Surabhi Singh

The Sparks Foundation

Exploratory Data Analysis

To find out the weak areas and other problems of business to make more profit

Python libraries being used

A. Data Analysis

In [1]:

```
import pandas as pd
import numpy as np
```

B. Data Visualization

In [2]:

```
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

Data Extraction

Importing Data from Superstore Data

In [4]:

```
import types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your c
redentials.
# You might want to remove those credentials before you share the notebook.
client_df59ba2aae8f4254afb2ed76c5028249 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='C0K_2AXL10Mc0wYapj_sm2un8kk7QWfw3bhgF6k6e3mK',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3-api.us-geo.objectstorage.service.networklayer.com')

body = client_df59ba2aae8f4254afb2ed76c5028249.get_object(Bucket='tsf-donotdelete-pr-spz8
z9iaocdhoc',Key='SampleSuperstore.csv')['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

df = pd.read_csv(body)
df.head(10)
```

Out[4]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discou
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	2	0.(
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731.9400	3	0.(
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	14.6200	2	0.(
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.5775	5	0.4
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.3680	2	0.2
5	Standard Class	Consumer	United States	Los Angeles	California	90032	West	Furniture	Furnishings	48.8600	7	0.(
6	Standard Class	Consumer	United States	Los Angeles	California	90032	West	Office Supplies	Art	7.2800	4	0.(
7	Standard Class	Consumer	United States	Los Angeles	California	90032	West	Technology	Phones	907.1520	6	0.2
8	Standard Class	Consumer	United States	Los Angeles	California	90032	West	Office Supplies	Binders	18.5040	3	0.2
9	Standard Class	Consumer	United States	Los Angeles	California	90032	West	Office Supplies	Appliances	114.9000	5	0.(

In [5]:

```
df.tail(10)
```

Out[5]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Dis
9984	Standard Class	Consumer	United States	Long Beach	New York	11561	East	Office Supplies	Labels	31.500	10	
9985	Standard Class	Consumer	United States	Long Beach	New York	11561	East	Office Supplies	Supplies	55.600	4	
9986	Standard Class	Consumer	United States	Los Angeles	California	90008	West	Technology	Accessories	36.240	1	
9987	Standard Class	Corporate	United States	Athens	Georgia	30605	South	Technology	Accessories	79.990	1	
9988	Standard Class	Corporate	United States	Athens	Georgia	30605	South	Technology	Phones	206.100	5	
9989	Second Class	Consumer	United States	Miami	Florida	33180	South	Furniture	Furnishings	25.248	3	
9990	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Furniture	Furnishings	91.960	2	
9991	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Technology	Phones	258.576	2	
9992	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Office Supplies	Paper	29.600	4	
9993	Second Class	Consumer	United States	Westminster	California	92683	West	Office Supplies	Appliances	243.160	2	

In [6]:

```
df.shape
```

Out[6]:

```
(9994, 13)
```

here, 13 columns and 9994 rows in this Data

In [7]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Ship Mode      9994 non-null   object 
 1   Segment        9994 non-null   object 
 2   Country        9994 non-null   object 
 3   City           9994 non-null   object 
 4   State          9994 non-null   object 
 5   Postal Code    9994 non-null   int64  
 6   Region         9994 non-null   object 
 7   Category       9994 non-null   object 
 8   Sub-Category   9994 non-null   object 
 9   Sales          9994 non-null   float64
 10  Quantity       9994 non-null   int64  
 11  Discount       9994 non-null   float64
 12  Profit         9994 non-null   float64
dtypes: float64(3), int64(2), object(8)
memory usage: 1015.1+ KB
```

In [8]:

```
duplicate=df.duplicated()
print(duplicate.sum())
df[duplicate]
```

```
17
```

Out[8]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Dis
950	Standard Class	Home Office	United States	Philadelphia	Pennsylvania	19120	East	Office Supplies	Paper	15.552	3	
3406	Standard Class	Home Office	United States	Columbus	Ohio	43229	East	Furniture	Chairs	281.372	2	
3670	Standard Class	Consumer	United States	Salem	Oregon	97301	West	Office Supplies	Paper	10.368	2	
4117	Standard Class	Consumer	United States	Los Angeles	California	90036	West	Office Supplies	Paper	19.440	3	
4553	Standard Class	Consumer	United States	San Francisco	California	94122	West	Office Supplies	Paper	12.840	3	
5905	Same Day	Home Office	United States	San Francisco	California	94122	West	Office Supplies	Labels	41.400	4	
6146	Standard Class	Corporate	United States	San Francisco	California	94122	West	Office Supplies	Art	11.760	4	
6334	Standard Class	Consumer	United States	New York City	New York	10011	East	Office Supplies	Paper	49.120	4	
6357	Standard Class	Corporate	United States	Seattle	Washington	98103	West	Office Supplies	Paper	25.920	4	
7608	Standard Class	Consumer	United States	San Francisco	California	94122	West	Office Supplies	Paper	25.920	4	
7735	Standard Class	Corporate	United States	Seattle	Washington	98105	West	Office Supplies	Paper	19.440	3	

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Dis
7759	Standard Class	Corporate	United States	Houston	Texas	77041	Central	Office Supplies	Paper	15.552	3	
8032	First Class	Consumer	United States	Houston	Texas	77041	Central	Office Supplies	Paper	47.952	3	
8095	Second Class	Consumer	United States	Seattle	Washington	98115	West	Office Supplies	Paper	12.960	2	
9262	Standard Class	Consumer	United States	Detroit	Michigan	48227	Central	Furniture	Chairs	389.970	3	
9363	Standard Class	Home Office	United States	Seattle	Washington	98105	West	Furniture	Furnishings	22.140	3	
9477	Second Class	Corporate	United States	Chicago	Illinois	60653	Central	Office Supplies	Binders	3.564	3	

Removing duplicate data

In [9]:

```
df.drop_duplicates(inplace = True)
```

Confirming if all duplicates are removed

In [10]:

```
dp = df.duplicated()
dp.sum()
```

Out[10]:

```
0
```

Checking null Data

In [11]:

```
df.isnull().sum()
```

Out[11]:

```
Ship Mode       0
Segment         0
Country         0
City            0
State           0
Postal Code     0
Region          0
Category        0
Sub-Category    0
Sales           0
Quantity        0
Discount        0
Profit          0
dtype: int64
```

Checking Unique Values

In [12]:

```
df.nunique()
```

Out[12]:

```
Ship Mode       4
Segment         3
Country         1
```

```
City              531
State              49
Postal Code       631
Region              4
Category            3
Sub-Category       17
Sales            5825
Quantity           14
Discount           12
Profit           7287
dtype: int64
```

Dropping Postal code column from the analysis

In [13]:

```
df1=df.drop(columns='Postal Code', axis=1)
```

Final Summary of the Dataset

In [14]:

```
df1.describe()
```

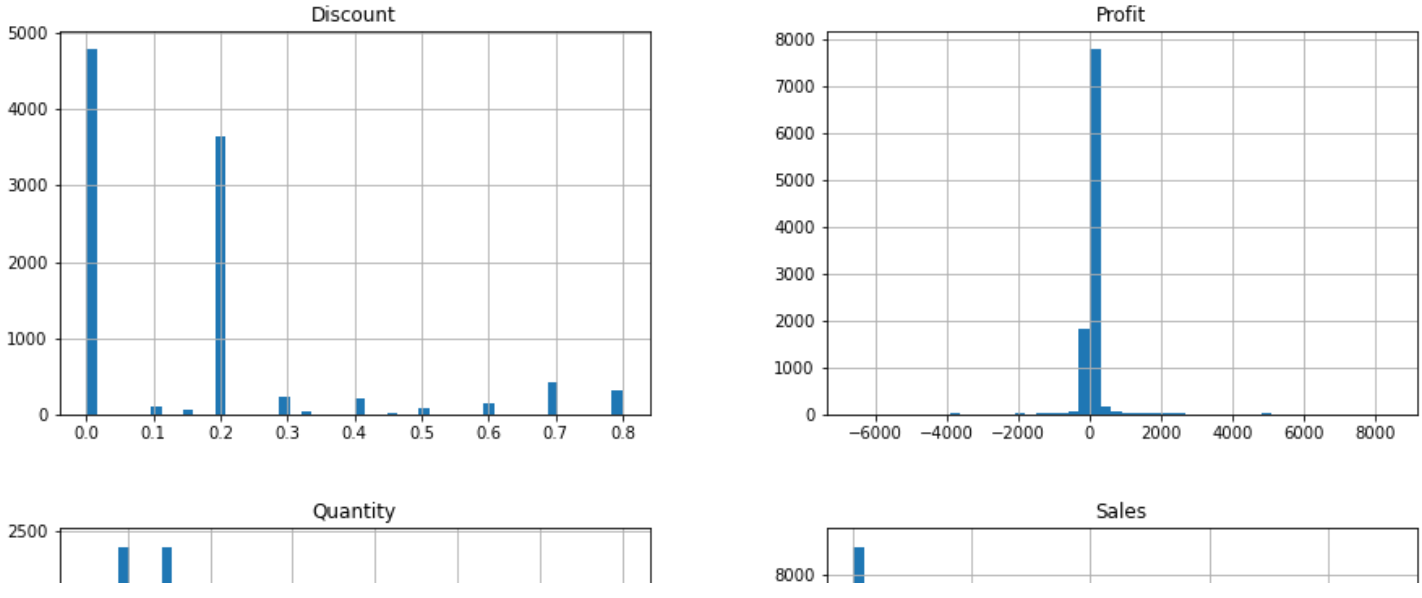
Out[14]:

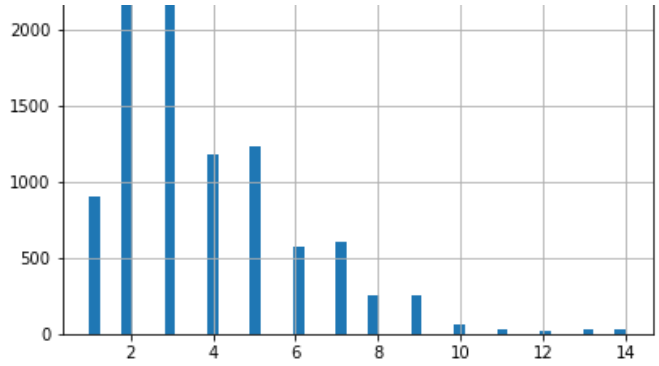
	Sales	Quantity	Discount	Profit
count	9977.000000	9977.000000	9977.000000	9977.00000
mean	230.148902	3.790719	0.156278	28.69013
std	623.721409	2.226657	0.206455	234.45784
min	0.444000	1.000000	0.000000	-6599.97800
25%	17.300000	2.000000	0.000000	1.72620
50%	54.816000	3.000000	0.200000	8.67100
75%	209.970000	5.000000	0.200000	29.37200
max	22638.480000	14.000000	0.800000	8399.97600

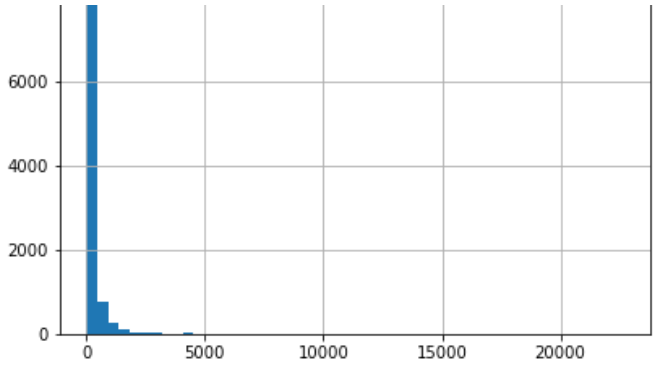
Presenting on Graph

In [15]:

```
df1.hist(figsize=(15, 10), bins=50)
plt.show()
```







Correlation between the sales, Quantities, Discount rate and profit

In [16]:

```
df1.corr()
```

Out[16]:

	Sales	Quantity	Discount	Profit
Sales	1.000000	0.200722	-0.028311	0.479067
Quantity	0.200722	1.000000	0.008678	0.066211
Discount	-0.028311	0.008678	1.000000	-0.219662
Profit	0.479067	0.066211	-0.219662	1.000000

Heat map on above Correlation

In [17]:

```
corr = df1.corr()
sns.heatmap(corr,annot=True)
```

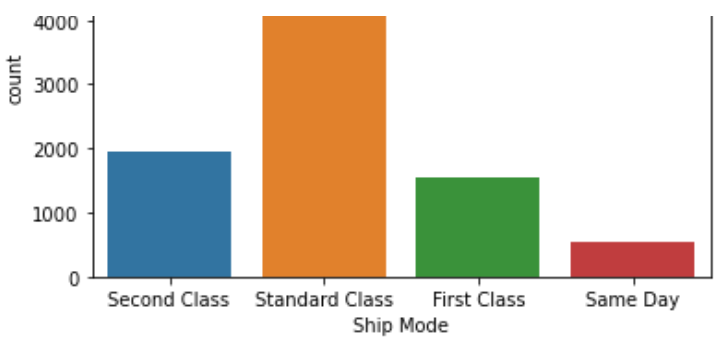
Out[17]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f58d4dd7d90>
```

Plotting the no. of orders for each Ship Mode

In [18]:

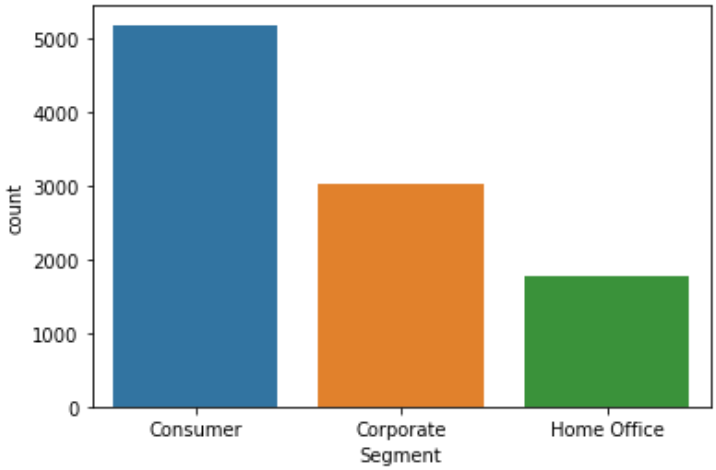
```
sns.countplot(df1['Ship Mode'])
plt.show()
```



Plotting the no. of orders for each Segment

In [19]:

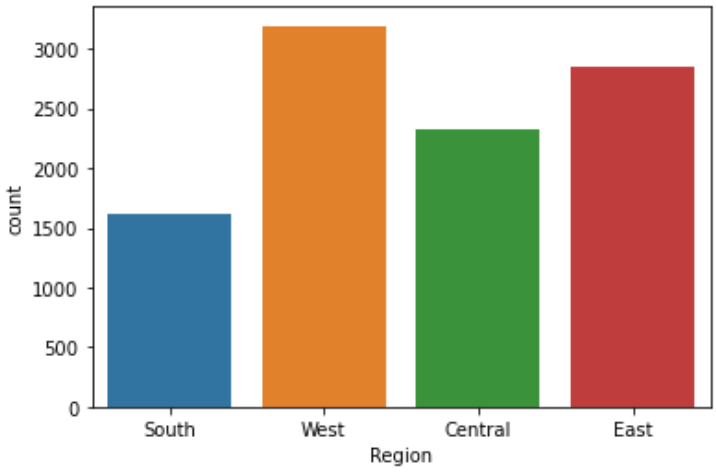
```
sns.countplot(df1['Segment'])
plt.show()
```



Plotting the no. of orderes for each region

In [20]:

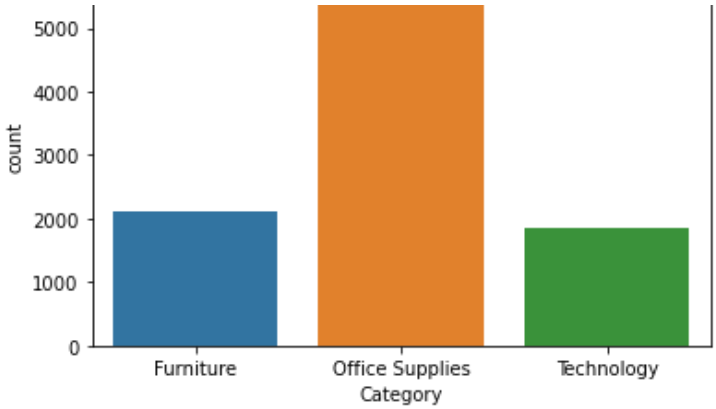
```
sns.countplot(df1['Region'])
plt.show()
```



Plotting the no. of orders for each category

In [21]:

```
sns.countplot(df1['Category'])
plt.show()
```



No. of orders for each Sub-Category

In [22]:

```
grouped = df1.groupby(['Category','Sub-Category'])
grouped.size()
```

Out[22]:

```
Category         Sub-Category
Furniture        Bookcases         228
                 Chairs            615
                 Furnishings       956
                 Tables            319
Office Supplies  Appliances        466
                 Art               795
                 Binders          1522
                 Envelopes         254
                 Fasteners         217
                 Labels            363
                 Paper            1359
                 Storage           846
                 Supplies          190
Technology       Accessories       775
                 Copiers            68
                 Machines          115
                 Phones            889
dtype: int64
```

In [23]:

```
subcategory_table = pd.crosstab(index=df1["Category"],
columns=df1["Sub-Category"])
subcategory_table
```

Out[23]:

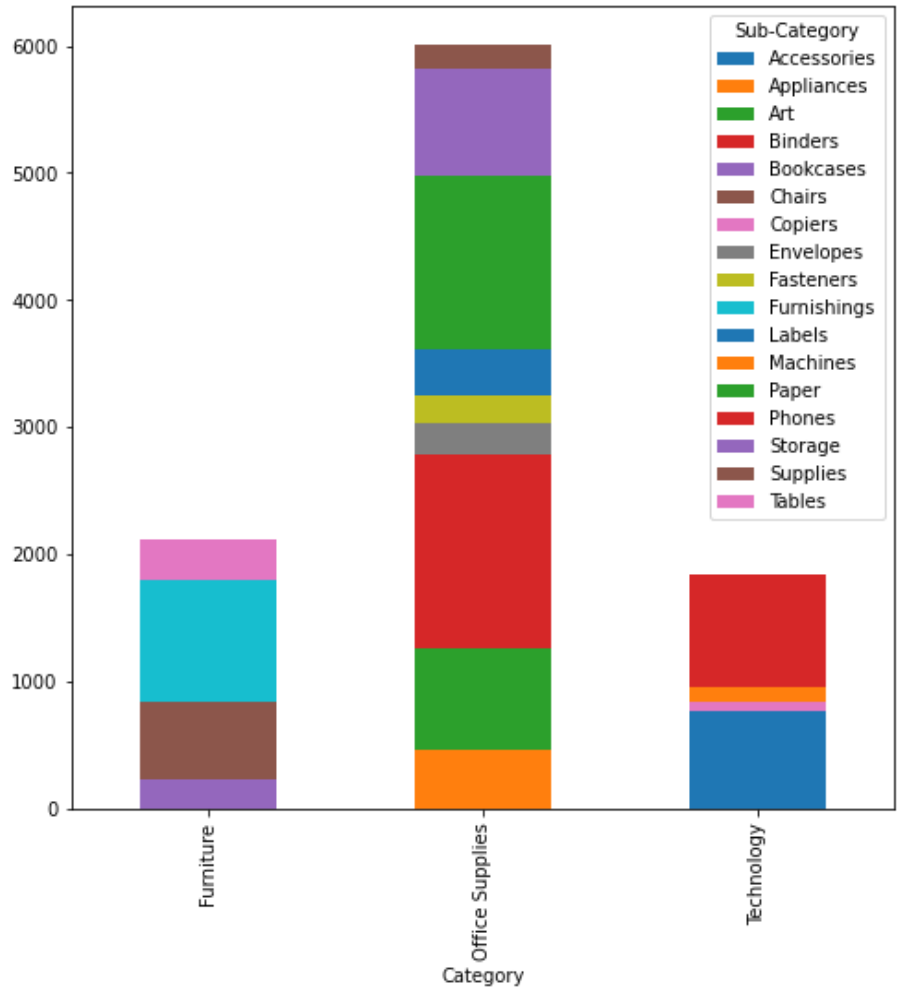
Sub-Category / Category	Accessories	Appliances	Art	Binders	Bookcases	Chairs	Copiers	Envelopes	Fasteners	Furnishings	Labels	I
Furniture	0	0	0	0	228	615	0	0	0	956	0	
Office Supplies	0	466	795	1522	0	0	0	254	217	0	363	
Technology	775	0	0	0	0	0	68	0	0	0	0	

In [24]:

```
subcategory_table.plot(kind="bar",
figsize=(8,8),
stacked=True)
```

Out[24]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f58d4dec3d0>
```



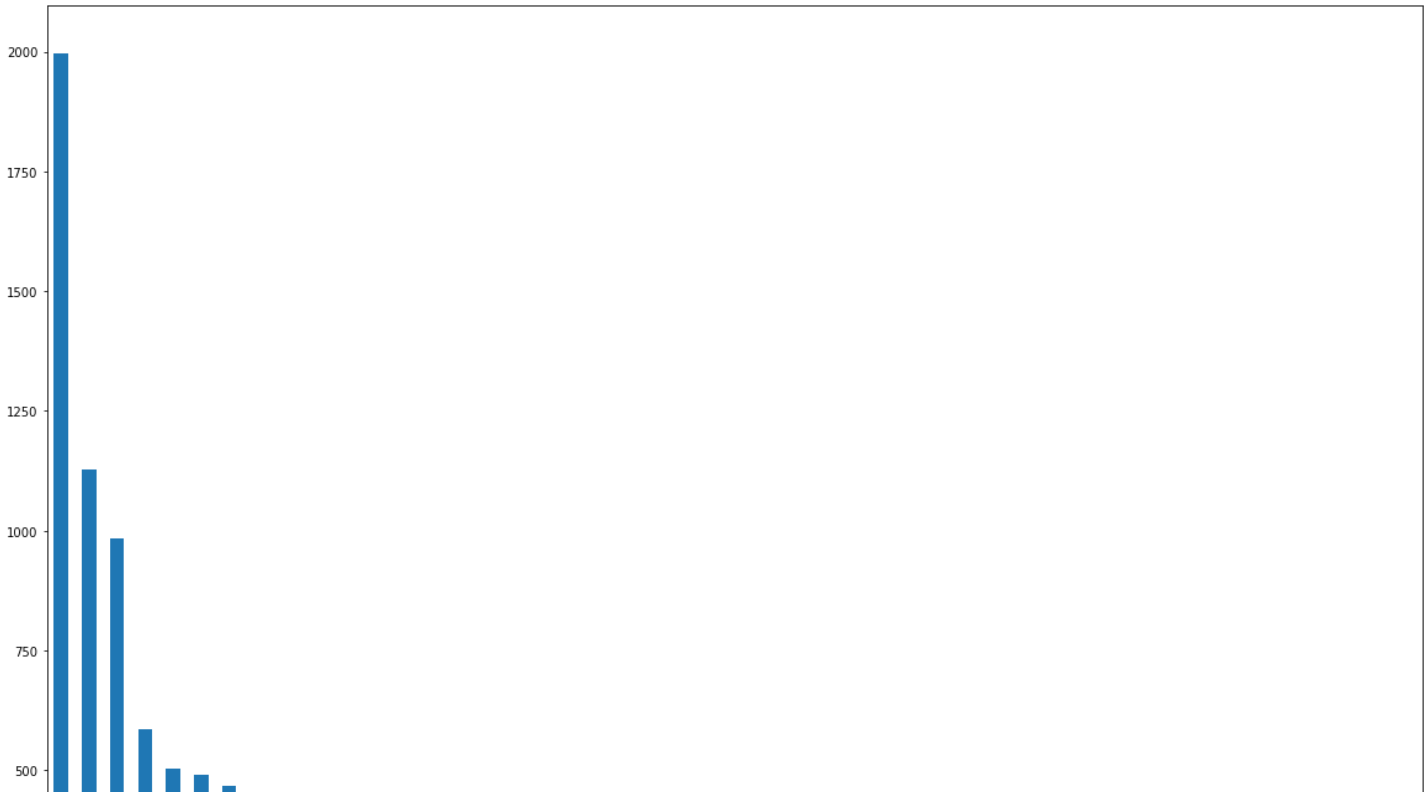
No. of orders for each State

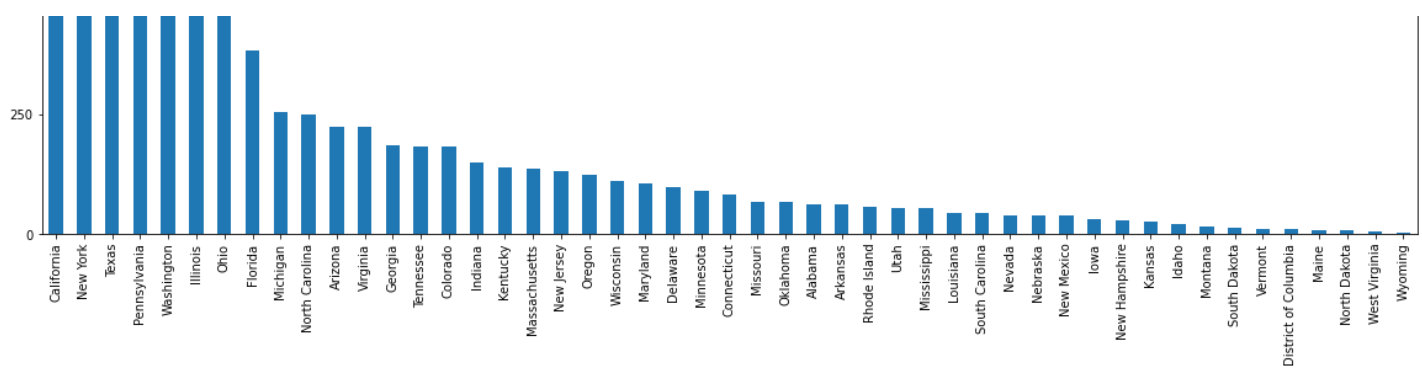
In [25]:

```
df1['State'].value_counts().plot(kind = 'bar', figsize=(20,15))
```

Out[25]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f58d505fb10>
```





No. of orders in each City

In [27]:

```
df1['City'].value_counts()
```

Out[27]:

```
New York City     914
Los Angeles       746
Philadelphia      536
San Francisco     506
Seattle           424
                 ...
Melbourne           1
Vacaville           1
San Mateo           1
Orland Park         1
Goldsboro           1
Name: City, Length: 531, dtype: int64
```

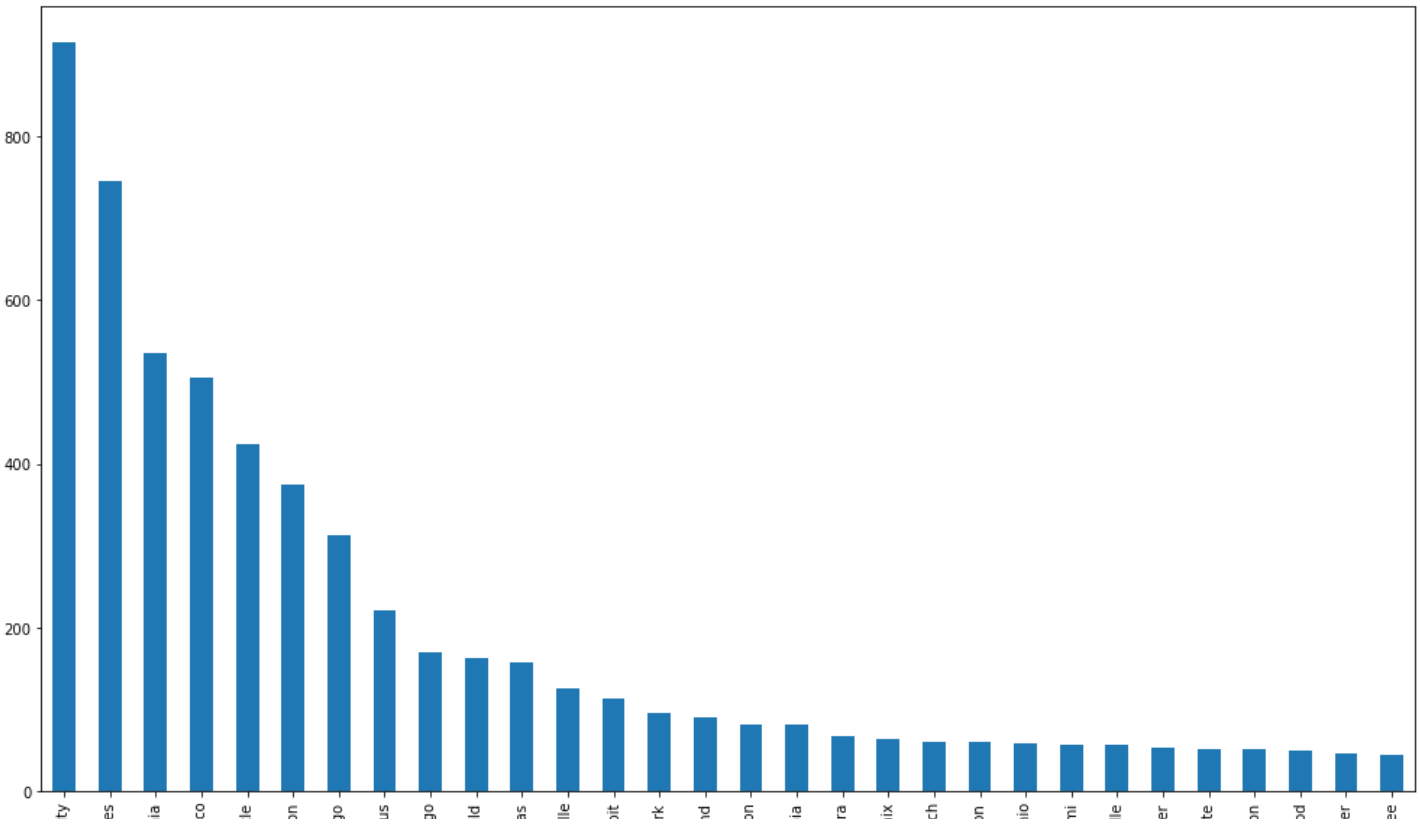
Top 30 cities with most no. of orders

In [28]:

```
df1['City'].value_counts().head(30).plot(kind = 'bar', figsize=(17,10))
```

Out[28]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f58d548cc10>
```



New York Ci Los Angeli Philadelph San Francis Seatt Houstc Chicag Columbi San Dieg Springfie Dalla Jacksonvil Detrc Newar Richmor Jacksc Columb Auror Phoen Long Beac Arlingtc San Anton Miar Louisvil Rochest Charlot Hendersc Lakewoc Lancast Milwauke

Quantities

Quantities Ordered by Ship Modes

In [30]:

```
df.shipmode = df.groupby('Ship Mode')['Quantity'].sum().reset_index()
print(df.shipmode)
```

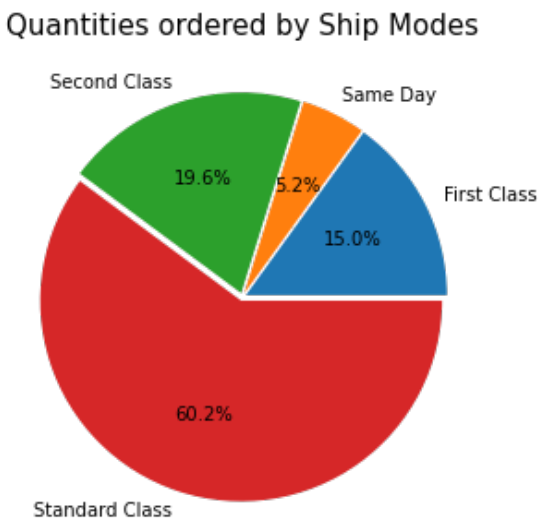
```
        Ship Mode  Quantity
0     First Class      5690
1        Same Day      1956
2    Second Class      7418
3  Standard Class     22756
```

In [32]:

```
shipmode_quantity=pd.DataFrame(df.groupby('Ship Mode').sum()['Quantity'])
labels=df.shipmode['Ship Mode'].unique()
plt.figure(figsize=(5,5))
plt.pie(df.shipmode['Quantity'], labels=shipmode_quantity.index, autopct='%1.1f%%', expl
ode=(0.02, 0.02, 0.02, 0.02),)
plt.title('Quantities ordered by Ship Modes', size=15)
```

Out[32]:

```
Text(0.5, 1.0, 'Quantities ordered by Ship Modes')
```



Quantities ordered by Segment

In [34]:

```
df.segment = df.groupby('Segment')['Quantity'].sum().reset_index()
print(df.segment)
```

```
       Segment  Quantity
0     Consumer     19497
1    Corporate     11591
2  Home Office      6732
```

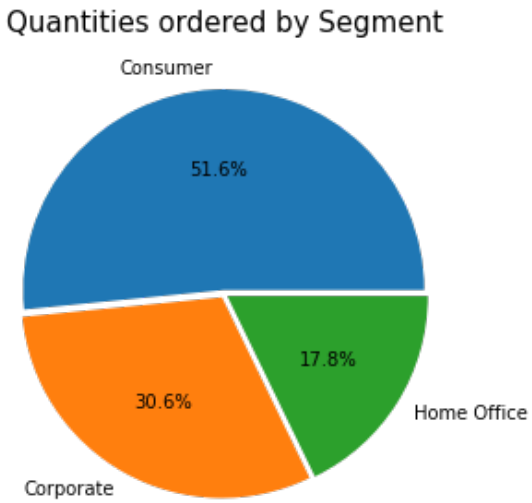
In [36]:

```
segment_quantity=pd.DataFrame(df.groupby('Segment').sum()['Quantity'])
labels=df.segment['Segment'].unique()
plt.figure(figsize=(5,5))
```

```
plt.pie(df.segment['Quantity'], labels=segment_quantity.index, autopct='%1.1f%%', explod
e=(0.02, 0.02, 0.02),)
plt.title('Quantities ordered by Segment', size=15)
```

Out[36]:

```
Text(0.5, 1.0, 'Quantities ordered by Segment')
```



Quantities ordered by Categories

In [38]:

```
df.category = df.groupby('Category')['Quantity'].sum().reset_index()
print(df.category)
```

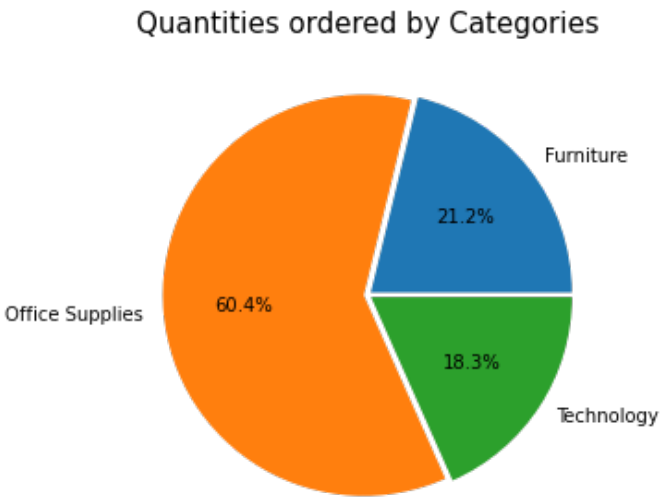
```
          Category  Quantity
0        Furniture      8020
1  Office Supplies     22861
2       Technology      6939
```

In [40]:

```
category_quantity=pd.DataFrame(df.groupby('Category').sum()['Quantity'])
labels=df.category['Category'].unique()
plt.figure(figsize=(5,5))
plt.pie(df.category['Quantity'], labels=category_quantity.index, autopct='%1.1f%%', expl
ode=(0.02, 0.02, 0.02),)
plt.title('Quantities ordered by Categories', size=15)
```

Out[40]:

```
Text(0.5, 1.0, 'Quantities ordered by Categories')
```



Quantities ordered by region

In [42]:

```
df.region = df.groupby('Region')['Quantity'].sum().reset_index()
print(df.region)
```

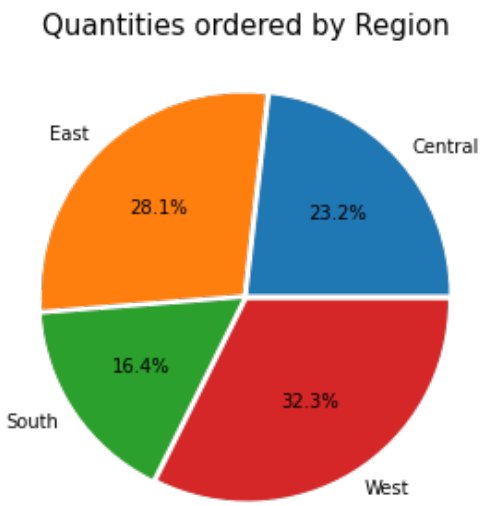
```
    Region  Quantity
0  Central      8768
1     East     10609
2    South      6209
3     West     12234
```

In [43]:

```
region_quantity=pd.DataFrame(df.groupby('Region').sum()['Quantity'])
labels=df.region['Region'].unique()
plt.figure(figsize=(5,5))
plt.pie(df.region['Quantity'], labels=region_quantity.index, autopct='%1.1f%%', explode=
(0.02, 0.02, 0.02, 0.02),)
plt.title('Quantities ordered by Region', size=15)
```

Out[43]:

```
Text(0.5, 1.0, 'Quantities ordered by Region')
```



Quantities ordered by States

In [45]:

```
states_quantity=df.groupby('State')['Quantity'].sum().reset_index().sort_values(by='Quan
tity', ascending=False)
```

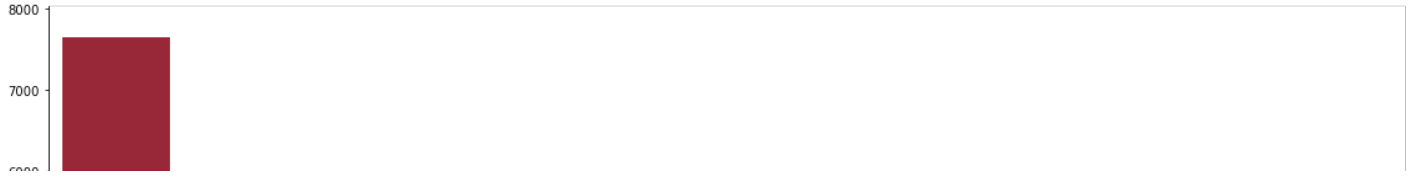
Top 10 States

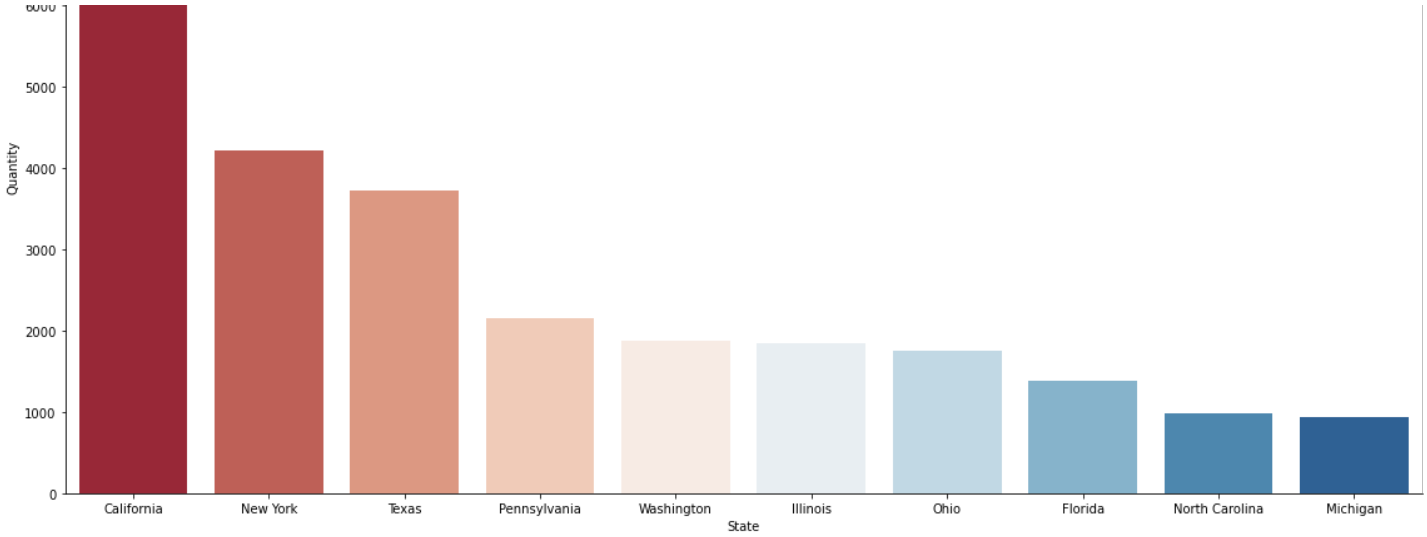
In [46]:

```
top10_states_quantity=states_quantity.head(10)
sns.catplot('State', 'Quantity', data=top10_states_quantity, kind='bar', aspect=2, heigh
t=8, palette="RdBu")
```

Out[46]:

```
<seaborn.axisgrid.FacetGrid at 0x7f58d5662d10>
```





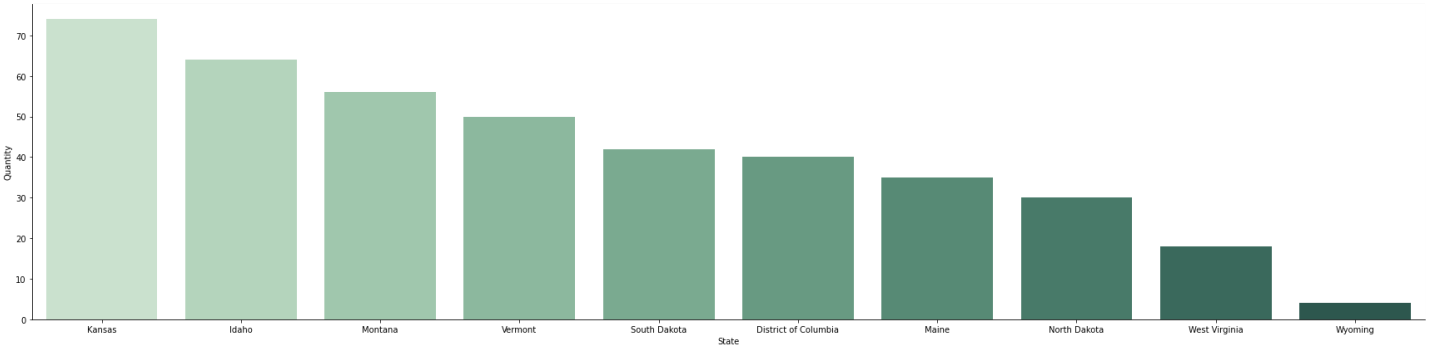
Bottom 10 States

In [47]:

```
bottom10_states_quantity=states_quantity.tail(10)
sns.catplot('State', 'Quantity', data=bottom10_states_quantity, kind='bar', aspect=4, he
ight=6, palette='ch:2.5,-.2,dark=.3')
```

Out[47]:

```
<seaborn.axisgrid.FacetGrid at 0x7f58d56fd1d0>
```



Quantities ordered by Sub-Categories

In [49]:

```
df.subcategory = df.groupby('Sub-Category')['Quantity'].sum().reset_index()
print(df.subcategory)
```

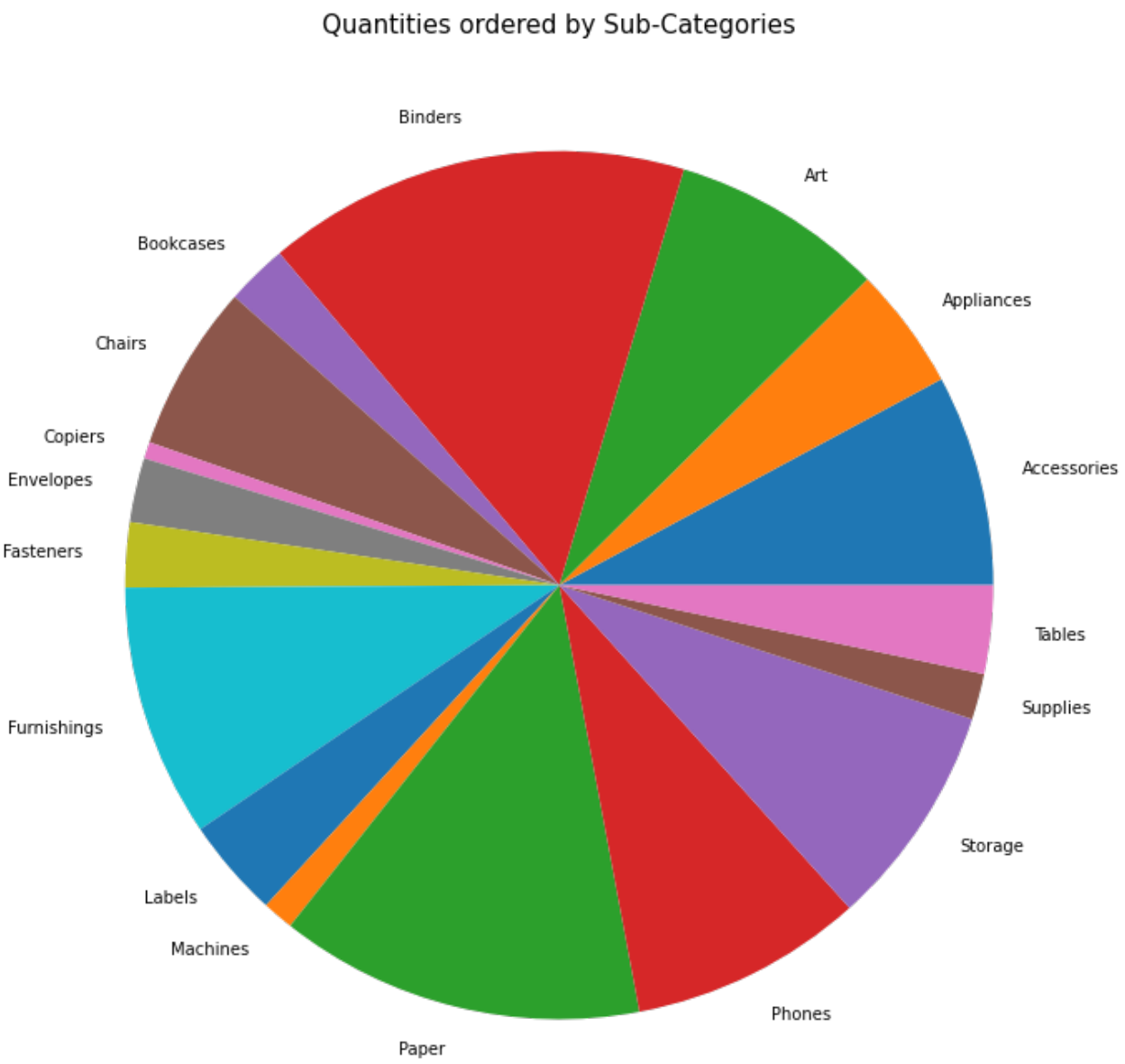
```
   Sub-Category  Quantity
0   Accessories      2976
1    Appliances      1729
2           Art      2996
3       Binders      5971
4     Bookcases       868
5        Chairs      2351
6       Copiers       234
7     Envelopes       906
8     Fasteners       914
9   Furnishings      3560
10       Labels      1396
11     Machines       440
12        Paper      5144
13       Phones      3289
14      Storage      3158
15     Supplies       647
16       Tables      1241
```

In [50]:

```
subcategory_quantity=pd.DataFrame(df.groupby('Sub-Category').sum()['Quantity'])
labels=df.subcategory['Sub-Category'].unique()
plt.figure(figsize=(12,12))
plt.pie(df.subcategory['Quantity'], labels=subcategory_quantity.index,)
plt.title('Quantities ordered by Sub-Categories', size=15)
```

Out[50]:

```
Text(0.5, 1.0, 'Quantities ordered by Sub-Categories')
```



Top 10 Cities

In [52]:

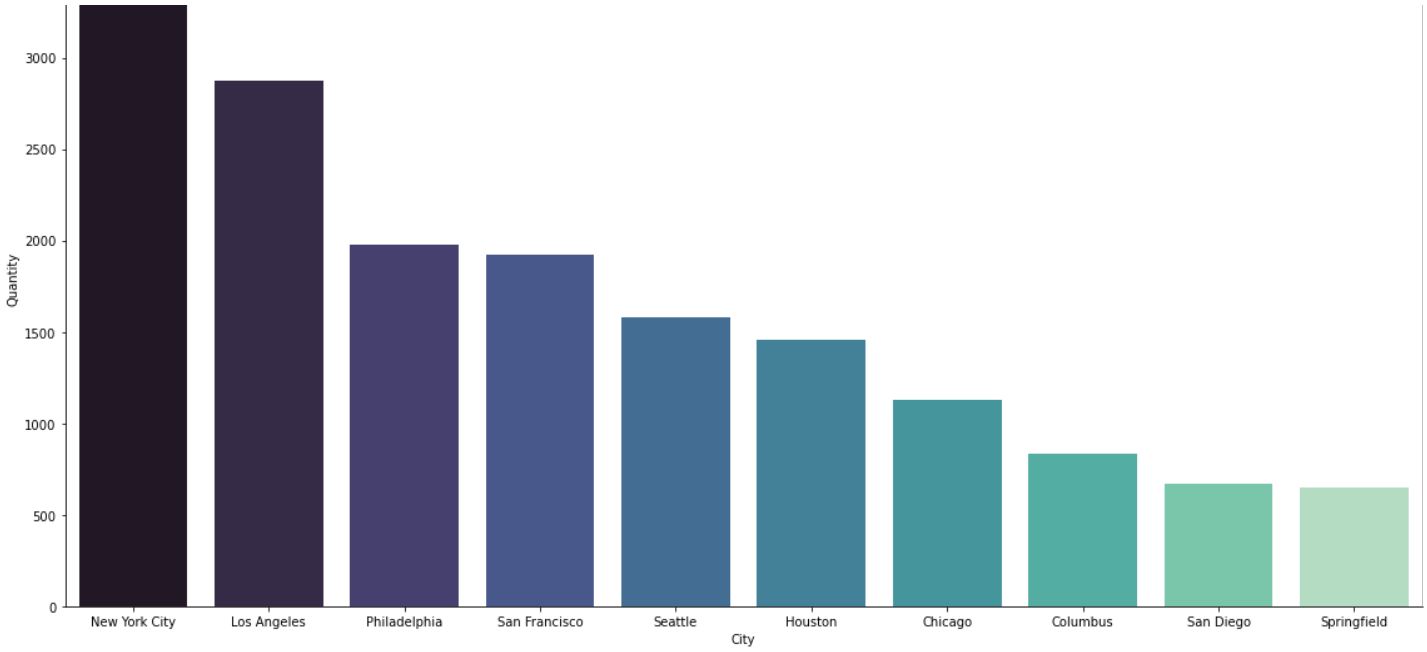
```
cities_quantity=df.groupby('City')['Quantity'].sum().reset_index().sort_values(by='Quantity', ascending=False)
```

In [53]:

```
top10_cities_quantity=cities_quantity.head(10)
sns.catplot('City', 'Quantity', data=top10_cities_quantity, kind='bar', aspect=2, height=8, palette="mako")
```

Out[53]:

```
<seaborn.axisgrid.FacetGrid at 0x7f58d43e8710>
```



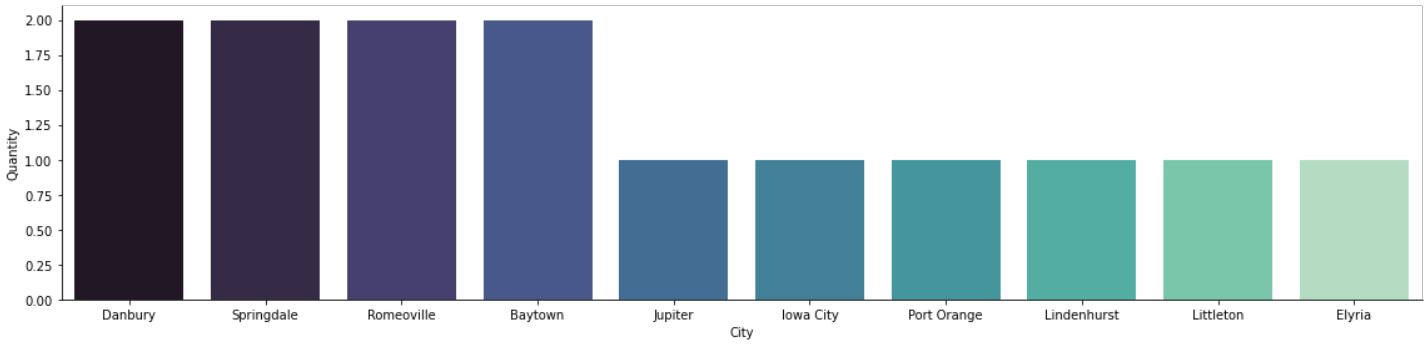
Bottom 10 Cities

In [55]:

```
bottom10_cities_quantity=cities_quantity.tail(10)
sns.catplot('City', 'Quantity', data=bottom10_cities_quantity, kind='bar', aspect=4, hei
ght=4, palette="mako")
```

Out[55]:

```
<seaborn.axisgrid.FacetGrid at 0x7f58d5117f50>
```



Sales

Shipmode wise sales

In [58]:

```
df.shipmodesales = df.groupby('Ship Mode')['Sales'].sum().reset_index()
print(df.shipmodesales)
```

```
        Ship Mode         Sales
0     First Class  3.513805e+05
1        Same Day  1.283217e+05
2    Second Class  4.591770e+05
3  Standard Class  1.357316e+06
```

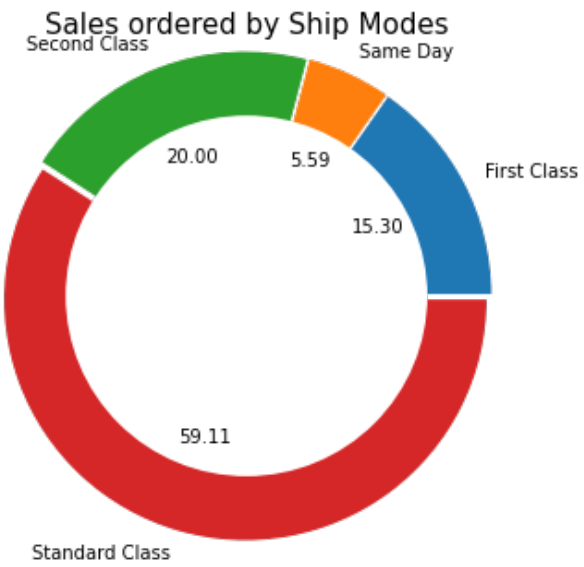
In [57]:

```
shipmode_sales=pd.DataFrame(df.groupby('Ship Mode').sum()['Sales'])
labels=df.shipmodesales['Ship Mode'].unique()
plt.figure(figsize=(5,5))
plt.pie(df.shipmodesales['Sales'], labels=shipmode_sales.index, autopct='%.2f', explode=
(0.02, 0.02, 0.02, 0.02), radius=1.2)
```

```
centre_circle=plt.Circle((0,0), 0.90, fc='white')
fig=plt.gcf()
fig.gca().add_artist(centre_circle)
plt.title('Sales ordered by Ship Modes', size=15)
```

Out[57]:

```
Text(0.5, 1.0, 'Sales ordered by Ship Modes')
```



Region Wise Sales

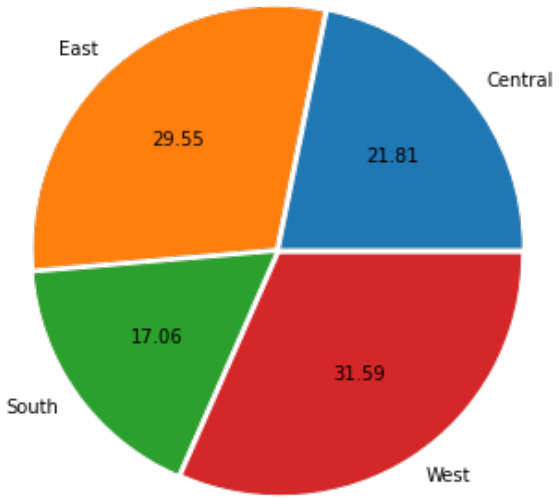
In [60]:

```
df.regionsales = df.groupby('Region')['Sales'].sum().reset_index()
print(df.regionsales)
```

```
    Region        Sales
0  Central  500782.8528
1     East  678435.1960
2    South  391721.9050
3     West  725255.6365
```

In [62]:

```
region_sales=pd.DataFrame(df.groupby('Region').sum()['Sales'])
labels=df.regionsales['Region'].unique()
plt.figure(figsize=(5,5))
plt.pie(df.regionsales['Sales'], labels=region_sales.index, autopct='%.2f', explode=(0.0
2, 0.02, 0.02, 0.02), radius=1.2)
centre_circle=plt.Circle((0,0), 0.90, fc='white')
fig=plt.gcf()
```



Category wise Sales

Category wise Sales

In [64]:

```
df.categorysales = df.groupby('Category')['Sales'].sum().reset_index()
print(df.categorysales)
```

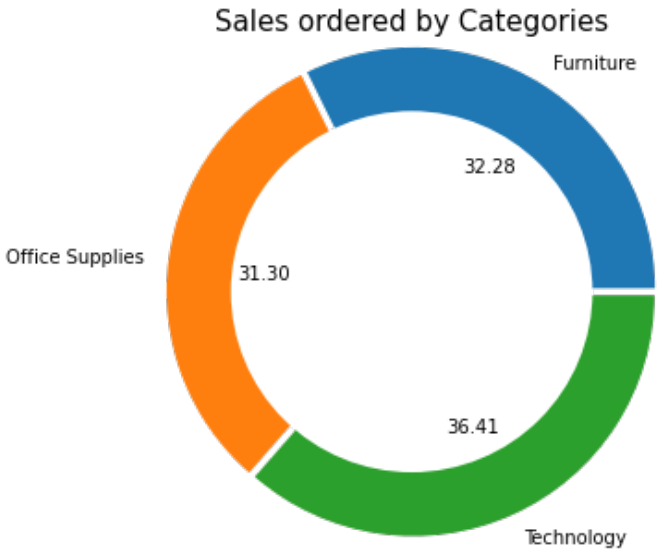
```
          Category        Sales
0        Furniture  741306.3133
1  Office Supplies  718735.2440
2       Technology  836154.0330
```

In [65]:

```
category_sales=pd.DataFrame(df.groupby('Category').sum()['Sales'])
labels=df.categorysales['Category'].unique()
plt.figure(figsize=(5,5))
plt.pie(df.categorysales['Sales'], labels=category_sales.index, autopct='%.2f', explode=
(0.02, 0.02, 0.02), radius=1.2)
centre_circle=plt.Circle((0,0), 0.90, fc='white')
fig=plt.gcf()
fig.gca().add_artist(centre_circle)
plt.title('Sales ordered by Categories', size=15)
```

Out[65]:

```
Text(0.5, 1.0, 'Sales ordered by Categories')
```



Segment wise Sales

In [67]:

```
df.segmentsales = df.groupby('Segment')['Sales'].sum().reset_index()
print(df.segmentsales)
```

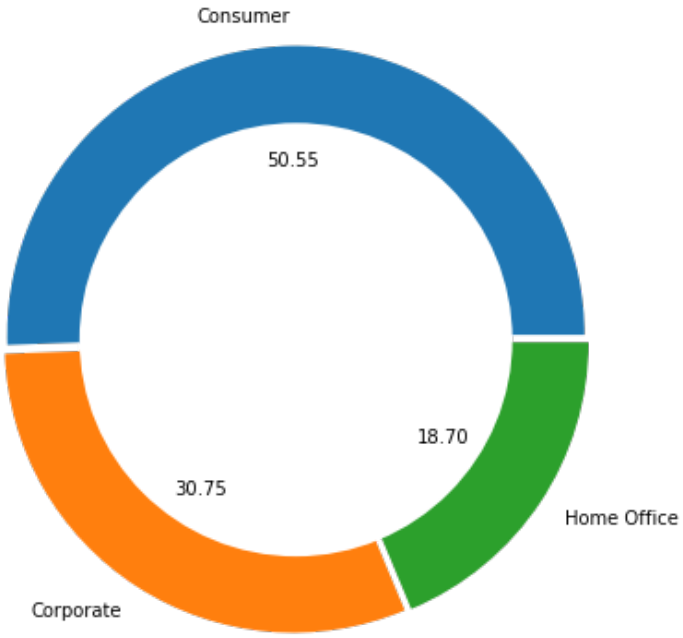
```
       Segment         Sales
0     Consumer  1.160833e+06
1    Corporate  7.060701e+05
2  Home Office  4.292927e+05
```

In [69]:

```
segment_sales=pd.DataFrame(df.groupby('Segment').sum()['Sales'])
labels=df.segmentsales['Segment'].unique()
plt.figure(figsize=(6,6))
plt.pie(df.segmentsales['Sales'], labels=segment_sales.index, autopct='%.2f', explode=(0
.02, 0.02, 0.02), radius=1.2)
centre_circle=plt.Circle((0,0), 0.90, fc='white')
fig=plt.gcf()
fig.gca().add_artist(centre_circle)
```

Out[69]:

<matplotlib.patches.Circle at 0x7f58d4e807d0>



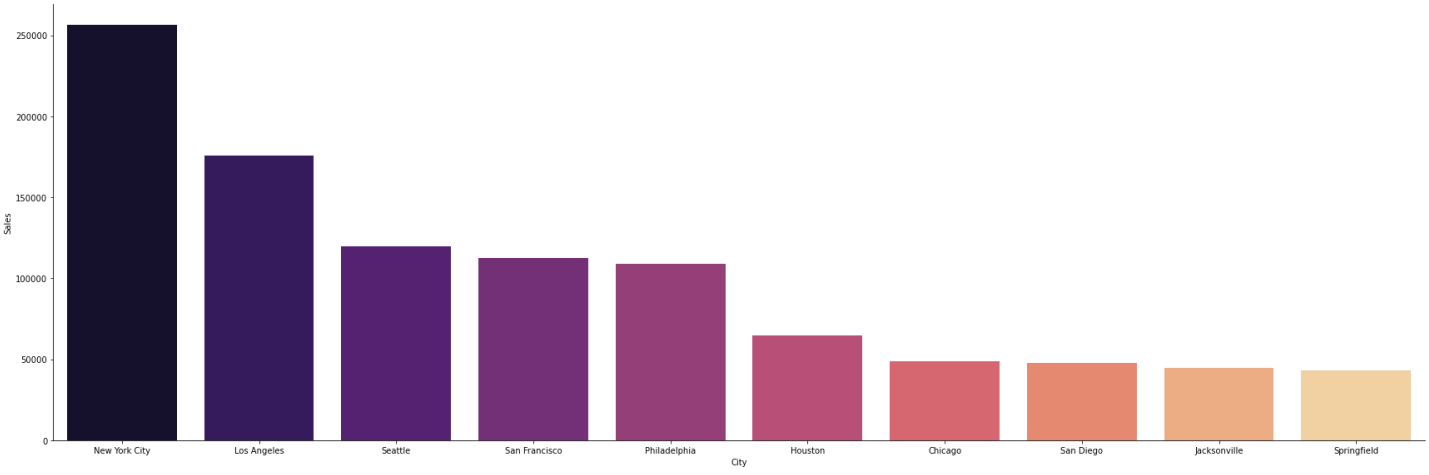
Top 10 Cities-sales wise

In [72]:

```
cities_sales=df.groupby('City')['Sales'].sum().reset_index().sort_values(by='Sales', asc
ending=False)
top10_cities_sales=cities_sales.head(10)
sns.catplot('City', 'Sales', data=top10_cities_sales, kind='bar', aspect=3, height=8, pa
lette="magma")
```

Out[72]:

<seaborn.axisgrid.FacetGrid at 0x7f58d4ea9390>



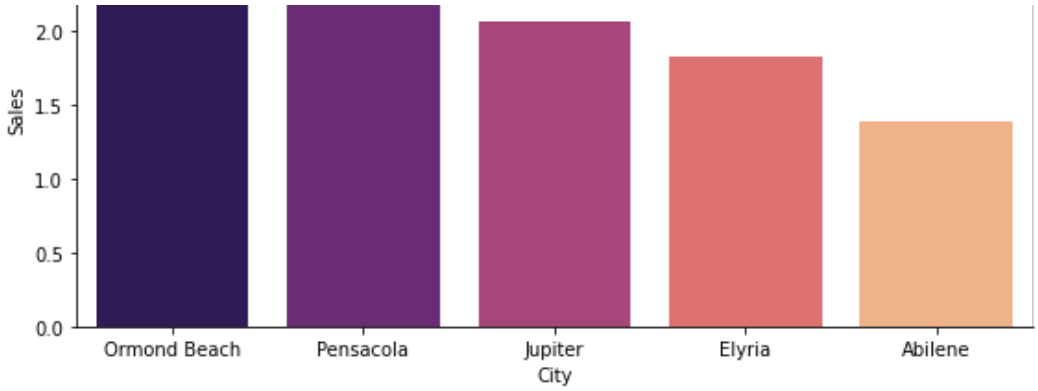
Bottom 5 Cities wise Sales

In [73]:

```
bottom5_cities_sales=cities_sales.tail(5)
sns.catplot('City', 'Sales', data=bottom5_cities_sales, kind='bar', aspect=2, height=4,
palette="magma")
```

Out[73]:

<seaborn.axisgrid.FacetGrid at 0x7f58d425da50>



Discount

Shipmode- Average Discount

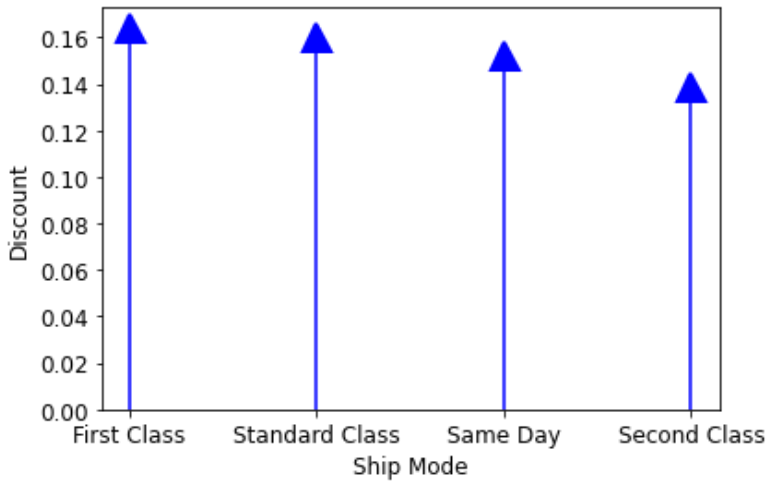
In [76]:

```
df.shipmodedisc = df.groupby('Ship Mode')['Discount'].agg(np.mean).reset_index().sort_va
lues(by='Discount', ascending=False)
print(df.shipmodedisc)
```

```
        Ship Mode  Discount
0     First Class  0.164587
3  Standard Class  0.160222
1        Same Day  0.152675
2    Second Class  0.138626
```

In [77]:

```
(markerline, stemlines, baseline) = plt.stem(df.shipmodedisc['Ship Mode'],
df.shipmodedisc['Discount'], use_line_collection=True)
plt.setp(markerline, marker='^', markersize=15,
markeredgewidth=2, color='blue')
plt.setp(stemlines, color='blue')
plt.setp(baseline, visible=False)
plt.tick_params(labelsize=12)
plt.xlabel('Ship Mode', size=12)
plt.ylabel('Discount', size=12)
plt.ylim(bottom=0)
plt.show()
```



Segments- Average Discount

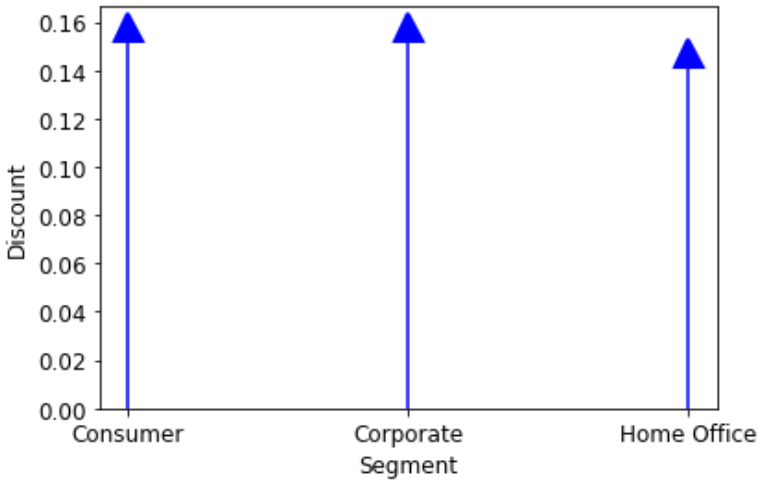
In [79]:

```
df.segmentdisc = df.groupby('Segment')['Discount'].agg(np.mean).reset_index().sort_value
s(by='Discount', ascending=False)
print(df.segmentdisc)
```

```
       Segment  Discount
0     Consumer  0.158308
1    Corporate  0.158159
2  Home Office  0.147178
```

In [81]:

```
(markerline, stemlines, baseline) = plt.stem(df.segmentdisc['Segment'],
df.segmentdisc['Discount'], use_line_collection=True)
plt.setp(markerline, marker='^', markersize=15,
markeredgewidth=2, color='blue')
plt.setp(stemlines, color='blue')
plt.setp(baseline, visible=False)
plt.tick_params(labelsize=12)
plt.xlabel('Segment', size=12)
plt.ylabel('Discount', size=12)
plt.ylim(bottom=0)
plt.show()
```



Regions- Average Discount

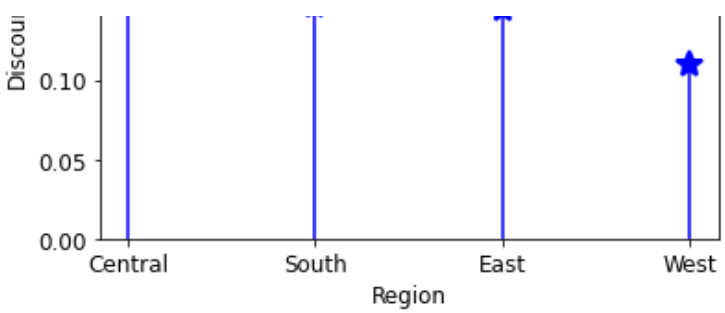
In [83]:

```
df.regiondisc = df.groupby('Region')['Discount'].agg(np.mean).reset_index().sort_values(
by='Discount', ascending=False)
print(df.regiondisc)
```

```
    Region  Discount
0  Central  0.240250
2    South  0.147253
1     East  0.145343
3     West  0.109615
```

In [84]:

```
(markerline, stemlines, baseline) = plt.stem(df.regiondisc['Region'],
df.regiondisc['Discount'], use_line_collection=True)
plt.setp(markerline, marker='*', markersize=15,
markeredgewidth=2, color='blue')
plt.setp(stemlines, color='blue')
plt.setp(baseline, visible=False)
plt.tick_params(labelsize=12)
plt.xlabel('Region', size=12)
plt.ylabel('Discount', size=12)
plt.ylim(bottom=0)
plt.show()
```



Categories- Average Discount

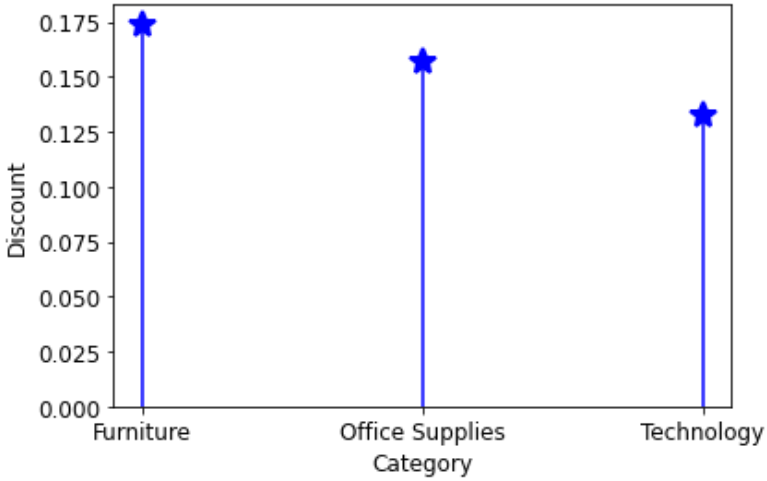
In [86]:

```
df.categorydisc = df.groupby('Category')['Discount'].agg(np.mean).reset_index().sort_values(by='Discount', ascending=False)
print(df.categorydisc)
```

```
          Category  Discount
0        Furniture  0.174027
1  Office Supplies  0.157385
2       Technology  0.132323
```

In [87]:

```
(markerline, stemlines, baseline) = plt.stem(df.categorydisc['Category'], df.categorydisc['Discount'], use_line_collection=True)
plt.setp(markerline, marker='*', markersize=15, markeredgewidth=2, color='blue')
plt.setp(stemlines, color='blue')
plt.setp(baseline, visible=False)
plt.tick_params(labelsize=12)
plt.xlabel('Category', size=12)
plt.ylabel('Discount', size=12)
plt.ylim(bottom=0)
plt.show()
```



Sub Categories- Average Discount

In [89]:

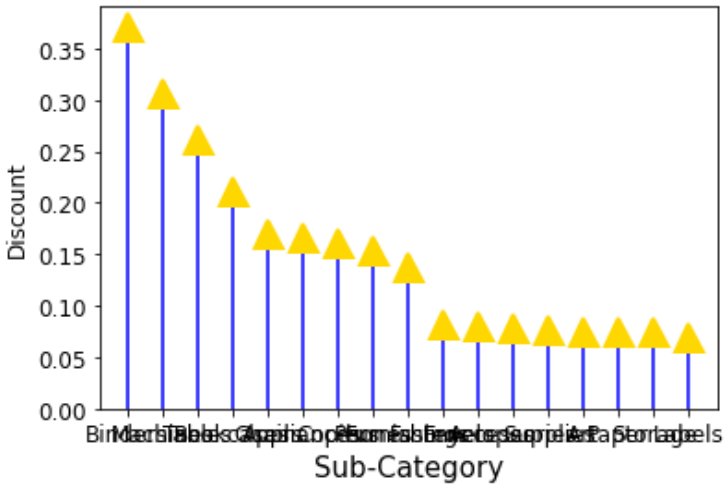
```
df.subcategorydisc = df.groupby('Sub-Category')['Discount'].agg(np.mean).reset_index().sort_values(by='Discount', ascending=False)
print(df.subcategorydisc)
```

```
   Sub-Category  Discount
3       Binders  0.372011
11     Machines  0.306087
16       Tables  0.261285
4     Bookcases  0.211140
5        Chairs  0.170244
1     Appliances  0.166524
```

```
6       Copiers  0.161765
13       Phones  0.154556
9    Furnishings  0.138494
8      Fasteners  0.082028
7      Envelopes  0.080315
0    Accessories  0.078452
15     Supplies  0.076842
2           Art  0.074969
12        Paper  0.074908
14      Storage  0.074704
10       Labels  0.068871
```

In [90]:

```
(markerline, stemlines, baseline) = plt.stem(df.subcategorydisc['Sub-Category'],
df.subcategorydisc['Discount'], use_line_collection=True)
plt.setp(markerline, marker='^', markersize=15,
markeredgewidth=2, color='gold')
plt.setp(stemlines, color='blue')
plt.setp(baseline, visible=False)
plt.tick_params(labelsize=12)
plt.xlabel('Sub-Category', size=15)
plt.ylabel('Discount', size=12)
plt.ylim(bottom=0)
plt.show()
```



Top 10 Sates having high average discount

In [92]:

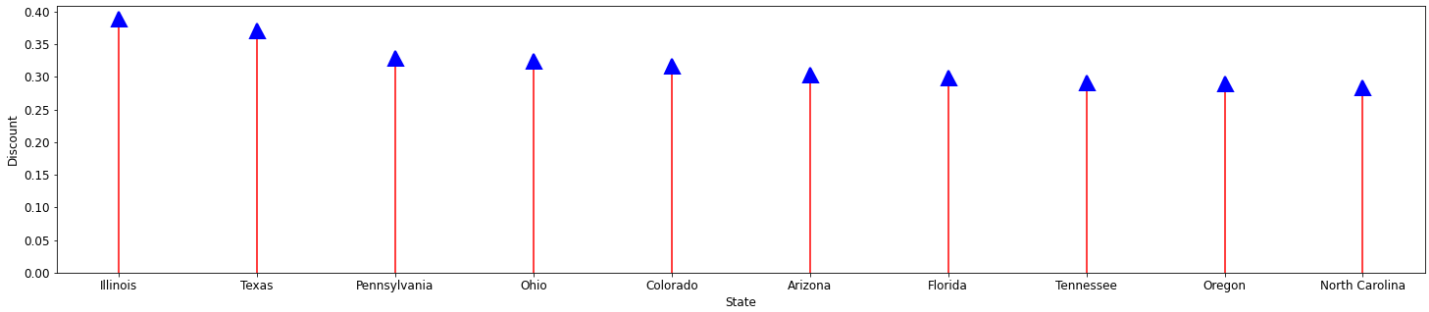
```
df.statedisc = df.groupby('State')['Discount'].agg(np.mean).reset_index().sort_values(by
='Discount', ascending=False)
top10_states_disc=df.statedisc.head(10)
print(top10_states_disc)
```

```
            State  Discount
11        Illinois  0.389206
41           Texas  0.370539
36    Pennsylvania  0.328840
33            Ohio  0.325000
4         Colorado  0.316484
1          Arizona  0.303571
8          Florida  0.299347
40       Tennessee  0.291257
35          Oregon  0.289431
31  North Carolina  0.283534
```

In [93]:

```
plt.figure(figsize=(25,5))
(markerline, stemlines, baseline) = plt.stem(top10_states_disc['State'],
top10_states_disc['Discount'], use_line_collection=True)
plt.setp(markerline, marker='^', markersize=15,
```

```
markeredgewidth=2, color='blue')
plt.setp(stemlines, color='red')
plt.setp(baseline, visible=False)
plt.tick_params(labelsize=12)
plt.xlabel('State', size=12)
plt.ylabel('Discount', size=12)
plt.ylim(bottom=0)
plt.show()
```



Top 10 Cities having high Average of Discounts

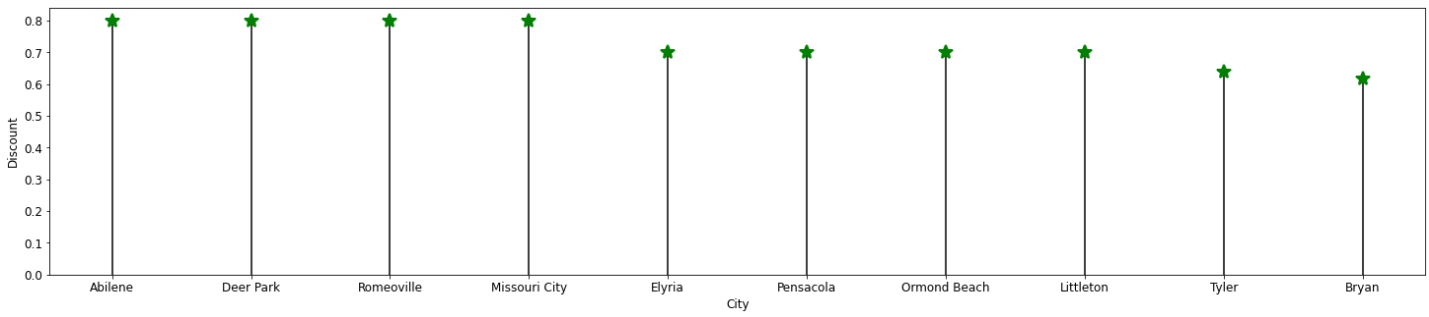
In [100]:

```
df.citydisc = df.groupby('City')['Discount'].agg(np.mean).reset_index().sort_values(by='
Discount', ascending=False)
top10_cities_disc=df.citydisc.head(10)
print(top10_cities_disc)
```

```
             City  Discount
1         Abilene  0.800000
117     Deer Park  0.800000
417    Romeoville  0.800000
305  Missouri City  0.800000
140        Elyria  0.700000
370     Pensacola  0.700000
354  Ormond Beach  0.700000
259     Littleton  0.700000
493         Tyler  0.640000
55          Bryan  0.616667
```

In [101]:

```
plt.figure(figsize=(25,5))
(markerline, stemlines, baseline) = plt.stem(top10_cities_disc['City'],
top10_cities_disc['Discount'], use_line_collection=True)
plt.setp(markerline, marker='*', markersize=15,
markeredgewidth=2, color='green')
plt.setp(stemlines, color='black')
plt.setp(baseline, visible=False)
plt.tick_params(labelsize=12)
plt.xlabel('City', size=12)
plt.ylabel('Discount', size=12)
plt.ylim(bottom=0)
plt.show()
```



Profit

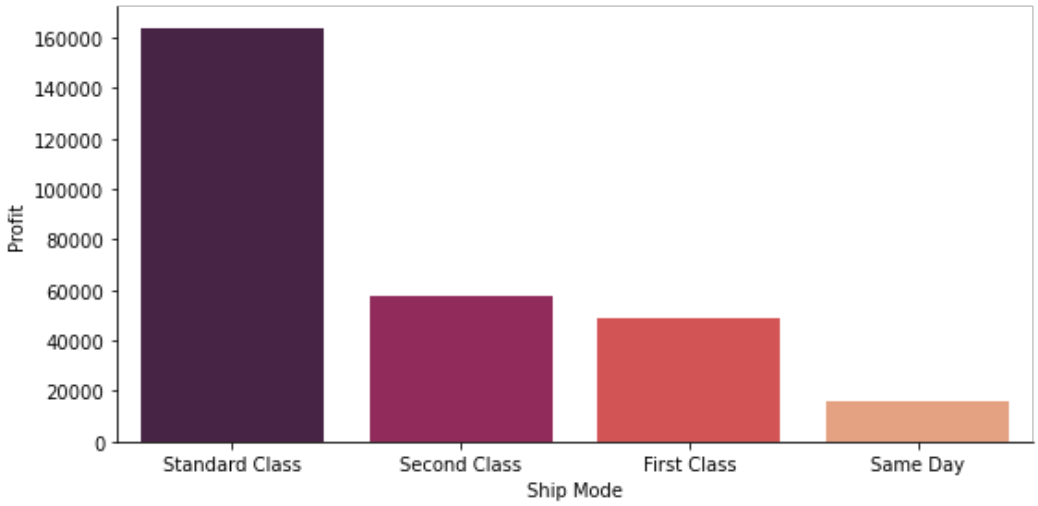
Profit by Ship Mode

In [102]:

```
shipmode_profit=df.groupby('Ship Mode')['Profit'].sum().reset_index().sort_values(by='Profit', ascending=False)
sns.catplot('Ship Mode', 'Profit', data=shipmode_profit, kind='bar', aspect=2, height=4, palette="rocket")
```

Out[102]:

```
<seaborn.axisgrid.FacetGrid at 0x7f58cc635490>
```



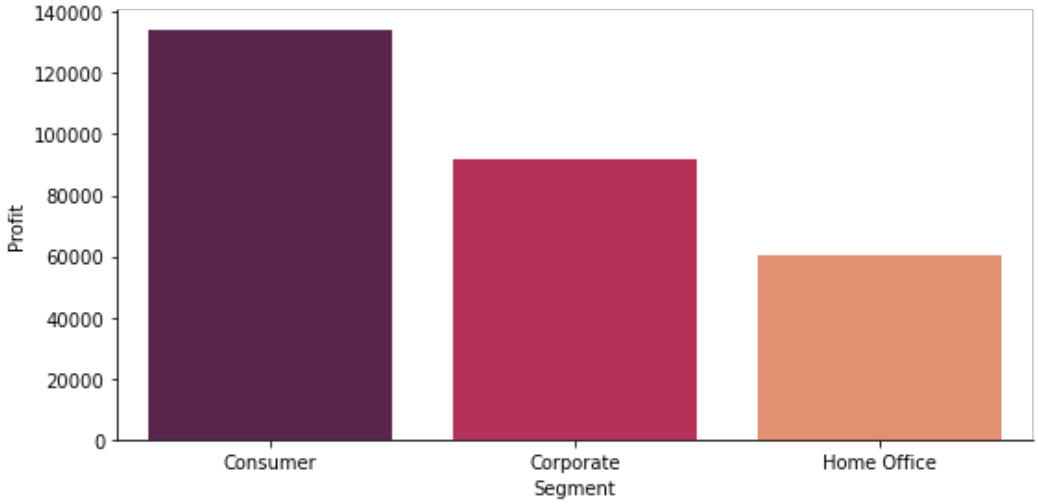
Profit by Segments

In [104]:

```
segment_profit=df.groupby('Segment')['Profit'].sum().reset_index().sort_values(by='Profit', ascending=False)
sns.catplot('Segment', 'Profit', data=segment_profit, kind='bar', aspect=2, height=4, palette="rocket")
```

Out[104]:

```
<seaborn.axisgrid.FacetGrid at 0x7f58cc7c9b90>
```



Profit by Region

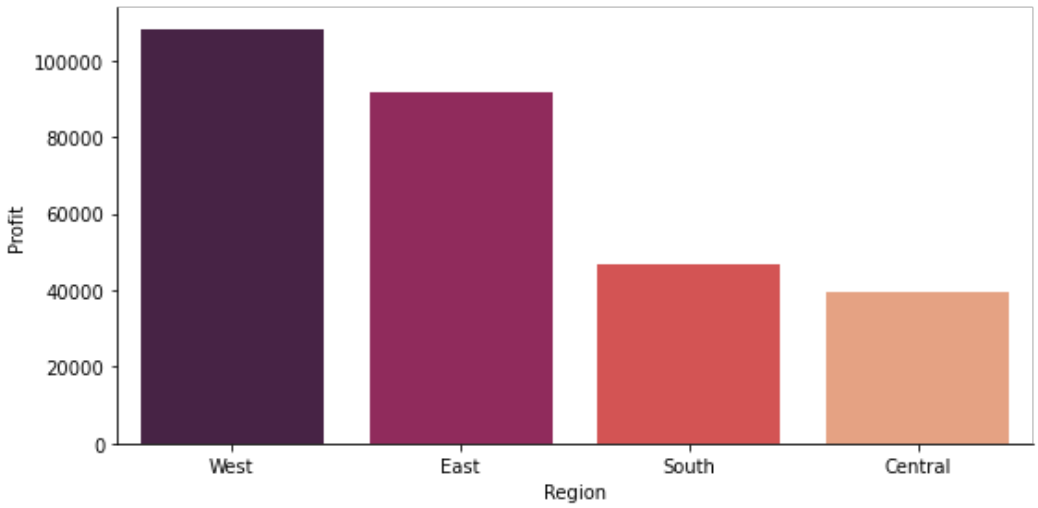
In [105]:

```
region_profit=df.groupby('Region')['Profit'].sum().reset_index().sort_values(by='Profit', ascending=False)
sns.catplot('Region', 'Profit', data=region_profit, kind='bar', aspect=2, height=4, pale
```

```
tte="rocket")
```

Out[105]:

```
<seaborn.axisgrid.FacetGrid at 0x7f58cc7aad10>
```



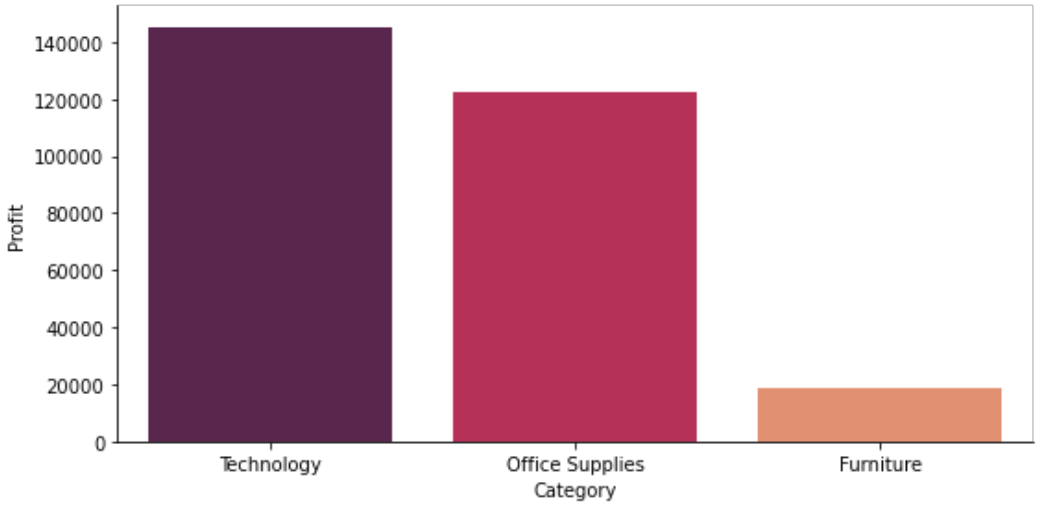
Profit by Categories

In [106]:

```
category_profit=df.groupby('Category')['Profit'].sum().reset_index().sort_values(by='Pro
fit', ascending=False)
sns.catplot('Category', 'Profit', data=category_profit, kind='bar', aspect=2, height=4,
palette="rocket")
```

Out[106]:

```
<seaborn.axisgrid.FacetGrid at 0x7f58cc70c490>
```



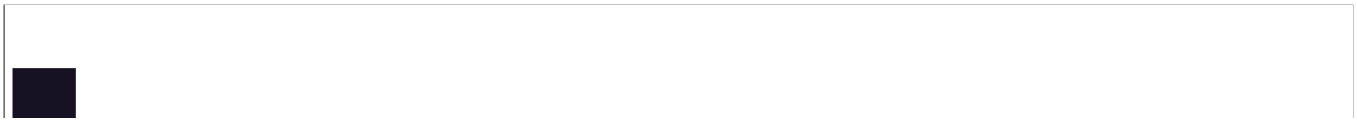
Profit by Sub-Categories

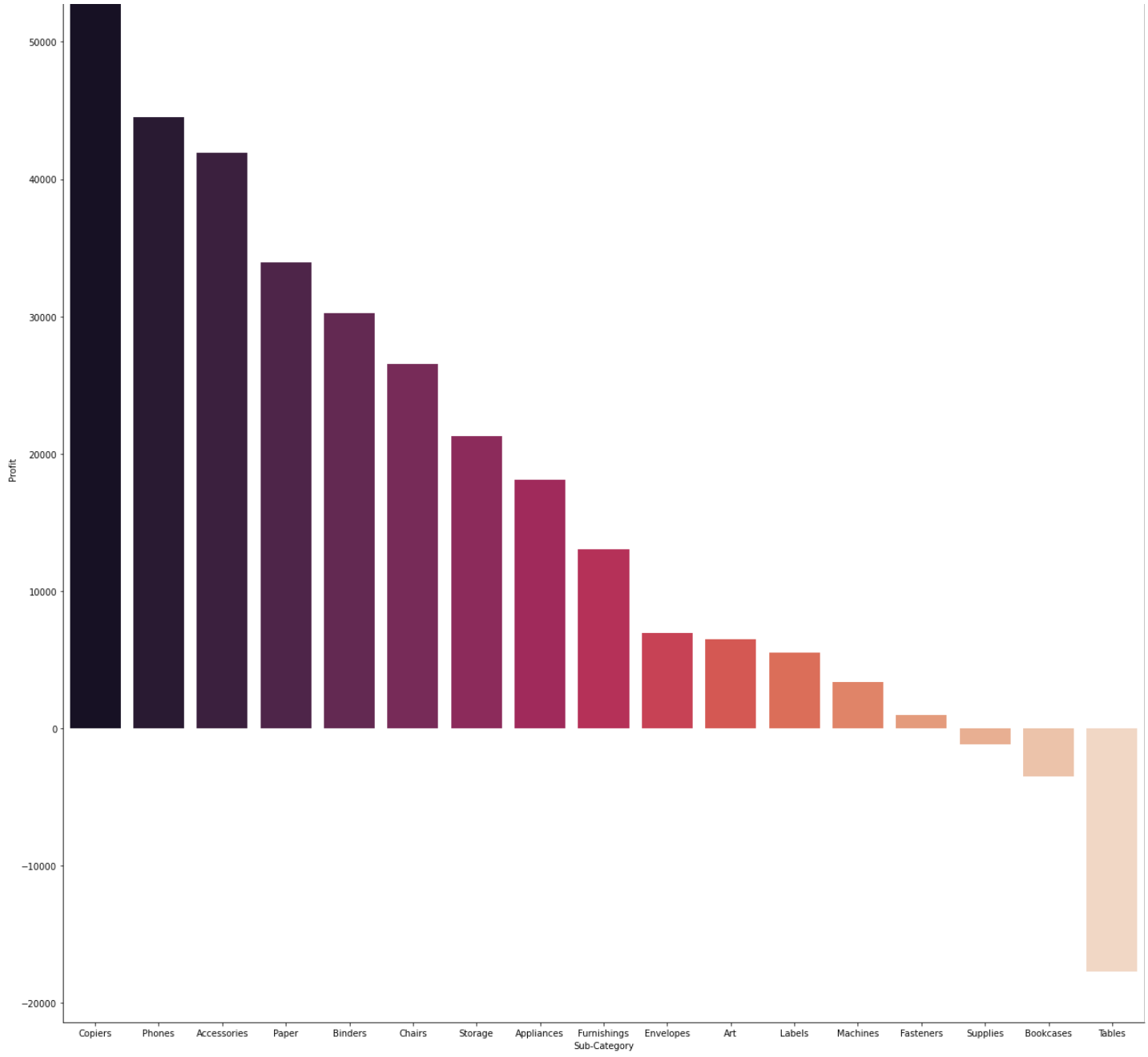
In [107]:

```
subcategory_profit=df.groupby('Sub-Category')['Profit'].sum().reset_index().sort_values(
by='Profit', ascending=False)
sns.catplot('Sub-Category', 'Profit', data=subcategory_profit, kind='bar', aspect=1, heig
ht=18, palette="rocket")
```

Out[107]:

```
<seaborn.axisgrid.FacetGrid at 0x7f58d4edc750>
```





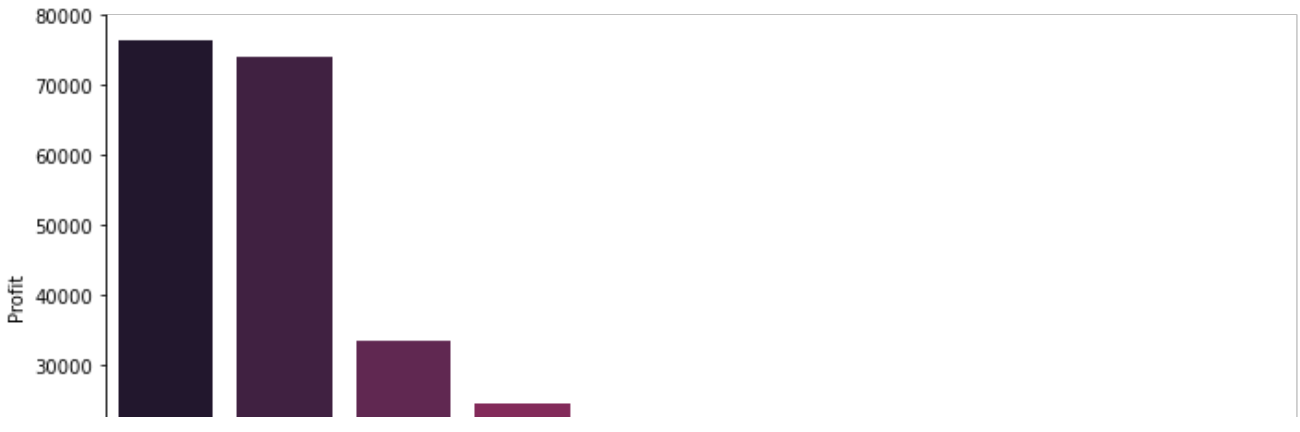
Top 10 States by Profit

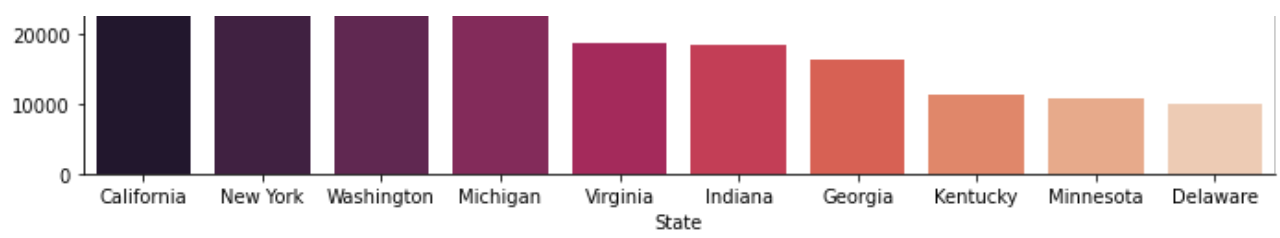
In [108]:

```
states_profit=df.groupby('State')['Profit'].sum().reset_index().sort_values(by='Profit',
ascending=False)
top10_states_profit=states_profit.head(10)
sns.catplot('State', 'Profit', data=top10_states_profit, kind='bar', aspect=2, height=5,
palette="rocket")
```

Out[108]:

```
<seaborn.axisgrid.FacetGrid at 0x7f58d4ed9fd0>
```





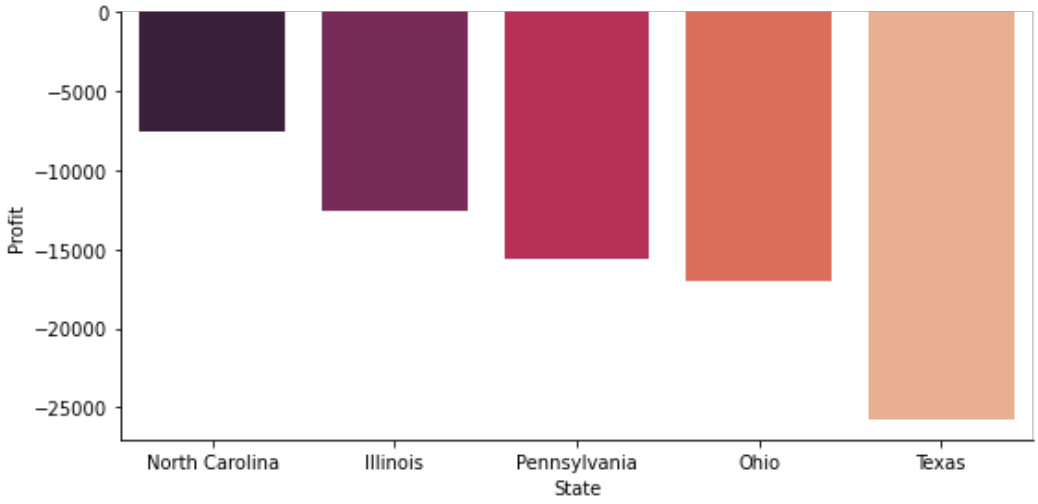
Bottom 5 Sates by Profit

In [109]:

```
bottom5_states_profit=states_profit.tail(5)
sns.catplot('State', 'Profit', data=bottom5_states_profit, kind='bar', aspect=2, height=
4, palette="rocket")
```

Out[109]:

```
<seaborn.axisgrid.FacetGrid at 0x7f58cc3733d0>
```



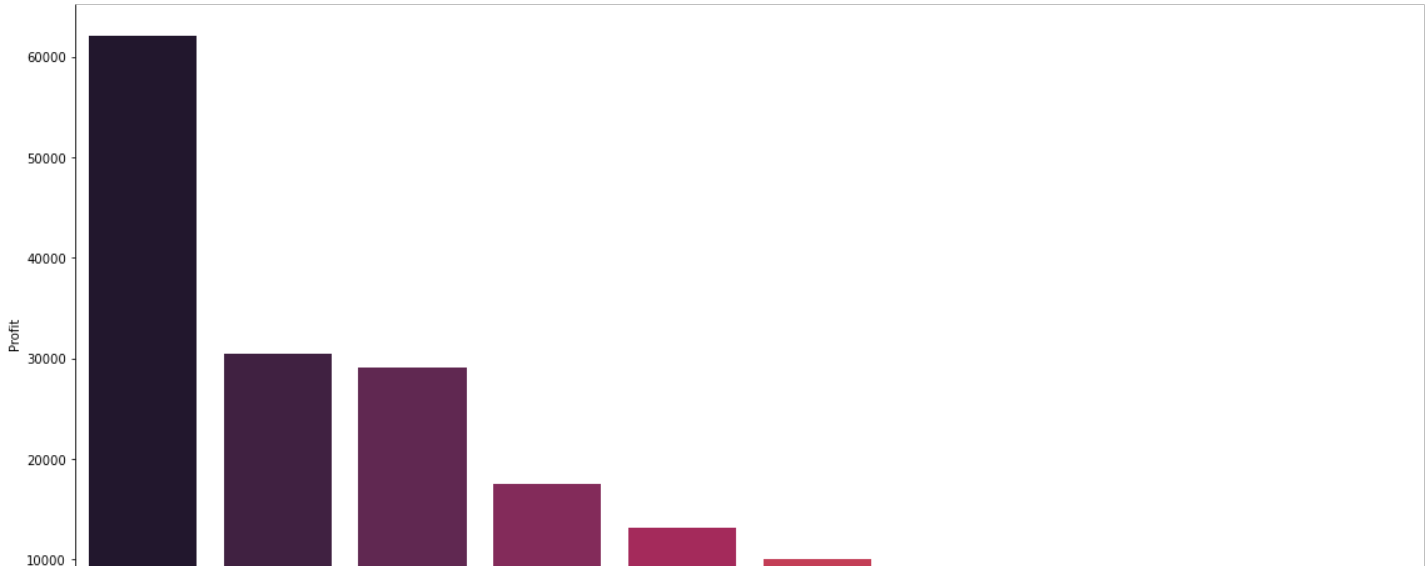
Top 10 Cities by Profit

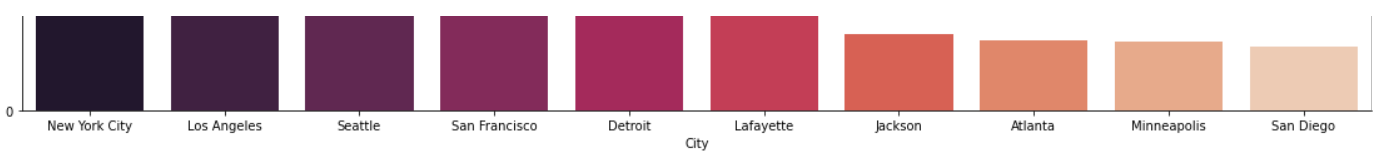
In [110]:

```
cities_profit=df.groupby('City')['Profit'].sum().reset_index().sort_values(by='Profit',
ascending=False)
top10_cities_profit=cities_profit.head(10)
sns.catplot('City', 'Profit', data=top10_cities_profit, kind='bar', aspect=2, height=8,
palette="rocket")
```

Out[110]:

```
<seaborn.axisgrid.FacetGrid at 0x7f58cc388090>
```





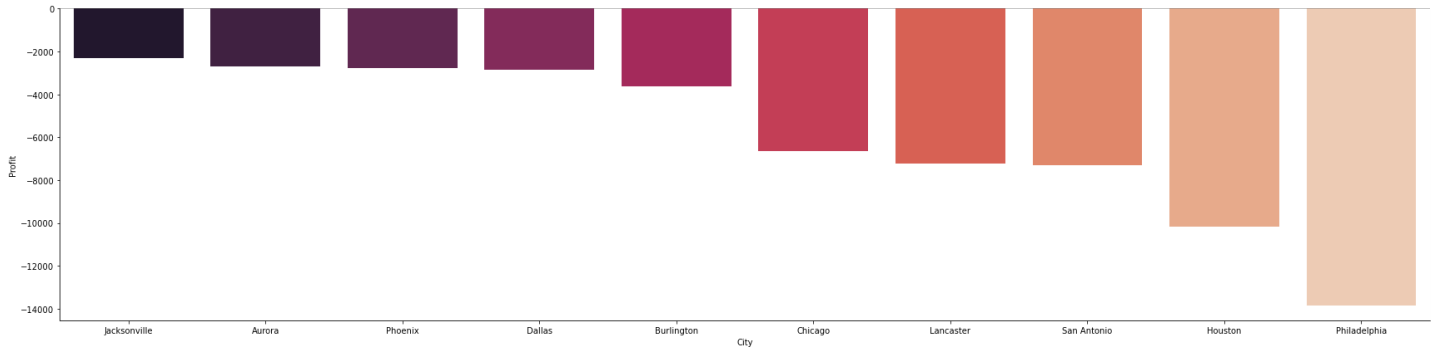
Bottom 10 Cities by Profit

In [111]:

```
bottom10_cities_profit=cities_profit.tail(10)
sns.catplot('City', 'Profit', data=bottom10_cities_profit, kind='bar', aspect=4, height=6, palette="rocket")
```

Out[111]:

```
<seaborn.axisgrid.FacetGrid at 0x7f58cc27be10>
```



CONCLUSION

The segment - 'Home office' generates lowest profit and lowest sale also it is the least ordered segment.

Central region generatres lowest profit despite being offered highest average discount.

Southern region generates the lowest sale among all the regions.

Lowest sale and quantities were ordered from Wyoming and West Virginia.

Texas and Illinois generated least sale even after offering high discounts.

Among citites Philadelphia and Houston recorded highest losses.

Standard class accounts for majority of profits.

Sales of bookcase and tables are good but the profit is negative. The Company is facing loss because of these two products.

Improvements should be made for the same day shipment mode.

Office Supplies are excellent. We have to work more on Furniture and Technology Category of business.

When the profits of a state are compared with the discount provided in each state, the states which offered more discount went in loss.

In [ ]: