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

```
# ============================================================
# SUPERSTORE FULL CODE (5 BEST VARIABLES)
# Best 5 variables for report:
#   1) sales
#   2) profit
#   3) discount
#   4) quantity
#   5) ship_days  (derived from ship_date - order_date)
```

```
superstore = pd.read_csv("Group Project Data-SuperStore-Sales_DATA.csv")
superstore.head()
```

	order_id	order_date	ship_date	customer	manufactory	product_name	segment	category	subcategory	region	...	city
0	US-2020-103800	2019-01-03	2019-01-07	Darren Powers	Message Book	Message Book, Wirebound, Four 5 1/2" X 4" Form...	Consumer	Office Supplies	Paper	Central	...	Houston
1	US-2020-112326	2019-01-04	2019-01-08	Phillina Ober	GBC	GBC Standard Plastic Binding Systems Combs	Home Office	Office Supplies	Binders	Central	...	Naperville
2	US-2020-112326	2019-01-04	2019-01-08	Phillina Ober	Avery	Avery 508	Home Office	Office Supplies	Labels	Central	...	Naperville
3	US-2020-112326	2019-01-04	2019-01-08	Phillina Ober	SAFCO	SAFCO Boltless Steel Shelving	Home Office	Office Supplies	Storage	Central	...	Naperville
4	US-2020-141817	2019-01-05	2019-01-12	Mick Brown	Avery	Avery Hi-Liter EverBold Pen Style Fluorescent ...	Consumer	Office Supplies	Art	East	...	Philadelphia

5 rows × 21 columns

```
#Cell 1 — Check columns
superstore.columns
# delete specific unnamed columns
superstore = superstore.drop(columns=["Unnamed: 19", "Unnamed: 20"], errors="ignore")

superstore.columns
```

```
Index(['order_id', 'order_date', 'ship_date', 'customer', 'manufactory',
       'product_name', 'segment', 'category', 'subcategory', 'region', 'zip',
       'city', 'state', 'country', 'discount', 'profit', 'quantity', 'sales',
       'profit_margin'],
      dtype='object')
```

```
#Cell 2 — Convert dates + create ship_days
superstore["order_date"] = pd.to_datetime(superstore["order_date"], errors="coerce")
superstore["ship_date"]  = pd.to_datetime(superstore["ship_date"], errors="coerce")

superstore["ship_days"] = (superstore["ship_date"] - superstore["order_date"]).dt.days

superstore[["order_date", "ship_date", "ship_days"]].head()
```

	order_date	ship_date	ship_days
0	2019-01-03	2019-01-07	4
1	2019-01-04	2019-01-08	4
2	2019-01-04	2019-01-08	4
3	2019-01-04	2019-01-08	4
4	2019-01-05	2019-01-12	7

```
#Cell 3 — Descriptive statistics for the BEST 5 variables
vars_5 = ["sales", "profit", "discount", "quantity", "ship_days"]

desc5 = superstore[vars_5].describe().T
desc5
```

	count	mean	std	min	25%	50%	75%	max
sales	9994.0	229.858001	623.245101	0.444	17.28000	54.4900	209.940	22638.480
profit	9994.0	28.656896	234.260108	-6599.978	1.72875	8.6665	29.364	8399.976
discount	9994.0	0.156203	0.206452	0.000	0.00000	0.2000	0.200	0.800
quantity	9994.0	3.789574	2.225110	1.000	2.00000	3.0000	5.000	14.000
ship_days	9994.0	3.958475	1.747603	0.000	3.00000	4.0000	5.000	7.000

```
#Cell 4 — Correlation table (supports insight)
corr5 = superstore[vars_5].corr(numeric_only=True)
corr5
```

	sales	profit	discount	quantity	ship_days
sales	1.000000	0.479064	-0.028190	0.200795	-0.007285
profit	0.479064	1.000000	-0.219487	0.066253	-0.004672
discount	-0.028190	-0.219487	1.000000	0.008623	0.000306
quantity	0.200795	0.066253	0.008623	1.000000	0.018494
ship_days	-0.007285	-0.004672	0.000306	0.018494	1.000000

```
#Cell 5 — Pivot tables (Category)
# Category summary
cat_summary = superstore.groupby("category").agg(
    orders=("order_id","count"),
    total_sales=("sales","sum"),
    total_profit=("profit","sum"),
    avg_discount=("discount","mean"),
    avg_ship_days=("ship_days","mean")
).sort_values("total_profit", ascending=False)

cat_summary
```

category	orders	total_sales	total_profit	avg_discount	avg_ship_days
Technology	1847	836154.0330	145454.9481	0.132323	3.924201
Office Supplies	6026	719047.0320	122490.8008	0.157285	3.983239
Furniture	2121	741999.7953	18451.2728	0.173923	3.917963

```
#Cell 6 — Pivot tables (Region)
# Region summary
reg_summary = superstore.groupby("region").agg(
    orders=("order_id","count"),
    total_sales=("sales","sum"),
    total_profit=("profit","sum"),
    avg_discount=("discount","mean"),
    avg_ship_days=("ship_days","mean")
```

```
).sort_values("total_profit", ascending=False)

reg_summary
```

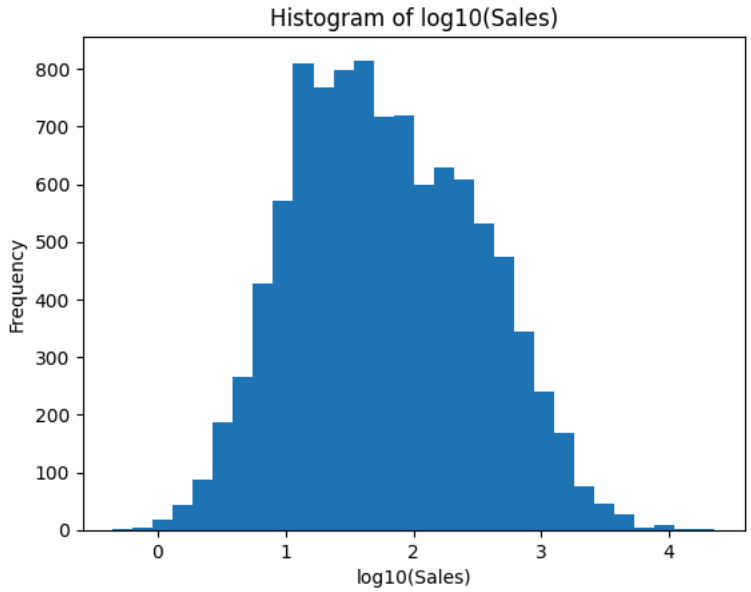
region	orders	total_sales	total_profit	avg_discount	avg_ship_days
West	3203	725457.8245	108418.4489	0.109335	3.930066
East	2848	678781.2400	91522.7800	0.145365	3.909410
South	1620	391721.9050	46749.4303	0.147253	3.959259
Central	2323	501239.8908	39706.3625	0.240353	4.057254

```
#Cell 7 — Pivot tables (Segment distribution)
# Segment distribution
seg_counts = superstore["segment"].value_counts()
seg_pct = superstore["segment"].value_counts(normalize=True) * 100
seg_table = pd.DataFrame({"count": seg_counts, "pct": seg_pct.round(2)})

seg_table
```

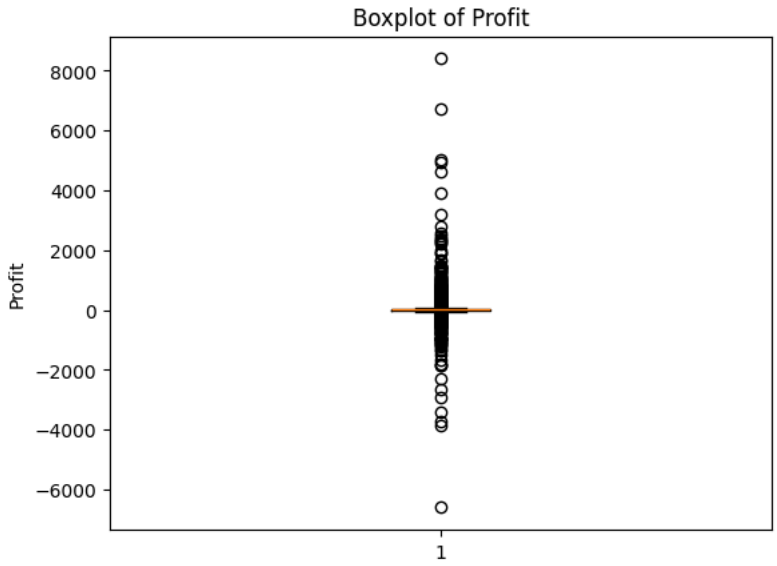
segment	count	pct
Consumer	5191	51.94
Corporate	3020	30.22
Home Office	1783	17.84

```
#Cell 8 — Plots
#Plot 1: Sales histogram (log10)
plt.figure()
sales = superstore["sales"].dropna()
plt.hist(np.log10(sales), bins=30)
plt.title("Histogram of log10(Sales)")
plt.xlabel("log10(Sales)")
plt.ylabel("Frequency")
plt.show()
```

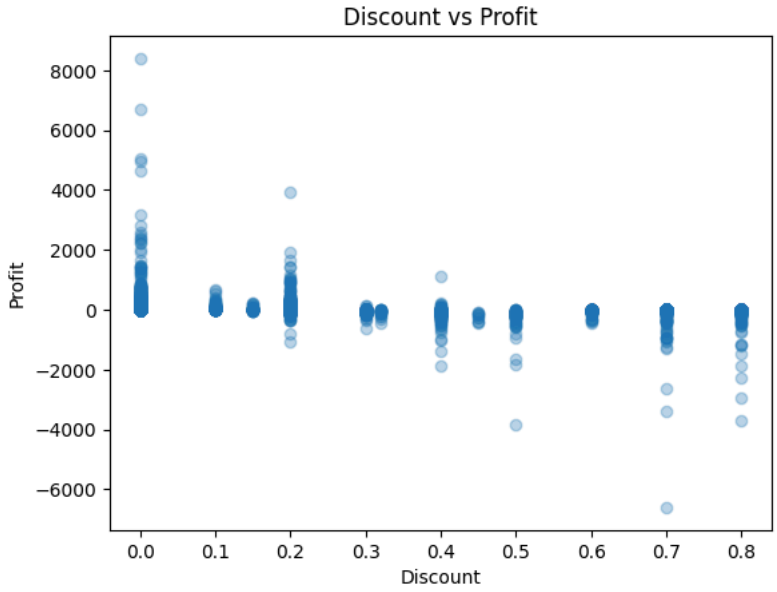



```
#Cell 9 — Plots
#Plot 2: Profit boxplot
plt.figure()
plt.boxplot(superstore["profit"].dropna())
plt.title("Boxplot of Profit")
```

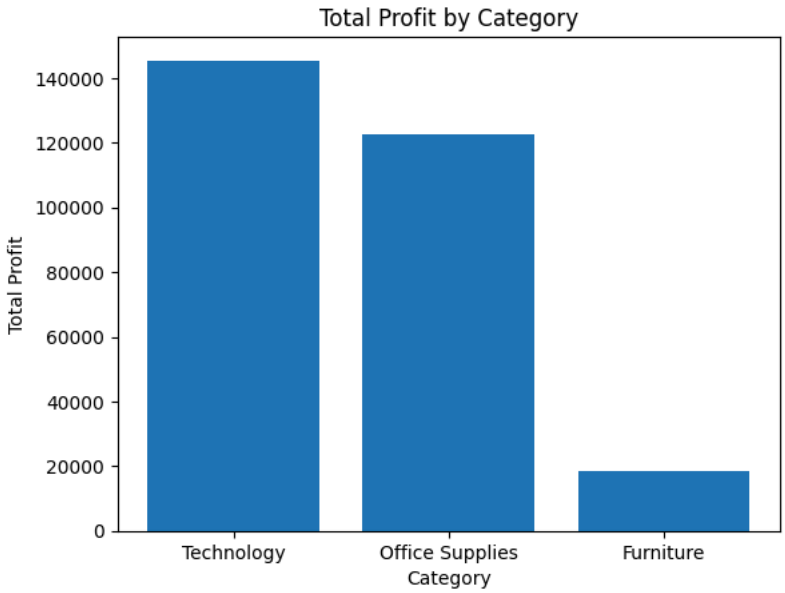
```
plt.ylabel("Profit")
plt.show()
```




```
#Cell 10 – Plots
#Plot 3: Discount vs Profit scatter
plt.figure()
plt.scatter(superstore["discount"], superstore["profit"], alpha=0.3)
plt.title("Discount vs Profit")
plt.xlabel("Discount")
plt.ylabel("Profit")
plt.show()
```




```
#Cell 11 – Plots
#Plot 4: Profit by Category bar
plt.figure()
plt.bar(cat_summary.index.astype(str), cat_summary["total_profit"])
plt.title("Total Profit by Category")
plt.xlabel("Category")
plt.ylabel("Total Profit")
plt.show()
```




```
#Cell 12 — Auto Insights
print("=== INSIGHTS ===")

# Insight 1: Sales skew (mean vs median)
sales_mean = superstore["sales"].mean()
sales_median = superstore["sales"].median()
print(f"1) Sales is right-skewed: mean= {sales_mean:.2f} > median= {sales_median:.2f}. "
      f"Outliers/high-value orders pull the mean up (supported by Sales histogram).")

# Insight 2: Discount vs Profit correlation
r_dp = superstore[["discount","profit"]].corr(numeric_only=True).iloc[0,1]
print(f"2) Discount impacts profit: corr(discount, profit)={r_dp:.3f}. "
      f"Higher discounts tend to reduce profit / increase loss risk (supported by scatter + correlation).")

# Insight 3: Category profit differences
best_cat = cat_summary["total_profit"].idxmax()
worst_cat = cat_summary["total_profit"].idxmin()
print(f"3) Category performance differs: best profit category = {best_cat}, worst = {worst_cat} "
      f"(supported by category pivot + profit bar chart).")
```

```
=== INSIGHTS ===
1) Sales is right-skewed: mean= 229.86 > median= 54.49. Outliers/high-value orders pull the mean up (supported by Sales histogra
2) Discount impacts profit: corr(discount, profit)=-0.219. Higher discounts tend to reduce profit / increase loss risk (supporte
3) Category performance differs: best profit category = Technology, worst = Furniture (supported by category pivot + profit bar
```