

```
[1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

[3]: df = pd.read_csv('Superstore.csv',encoding='latin1')

[4]: df.head()
```

Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Product ID	Category	Sub-Category	Product Name	Sales	Quantity	Discount	Profit
Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-BO-10001798	Furniture	Bookcases	Bush Somerset Collection Bookcase	261.9600	2	0.00	41.9136
Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-CH-10000454	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs,...	731.9400	3	0.00	219.5820
Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	90036	West	OFF-LA-10000240	Office Supplies	Labels	Self-Adhesive Address Labels for Typewriters b...	14.6200	2	0.00	6.8714
Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	FUR-TA-10000577	Furniture	Tables	Bretford CR4500 Series Slim Rectangular Table	957.5775	5	0.45	-383.0310
Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	OFF-ST-10000760	Office Supplies	Storage	Eldon Fold 'N Roll Cart System	22.3680	2	0.20	2.5164

Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Product ID	Category	Sub-Category	
0	1	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-BO-10001798	Furniture	Bookcases
1	2	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-CH-10000454	Furniture	Chairs
2	3	CA-2016-138688	6/12/2016	6/16/2016	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	90036	West	OFF-LA-10000240	Office Supplies	Labels
3	4	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	FUR-TA-10000577	Furniture	Tables
4	5	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	OFF-ST-10000760	Office Supplies	Storage
...
9989	9990	CA-2014-110422	1/21/2014	1/23/2014	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States	Miami	...	33180	South	FUR-FU-10001889	Furniture	Furnishings
9990	9991	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	92627	West	FUR-FU-10000747	Furniture	Furnishings
9991	9992	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	92627	West	TEC-PH-10003645	Technology	Phones
9992	9993	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	92627	West	OFF-PA-10004041	Office Supplies	Paper
9993	9994	CA-2017-119914	5/4/2017	5/9/2017	Second Class	CC-12220	Chris Cortes	Consumer	United States	Westminster	...	92683	West	OFF-AP-10002684	Office Supplies	Appliances

9994 rows × 21 columns

```
[7]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Row ID      9994 non-null   int64  
 1   Order ID    9994 non-null   object  
 2   Order Date  9994 non-null   object  
 3   Ship Date   9994 non-null   object  
 4   Ship Mode   9994 non-null   object  
 5   Customer ID 9994 non-null   object  
 6   Customer Name 9994 non-null   object  
 7   Segment     9994 non-null   object  
 8   Country     9994 non-null   object  
 9   City        9994 non-null   object  
 10  State        9994 non-null   object  
 11  Postal Code 9994 non-null   int64  
 12  Region      9994 non-null   object  
 13  Product ID  9994 non-null   object  
 14  Category    9994 non-null   object  
 15  Sub-Category 9994 non-null   object  
 16  Product Name 9994 non-null   object  
 17  Sales        9994 non-null   float64 
 18  Quantity    9994 non-null   int64  
 19  Discount    9994 non-null   float64 
 20  Profit       9994 non-null   float64 
dtypes: float64(3), int64(3), object(15)
memory usage: 1.6+ MB

[8]: df['Order Date'] = pd.to_datetime(df['Order Date'])
df['Ship Date'] = pd.to_datetime(df['Ship Date'])

[9]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Row ID      9994 non-null   int64  
 1   Order ID    9994 non-null   object  
 2   Order Date  9994 non-null   datetime64[ns] 
 3   Ship Date   9994 non-null   datetime64[ns] 
 4   Ship Mode   9994 non-null   object  
 5   Customer ID 9994 non-null   object  
 6   Customer Name 9994 non-null   object  
 7   Segment     9994 non-null   object  
 8   Country     9994 non-null   object  
 9   City        9994 non-null   object  
 10  State        9994 non-null   object  
 11  Postal Code 9994 non-null   int64  
 12  Region      9994 non-null   object  
 13  Product ID  9994 non-null   object  
 14  Category    9994 non-null   object  
 15  Sub-Category 9994 non-null   object  
 16  Product Name 9994 non-null   object  
 17  Sales        9994 non-null   float64 
 18  Quantity    9994 non-null   int64  
 19  Discount    9994 non-null   float64 
 20  Profit       9994 non-null   float64 
dtypes: datetime64[ns](2), float64(3), int64(3), object(13)
memory usage: 1.6+ MB

[11]: df.describe()

[11]:   Row ID      Order Date      Ship Date  Postal Code    Sales  Quantity  Discount  Profit
  count  9994.000000  9994.000000  9994.000000  9994.000000  9994.000000  9994.000000  9994.000000
  mean   4997.500000  2016-04-30 00:07:12.259355648  2016-05-03 23:06:58.571142912  55190.379428  229.858001  3.789574  0.156203  28.656896
  min    1.000000   2014-01-03 00:00:00  2014-01-07 00:00:00  1040.000000  0.444400  1.000000  0.000000 -6599.978000
  25%   2499.250000   2015-05-23 00:00:00  2015-05-27 00:00:00  23223.000000  17.280000  2.000000  0.000000  1.728750
  50%   4997.500000   2016-06-26 00:00:00  2016-06-29 00:00:00  56430.500000  54.490000  3.000000  0.200000  8.666500
  75%   7495.750000   2017-05-14 00:00:00  2017-05-18 00:00:00  90008.000000  209.940000  5.000000  0.200000  29.364000
  max   9994.000000   2017-12-30 00:00:00  2018-01-05 00:00:00  99301.000000  22638.480000  14.000000  0.800000  8399.976000
  std   2885.163629          NaN          NaN  32063.693350  623.245101  2.225110  0.206452  234.260108

[12]: df.duplicated().sum()

[12]: np.int64(0)

[13]: df[df['Profit'] == df['Profit'].min()]

[13]:   Row ID      Order ID      Order Date    Ship Date  Ship Mode  Customer ID  Customer Name  Segment  Country  City  ...  Postal Code  Region  Product ID  Category  Sub-Category  Product Name  Sales
  7772  7773      CA-2016-108196  2016-11-25  2016-12-02  Standard Class  CS-12505  Cindy Stewart  Consumer  United States  Lancaster  ...  43130  East  TEC-MA-10000418  Technology  Machines  Cubify CubeX  3D Printer Double Head Print
  1 rows x 21 columns

[14]: df.loc[df['Profit'].idxmin()]

[14]:   Row ID      Order ID      Order Date    Ship Date  Ship Mode  Customer ID  Customer Name  Segment  Country  City  ...  Postal Code  Region  Product ID  Category  Sub-Category  Product Name  Sales
  7773      CA-2016-108196  2016-11-25  2016-12-02  Standard Class  CS-12505  Cindy Stewart  Consumer  United States  Lancaster  ...  43130  East  TEC-MA-10000418  Technology  Machines  Cubify CubeX  3D Printer Double Head Print
  1 rows x 21 columns
```

```

Postal Code 43130
Region East
Product ID TEC-MA-10000418
Category Technology
Sub-Category Machines
Product Name Cubify CubeX 3D Printer Double Head Print
Sales 4499.985
Quantity 5
Discount 0.7
Profit -6599.978
Name: 7772, dtype: object

```

```
[15]: df.sort_values(by='Profit', ascending=True).head(10)
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Product ID	Category	Sub-Category	Product Name
7772	7773	CA-2016-108196	2016-11-25	2016-12-02	Standard Class	CS-12505	Cindy Stewart	Consumer	United States	Lancaster	...	43130	East	TEC-MA-10000418	Technology	Machines	Cubify CubeX 3D Printer Double Head Print
683	684	US-2017-168116	2017-11-04	2017-11-04	Same Day	GT-14635	Grant Thornton	Corporate	United States	Burlington	...	27217	South	TEC-MA-10004125	Technology	Machines	Cubify CubeX 3D Printer Triple Head Print
9774	9775	CA-2014-169019	2014-07-26	2014-07-30	Standard Class	LF-17185	Luke Foster	Consumer	United States	San Antonio	...	78207	Central	OFF-BI-10004995	Office Supplies	Binders	GBC DocuBind P400 Electric Binding System
3011	3012	CA-2017-134845	2017-04-17	2017-04-23	Standard Class	SR-20425	Sharelle Roach	Home Office	United States	Louisville	...	80027	West	TEC-MA-10000822	Technology	Machines	Lexmark MX611dhe Monochrome Laser Printer
4991	4992	US-2017-122714	2017-12-07	2017-12-13	Standard Class	HG-14965	Henry Goldwyn	Corporate	United States	Chicago	...	60653	Central	OFF-BI-10001120	Office Supplies	Binders	Ibico EPK-21 Electric Binding System
3151	3152	CA-2015-147830	2015-12-15	2015-12-18	First Class	NF-18385	Natalie Fritzler	Consumer	United States	Newark	...	43055	East	TEC-MA-10000418	Technology	Machines	Cubify CubeX 3D Printer Double Head Print
5310	5311	CA-2017-131254	2017-11-19	2017-11-21	First Class	NC-18415	Nathan Cano	Consumer	United States	Houston	...	77095	Central	OFF-BI-10003527	Office Supplies	Binders	Fellowes PB500 Electric Punch Plastic Comb Bin...
9639	9640	CA-2015-116638	2015-01-28	2015-01-31	Second Class	JH-15985	Joseph Holt	Consumer	United States	Concord	...	28027	South	FUR-TA-10000198	Furniture	Tables	Chromcraft Bull-Nose Wood Oval Conference Tabl...
1199	1200	CA-2016-130946	2016-04-08	2016-04-12	Standard Class	ZC-21910	Zuschuss Carroll	Consumer	United States	Houston	...	77041	Central	OFF-BI-10004995	Office Supplies	Binders	GBC DocuBind P400 Electric Binding System
2697	2698	CA-2014-145317	2014-03-18	2014-03-23	Standard Class	SM-20320	Sean Miller	Home Office	United States	Jacksonville	...	32216	South	TEC-MA-10002412	Technology	Machines	Cisco TelePresence System EX90 Videoconferenc...

10 rows x 21 columns

```
[16]: df.sort_values(by='Profit', ascending=True)[['Category', 'Sub-Category', 'Sales', 'Discount', 'Profit']].head(10)
```

	Category	Sub-Category	Sales	Discount	Profit
7772	Technology	Machines	4499.985	0.7	-6599.9780
683	Technology	Machines	7999.980	0.5	-3839.9904
9774	Office Supplies	Binders	2177.584	0.8	-3701.8928
3011	Technology	Machines	2549.985	0.7	-3399.9800
4991	Office Supplies	Binders	1889.990	0.8	-2929.4845
3151	Technology	Machines	1799.994	0.7	-2639.9912
5310	Office Supplies	Binders	1525.188	0.8	-2287.7820
9639	Furniture	Tables	4297.644	0.4	-1862.3124
1199	Office Supplies	Binders	1088.792	0.8	-1850.9464
2697	Technology	Machines	22638.480	0.5	-1811.0784

```
[17]: df.sort_values(by='Profit', ascending=False)[['Category', 'Sub-Category', 'Sales', 'Discount', 'Profit']].head(10)
```

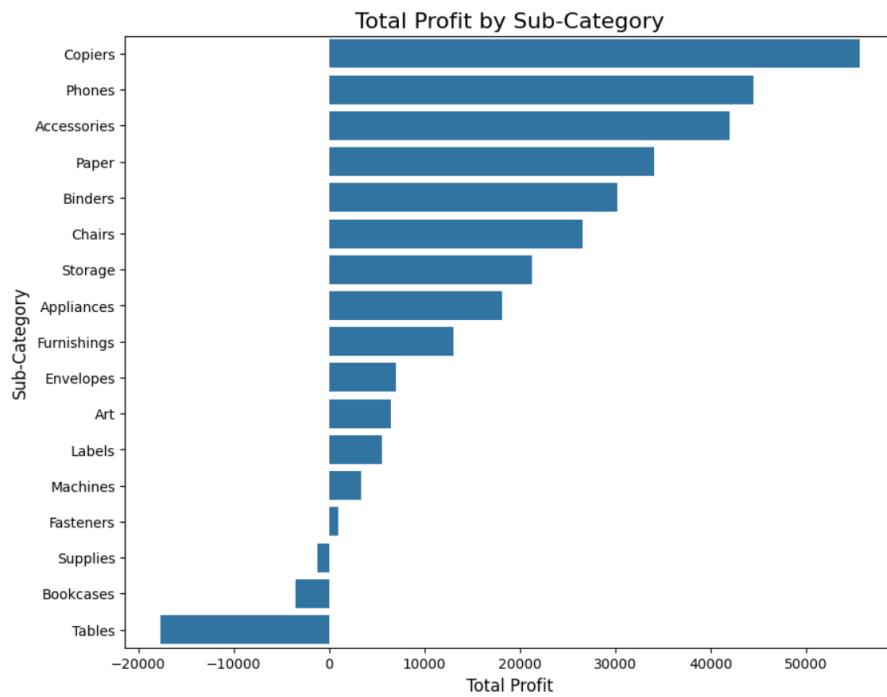
	Category	Sub-Category	Sales	Discount	Profit
6826	Technology	Copiers	17499.950	0.0	8399.9760
8153	Technology	Copiers	13999.960	0.0	619.9908
4190	Technology	Copiers	10499.970	0.0	5039.9856
9039	Office Supplies	Binders	9892.740	0.0	4946.3700
4098	Office Supplies	Binders	9449.950	0.0	4630.4755
2623	Technology	Copiers	11199.968	0.2	3919.9888
509	Office Supplies	Binders	6354.950	0.0	3177.4750
8488	Technology	Machines	8749.950	0.0	2799.9840
7666	Technology	Copiers	5399.910	0.0	2591.9568
6520	Office Supplies	Binders	5443.960	0.0	2504.2216

```
[18]: df.groupby('Sub-Category')['Profit'].sum().sort_values()
```

```
[18]: Sub-Category
Tables -17725.4811
Bookcases -3472.5560
Supplies -1189.0995
Fasteners 949.5182
Machines 3384.7569
Labels 5546.2548
etc 6597.7870
```

```
.. ~ .....  
Envelopes      6964.1767  
Furnishings    13059.1436  
Appliances     18138.0054  
Storage        21278.8264  
Chairs          26590.1663  
Binders         30221.7633  
Paper            34053.5693  
Accessories     41936.6357  
Phones           44515.7306  
Copiers          55617.8249  
Name: Profit, dtype: float64
```

```
[19]: profit_by_subcat = df.groupby('Sub-Category')['Profit'].sum().reset_index()  
  
profit_by_subcat = profit_by_subcat.sort_values(by='Profit', ascending=False)  
  
plt.figure(figsize=(10, 8))  
sns.barplot(  
    x='Profit',  
    y='Sub-Category',  
    data=profit_by_subcat  
)  
  
plt.title('Total Profit by Sub-Category', fontsize=16)  
plt.xlabel('Total Profit', fontsize=12)  
plt.ylabel('Sub-Category', fontsize=12)  
plt.show() # Display the chart
```



```
[ ]:
```