

```
from google.colab import files
uploaded = files.upload()
```

Choose Files

Sample - Superstore.csv

• **Sample - Superstore.csv**(text/csv) - 2287806 bytes, last modified: 4/27/2025 - 100% done

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

%matplotlib inline
```

```
# No code provided to correct. This appears to be a shell command or a comment.
# Saving Sample - Superstore.csv to Sample - Superstore.csv
```

```
df = pd.read_csv('Sample - Superstore.csv', encoding='latin-1') # or 'cp1252'
df.head()
```

🔄

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Product ID	Category	Cat
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	Book
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	
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	I
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	
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	S

5 rows × 21 columns

```
print("Dataset Shape:", df.shape)

print("Columns:", df.columns)

print("Missing Values:\n", df.isnull().sum())

df.describe()
```

Dataset Shape: (9994, 21)
Columns: Index(['Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode', 'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'State', 'Postal Code', 'Region', 'Product ID', 'Category', 'Sub-Category', 'Product Name', 'Sales', 'Quantity', 'Discount', 'Profit'], dtype='object')
Missing Values:
Row ID 0
Order ID 0
Order Date 0
Ship Date 0
Ship Mode 0
Customer ID 0
Customer Name 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
Quantity 0
Discount 0
Profit 0
dtype: int64

	Row ID	Postal Code	Sales	Quantity	Discount	Profit
count	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000
mean	4997.500000	55190.379428	229.858001	3.789574	0.156203	28.656896
std	2885.163629	32063.693350	623.245101	2.225110	0.206452	234.260108
min	1.000000	1040.000000	0.444000	1.000000	0.000000	-6599.978000
25%	2499.250000	23223.000000	17.280000	2.000000	0.000000	1.728750
50%	4997.500000	56430.500000	54.490000	3.000000	0.200000	8.666500
75%	7495.750000	90008.000000	209.940000	5.000000	0.200000	29.364000
max	9994.000000	99901.000000	22638.480000	14.000000	0.800000	8399.976000

```
category_profit = df.groupby('Category')['Profit'].sum().sort_values(ascending=False)
print(category_profit)

subcategory_profit = df.groupby('Sub-Category')['Profit'].sum().sort_values(ascending=False)
print(subcategory_profit)
```

Category
Technology 145454.9481
Office Supplies 122490.8008
Furniture 18451.2728
Name: Profit, dtype: float64
Sub-Category
Copiers 55617.8249
Phones 44515.7306
Accessories 41936.6357
Paper 34053.5693
Binders 30221.7633
Chairs 26590.1663
Storage 21278.8264
Appliances 18138.0054
Furnishings 13059.1436
Envelopes 6964.1767
Art 6527.7870
Labels 5546.2540
Machines 3384.7569
Fasteners 949.5182
Supplies -1189.0995
Bookcases -3472.5560
Tables -17725.4811
Name: Profit, dtype: float64

```
category_profit = df.groupby('Category')['Profit'].sum().sort_values(ascending=False)
print(category_profit)

subcategory_profit = df.groupby('Sub-Category')['Profit'].sum().sort_values(ascending=False)
print(subcategory_profit)
```

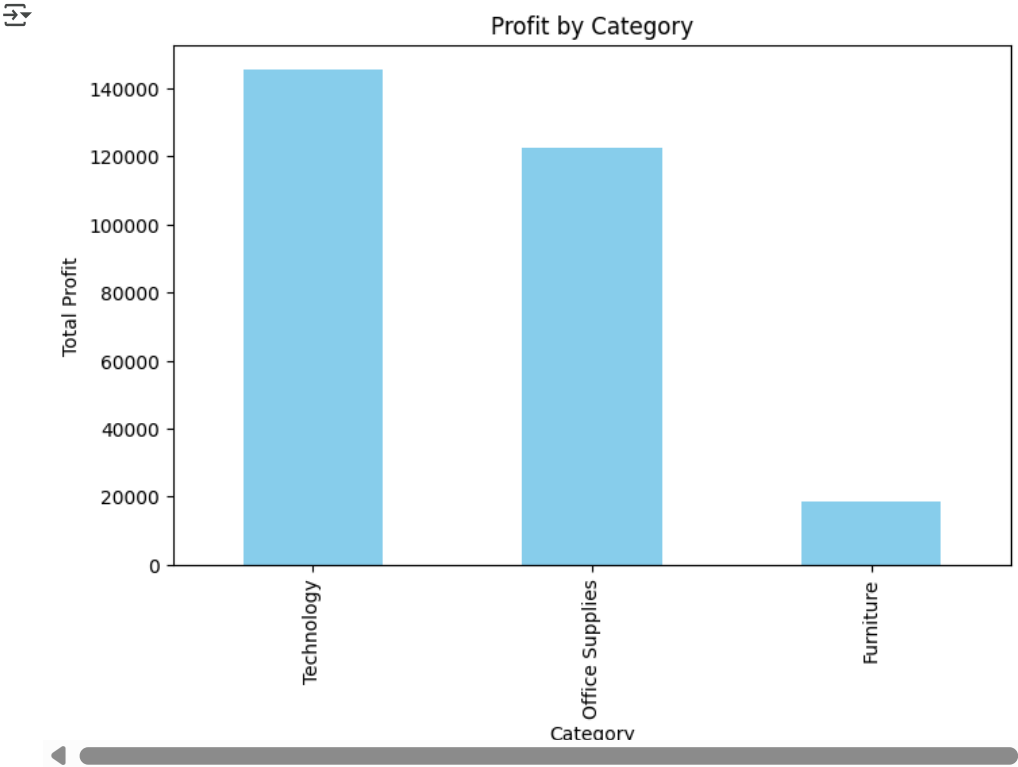
Category
Technology 145454.9481
Office Supplies 122490.8008
Furniture 18451.2728

Name: Profit, dtype: float64

Sub-Category	Profit
Copiers	55617.8249
Phones	44515.7306
Accessories	41936.6357
Paper	34053.5693
Binders	30221.7633
Chairs	26590.1663
Storage	21278.8264
Appliances	18138.0054
Furnishings	13059.1436
Envelopes	6964.1767
Art	6527.7870
Labels	5546.2540
Machines	3384.7569
Fasteners	949.5182
Supplies	-1189.0995
Bookcases	-3472.5560
Tables	-17725.4811

Name: Profit, dtype: float64

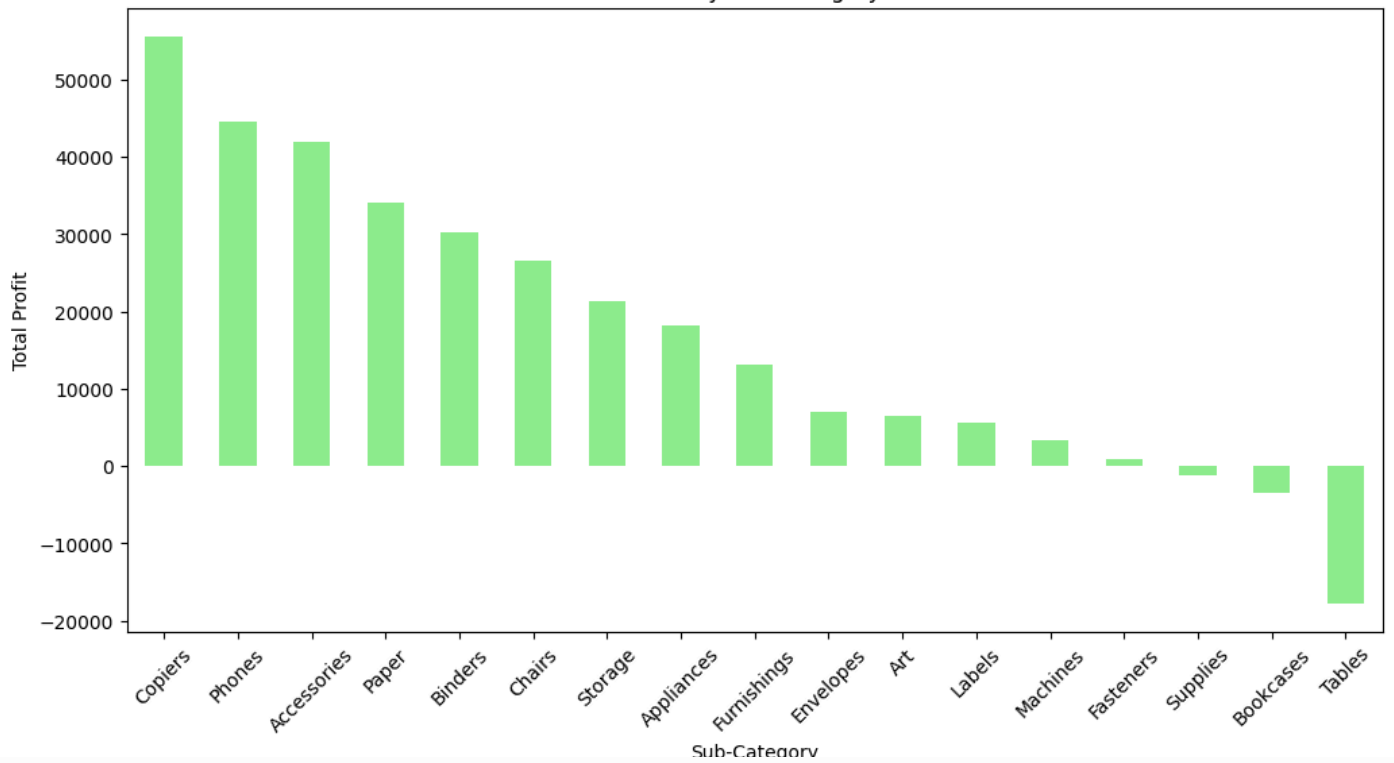
```
plt.figure(figsize=(8,5))
category_profit.plot(kind='bar', color='skyblue')
plt.title('Profit by Category')
plt.xlabel('Category')
plt.ylabel('Total Profit')
plt.show()
```



```
plt.figure(figsize=(12,6))
subcategory_profit.plot(kind='bar', color='lightgreen')
plt.title('Profit by Sub-Category')
plt.xlabel('Sub-Category')
plt.ylabel('Total Profit')
plt.xticks(rotation=45)
plt.show()
```



Profit by Sub-Category



```
df['Inventory_Turnover'] = df['Quantity'] / (df['Sales'] + 1)

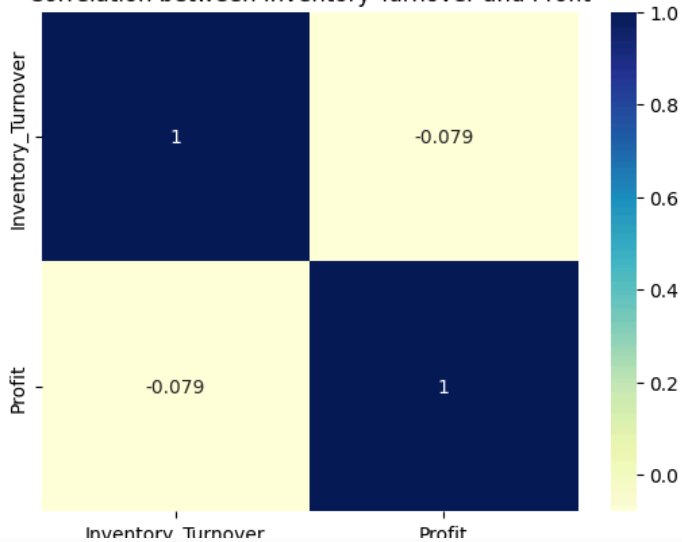
correlation = df[['Inventory_Turnover', 'Profit']].corr()
print(correlation)

sns.heatmap(correlation, annot=True, cmap="YlGnBu")
plt.title('Correlation between Inventory Turnover and Profit')
plt.show()
```



```
Inventory_Turnover    Inventory_Turnover    Profit
Inventory_Turnover    1.000000    -0.079277
Profit                -0.079277    1.000000
```

Correlation between Inventory Turnover and Profit



```
df['Order Date'] = pd.to_datetime(df['Order Date'])

df['Month'] = df['Order Date'].dt.month

monthly_profit = df.groupby('Month')['Profit'].sum()

plt.figure(figsize=(10,5))
monthly_profit.plot(kind='line', marker='o', color='red')
plt.title('Monthly Profit Trend')
plt.xlabel('Month')
plt.ylabel('Profit')
plt.grid(True)
plt.show()
```

Monthly Profit Trend

