In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
df_sales = pd.read_csv("MegaMart_sales.csv")
```

In [3]:
```
df_newsales = pd.read_csv("MegaMart_newsales.csv")
```

In [4]:
```
combined = df_sales.append(df_newsales)
len(combined)
```

Out[4]:
```
68
```

In [5]:
```
combined
```

Out[5]:

	Order ID	Product Name	Discount	Sales	Profit	Quantity	Category	Sub-Category
0	AZ-2011-1029887	Novimex Color Coded Labels, 5000 Label Set	0.0	26	7	2	Office Supplies	Labels
1	AZ-2011-107716	Deflect-O Door Stop, Erganomic	0.0	85	15	2	Furniture	Furnishings
2	AZ-2011-1087704	Belkin Flash Drive, Bluetooth	0.0	294	109	7	Technology	Accessories
3	AZ-2011-1372644	Panasonic Printer, Durable	0.0	800	168	3	Technology	Machines
4	AZ-2011-1362199	Sanford Pens, Fluorescent	0.5	25	-11	4	Office Supplies	Art
...	...	...	...	...	...	...	...	...
4	AZ-2011-1967754	Logitech Numeric Keypad, USB	0.0	93	40	2	Technology	Accessories
5	AZ-2011-1976919	Boston Markers, Blue	0.0	132	54	5	Office Supplies	Art
6	AZ-2011-2001312	Avery Binding Machine, Clear	0.0	97	12	2	Office Supplies	Binders
7	AZ-2011-2002251	SanDisk Computer Printout Paper, 8.5 x 11	0.0	136	15	4	Office Supplies	Paper
8	AZ-2011-201891	Cameo Clasp Envelope, with clear poly window	0.0	52	19	4	Office Supplies	Envelopes

68 rows × 8 columns

In [6]:
```
combined.columns
```

Out[6]:
```
Index(['Order ID', 'Product Name', 'Discount', 'Sales', 'Profit', 'Quantity',
       'Category', 'Sub-Category'],
      dtype='object')
```

1. Combining two data frames

Some of the orders are stored in another csv file named `megamart_new sales`. Read the csv file, store it in a data frame and add it to the `megamart_sales` data frame.

Question 1: Find the total sales value of the category 'Office Supplies' after combining the dataframes

a)7970
b)6964
c)7494
d)6076

In [7]:
```
combined["Category"].unique()
```

Out[7]:
```
array(['Office Supplies', 'Furniture', 'Technology'], dtype=object)
```

In [8]:
```
combined[combined["Category"]=="Office Supplies"]["Sales"].sum()
```

Out[8]:
```
7970
```

---

2. Dropping duplicates

Question 2: There are some duplicate rows in the data frame. Drop these rows and calculate the total sales value of the category Office Supplies.

a)7156
b)6496
c)6964
d)6023

In [9]:
```
combined_no_dups = combined.drop_duplicates()
len(combined_no_dups)
```

Out[9]:
```
61
```

In [10]:
```
combined_no_dups[combined_no_dups["Category"]=="Office Supplies"]["Sales"].sum()
```

Out[10]:
```
6964
```

---

3. Best category-sub category

Question 3: Find the most profitable category and sub category combination based on the net profit.

a)Furniture-Bookcases
b)Office supplies-Appliances
c)Office supplies-Storage
d)Technology-Phones

In [11]:
```
grp_cat_subcat_mean = combined.groupby(["Category","Sub-Category"])["Profit"].sum()
grp_cat_subcat_mean
```

Out[11]:
```
Category         Sub-Category
Furniture        Bookcases        308
                 Chairs           -49
                 Furnishings       74
Office Supplies  Appliances       229
                 Art              283
                 Binders          158
                 Envelopes         35
```

```
                Fasteners         10
                Labels            30
                Paper             15
                Storage          475
                Supplies         103
Technology      Accessories      324
                Copiers            0
                Machines         336
                Phones          1618
Name: Profit, dtype: int64
```

In [12]:
```
grp_cat_subcat_mean.sort_values(ascending=False).head(1)
```

Out[12]:
```
Category    Sub-Category
Technology  Phones          1618
Name: Profit, dtype: int64
```

---

4. Invalid order IDs

Question 4: How many invalid order IDs are there in the data frame? An order id is of the form AZ-2011-Y where Y represents a whole number. A Order ID is said to be valid only if Y consists of 7 digits. Find the number of invalid order IDs in the data frame.

a)6
b)7
c)8
d)9

In [13]:
```
def check_order_id(order_id):
    y = order_id.split("-")[-1]
    if (y.isdigit()) and (len(y)==7):
        return False
    else:
        return True
invalid_order_ids = combined_no_dups["Order ID"].apply(check_order_id)
combined_no_dups["Order ID"][invalid_order_ids]
```

Out[13]:
```
1     AZ-2011-107716
9     AZ-2011-122598
```

```
17    AZ-2011-130330
31    AZ-2011-144325
34    AZ-2011-145488
58    AZ-2011-176674
8     AZ-2011-201891
Name: Order ID, dtype: object
```

In [14]:
```
len(combined_no_dups["Order ID"][invalid_order_ids])
```

Out[14]:
```
7
```

---

5. Occurence of furniture in top 25 sales

Question 5: Find the top 25 orders based on sales value and find the number of orders which belong to furniture category.

a)2
b)3
c)4
d)5

In [15]:
```
top25 = combined_no_dups.sort_values(by="Sales",ascending=False)[:25]
top25[top25["Category"]=="Furniture"]
```

Out[15]:

	Order ID	Product Name	Discount	Sales	Profit	Quantity	Category	Sub-Category
31	AZ-2011-144325	Bush Stackable Bookrack, Pine	0.0	630	132	5	Furniture	Bookcases
12	AZ-2011-1253407	Safco Stackable Bookrack, Pine	0.1	541	156	4	Furniture	Bookcases
17	AZ-2011-130330	Office Star Chairmat, Adjustable	0.1	307	99	5	Furniture	Chairs
1	AZ-2011-1916360	Dania 3-Shelf Cabinet, Mobile	0.0	288	20	2	Furniture	Bookcases
43	AZ-2011-1589827	Novimex Steel Folding Chair, Red	0.6	164	-70	5	Furniture	Chairs

In [16]:
```
top25["Category"].value_counts()
```

Out[16]:
```
Office Supplies    11
Technology          9
Furniture           5
Name: Category, dtype: int64
```

---

6. And operation

Question 6: Among the orders with sales>250 and profit>50, find the product name of the fourth highest order based on sales value.

a)Motorola Headset, with Caller ID
b)Panasonic Printer, Durable
c)Hoover Microwave, Red
d)Fellowes Lockers, Industrial

In [17]:
```
combined_no_dups[(combined_no_dups["Sales"]>250) & (combined_no_dups["Profit"]>50)].sort_values(by="Sales",ascending=False)["Produ
```

Out[17]:
```
'Motorola Headset, with Caller ID'
```

7. Column manipulation

Question 7: Remove the orders with negative profit by dropping the corresponding rows with negative `Profit`. Find the product that makes the lowest profit per Quantity in the Technology category.

a) Nokia Audio Dock, with Caller ID
b) Logitech Keyboard, Programmable
c) Motorola Headset, with Caller ID
d) Belkin Flash Drive, Bluetooth

In [18]:
```
# combined_no_dups.assign(Profit_per_qty=0)  # creates a column with default values
df_positive_profit = combined_no_dups[combined_no_dups["Profit"]>0]
df_positive_profit_tech = df_positive_profit[df_positive_profit["Category"]=="Technology"]
# df_positive_profit_tech_cp = df_positive_profit_tech.copy()
df_positive_profit_tech["Profit per qty"] = df_positive_profit_tech["Profit"]/df_positive_profit_tech["Quantity"]
df_positive_profit_tech.sort_values(by="Profit per qty").head(1)
```

```
C:\Users\viren\AppData\Local\Temp/ipykernel_30396/2509391348.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_positive_profit_tech["Profit per qty"] = df_positive_profit_tech["Profit"]/df_positive_profit_tech["Quantity"]
```

Out[18]:

	Order ID	Product Name	Discount	Sales	Profit	Quantity	Category	Sub-Category	Profit per qty
39	AZ-2011-1536006	Logitech Keyboard, Programmable	0.0	666	66	9	Technology	Accessories	7.333333

In [ ]: