# Project & Presentation

April 16, 2022

[2]:
```
import pandas as pd
Orders = pd.read_excel("Sample - Superstore (1).xlsx",index_col="Row␣
↪ID",sheet_name="Orders")
```

## 1 Categorical Columns

[48]:
```
categorical_columns = [cname for cname in Orders.columns if Orders[cname].dtype␣
↪== "object"]
categorical_columns
```

[48]:
```
['Order ID',
 'Ship Mode',
 'Customer ID',
 'Customer Name',
 'Segment',
 'Country',
 'City',
 'State',
 'Region',
 'Product ID',
 'Category',
 'Sub-Category',
 'Product Name']
```

## 2 overview of the data

[7]:
```
Orders.describe()
```

[7]:

| | Postal Code | Sales | Quantity | Discount | Profit |
|---|---|---|---|---|---|
| count | 9994.000000 | 9994.000000 | 9994.000000 | 9994.000000 | 9994.000000 |
| mean | 55190.379428 | 229.858001 | 3.789574 | 0.156203 | 28.656896 |
| std | 32063.693350 | 623.245101 | 2.225110 | 0.206452 | 234.260108 |
| min | 1040.000000 | 0.444000 | 1.000000 | 0.000000 | -6599.978000 |
| 25% | 23223.000000 | 17.280000 | 2.000000 | 0.000000 | 1.728750 |
| 50% | 56430.500000 | 54.490000 | 3.000000 | 0.200000 | 8.666500 |

```
75%     90008.000000    209.940000     5.000000     0.200000    29.364000
max     99301.000000  22638.480000    14.000000     0.800000  8399.976000
```

## 3 Unique values in different columns

[ ]:
```
len(Orders['City'].unique())
len(Orders['State'].unique())
len(Orders['Country'].unique())
len(Orders['Region'].unique())
len(Orders['Product Name'].unique())
len(Orders['Category'].unique())
len(Orders['Sub-Category'].unique())
```

## 4 Most Profit Making Cities

[61]:
```
Orders.groupby(['City'])['Profit'].sum().sort_values(ascending = False).head()
```

[61]:
```
City
New York City    62036.9837
Los Angeles      30440.7579
Seattle          29156.0967
San Francisco    17507.3854
Detroit          13181.7908
Name: Profit, dtype: float64
```

## 5 Least Profit Making Cities

[62]:
```
Orders.groupby(['City'])['Profit'].sum().sort_values(ascending = False).tail()
```

[62]:
```
City
Chicago         -6654.5688
Lancaster       -7239.0684
San Antonio     -7299.0502
Houston        -10153.5485
Philadelphia   -13837.7674
Name: Profit, dtype: float64
```

## 6 Cities in which sales are highest

[57]:
```
Orders.groupby(['City'])['Sales'].sum().sort_values(ascending = False).head()
```

[57]:
```
City
New York City    256368.161
```

```
Los Angeles      175851.341
Seattle          119540.742
San Francisco    112669.092
Philadelphia     109077.013
Name: Sales, dtype: float64
```

## 7 Cities in which sales are lowest

[63]:
```
Orders.groupby(['City'])['Sales'].sum().sort_values(ascending = False).tail()
```

[63]:
```
City
Ormond Beach    2.808
Pensacola       2.214
Jupiter         2.064
Elyria          1.824
Abilene         1.392
Name: Sales, dtype: float64
```

## 8 Most Profitable Sub-Categories

[69]:
```
Orders.groupby(['Sub-Category'])['Profit'].sum().sort_values(ascending = False).
↪head()
```

[69]:
```
Sub-Category
Copiers        55617.8249
Phones         44515.7306
Accessories    41936.6357
Paper          34053.5693
Binders        30221.7633
Name: Profit, dtype: float64
```

## 9 Least Profitable Sub-Categories

[67]:
```
Orders.groupby(['Sub-Category'])['Profit'].sum().sort_values(ascending = False).
↪tail()
```

[67]:
```
Sub-Category
Machines      3384.7569
Fasteners      949.5182
Supplies     -1189.0995
Bookcases    -3472.5560
Tables      -17725.4811
Name: Profit, dtype: float64
```

## 10 Profit by Categories

[68]:
```
Orders.groupby(['Category'])['Profit'].sum().sort_values(ascending = False)
```

[68]:
```
Category
Technology         145454.9481
Office Supplies    122490.8008
Furniture           18451.2728
Name: Profit, dtype: float64
```

## 11 Profit by Segments

[4]:
```
Orders.groupby(['Segment'])['Profit'].sum().sort_values(ascending = False)
```

[4]:
```
Segment
Consumer       134119.2092
Corporate       91979.1340
Home Office     60298.6785
Name: Profit, dtype: float64
```

## 12 Converted order date into days month years

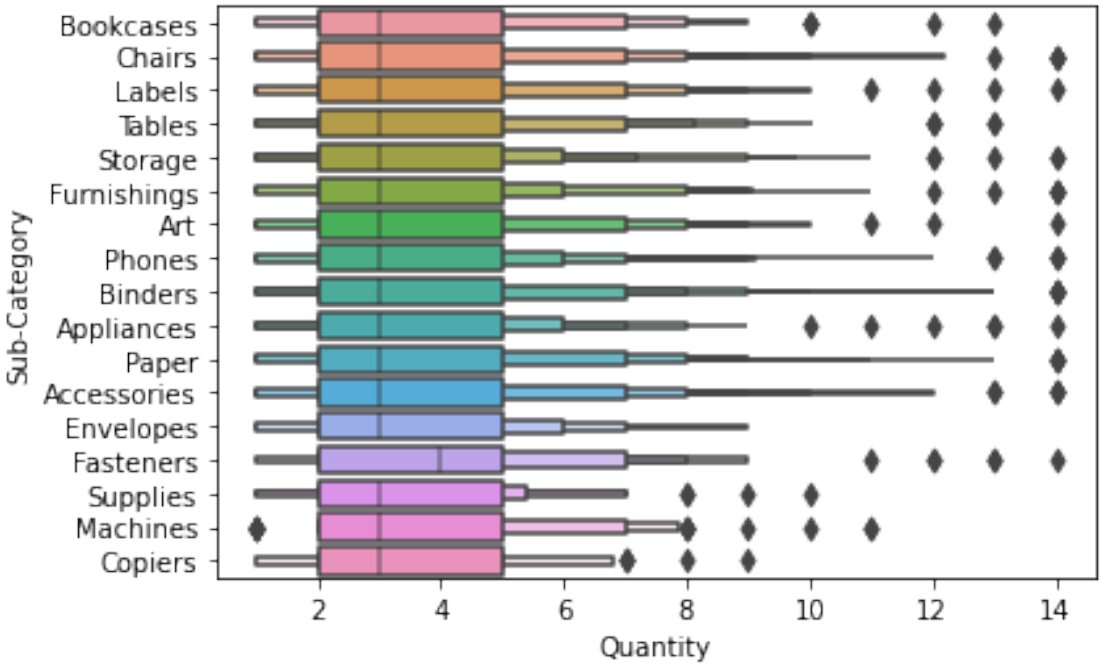
[15]:
```
Orders['Order Date'] = pd.to_datetime(Orders['Order Date'])
Orders['day'] = (Orders['Order Date']).dt.day
Orders['month'] = (Orders['Order Date']).dt.month
Orders['year'] = (Orders['Order Date']).dt.year
Orders['year'].unique()
```

[15]:
```
array([2016, 2015, 2014, 2017], dtype=int64)
```

## 13 Which sub-category has most quantity sold

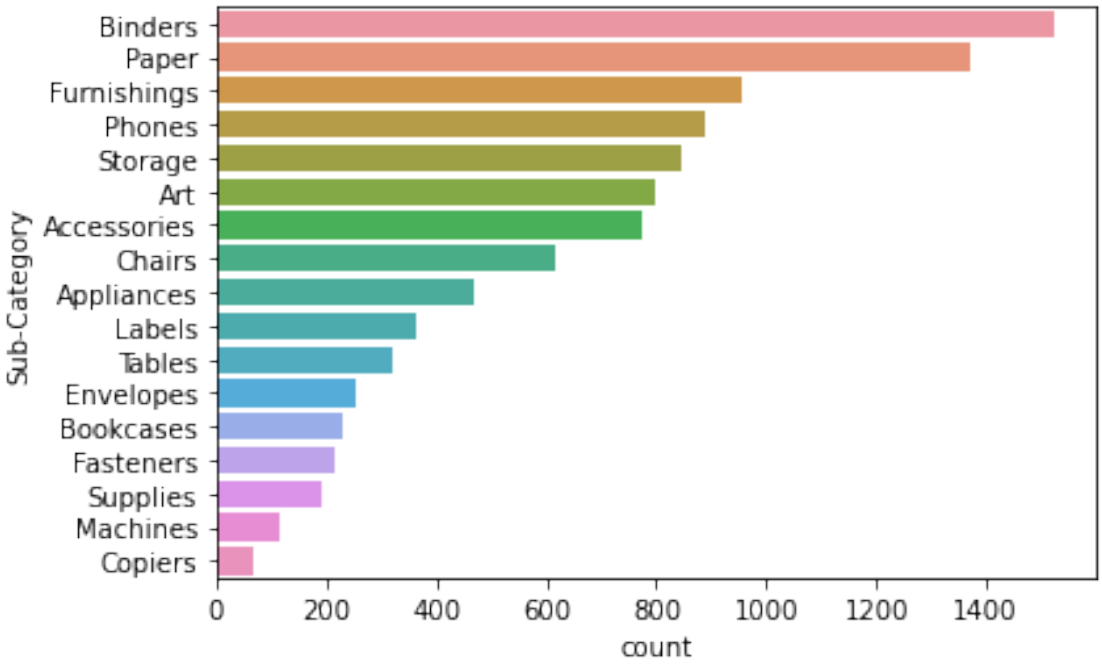
[30]:
```
import seaborn as sns
sns.boxenplot(y = 'Sub-Category', x = 'Quantity', data=Orders )
```

[30]:
```
<AxesSubplot:xlabel='Quantity', ylabel='Sub-Category'>
```




[34]:
```
sns.countplot(y = 'Sub-Category', data=Orders, order = Orders['Sub-Category'].
↪value_counts().index )
```

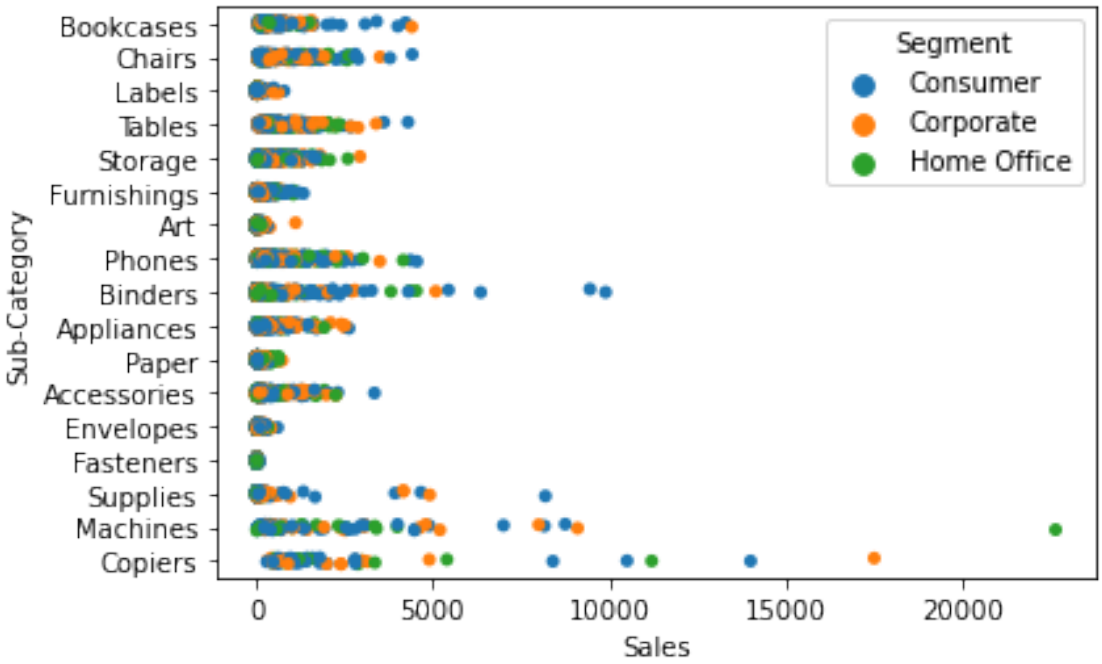
[34]: <AxesSubplot:xlabel='count', ylabel='Sub-Category'>



## 14 Demand according to segments

[38]:
```
sns.stripplot(y = 'Sub-Category', x = 'Sales', hue = 'Segment', data=Orders )
```

[38]:
```
<AxesSubplot:xlabel='Sales', ylabel='Sub-Category'>
```



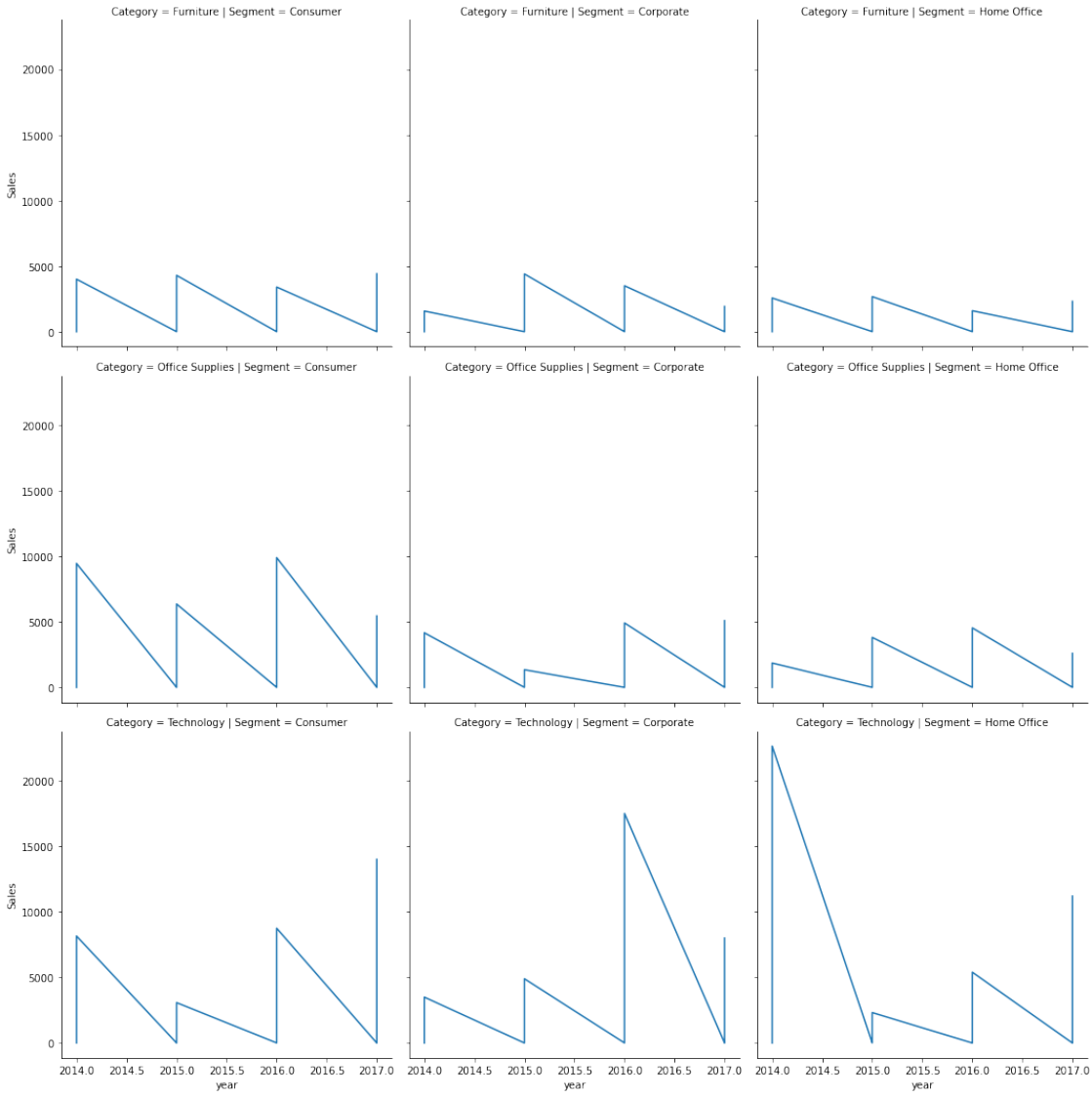

## 15 Profit according to sub categories

[40]:
```
sns.relplot(y = 'Sub-Category', x = 'Profit', data=Orders )
```

[40]:
```
<seaborn.axisgrid.FacetGrid at 0x21a3d522700>
```


## 16 yearly sales w.r.t categories and segments

[45]:
```
productCount  = sns.relplot(x="year",  y = 'Sales', col= 'Segment' , row=␣
↪'Category', estimator = None, kind="line", data =Orders)
```


Category = Furniture | Segment = Consumer
Category = Furniture | Segment = Corporate
Category = Furniture | Segment = Home Office
Category = Office Supplies | Segment = Consumer
Category = Office Supplies | Segment = Corporate
Category = Office Supplies | Segment = Home Office
Category = Technology | Segment = Consumer
Category = Technology | Segment = Corporate
Category = Technology | Segment = Home Office
Sales
0
5000
10000
15000
20000
year
2014.0
2014.5
2015.0
2015.5
2016.0
2016.5
2017.0