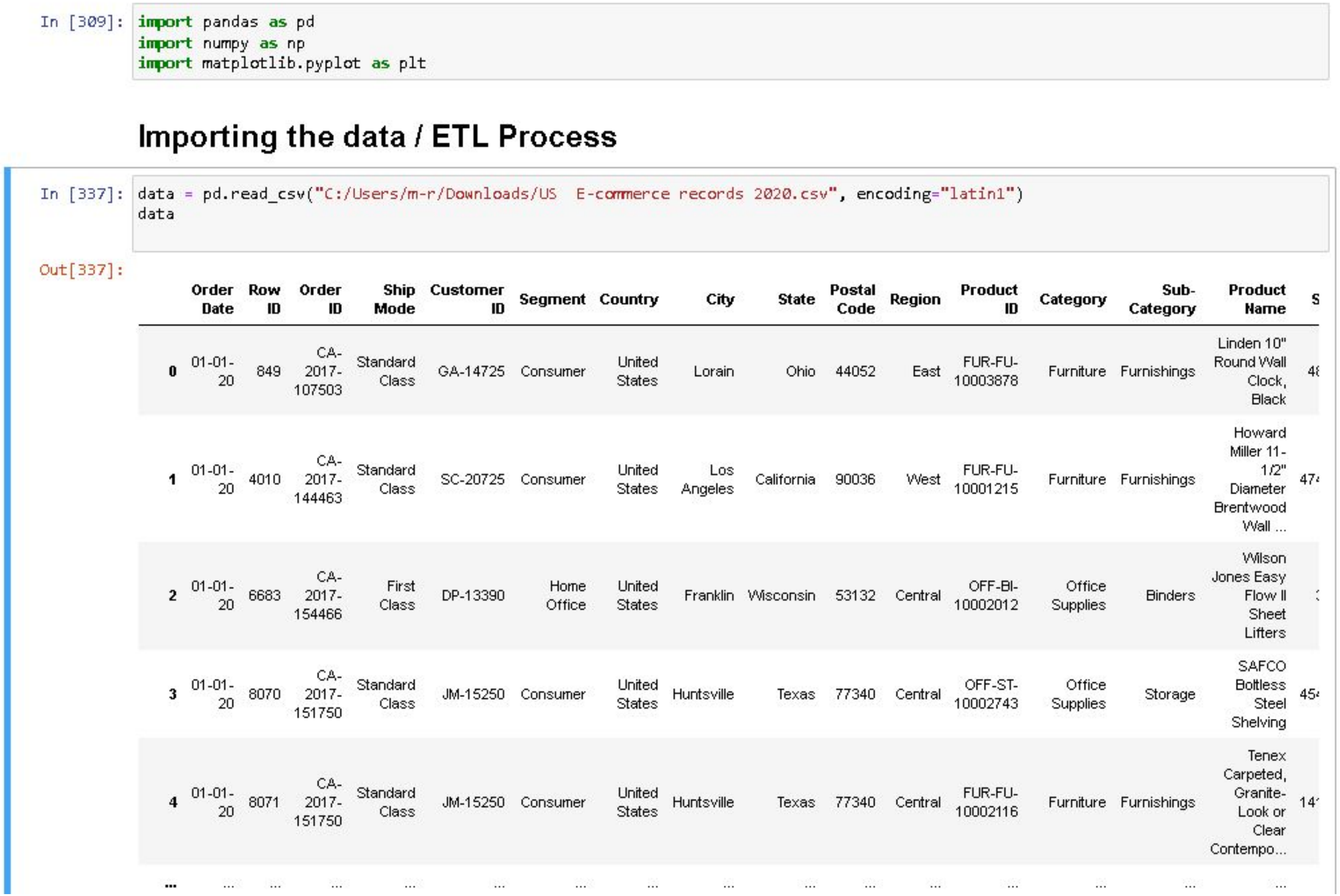
E-Commerce Python Project

Summary

The project in e-commerce involved extracting, transforming, and loading (ETL) data from various sources. The data was then cleaned and analyzed to extract insights crucial for decision-making.

By synthesizing this information, we extracted actionable insights to guide strategic business decisions, optimize operations, and enhance overall performance in the e-commerce landscape.



In [309]:

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

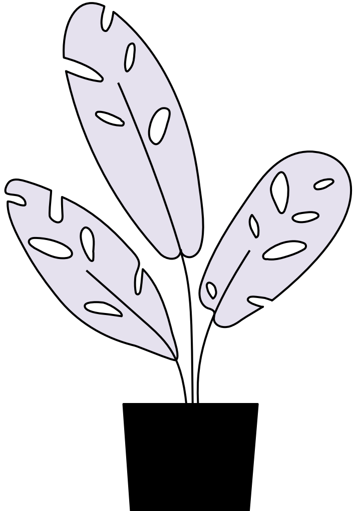
Importing the data / ETL Process

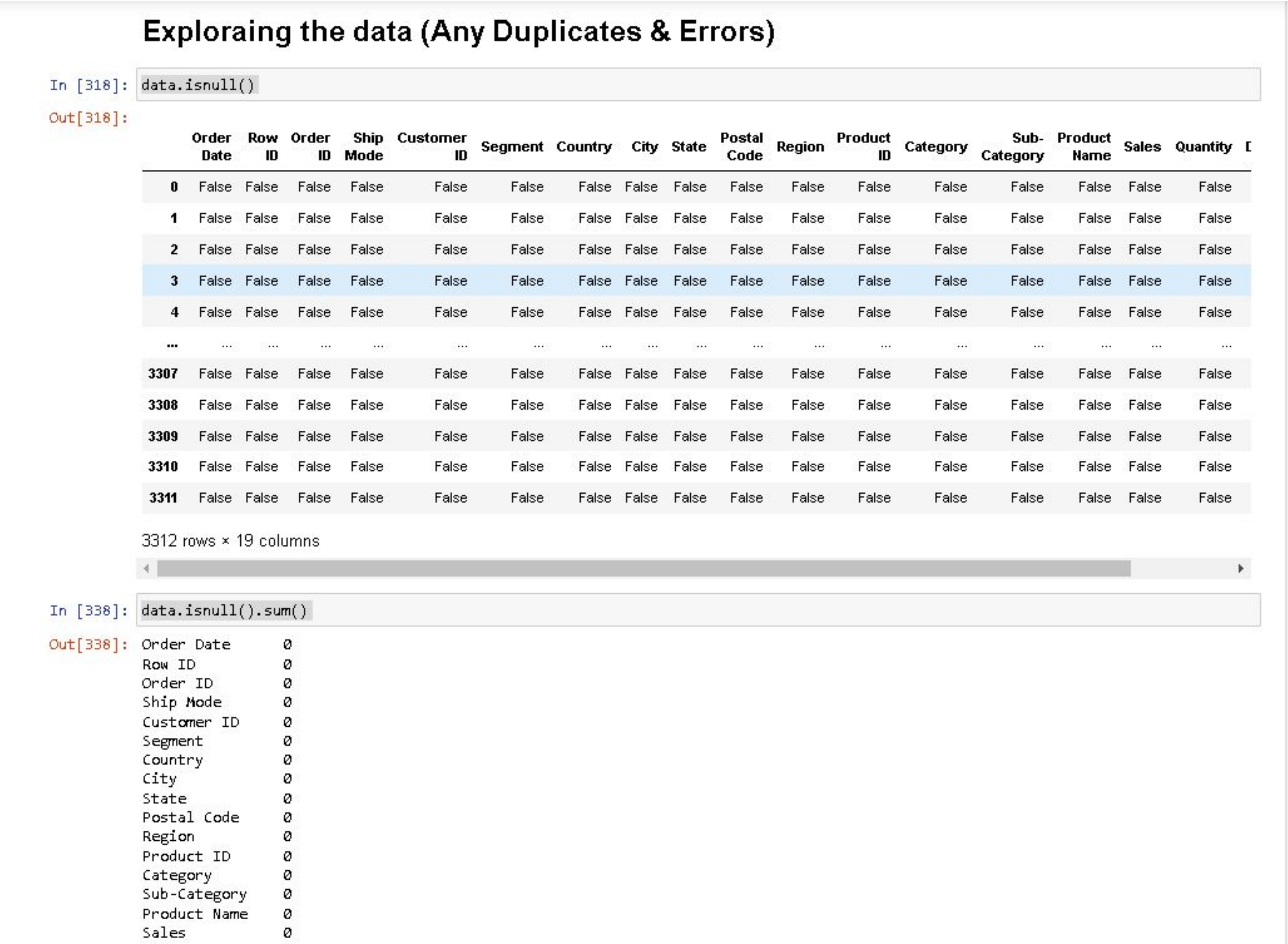
In [337]:

```
data = pd.read_csv("C:/Users/m-r/Downloads/US  E-commerce records 2020.csv", encoding="latin1")
data
```

Out[337]:

	Order Date	Row ID	Order ID	Ship Mode	Customer ID	Segment	Country	City	State	Postal Code	Region	Product ID	Category	Sub-Category	Product Name	S
0	01-01-20	849	CA-2017-107503	Standard Class	GA-14725	Consumer	United States	Lorain	Ohio	44052	East	FUR-FU-10003878	Furniture	Furnishings	Linden 10" Round Wall Clock, Black	48
1	01-01-20	4010	CA-2017-144463	Standard Class	SC-20725	Consumer	United States	Los Angeles	California	90036	West	FUR-FU-10001215	Furniture	Furnishings	Howard Miller 11-1/2" Diameter Brentwood Wall ...	47
2	01-01-20	6683	CA-2017-154466	First Class	DP-13390	Home Office	United States	Franklin	Wisconsin	53132	Central	OFF-BI-10002012	Office Supplies	Binders	Wilson Jones Easy Flow II Sheet Lifters	
3	01-01-20	8070	CA-2017-151750	Standard Class	JM-15250	Consumer	United States	Huntsville	Texas	77340	Central	OFF-ST-10002743	Office Supplies	Storage	SAFCO Boltless Steel Shelving	45
4	01-01-20	8071	CA-2017-151750	Standard Class	JM-15250	Consumer	United States	Huntsville	Texas	77340	Central	FUR-FU-10002116	Furniture	Furnishings	Tenex Carpeted, Granite-Look or Clear Contempo...	14
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	





Exploraing the data (Any Duplicates & Errors)

```
In [318]: data.isnull()
```

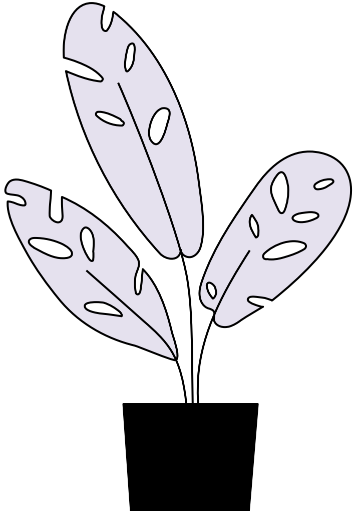
Out[318]:

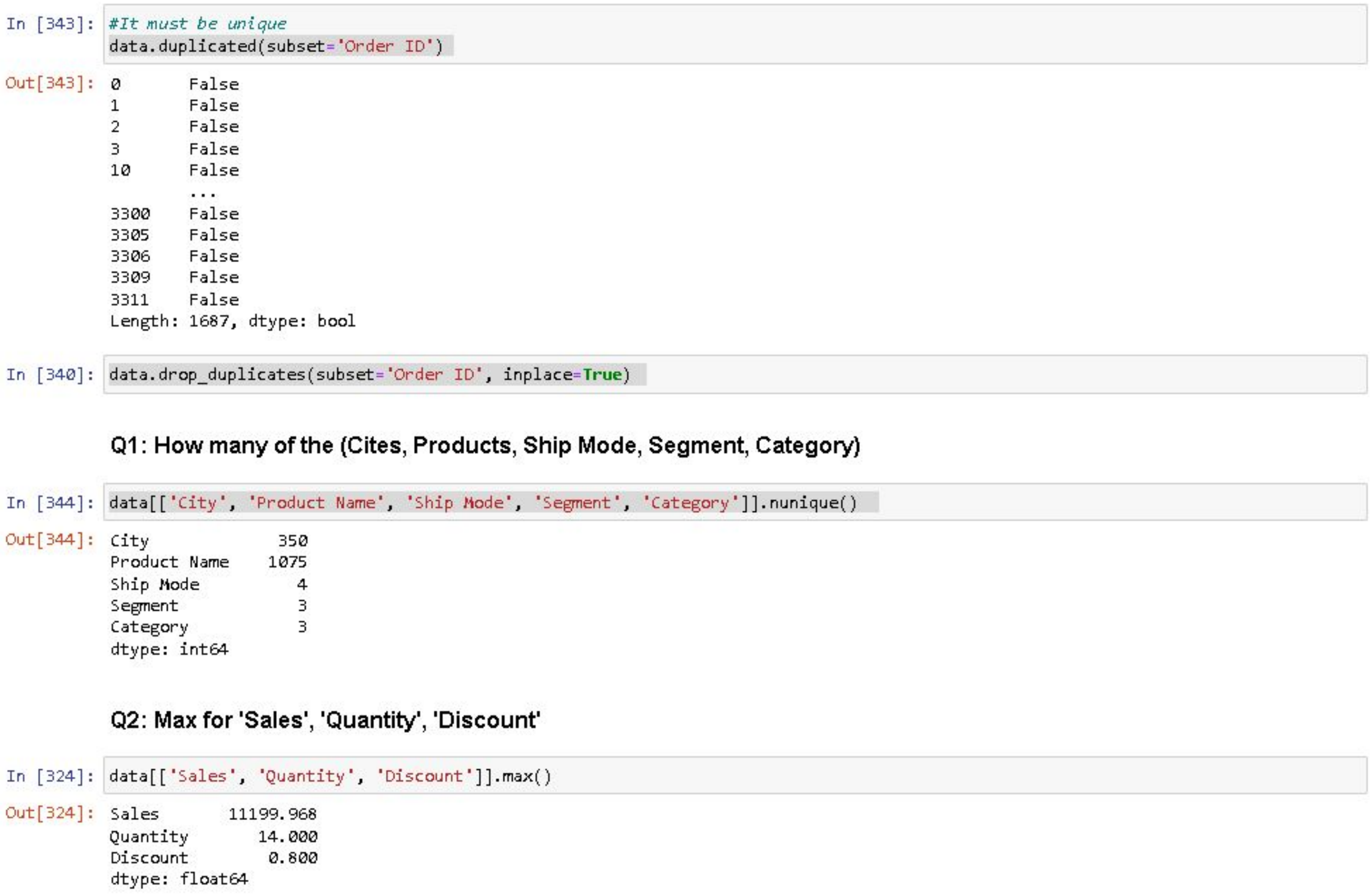
	Order Date	Row ID	Order ID	Ship Mode	Customer ID	Segment	Country	City	State	Postal Code	Region	Product ID	Category	Sub-Category	Product Name	Sales	Quantity	D
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
3307	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	
3308	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	
3309	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	
3310	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	
3311	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	

3312 rows × 19 columns

```
In [338]: data.isnull().sum()
```

```
Out[338]: Order Date       0
          Row ID           0
          Order ID         0
          Ship Mode        0
          Customer ID      0
          Segment          0
          Country          0
          City             0
          State            0
          Postal Code      0
          Region           0
          Product ID       0
          Category         0
          Sub-Category     0
          Product Name     0
          Sales            0
```





```
In [343]: #It must be unique
          data.duplicated(subset='Order ID')
```

```
Out[343]: 0       False
          1       False
          2       False
          3       False
          10      False
                  ...  
          3300    False
          3305    False
          3306    False
          3309    False
          3311    False
          Length: 1687, dtype: bool
```

```
In [340]: data.drop_duplicates(subset='Order ID', inplace=True)
```

Q1: How many of the (Cites, Products, Ship Mode, Segment, Category)

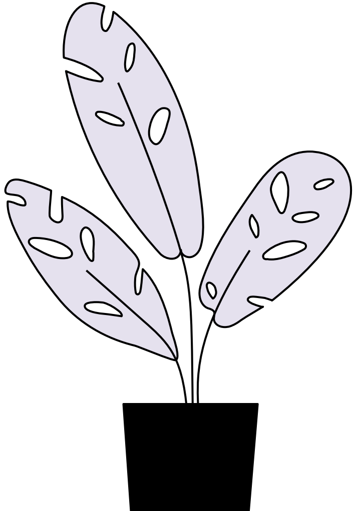
```
In [344]: data[['City', 'Product Name', 'Ship Mode', 'Segment', 'Category']].nunique()
```

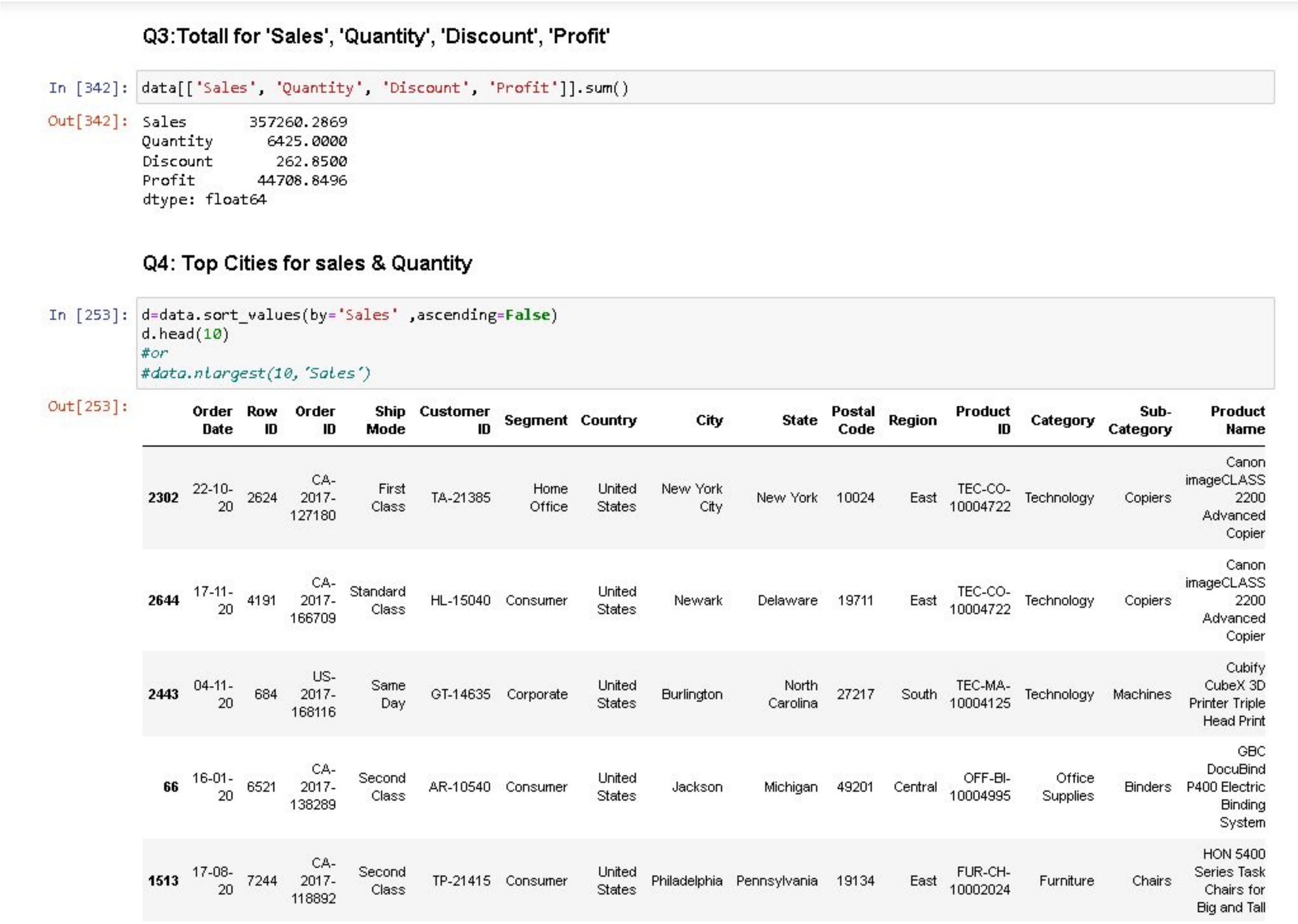
```
Out[344]: City             350
          Product Name    1075
          Ship Mode          4
          Segment            3
          Category           3
          dtype: int64
```

Q2: Max for 'Sales', 'Quantity', 'Discount'

```
In [324]: data[['Sales', 'Quantity', 'Discount']].max()
```

```
Out[324]: Sales       11199.968
          Quantity       14.000
          Discount        0.800
          dtype: float64
```





Q3:Totall for 'Sales', 'Quantity', 'Discount', 'Profit'

In [342]:
```
data[['Sales', 'Quantity', 'Discount', 'Profit']].sum()
```

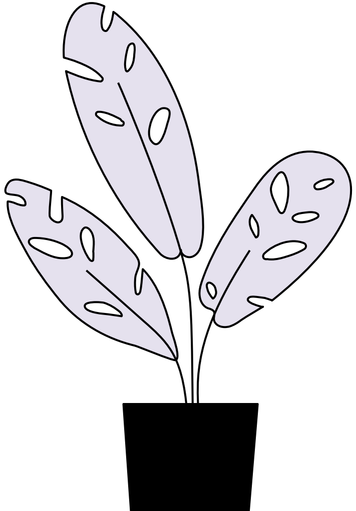
Out[342]:
```
Sales       357260.2869
Quantity      6425.0000
Discount       262.8500
Profit       44708.8496
dtype: float64
```

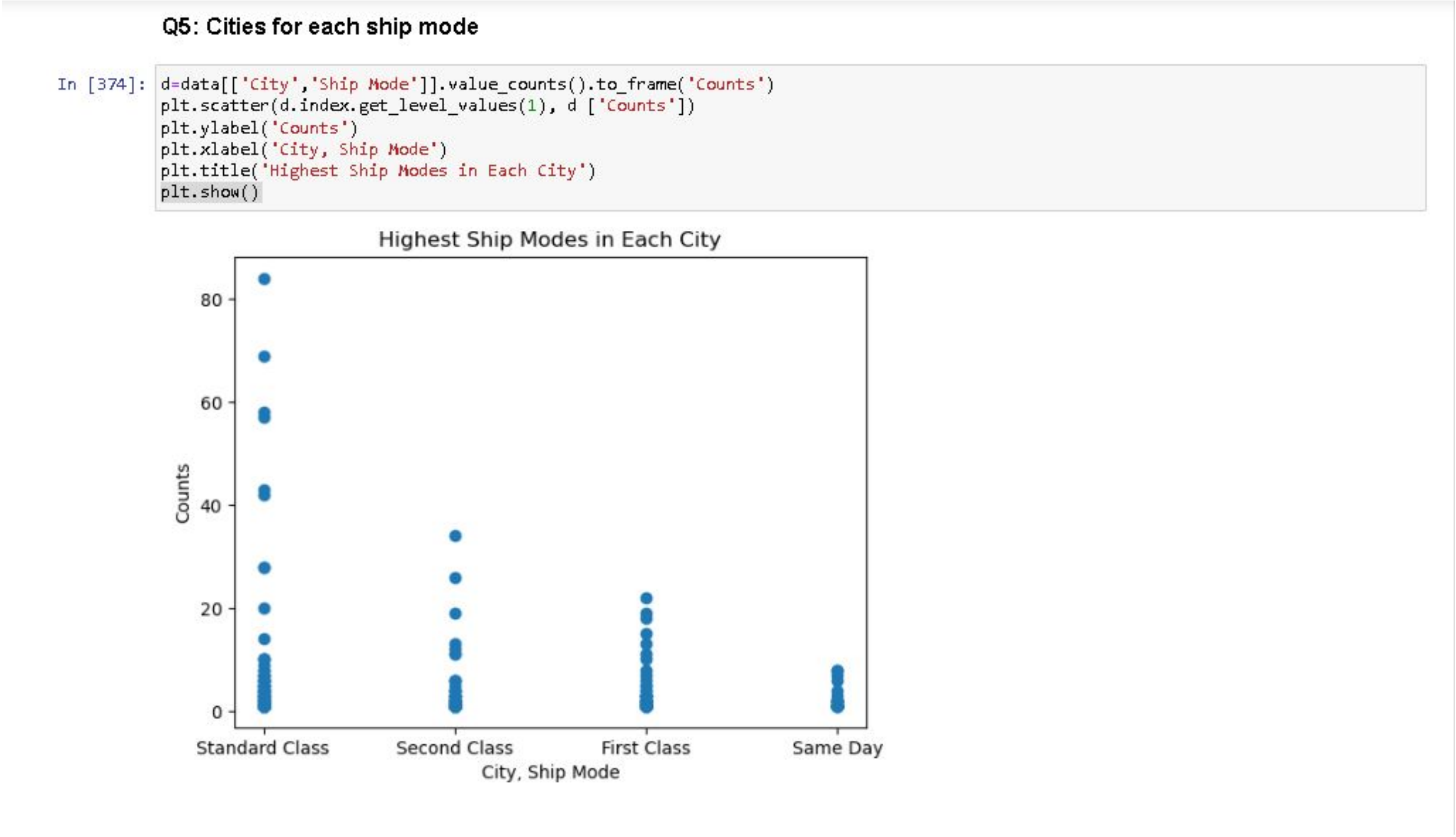
Q4: Top Cities for sales & Quantity

In [253]:
```
d=data.sort_values(by='Sales' ,ascending=False)
d.head(10)
#or
#data.nlargest(10,'Sales')
```

Out[253]:

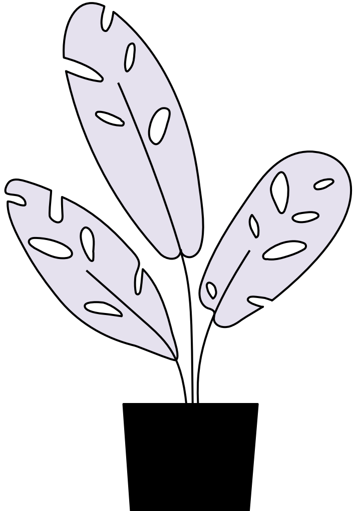
	Order Date	Row ID	Order ID	Ship Mode	Customer ID	Segment	Country	City	State	Postal Code	Region	Product ID	Category	Sub-Category	Product Name
2302	22-10-20	2624	CA-2017-127180	First Class	TA-21385	Home Office	United States	New York City	New York	10024	East	TEC-CO-10004722	Technology	Copiers	Canon imageCLASS 2200 Advanced Copier
2644	17-11-20	4191	CA-2017-166709	Standard Class	HL-15040	Consumer	United States	Newark	Delaware	19711	East	TEC-CO-10004722	Technology	Copiers	Canon imageCLASS 2200 Advanced Copier
2443	04-11-20	684	US-2017-168116	Same Day	GT-14635	Corporate	United States	Burlington	North Carolina	27217	South	TEC-MA-10004125	Technology	Machines	Cubify CubeX 3D Printer Triple Head Print
66	16-01-20	6521	CA-2017-138289	Second Class	AR-10540	Consumer	United States	Jackson	Michigan	49201	Central	OFF-BI-10004995	Office Supplies	Binders	GBC DocuBind P400 Electric Binding System
1513	17-08-20	7244	CA-2017-118892	Second Class	TP-21415	Consumer	United States	Philadelphia	Pennsylvania	19134	East	FUR-CH-10002024	Furniture	Chairs	HON 5400 Series Task Chairs for Big and Tall

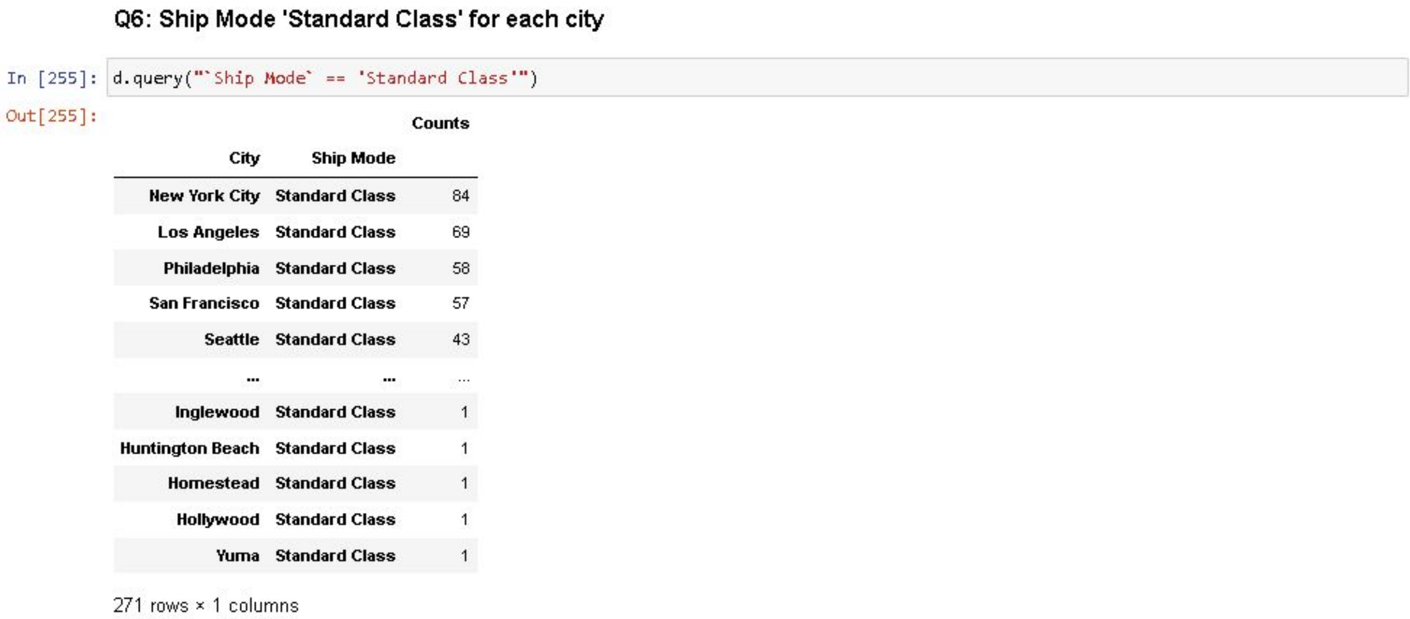




Q5: Cities for each ship mode

```
In [374]: d=data[['City','Ship Mode']].value_counts().to_frame('Counts')
          plt.scatter(d.index.get_level_values(1), d ['Counts'])
          plt.ylabel('Counts')
          plt.xlabel('City, Ship Mode')
          plt.title('Highest Ship Modes in Each City')
          plt.show()
```



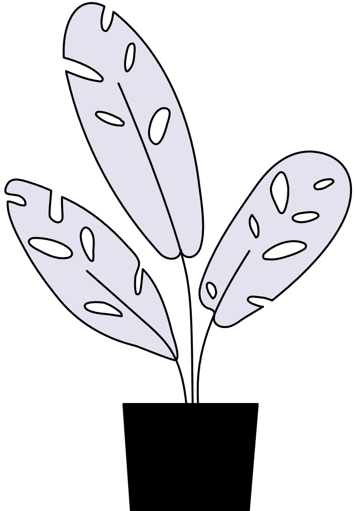


Q6: Ship Mode 'Standard Class' for each city

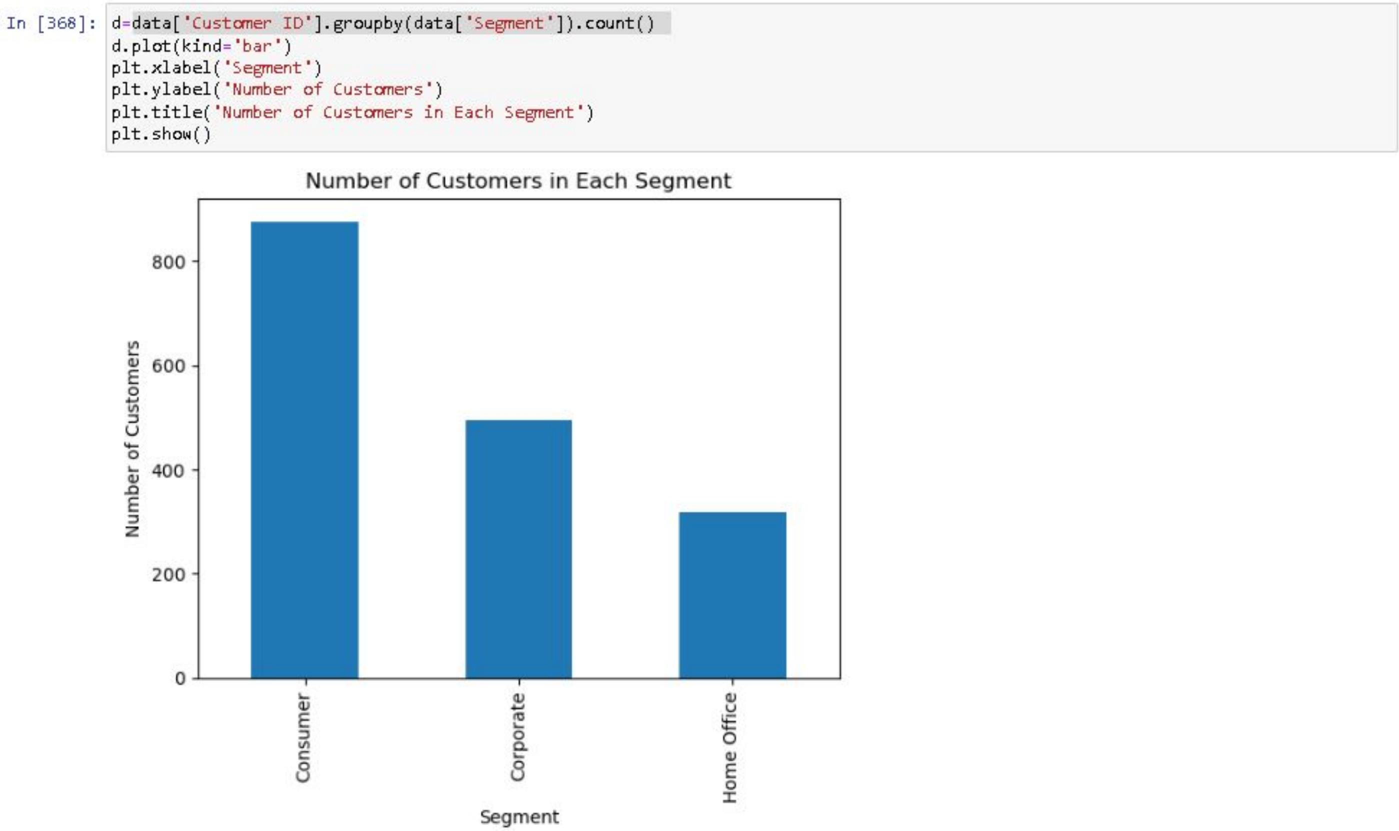
```
In [255]: d.query("`Ship Mode` == 'Standard Class'")
```

```
Out[255]:
                                        Counts
                City       Ship Mode
        New York City  Standard Class       84
          Los Angeles  Standard Class       69
         Philadelphia  Standard Class       58
        San Francisco  Standard Class       57
              Seattle  Standard Class       43
                  ...             ...      ...
            Inglewood  Standard Class        1
     Huntington Beach  Standard Class        1
            Homestead  Standard Class        1
            Hollywood  Standard Class        1
                 Yuma  Standard Class        1

271 rows × 1 columns
```

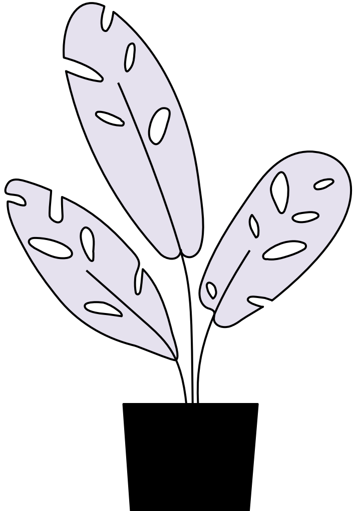


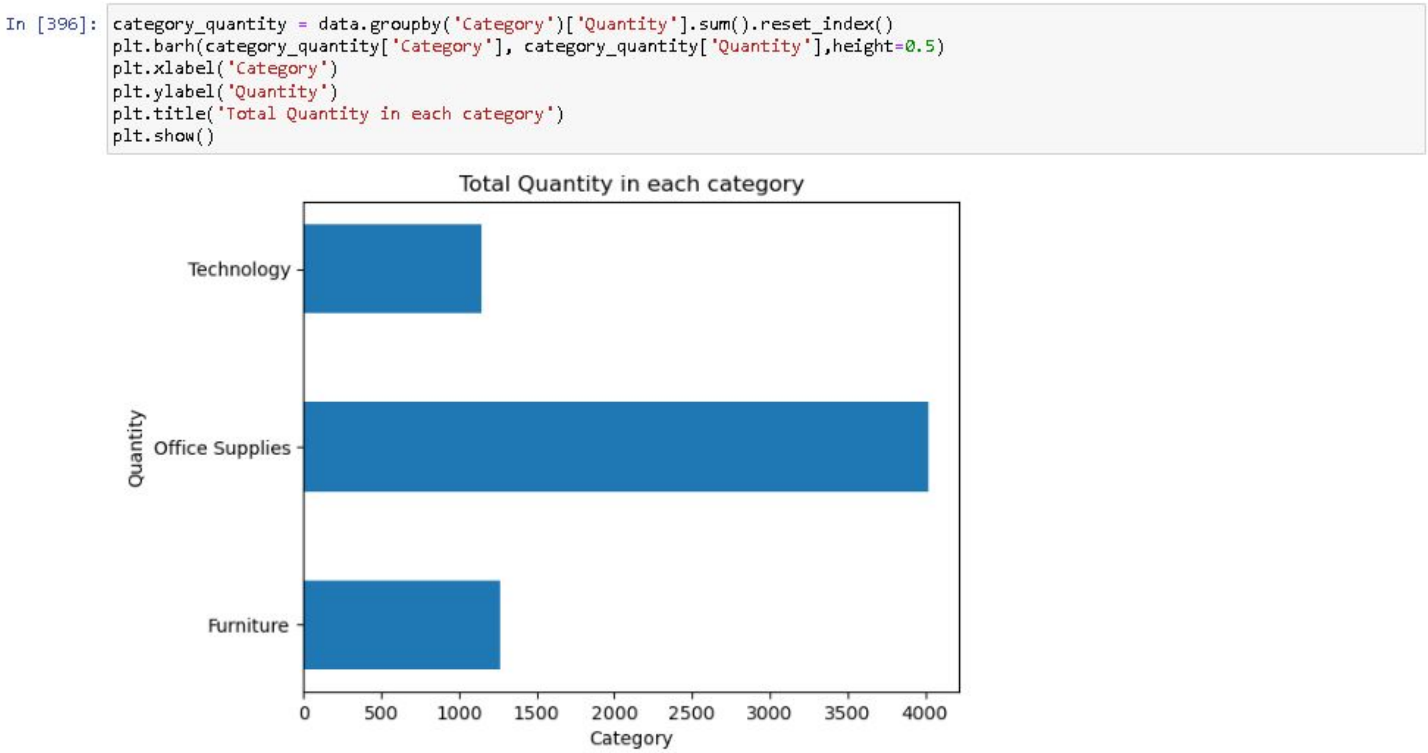
Q7:Segment for total Customer



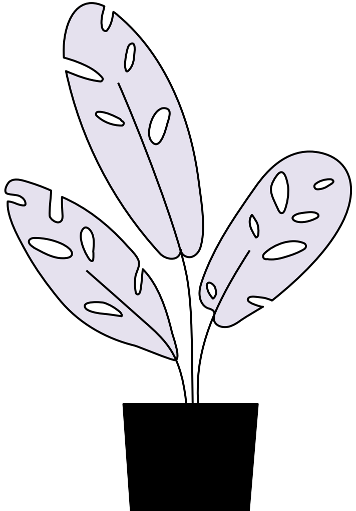
In [368]:

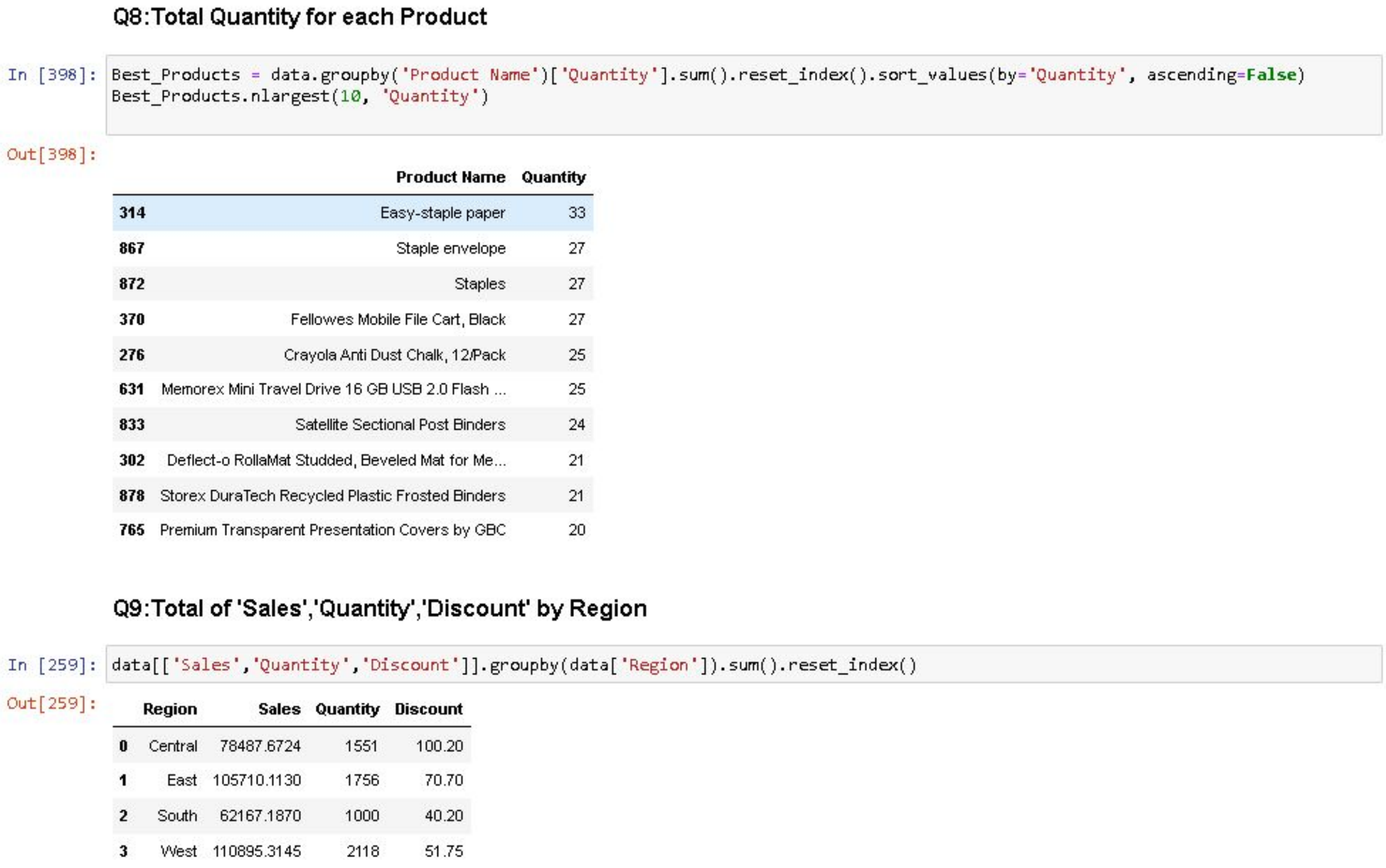
```
d=data['Customer ID'].groupby(data['Segment']).count()
d.plot(kind='bar')
plt.xlabel('Segment')
plt.ylabel('Number of Customers')
plt.title('Number of Customers in Each Segment')
plt.show()
```





```
In [396]: category_quantity = data.groupby('Category')['Quantity'].sum().reset_index()
          plt.barh(category_quantity['Category'], category_quantity['Quantity'],height=0.5)
          plt.xlabel('Category')
          plt.ylabel('Quantity')
          plt.title('Total Quantity in each category')
          plt.show()
```





Q8:Total Quantity for each Product

```
In [398]: Best_Products = data.groupby('Product Name')['Quantity'].sum().reset_index().sort_values(by='Quantity', ascending=False)
          Best_Products.nlargest(10, 'Quantity')
```

Out[398]:

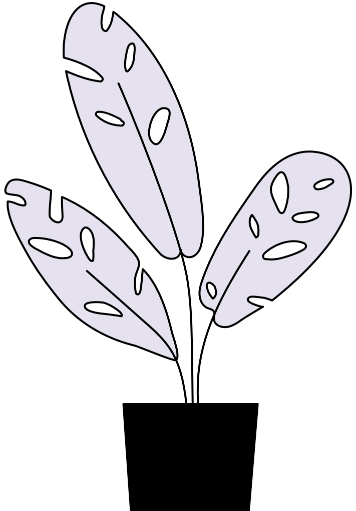
	Product Name	Quantity
314	Easy-staple paper	33
867	Staple envelope	27
872	Staples	27
370	Fellowes Mobile File Cart, Black	27
276	Crayola Anti Dust Chalk, 12/Pack	25
631	Memorex Mini Travel Drive 16 GB USB 2.0 Flash ...	25
833	Satellite Sectional Post Binders	24
302	Deflect-o RollaMat Studded, Beveled Mat for Me...	21
878	Storex DuraTech Recycled Plastic Frosted Binders	21
765	Premium Transparent Presentation Covers by GBC	20

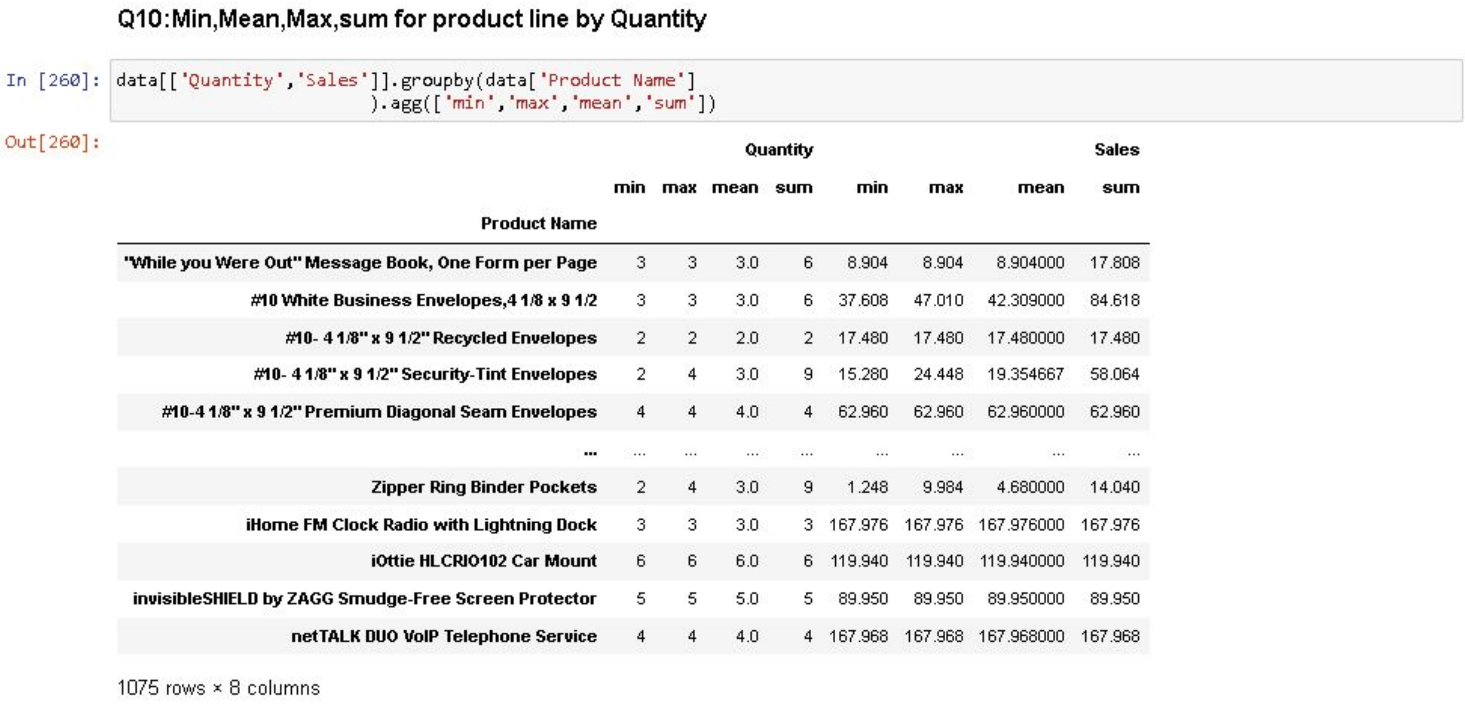
Q9:Total of 'Sales','Quantity','Discount' by Region

```
In [259]: data[['Sales','Quantity','Discount']].groupby(data['Region']).sum().reset_index()
```

Out[259]:

	Region	Sales	Quantity	Discount
0	Central	78487.6724	1551	100.20
1	East	105710.1130	1756	70.70
2	South	62167.1870	1000	40.20
3	West	110895.3145	2118	51.75





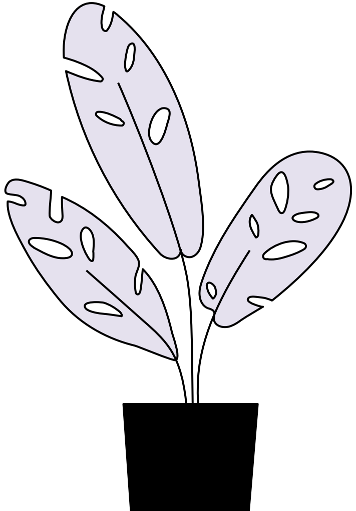
Q10:Min,Mean,Max,sum for product line by Quantity

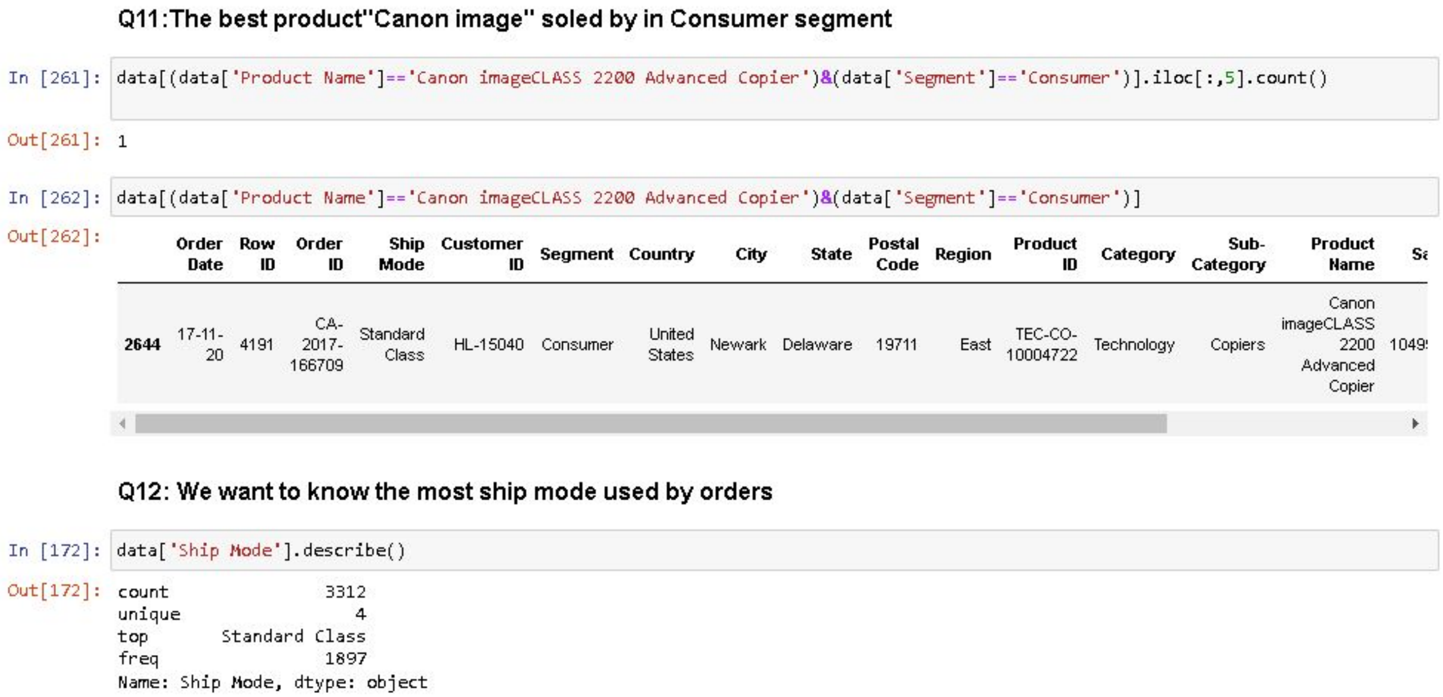
```
In [260]: data[['Quantity','Sales']].groupby(data['Product Name']
                                 ).agg(['min','max','mean','sum'])
```

Out[260]:

	Quantity				Sales			
	min	max	mean	sum	min	max	mean	sum
Product Name								
"While you Were Out" Message Book, One Form per Page	3	3	3.0	6	8.904	8.904	8.904000	17.808
#10 White Business Envelopes,4 1/8 x 9 1/2	3	3	3.0	6	37.608	47.010	42.309000	84.618
#10- 4 1/8" x 9 1/2" Recycled Envelopes	2	2	2.0	2	17.480	17.480	17.480000	17.480
#10- 4 1/8" x 9 1/2" Security-Tint Envelopes	2	4	3.0	9	15.280	24.448	19.354667	58.064
#10-4 1/8" x 9 1/2" Premium Diagonal Seam Envelopes	4	4	4.0	4	62.960	62.960	62.960000	62.960
...	...	...	...	...	...	...	...	...
Zipper Ring Binder Pockets	2	4	3.0	9	1.248	9.984	4.680000	14.040
iHome FM Clock Radio with Lightning Dock	3	3	3.0	3	167.976	167.976	167.976000	167.976
iOttie HLCRIO102 Car Mount	6	6	6.0	6	119.940	119.940	119.940000	119.940
invisibleSHIELD by ZAGG Smudge-Free Screen Protector	5	5	5.0	5	89.950	89.950	89.950000	89.950
netTALK DUO VoIP Telephone Service	4	4	4.0	4	167.968	167.968	167.968000	167.968

1075 rows × 8 columns





Q11:The best product"Canon image" soled by in Consumer segment

```
In [261]: data[(data['Product Name']=='Canon imageCLASS 2200 Advanced Copier')&(data['Segment']=='Consumer')].iloc[:,5].count()

Out[261]: 1
```

```
In [262]: data[(data['Product Name']=='Canon imageCLASS 2200 Advanced Copier')&(data['Segment']=='Consumer')]
```

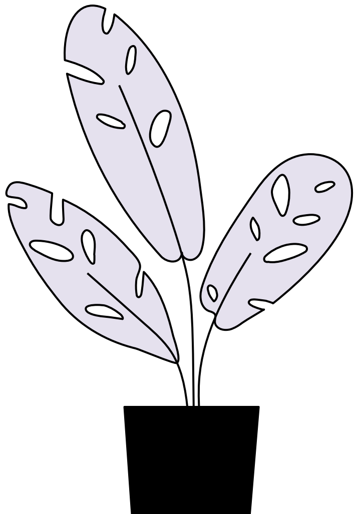
Out[262]:

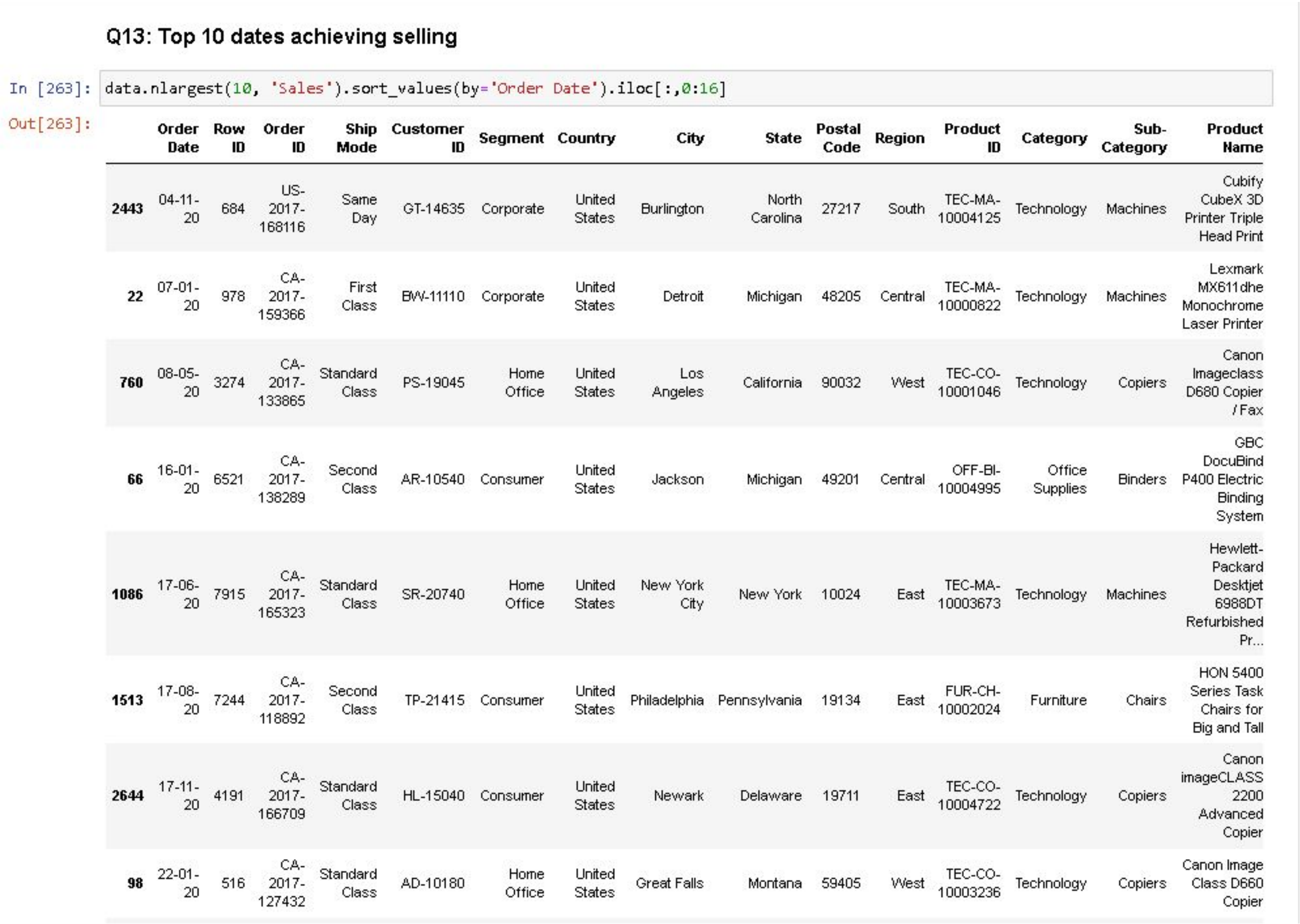
	Order Date	Row ID	Order ID	Ship Mode	Customer ID	Segment	Country	City	State	Postal Code	Region	Product ID	Category	Sub-Category	Product Name	Sa
2644	17-11-20	4191	CA-2017-166709	Standard Class	HL-15040	Consumer	United States	Newark	Delaware	19711	East	TEC-CO-10004722	Technology	Copiers	Canon imageCLASS 2200 Advanced Copier	1049

Q12: We want to know the most ship mode used by orders

```
In [172]: data['Ship Mode'].describe()

Out[172]: count                3312
          unique                  4
          top        Standard Class
          freq                 1897
          Name: Ship Mode, dtype: object
```

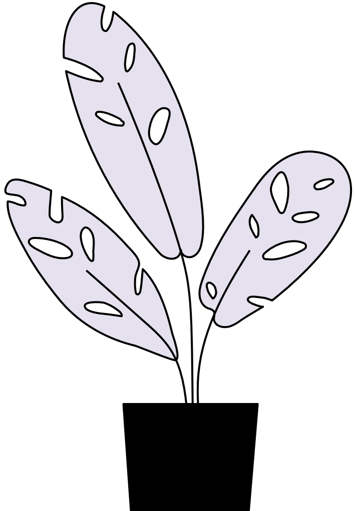


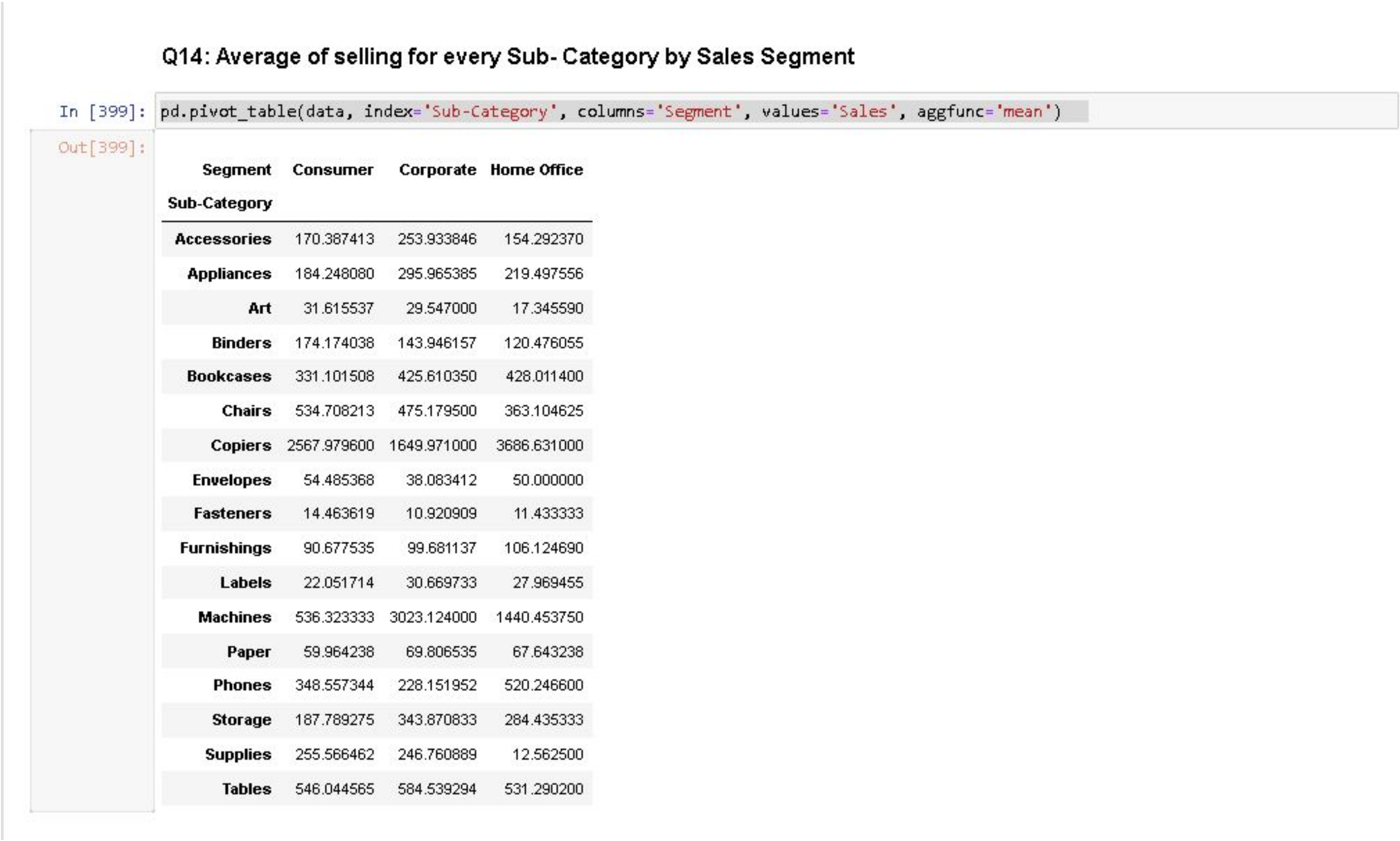


Q13: Top 10 dates achieving selling

```
In [263]: data.nlargest(10, 'Sales').sort_values(by='Order Date').iloc[:,0:16]
Out[263]:
```

	Order Date	Row ID	Order ID	Ship Mode	Customer ID	Segment	Country	City	State	Postal Code	Region	Product ID	Category	Sub-Category	Product Name
2443	04-11-20	684	US-2017-168116	Same Day	GT-14635	Corporate	United States	Burlington	North Carolina	27217	South	TEC-MA-10004125	Technology	Machines	Cubify CubeX 3D Printer Triple Head Print
22	07-01-20	978	CA-2017-159366	First Class	BW-11110	Corporate	United States	Detroit	Michigan	48205	Central	TEC-MA-10000822	Technology	Machines	Lexmark MX611dhe Monochrome Laser Printer
760	08-05-20	3274	CA-2017-133865	Standard Class	PS-19045	Home Office	United States	Los Angeles	California	90032	West	TEC-CO-10001046	Technology	Copiers	Canon Imageclass D680 Copier / Fax
66	16-01-20	6521	CA-2017-138289	Second Class	AR-10540	Consumer	United States	Jackson	Michigan	49201	Central	OFF-BI-10004995	Office Supplies	Binders	GBC DocuBind P400 Electric Binding System
1086	17-06-20	7915	CA-2017-165323	Standard Class	SR-20740	Home Office	United States	New York City	New York	10024	East	TEC-MA-10003673	Technology	Machines	Hewlett-Packard Desktjet 6988DT Refurbished Pr...
1513	17-08-20	7244	CA-2017-118892	Second Class	TP-21415	Consumer	United States	Philadelphia	Pennsylvania	19134	East	FUR-CH-10002024	Furniture	Chairs	HON 5400 Series Task Chairs for Big and Tall
2644	17-11-20	4191	CA-2017-166709	Standard Class	HL-15040	Consumer	United States	Newark	Delaware	19711	East	TEC-CO-10004722	Technology	Copiers	Canon imageCLASS 2200 Advanced Copier
98	22-01-20	516	CA-2017-127432	Standard Class	AD-10180	Home Office	United States	Great Falls	Montana	59405	West	TEC-CO-10003236	Technology	Copiers	Canon Image Class D660 Copier



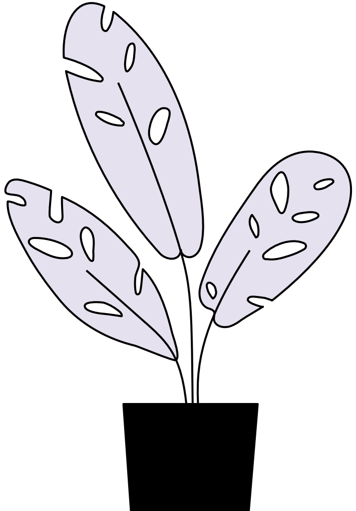


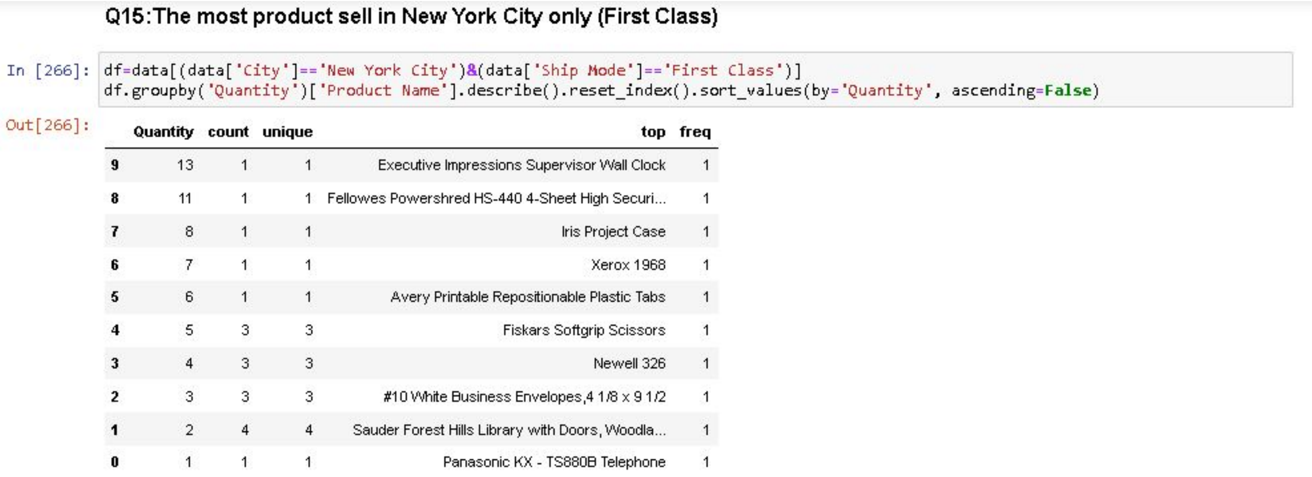
Q14: Average of selling for every Sub- Category by Sales Segment

```
In [399]: pd.pivot_table(data, index='Sub-Category', columns='Segment', values='Sales', aggfunc='mean')
```

Out[399]:

Segment	Consumer	Corporate	Home Office
Sub-Category			
Accessories	170.387413	253.933846	154.292370
Appliances	184.248080	295.965385	219.497556
Art	31.615537	29.547000	17.345590
Binders	174.174038	143.946157	120.476055
Bookcases	331.101508	425.610350	428.011400
Chairs	534.708213	475.179500	363.104625
Copiers	2567.979600	1649.971000	3686.631000
Envelopes	54.485368	38.083412	50.000000
Fasteners	14.463619	10.920909	11.433333
Furnishings	90.677535	99.681137	106.124690
Labels	22.051714	30.669733	27.969455
Machines	536.323333	3023.124000	1440.453750
Paper	59.964238	69.806535	67.643238
Phones	348.557344	228.151952	520.246600
Storage	187.789275	343.870833	284.435333
Supplies	255.566462	246.760889	12.562500
Tables	546.044565	584.539294	531.290200



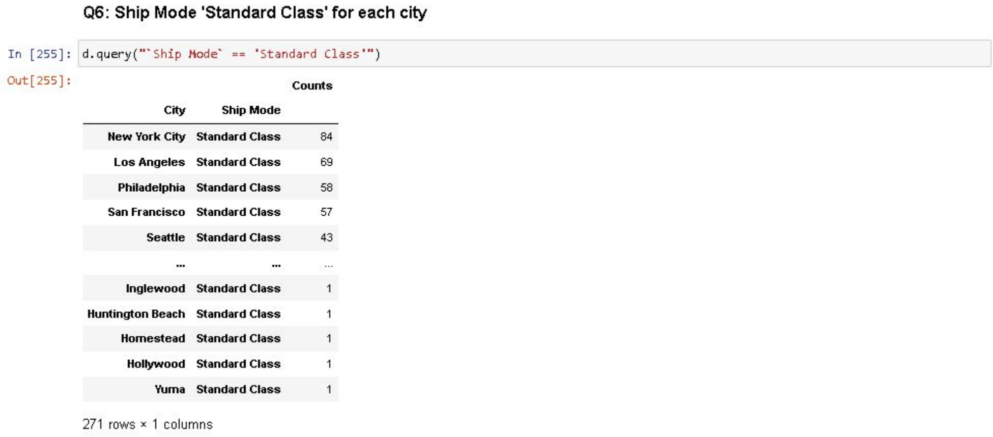


Q15:The most product sell in New York City only (First Class)

```
In [266]: df=data[(data['City']=='New York City')&(data['Ship Mode']=='First Class')]
          df.groupby('Quantity')['Product Name'].describe().reset_index().sort_values(by='Quantity', ascending=False)
```

Out[266]:

	Quantity	count	unique	top	freq
9	13	1	1	Executive Impressions Supervisor Wall Clock	1
8	11	1	1	Fellowes Powershred HS-440 4-Sheet High Securi...	1
7	8	1	1	Iris Project Case	1
6	7	1	1	Xerox 1968	1
5	6	1	1	Avery Printable Repositionable Plastic Tabs	1
4	5	3	3	Fiskars Softgrip Scissors	1
3	4	3	3	Newell 326	1
2	3	3	3	#10 White Business Envelopes,4 1/8 x 9 1/2	1
1	2	4	4	Sauder Forest Hills Library with Doors, Woodla...	1
0	1	1	1	Panasonic KX - TS880B Telephone	1



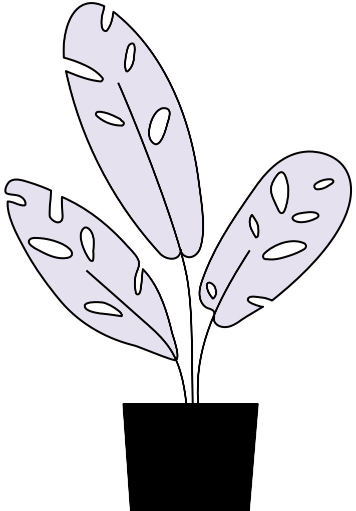
Q6: Ship Mode 'Standard Class' for each city

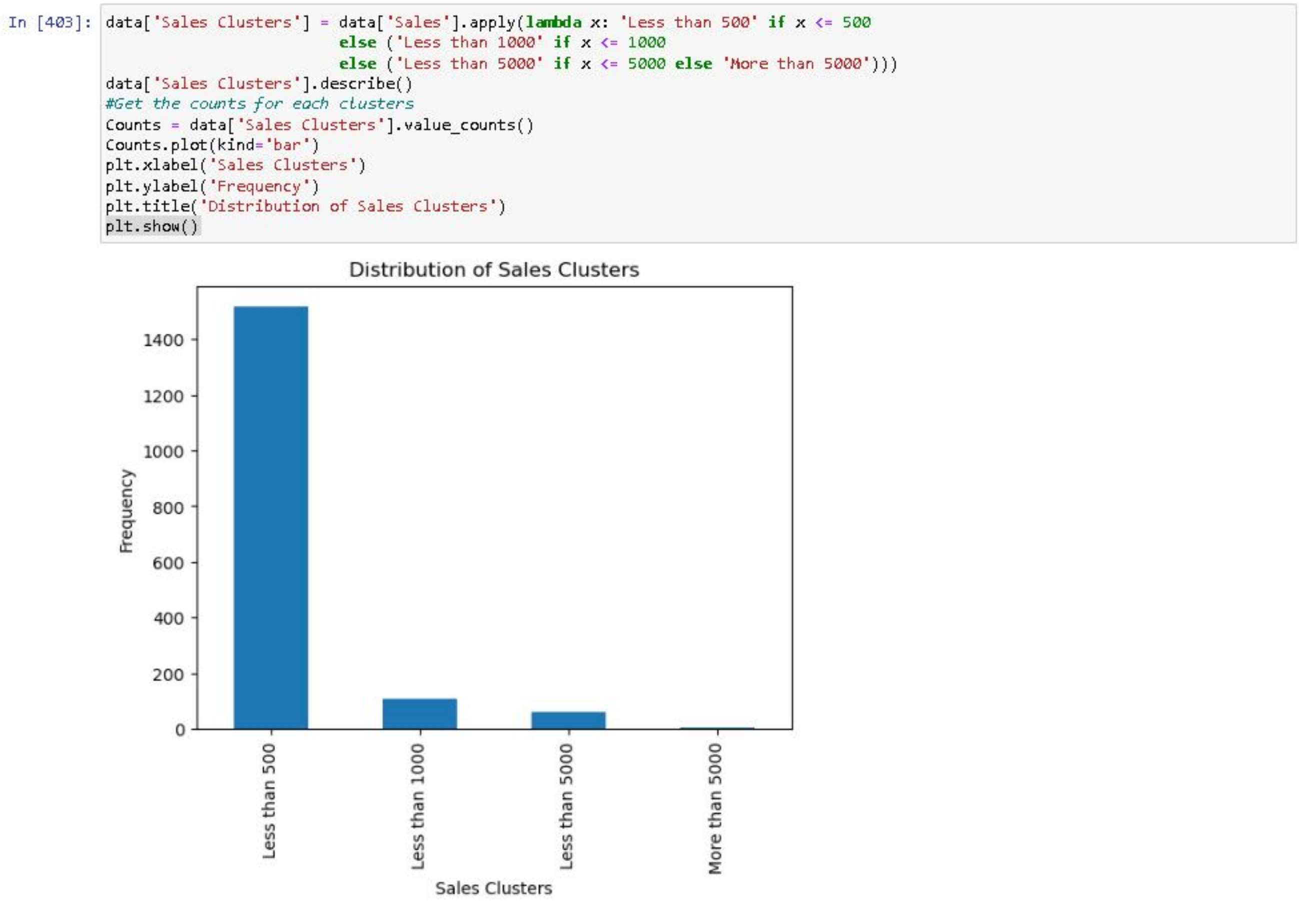
```
In [255]: d.query("`Ship Mode` == 'Standard Class'")
```

Out[255]:

		Counts
City	**Ship Mode**	
New York City	**Standard Class**	84
Los Angeles	**Standard Class**	69
Philadelphia	**Standard Class**	58
San Francisco	**Standard Class**	57
Seattle	**Standard Class**	43
...	**...**	...
Inglewood	**Standard Class**	1
Huntington Beach	**Standard Class**	1
Homestead	**Standard Class**	1
Hollywood	**Standard Class**	1
Yuma	**Standard Class**	1

271 rows × 1 columns





```
In [403]: data['Sales Clusters'] = data['Sales'].apply(lambda x: 'Less than 500' if x <= 500
                                        else ('Less than 1000' if x <= 1000
                                        else ('Less than 5000' if x <= 5000 else 'More than 5000')))
          data['Sales Clusters'].describe()
          #Get the counts for each clusters
          Counts = data['Sales Clusters'].value_counts()
          Counts.plot(kind='bar')
          plt.xlabel('Sales Clusters')
          plt.ylabel('Frequency')
          plt.title('Distribution of Sales Clusters')
          plt.show()
```