	index	Qty	Amount	ship-postal-code	year	month	Day
count	109335.000000	109335.000000	109335.000000	109309.000000	109335.0	109335.000000	109335.000000
mean	64314.049316	1.003741	649.049746	462641.405017	2022.0	4.904614	14.826734
std	37303.325459	0.073052	283.162162	191229.675133	0.0	0.812920	8.696296
min	1.000000	1.000000	0.000000	110001.000000	2022.0	4.000000	1.000000
25%	32087.500000	1.000000	449.000000	382340.000000	2022.0	4.000000	7.000000
50%	64003.000000	1.000000	605.000000	500030.000000	2022.0	5.000000	15.000000
75%	96935.500000	1.000000	788.000000	600015.000000	2022.0	6.000000	22.000000
max	128974.000000	8.000000	5584.000000	855117.000000	2022.0	6.000000	31.000000

In [ ]:

```
df.info()
```

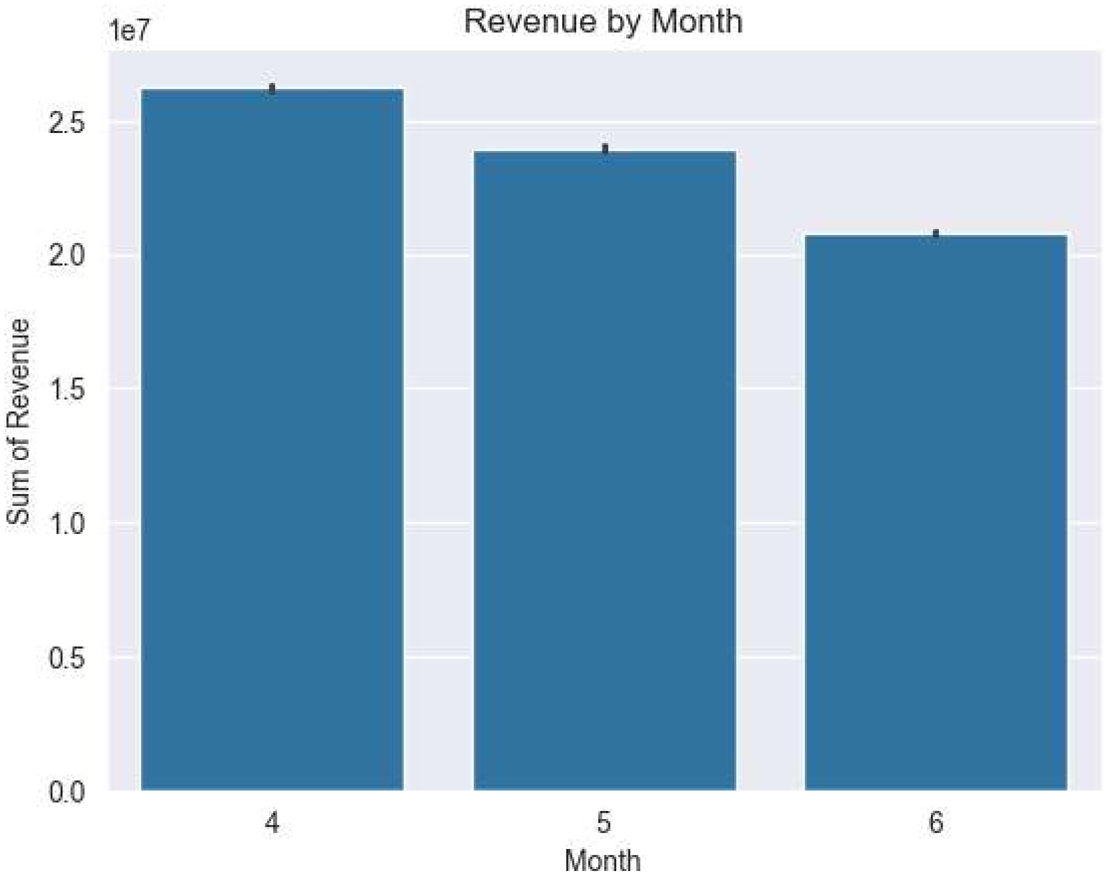
```
<class 'pandas.core.frame.DataFrame'>
Index: 109487 entries, 1 to 128974
Data columns (total 23 columns):
 #   Column              Non-Null Count   Dtype  
---  ------              --------------   -----  
 0   index               109487 non-null  int64  
 1   Order ID            109487 non-null  object 
 2   Date                109487 non-null  object 
 3   Status              109487 non-null  object 
 4   Fulfilment          109487 non-null  object 
 5   Sales_Channel       109487 non-null  object 
 6   ship-service-level  109487 non-null  object 
 7   Style               109487 non-null  object 
 8   SKU                 109487 non-null  object 
 9   Category            109487 non-null  object 
 10  Size                109487 non-null  object 
 11  ASIN                109487 non-null  object 
 12  Courier_Status      109487 non-null  object 
 13  Qty                 109487 non-null  int64  
 14  currency            109487 non-null  object 
 15  Amount              109487 non-null  float64
 16  ship-city           109461 non-null  object 
 17  ship-state          109461 non-null  object 
 18  ship-postal-code    109461 non-null  float64
 19  ship-country        109461 non-null  object 
 20  year                109487 non-null  int32  
 21  month               109487 non-null  int32  
 22  Day                 109487 non-null  int32  
dtypes: float64(2), int32(3), int64(2), object(16)
memory usage: 18.8+ MB
```

In [ ]:

```
sns.set_style("darkgrid")
sns.barplot(x="month",y="Amount",data=df,estimator=sum)
plt.xlabel("Month")
plt.ylabel("Sum of Revenue")
plt.title("Revenue by Month")
```

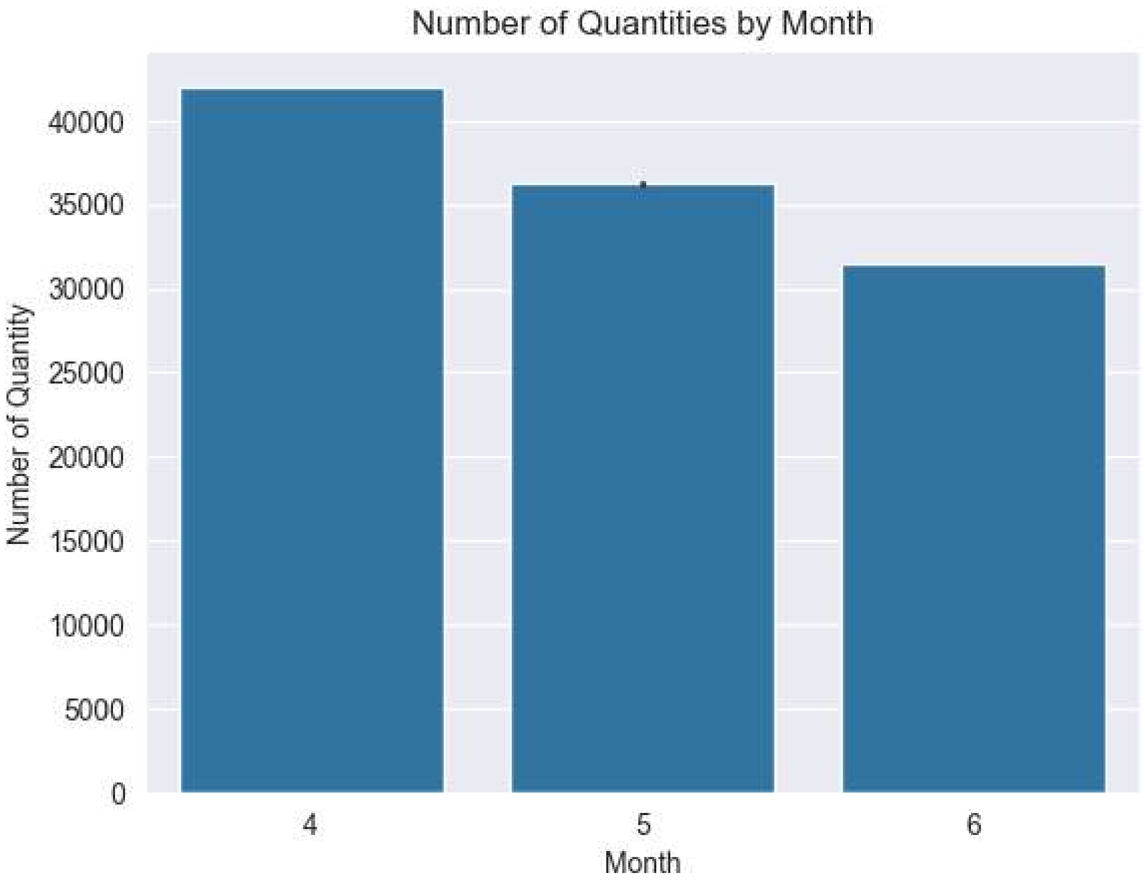
Out[ ]:

```
Text(0.5, 1.0, 'Revenue by Month')
```



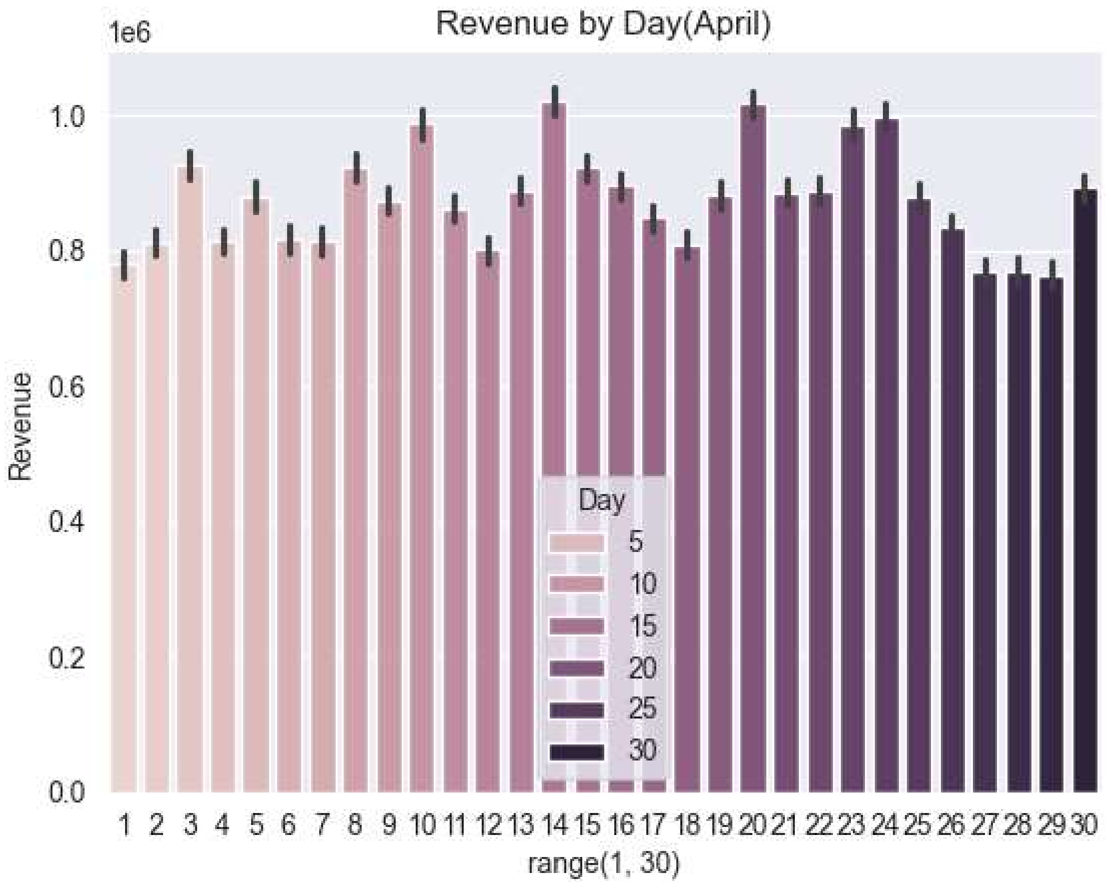
```
In [ ]: sns.set_style("darkgrid")
        sns.barplot(x="month",y="Qty",data=df,estimator=sum)
        plt.xlabel("Month")
        plt.ylabel("Number of Quantity")
        plt.title("Number of Quantities by Month")
```

```
Out[ ]: Text(0.5, 1.0, 'Number of Quantities by Month')
```



```
In [ ]: df4=df[df["month"]==4]
        sns.set_style("darkgrid")
        sns.barplot(x="Day",y="Amount",hue="Day",data=df2,estimator=sum)
        plt.xlabel("Day")
        plt.ylabel("Revenue")
        plt.title("Revenue by Day(April)")
        plt.figure(figsize=(20,10))
```

```
Out[ ]: Text(0.5, 1.0, 'Revenue by Day(April)')
```

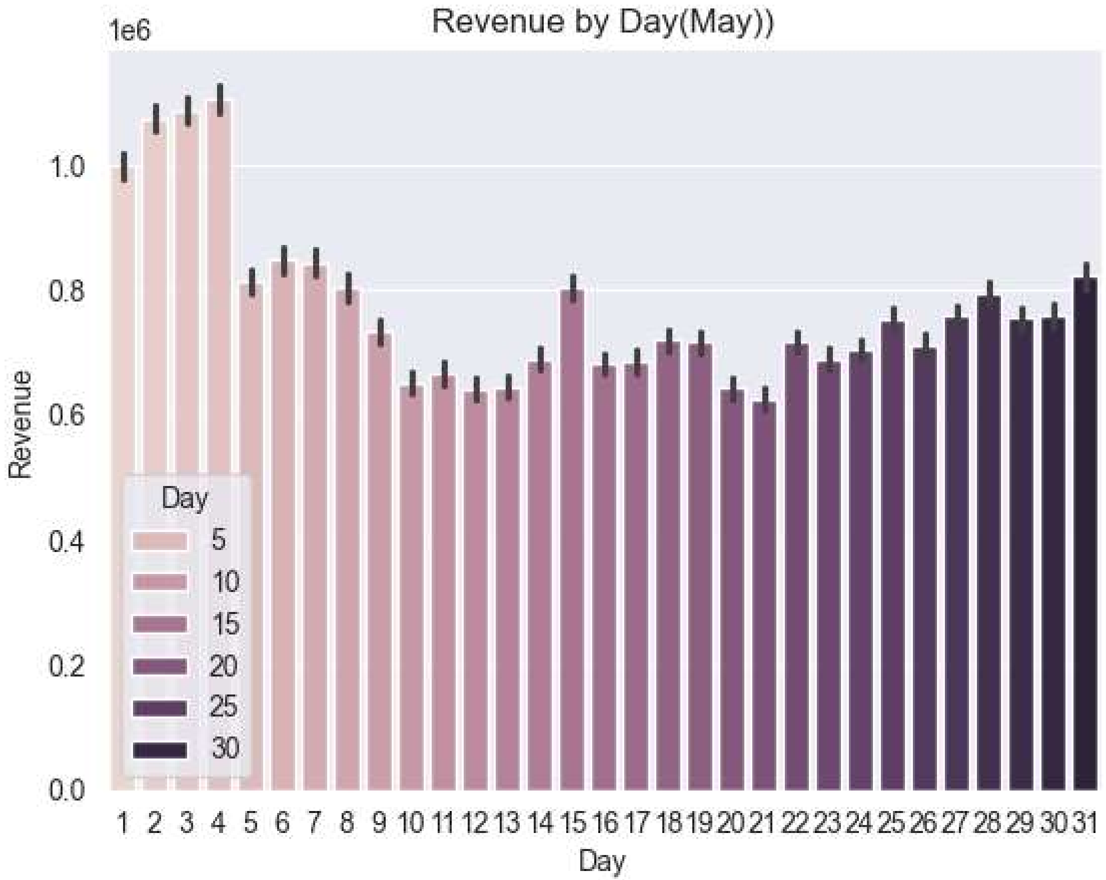


In [ ]:

```
df5=df[df["month"]==5]
sns.set_style("darkgrid")
sns.barplot(x="Day",y="Amount",hue="Day",data=df5,estimator=sum)
plt.xlabel("Day")
plt.ylabel("Revenue")
plt.title("Revenue by Day(May))")
plt.figure(figsize=(20,10))
```

Out[ ]:

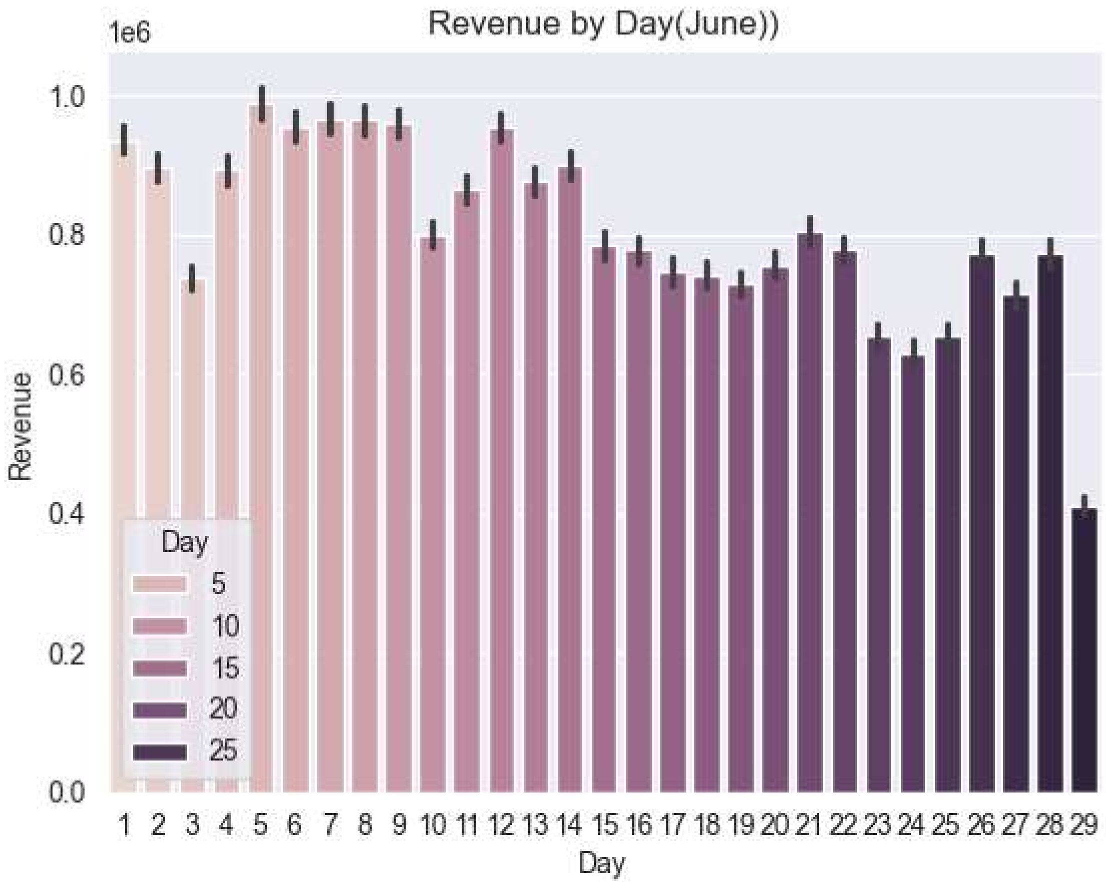
```
<Figure size 2000x1000 with 0 Axes>
```



```
<Figure size 2000x1000 with 0 Axes>
```

```
In [ ]: df6=df[df["month"]==6]
        sns.set_style("darkgrid")
        sns.barplot(x="Day",y="Amount",hue="Day",data=df6,estimator=sum)
        plt.xlabel("Day")
        plt.ylabel("Revenue")
        plt.title("Revenue by Day(June))")
        plt.figure(figsize=(20,10))
```

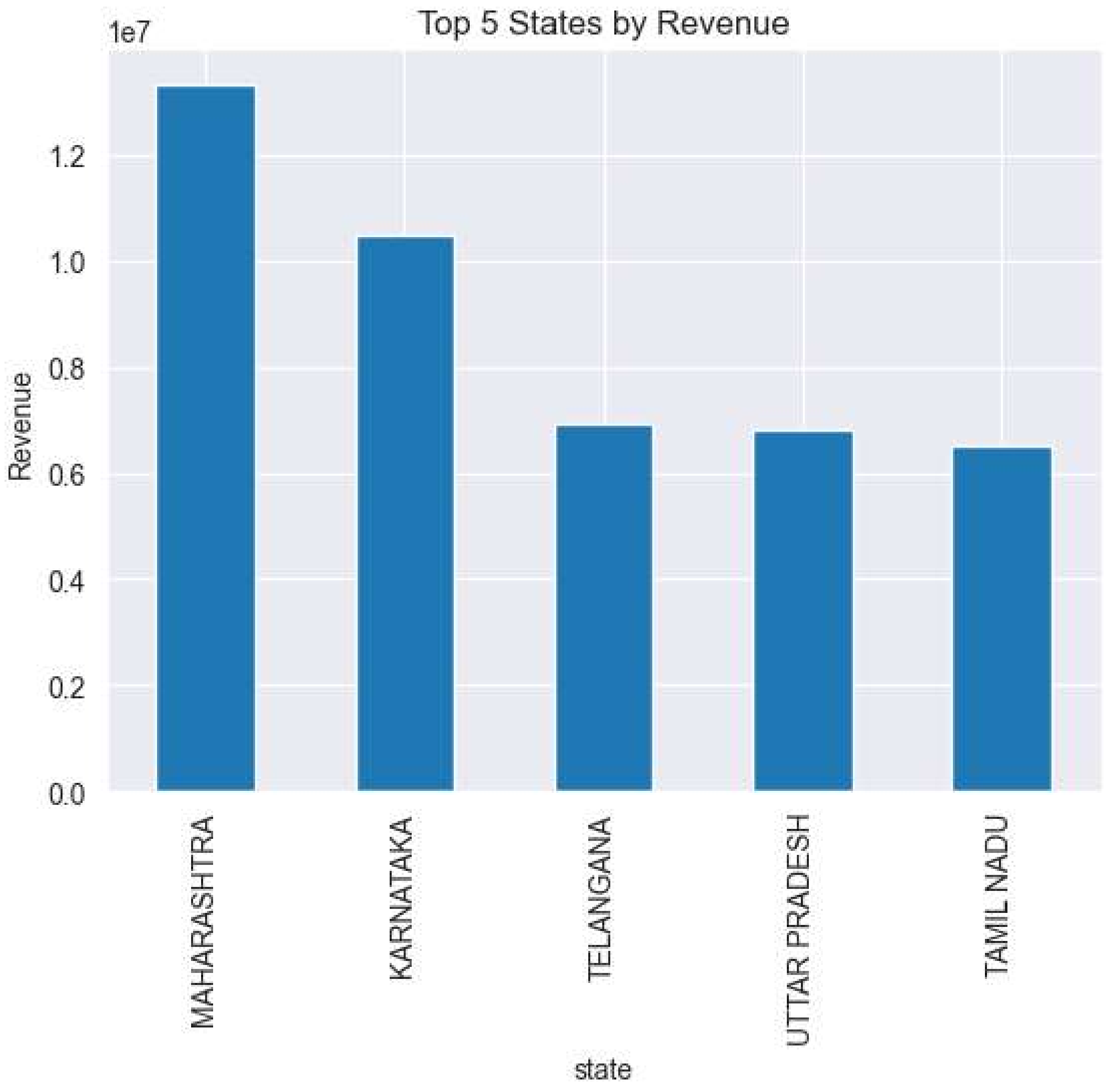
```
Out[ ]: <Figure size 2000x1000 with 0 Axes>
```



```
<Figure size 2000x1000 with 0 Axes>
```

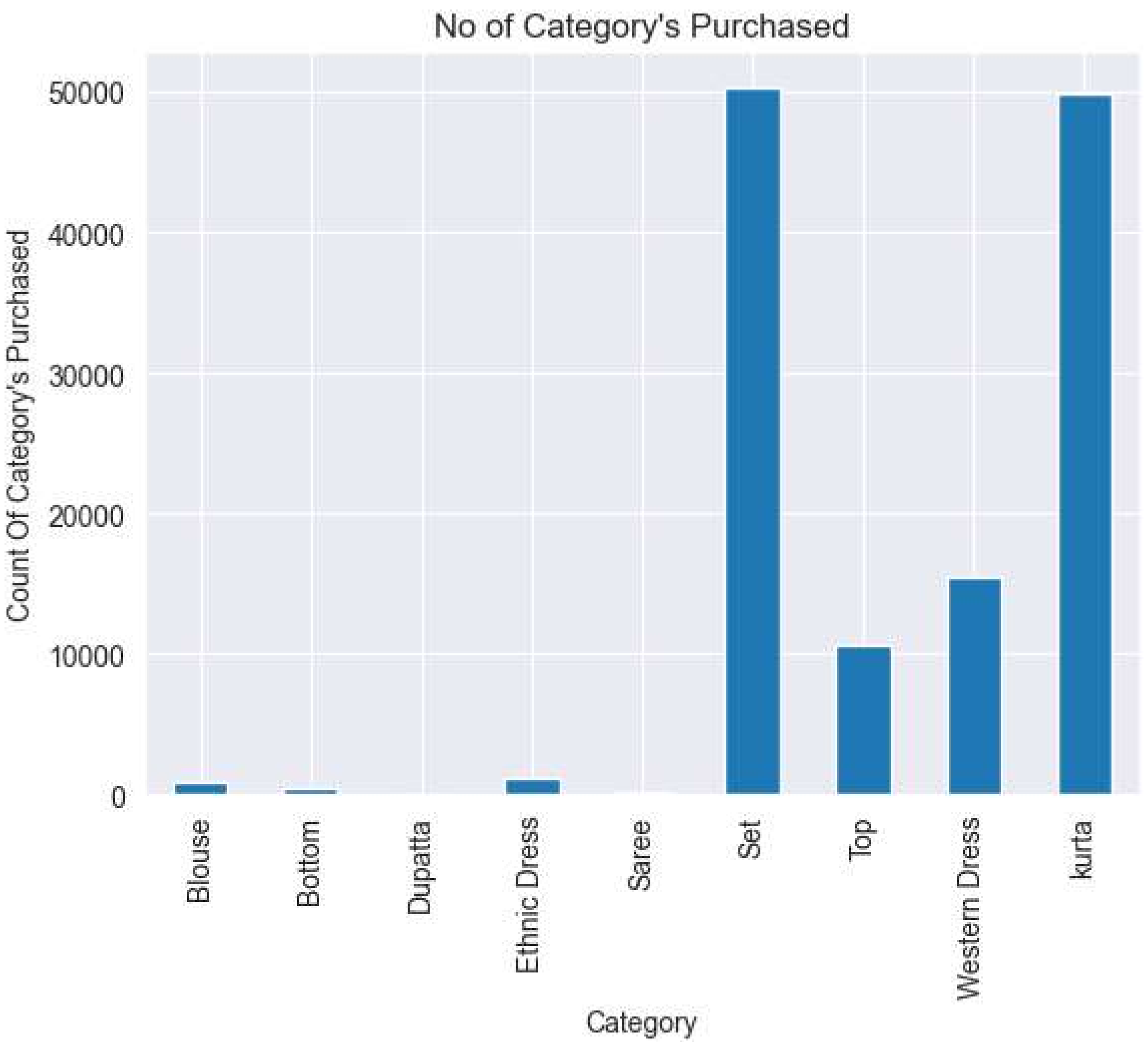
```
In [ ]: df_state=df.groupby("ship-state")["Amount"].sum().nlargest(5)
        df_state.plot(kind="bar")
        plt.title("Top 5 States by Revenue")
        plt.xlabel("state")
        plt.ylabel("Revenue")
```

```
Out[ ]: Text(0, 0.5, 'Revenue')
```



```
In [ ]: df_Category_bar=df.groupby("Category")["Category"].count()
        df_Category_bar.plot(kind="bar")
        plt.title("No of Category's Purchased")
        plt.ylabel("Count Of Category's Purchased")
```

```
Out[ ]: Text(0, 0.5, "Count Of Category's Purchased")
```

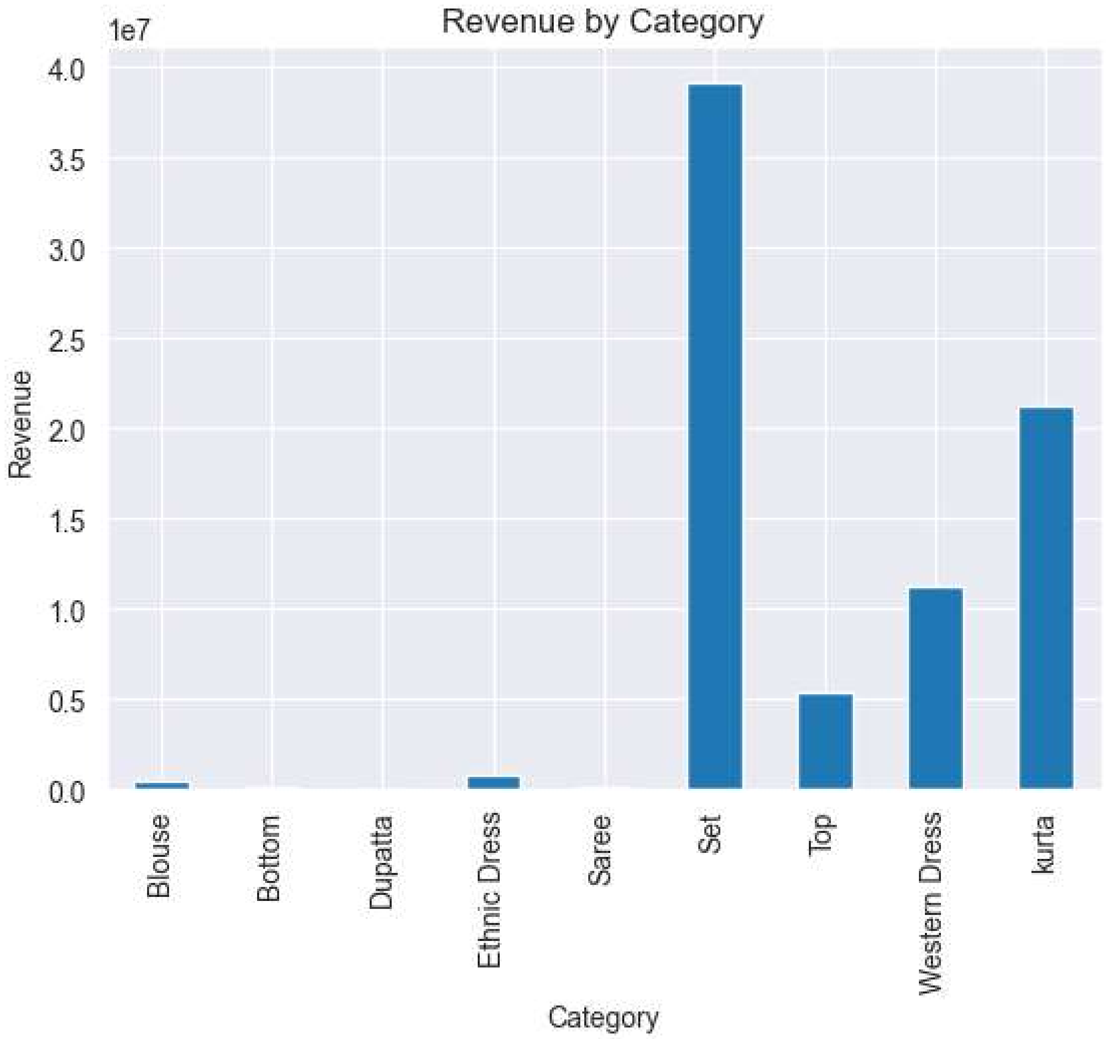


In [ ]:

```
sns.set_style("darkgrid")
df_category=df.groupby("Category")["Amount"].sum()
df_category.plot(kind="bar")
plt.title("Revenue by Category")
plt.xlabel("Category")
plt.ylabel("Revenue")
```

Out[ ]:

```
Text(0, 0.5, 'Revenue')
```

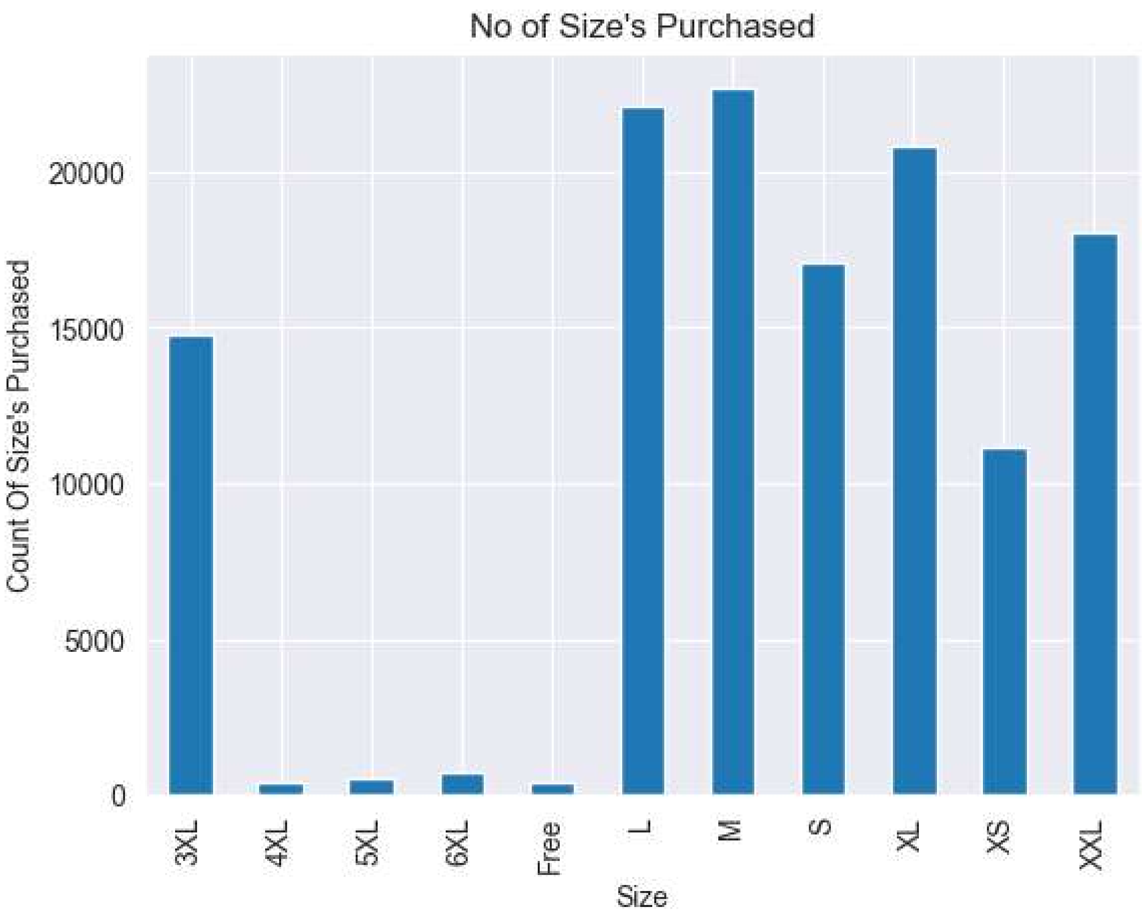


In [ ]:

```
df_size_bar=df.groupby("Size")["Size"].count()
df_size_bar.plot(kind="bar")
plt.title("No of Size's Purchased")
plt.ylabel("Count Of Size's Purchased")
```

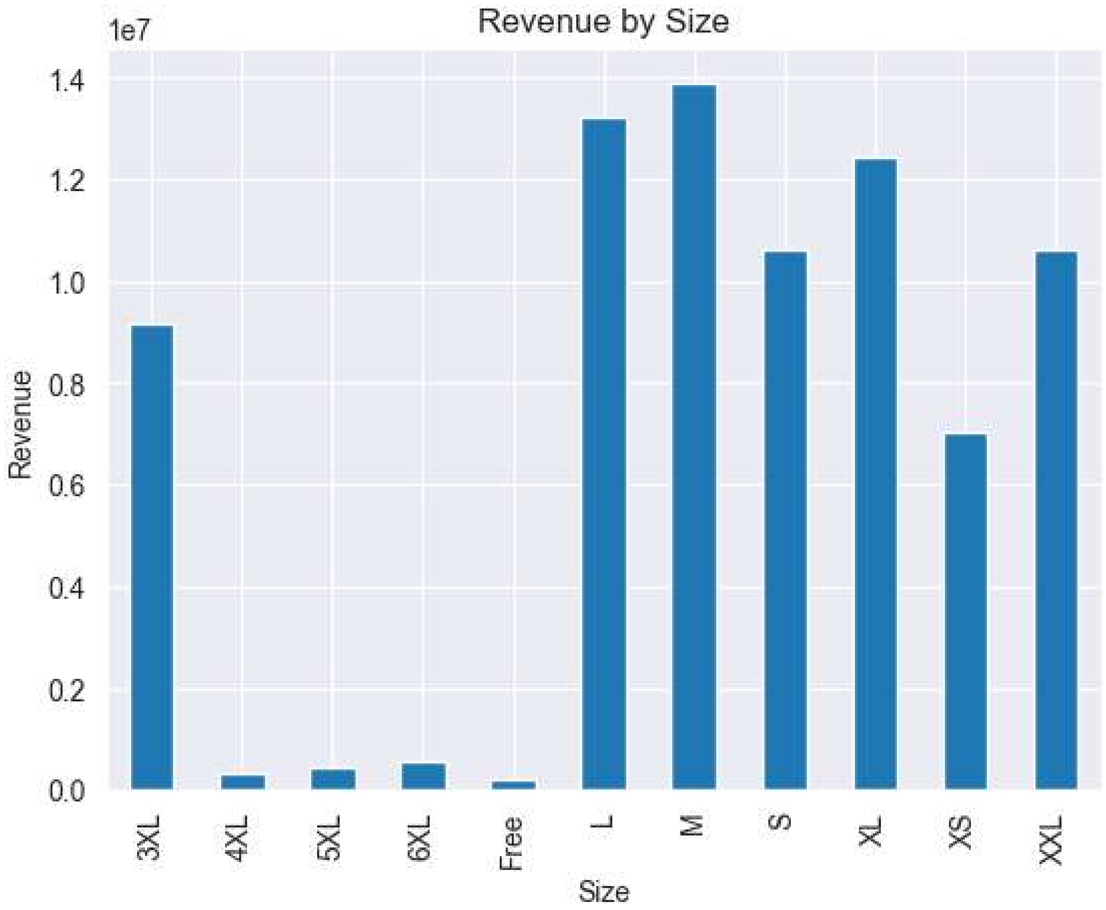
Out[ ]:

```
Text(0, 0.5, "Count Of Size's Purchased")
```



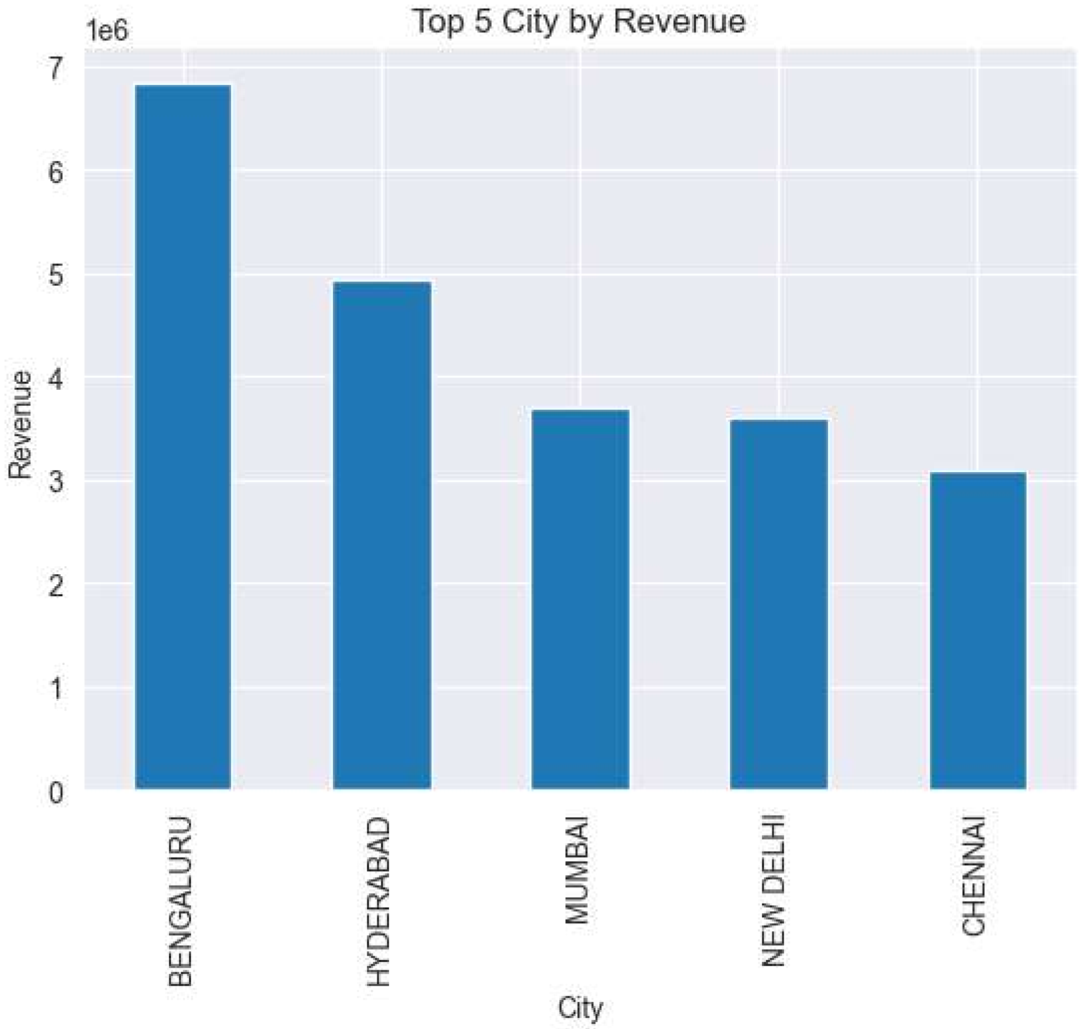
```
In [ ]: sns.set_style("darkgrid")
        df_size=df.groupby("Size")["Amount"].sum()
        df_size.plot(kind="bar")
        plt.title("Revenue by Size")
        plt.ylabel("Revenue")
```

```
Out[ ]: Text(0, 0.5, 'Revenue')
```



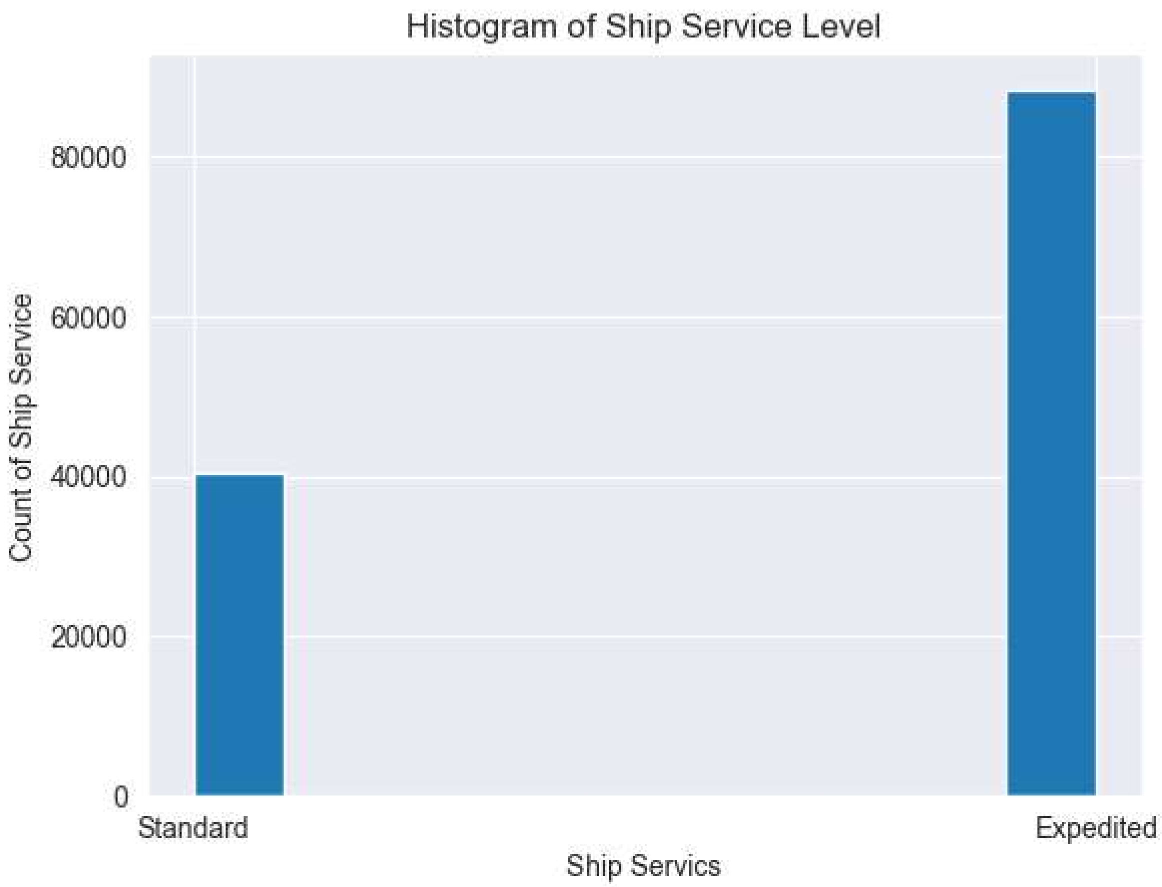
```
In [ ]: df_city=df.groupby("ship-city")["Amount"].sum().nlargest(5)
        df_city.plot(kind="bar")
        plt.title("Top 5 City by Revenue")
        plt.xlabel("City")
        plt.ylabel("Revenue")

Out[ ]:  Text(0, 0.5, 'Revenue')
```



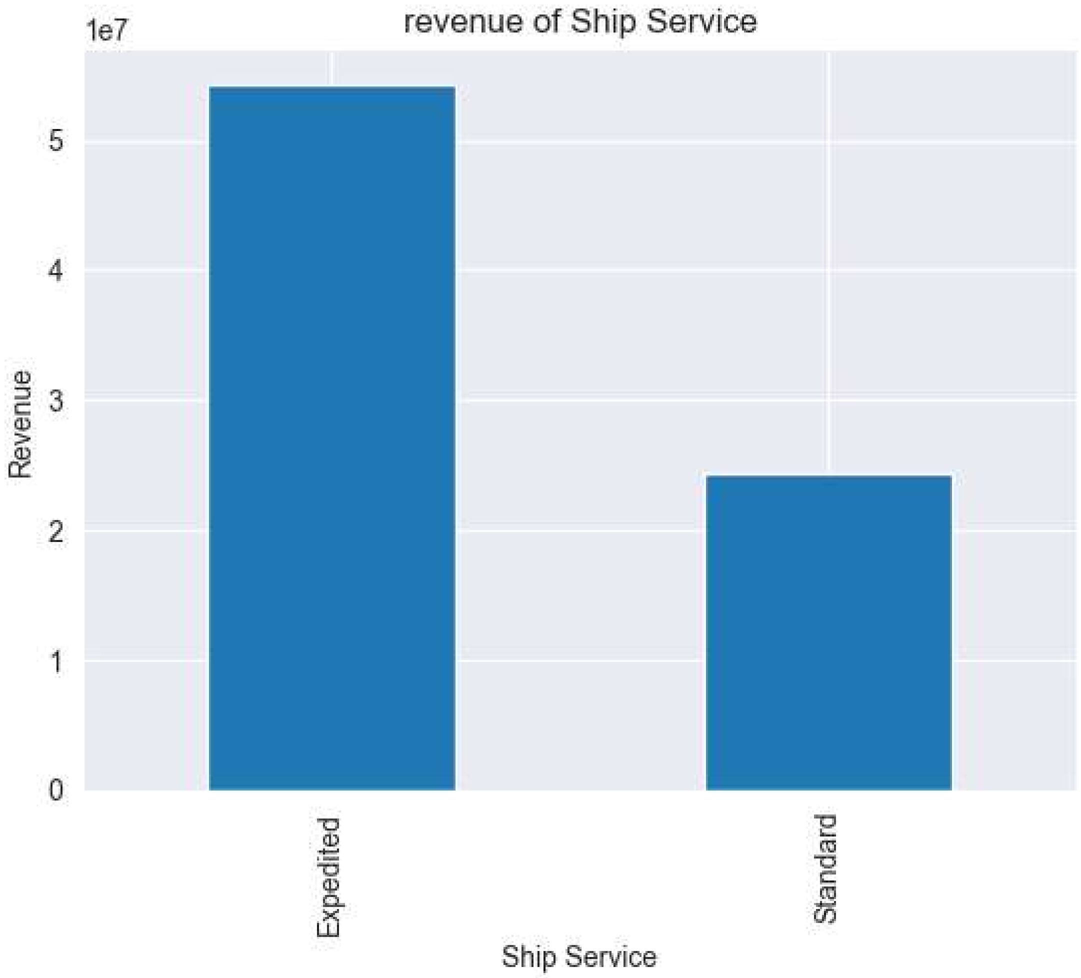
```
In [ ]: plt.hist("ship-service-level",data=df)
        plt.xlabel("Ship Servics")
        plt.ylabel("Count of Ship Service")
        plt.title("Histogram of Ship Service Level")
```

```
Out[ ]: Text(0.5, 1.0, 'Histogram of Ship Service Level')
```



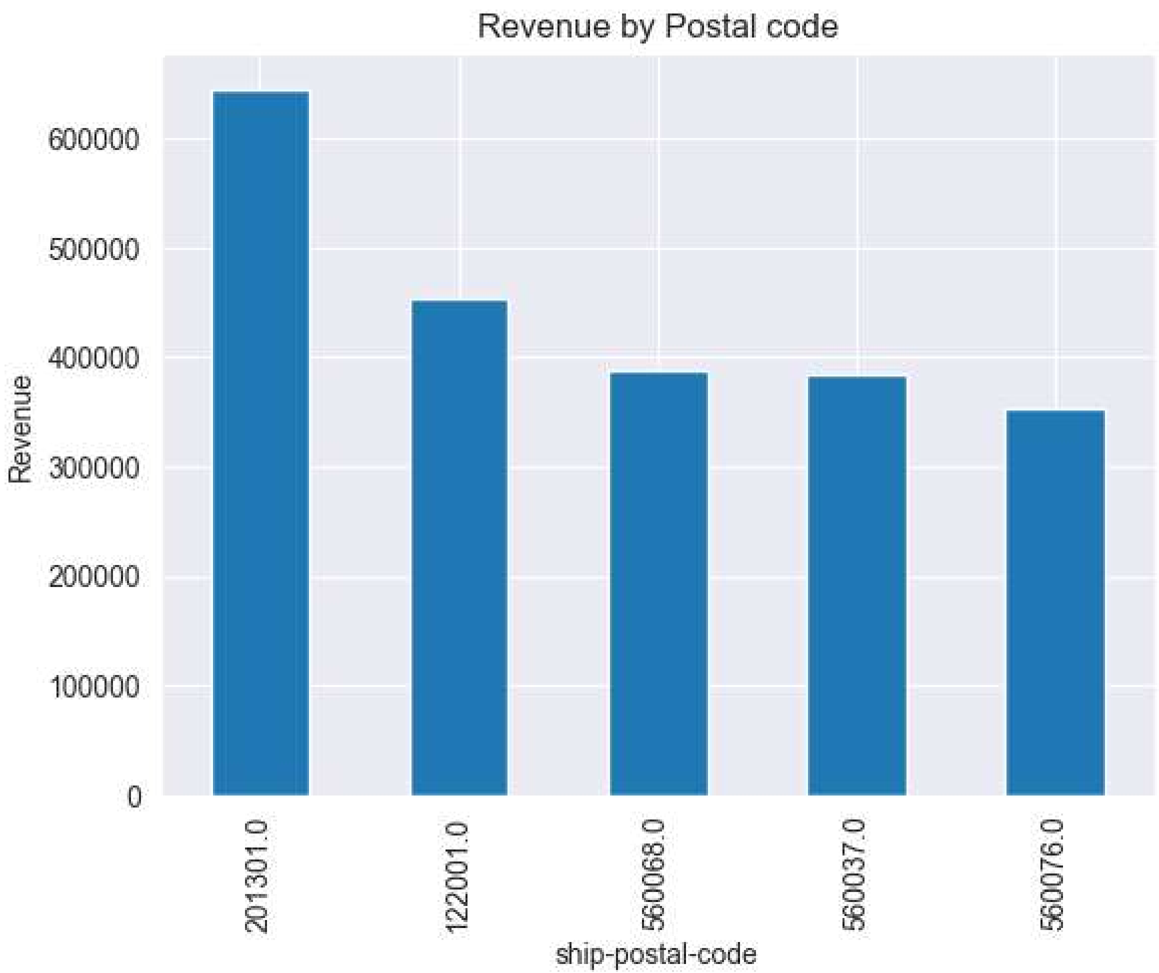
```
In [ ]: df_service=df.groupby("ship-service-level")["Amount"].sum()
        df_service.plot(kind="bar")
        plt.title("revenue of Ship Service")
        plt.xlabel("Ship Service")
        plt.ylabel("Revenue")
```

```
Out[ ]: Text(0, 0.5, 'Revenue')
```



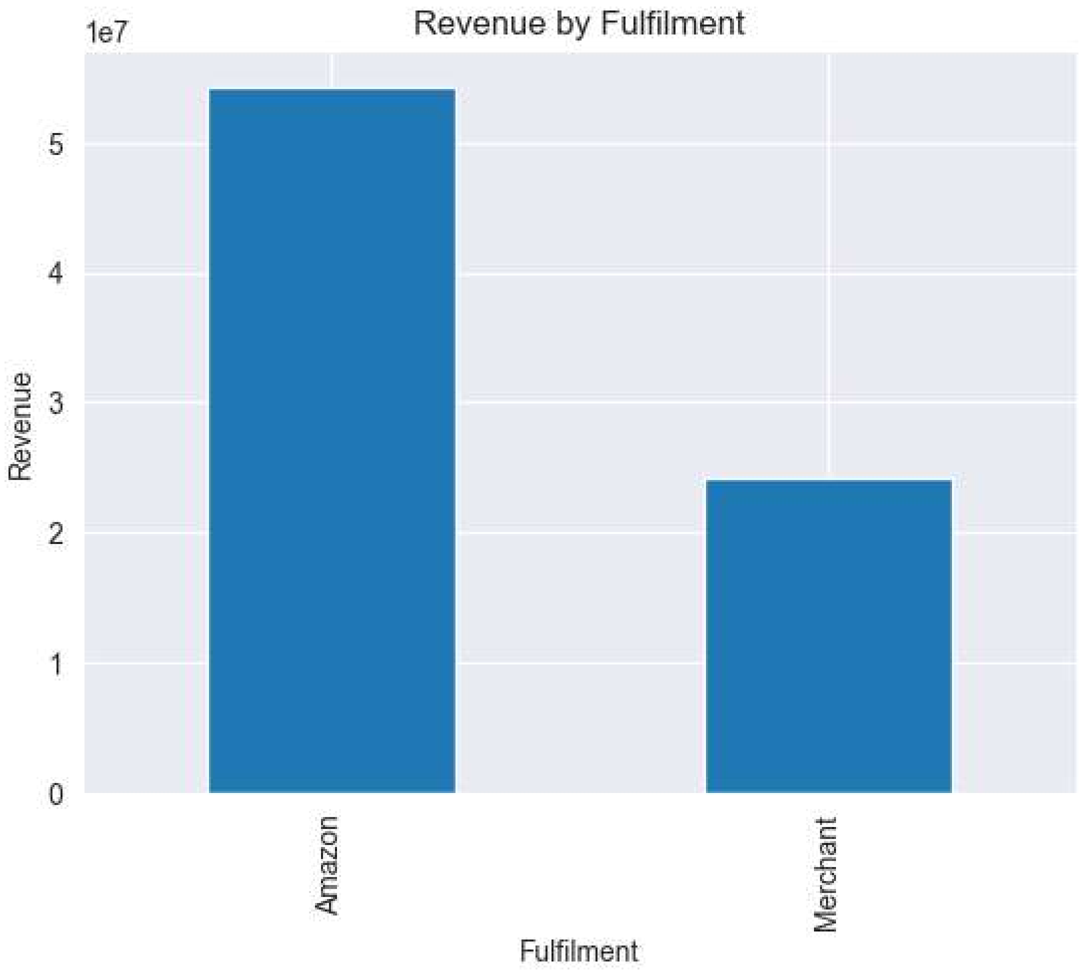
```
In [ ]: sns.set_style("darkgrid")
        df_postal=df.groupby("ship-postal-code")["Amount"].sum().nlargest(5)
        df_postal.plot(kind="bar")
        plt.title("Revenue by Postal code")
        plt.ylabel("Revenue")
```

```
Out[ ]: Text(0, 0.5, 'Revenue')
```



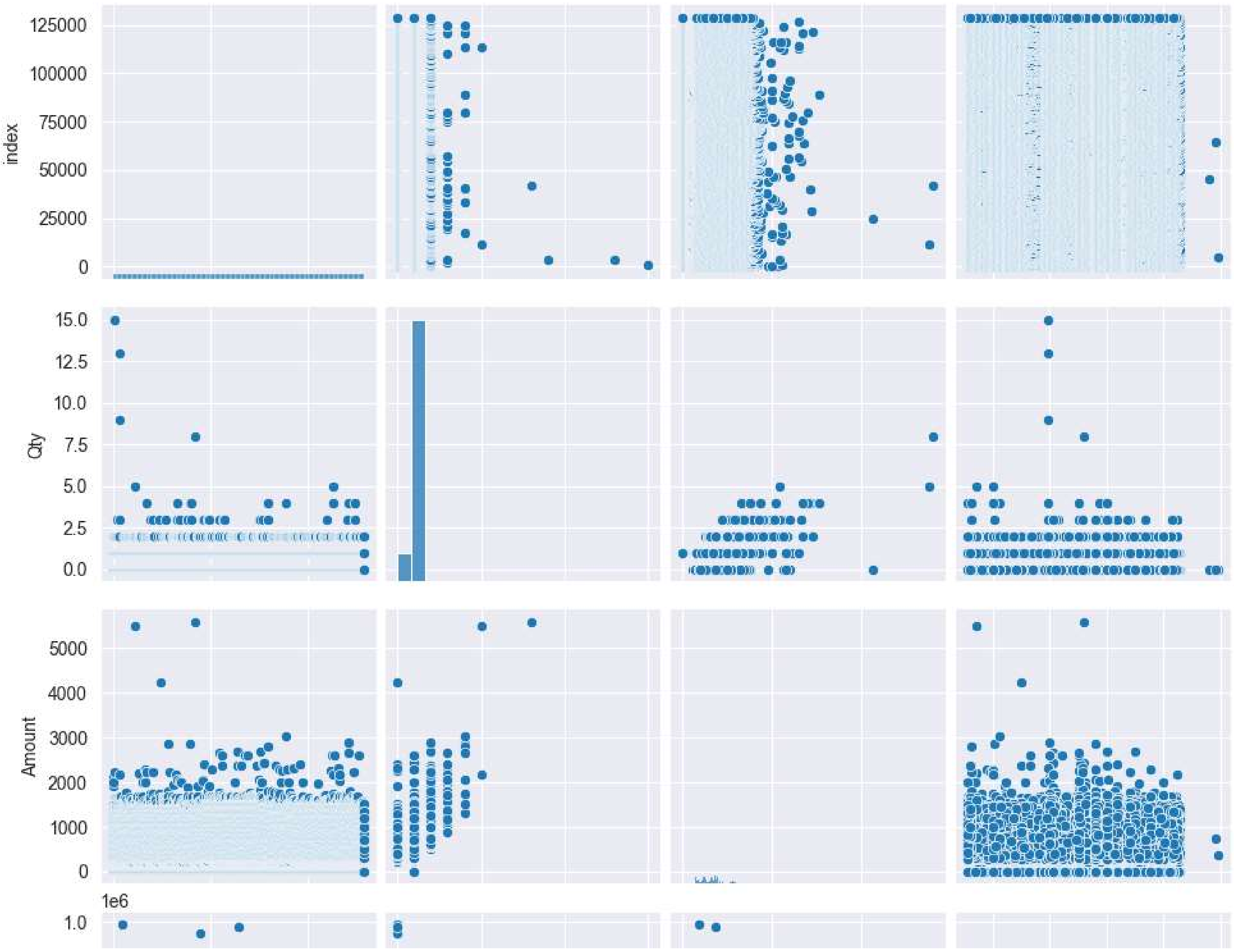
```
In [ ]: sns.set_style("darkgrid")
        df_fulfillment=df.groupby("Fulfilment")["Amount"].sum()
        df_fulfillment.plot(kind="bar")
        plt.title("Revenue by Fulfilment")
        plt.ylabel("Revenue")
```

```
Out[ ]: Text(0, 0.5, 'Revenue')
```

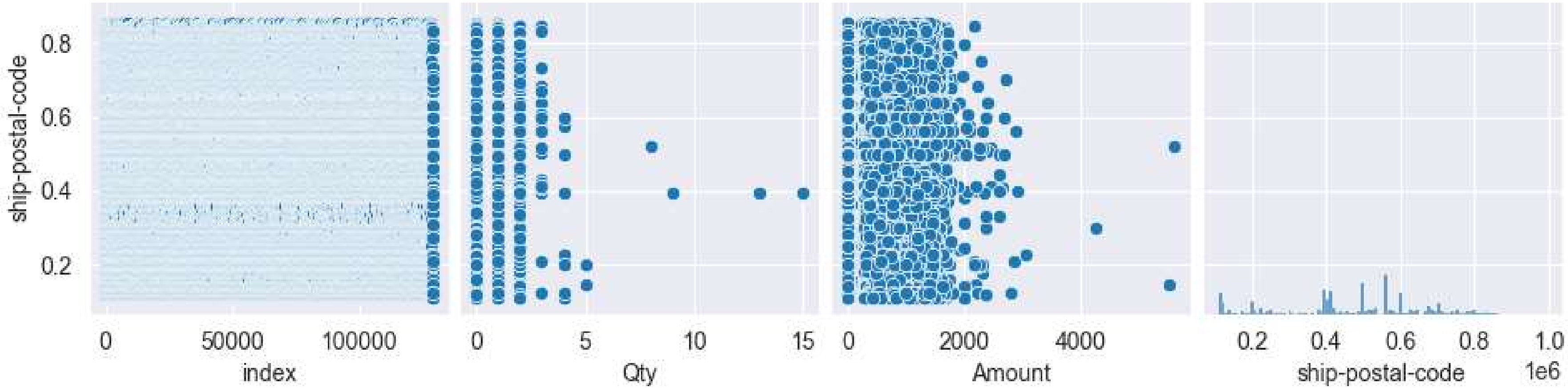


```
In [ ]: sns.pairplot(df)
```

```
Out[ ]: <seaborn.axisgrid.PairGrid at 0x2894fe52c60>
```

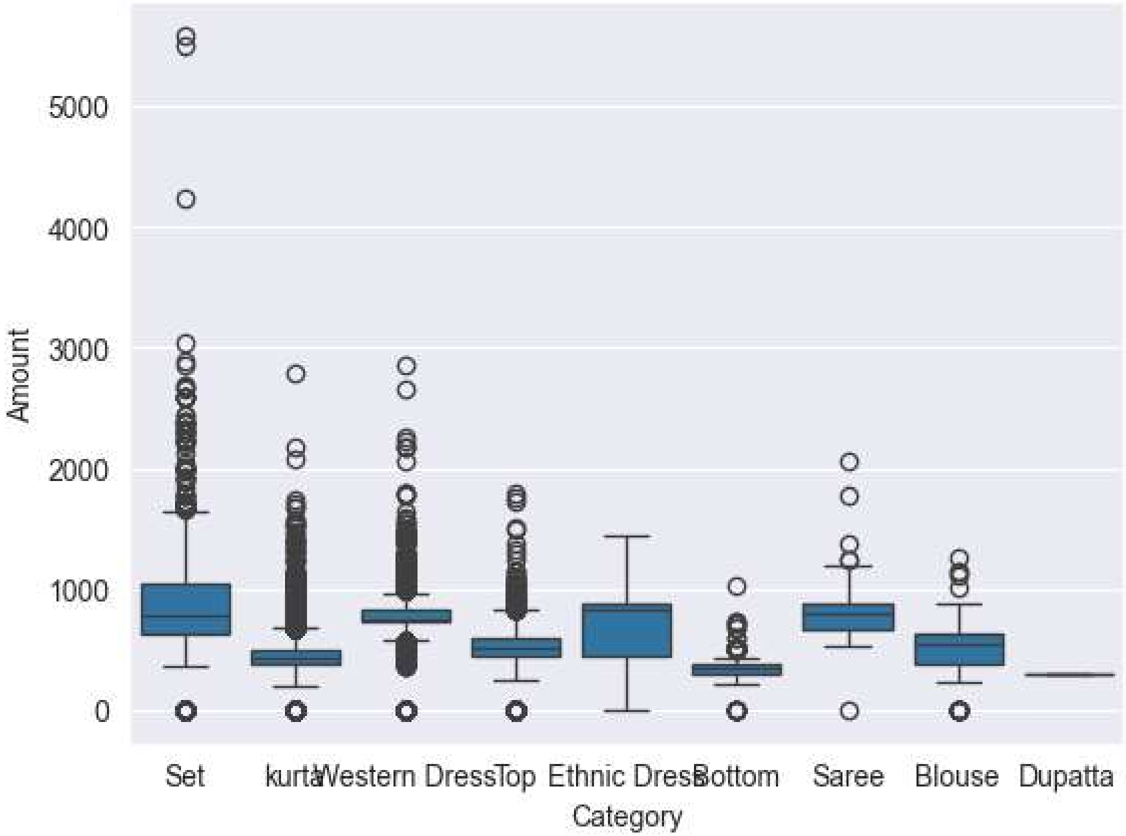


index
125000
100000
75000
50000
25000
0
Qty
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Amount
5000
4000
3000
2000
1000
0
1e6
1.0



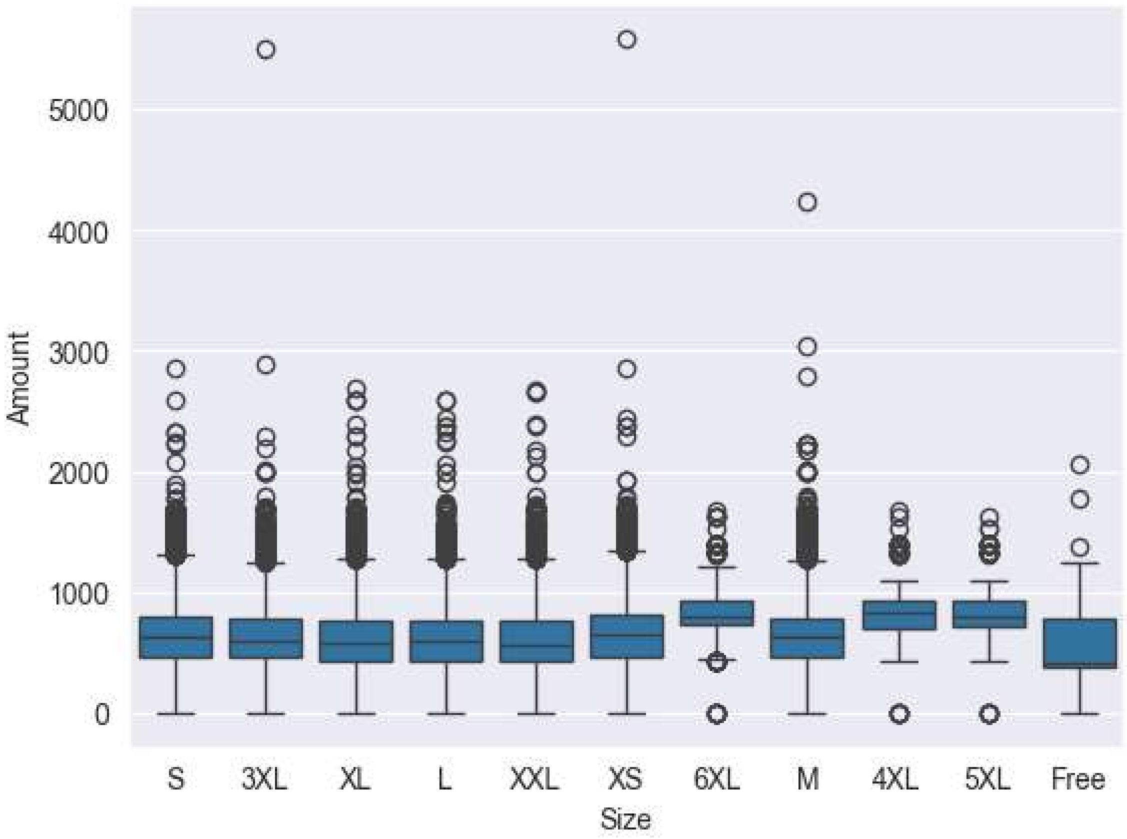
```
In [ ]: sns.boxplot(x="Category",y="Amount",data=df)
```

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



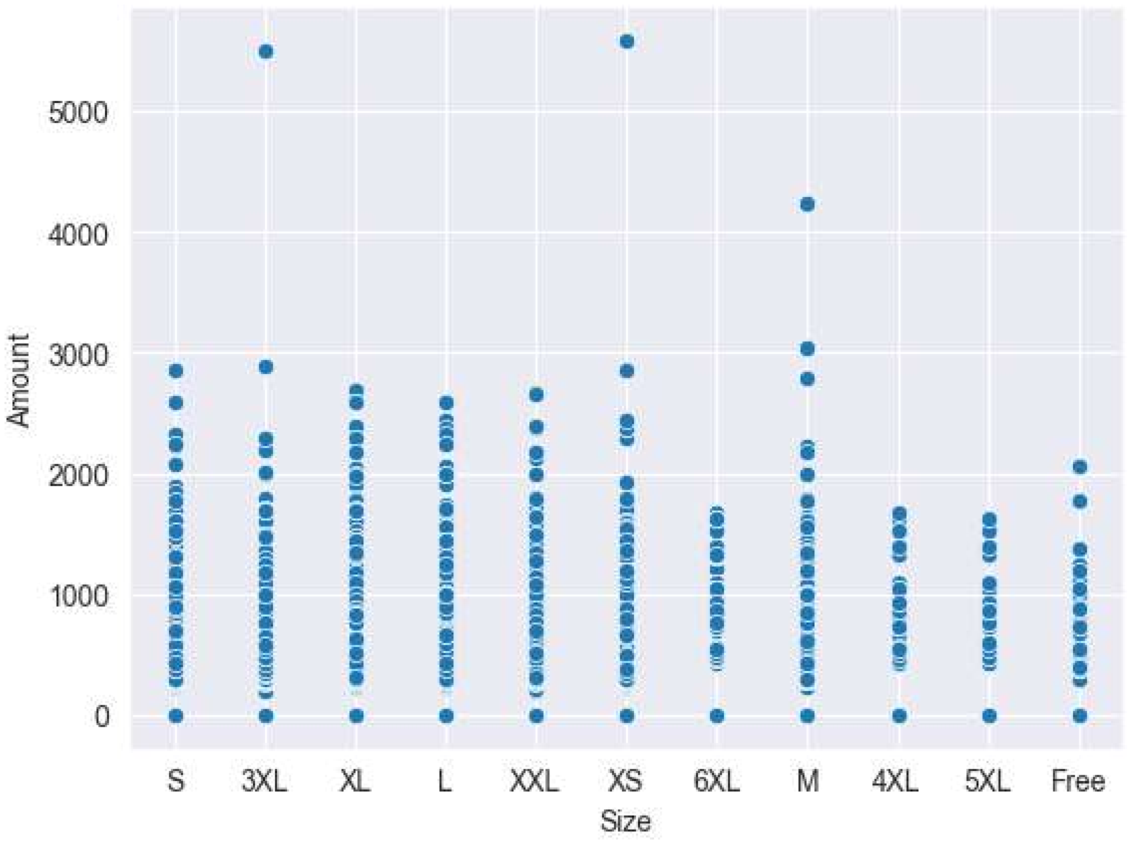
```
In [ ]: sns.boxplot(x="Size",y="Amount",data=df)
```

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



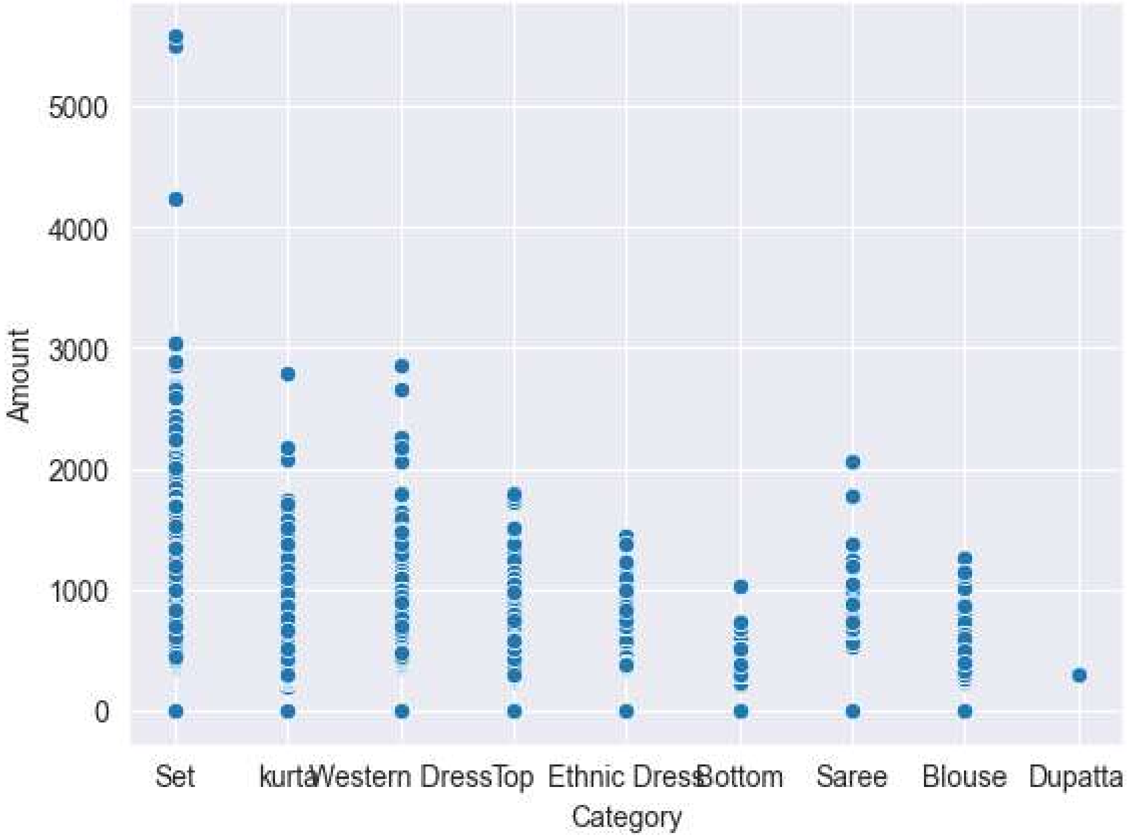
```
In [ ]: sns.scatterplot(x="Size",y="Amount",data=df)
```

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



```
In [ ]: sns.scatterplot(x="Category",y="Amount",data=df)

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


Amount
0
1000
2000
3000
4000
5000
Set
kurta
Western Dress
Top
Ethnic Dress
Bottom
Saree
Blouse
Dupatta
Category

```
In [ ]:
```