In [193…
```
import pandas as pd
import numpy as np
```

In [194…
```
df=pd.read_csv('/content/superstore.csv')
df
```

Out[194]:

	row_id	order_id	order_date	ship_date	ship_mode	customer_id	customer_name	se
0	42433	AG-2011-2040	1/1/2011	6/1/2011	Standard Class	TB-11280	Toby Braunhardt	Co
1	22253	IN-2011-47883	1/1/2011	8/1/2011	Standard Class	JH-15985	Joseph Holt	Co
2	48883	HU-2011-1220	1/1/2011	5/1/2011	Second Class	AT-735	Annie Thurman	Co
3	11731	IT-2011-3647632	1/1/2011	5/1/2011	Second Class	EM-14140	Eugene Moren	
4	22255	IN-2011-47883	1/1/2011	8/1/2011	Standard Class	JH-15985	Joseph Holt	Co
...	...	...	...	...	...	...	...	
51285	32593	CA-2014-115427	31-12-2014	4/1/2015	Standard Class	EB-13975	Erica Bern	Co
51286	47594	MO-2014-2560	31-12-2014	5/1/2015	Standard Class	LP-7095	Liz Preis	Co
51287	8857	MX-2014-110527	31-12-2014	2/1/2015	Second Class	CM-12190	Charlotte Melton	Co
51288	6852	MX-2014-114783	31-12-2014	6/1/2015	Standard Class	TD-20995	Tamara Dahlen	Co
51289	36388	CA-2014-156720	31-12-2014	4/1/2015	Standard Class	JM-15580	Jill Matthias	Co

51290 rows × 24 columns

Are there any regular costumers? Iif so, are they the most profitable ones?

The following two graphs indicate that just because customers purchase more frequently doesn't necessarily mean they are more profitable. Therefore, I have created another figure

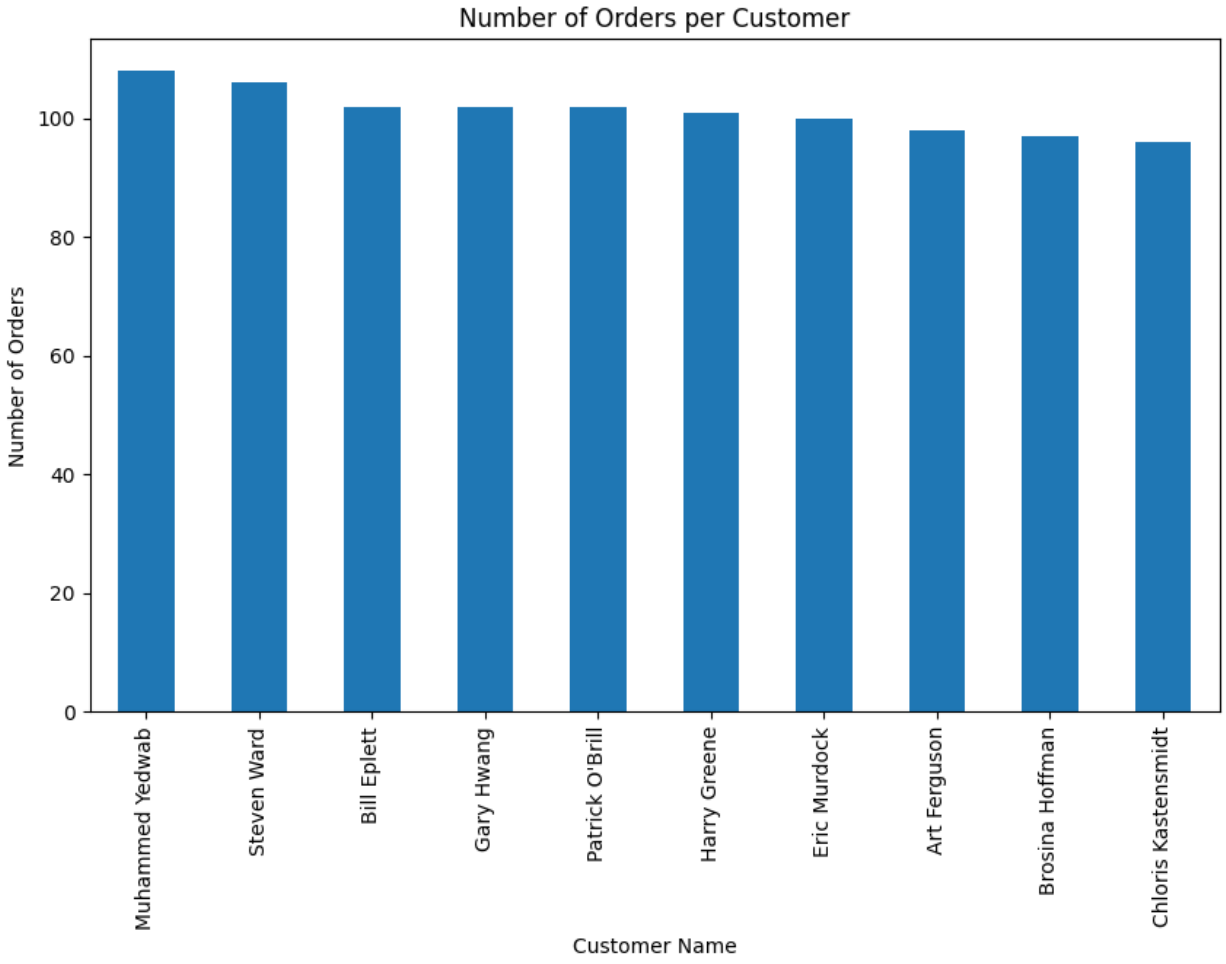
for the top 10 most profitable customers

In [195…
```
customer_order_counts
```

Out[195]:
```
Muhammed Yedwab    108
Steven Ward        106
Bill Eplett        102
Gary Hwang         102
Patrick O'Brill    102
                  ...
Andy Reiter         35
David Bremer        34
Darren Budd         31
Nicole Brennan      31
Michael Oakman      29
Name: customer_name, Length: 795, dtype: int64
```

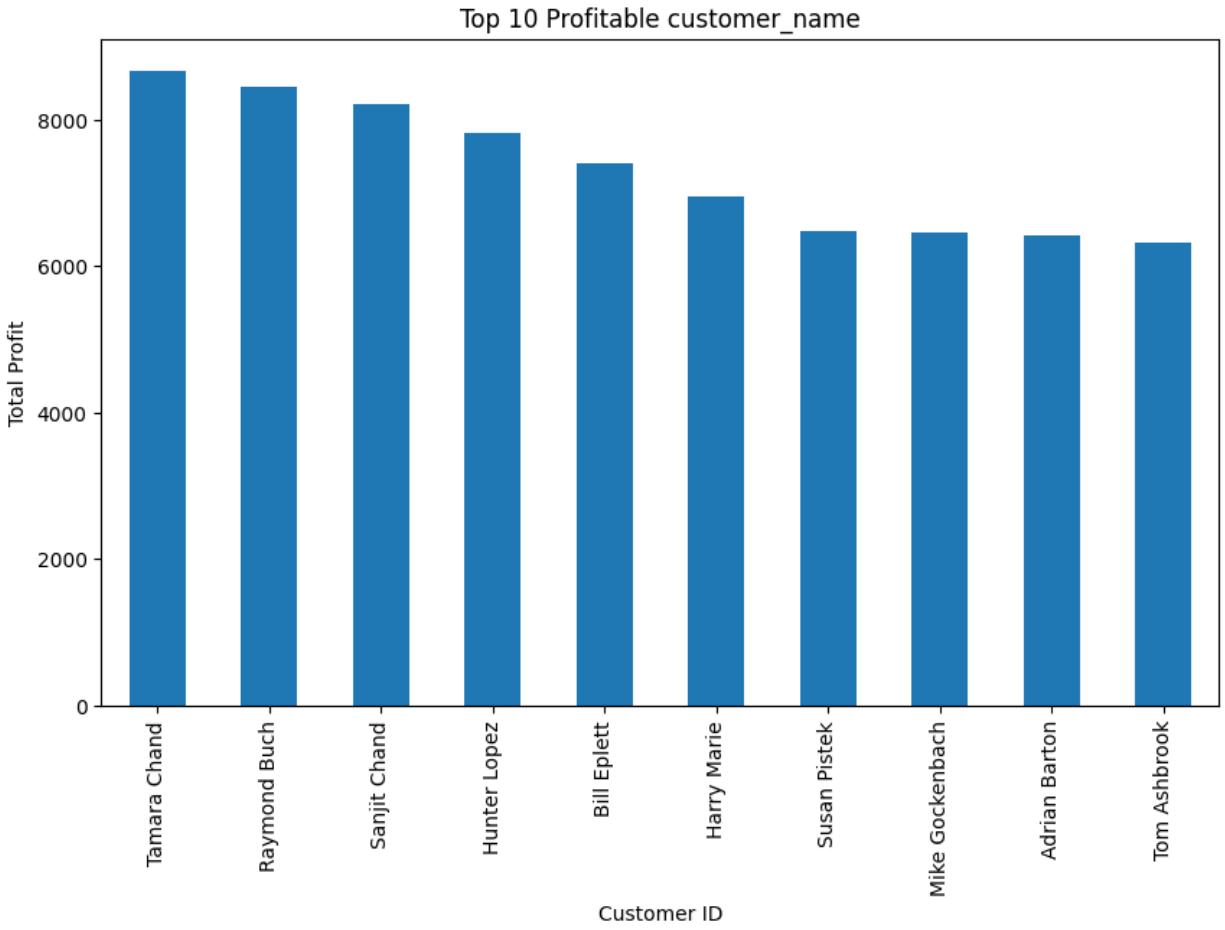
In [196…
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv('/content/superstore.csv')

df['order_date'] = pd.to_datetime(df['order_date'])


customer_order_counts = df['customer_name'].value_counts().head(10)

plt.figure(figsize=(10, 6))
customer_order_counts.plot(kind='bar')

plt.title('Number of Orders per Customer')
plt.xlabel('Customer Name')
plt.ylabel('Number of Orders')
plt.show()
```

```
<ipython-input-196-ec9dc7f5e793>:6: UserWarning: Parsing dates in DD/MM/YYYY f
ormat when dayfirst=False (the default) was specified. This may lead to incons
istently parsed dates! Specify a format to ensure consistent parsing.
  df['order_date'] = pd.to_datetime(df['order_date'])
```



In [197…

```
df['order_date'] = pd.to_datetime(df['order_date'])

customer_order_counts = df['customer_name'].value_counts()

customer_profits = df.groupby('customer_name')['profit'].sum().sort_values(asce

# grafica los 10 top
top_customers = customer_profits.iloc[:10]
plt.figure(figsize=(10, 6))
top_customers.plot(kind='bar')
plt.title('Top 10 Profitable customer_name')
plt.ylabel('Total Profit')
plt.xlabel('Customer ID')
plt.show()
```



In [198…
```
df.groupby('sub-category')['profit'].sum()
```

Out[198]:
```
sub-category
Accessories    129626.30620
Appliances     141680.58940
Art             57953.91090
Binders         72449.84600
Bookcases      161924.41950
Chairs         140396.26750
Copiers        258567.54818
Envelopes       29601.11630
Fasteners       11525.42410
Furnishings     46967.42550
Labels          15010.51200
Machines        58867.87300
Paper           59207.68270
Phones         216717.00580
Storage        108461.48980
Supplies        22583.26310
Tables         -64083.38870
Name: profit, dtype: float64
```

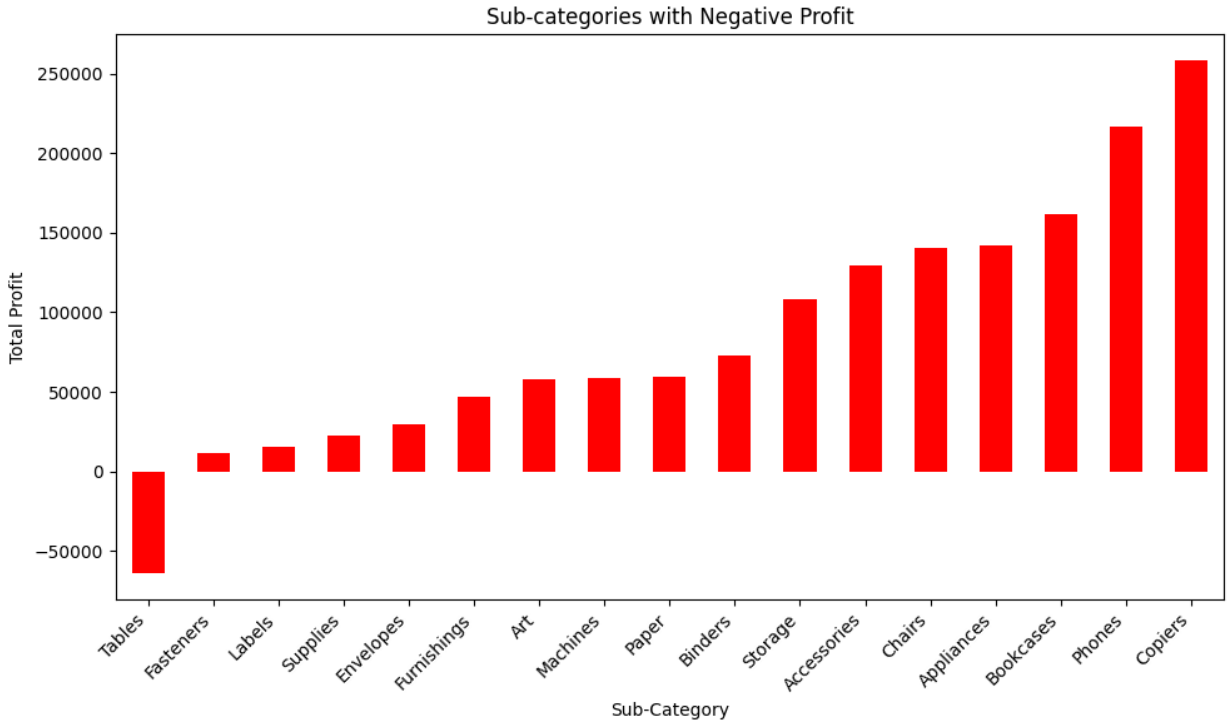
In [199…
```
negative_profit_subcategories = df.groupby('sub-category')['profit'].sum().sort
negative_profit_subcategories
```

Out[199]:
```
sub-category
Tables          -64083.38870
Fasteners        11525.42410
Labels           15010.51200
Supplies         22583.26310
Envelopes        29601.11630
Furnishings      46967.42550
Art              57953.91090
Machines         58867.87300
Paper            59207.68270
Binders          72449.84600
Storage         108461.48980
Accessories     129626.30620
Chairs          140396.26750
Appliances      141680.58940
Bookcases       161924.41950
Phones          216717.00580
Copiers         258567.54818
Name: profit, dtype: float64
```

Which product subcategories are responsible for most negative profit from sales?

Just one Tables

In [200…
```
negative_profit_subcategories = df.groupby('sub-category')['profit'].sum().sort

negative_profit_subcategories = negative_profit_subcategories
plt.figure(figsize=(10, 6))

negative_profit_subcategories.plot(kind='bar', color='red', x=negative_profit_s

plt.title('Sub-categories with Negative Profit')
plt.ylabel('Total Profit')
plt.xlabel('Sub-Category')

plt.xticks(rotation=45, ha='right')

plt.tight_layout()
plt.show()
```



Is there any trand with reagard to how different product categories sell over time?

The provided line graph titled "Sales by Category Over Time" shows sales in U.S. dollars on the y-axis and time from January 2011 to some point after July 2014 on the x-axis. It represents three categories: Furniture, Office Supplies, and Technology.

From the graph, we can observe several trends regarding how different product categories sell over time:

1. Seasonal Trends: There appear to be peaks and troughs that correspond with certain times of the year, which could suggest seasonal variations in sales. For example, sales for all categories seem to rise around the start of the year and in some cases mid-year, which may correspond to times when businesses are purchasing more office supplies or technology, such as during fiscal year beginnings or during back-to-school seasons.

2. Overall Growth: All categories show a general upward trend in sales over the years, indicating growth in each product category.

3. Category Performance: Technology products generally have the highest peaks, suggesting that when technology sales do well, they can significantly outperform the other categories. Office Supplies sales are the most consistent, with fewer dramatic peaks and valleys. Furniture sales also show growth but are less than technology and more volatile than office supplies.

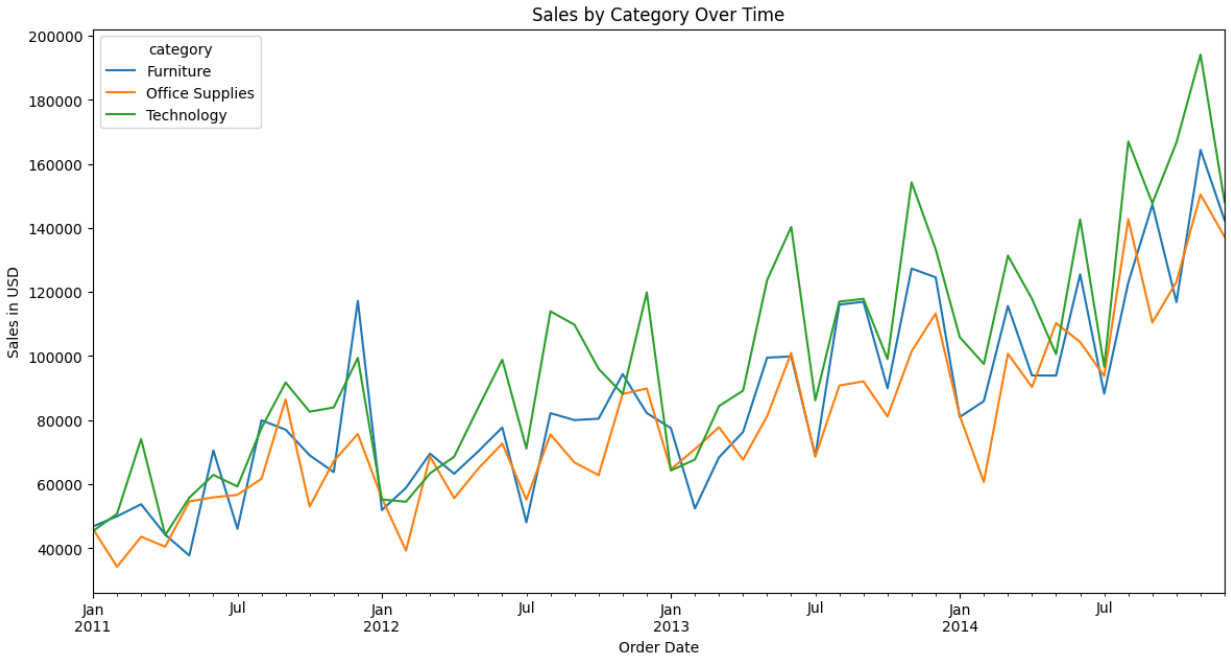
4. Peak Comparison: The highest peak for technology sales occurs in the latest part of the graph, indicating a particularly strong sales period, potentially stronger than any

previous period within the graph's timeframe.

5. Technology Sales Spikes: The Technology category has more pronounced spikes compared to the other categories, which could imply that technology sales are more susceptible to certain events or product releases.

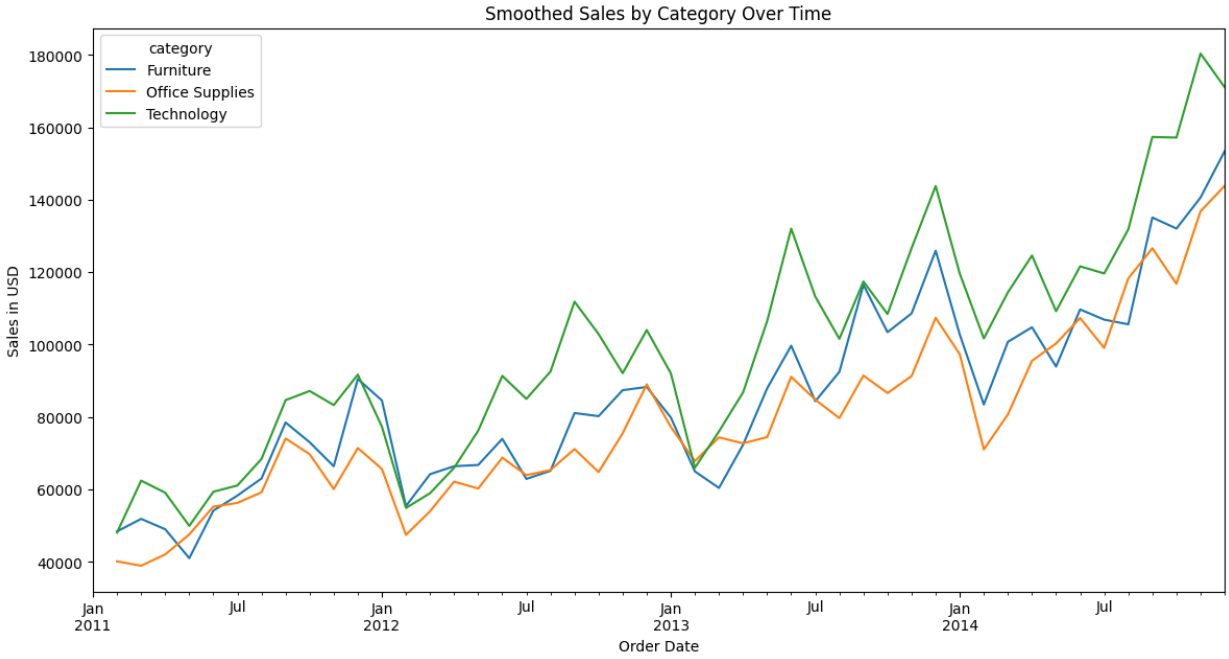
In [201…

```
df.set_index('order_date', inplace=True)
categories_time_series = df.groupby([pd.Grouper(freq='M'), 'category'])['sales

categories_time_series.plot(figsize=(14, 7), title='Sales by Category Over Time
plt.ylabel('Sales in USD')
plt.xlabel('Order Date')
plt.show()
```



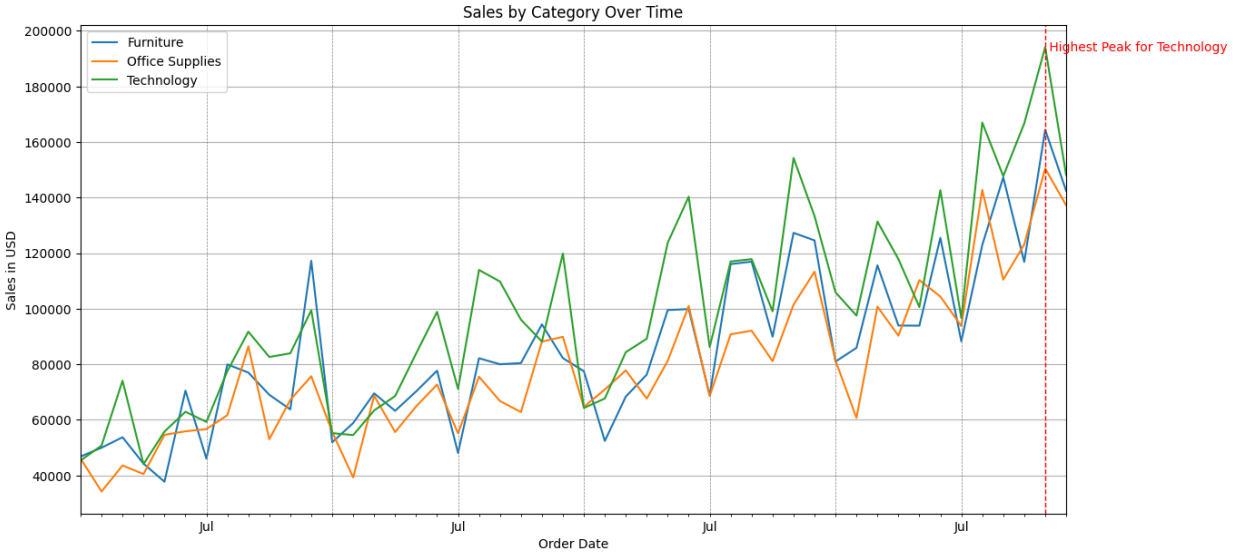
In [202…

```
import matplotlib.pyplot as plt
import pandas as pd
df.reset_index('order_date',inplace=True)
# Suponiendo que 'df' es tu DataFrame y ya tiene una columna 'order_date' que e
# Configuración del índice y agrupación por mes y categoría
df['order_date'] = pd.to_datetime(df['order_date'])  # Asegurándonos de que 'or
df.set_index('order_date', inplace=True)
categories_time_series = df.groupby([pd.Grouper(freq='M'), 'category'])['sales

# Cálculo del promedio móvil (por ejemplo, un promedio móvil de 3 meses)
window_size = 2
rolling_categories = categories_time_series.rolling(window=window_size).mean()

# Plot de las series de tiempo suavizadas por categoría
rolling_categories.plot(figsize=(14, 7), title='Smoothed Sales by Category Over
plt.ylabel('Sales in USD')
plt.xlabel('Order Date')
plt.show()
```



In [203…

```
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.dates as mdates


ax = categories_time_series.plot(figsize=(14, 7), title='Sales by Category Over
plt.ylabel('Sales in USD')
plt.xlabel('Order Date')


latest_technology_peak_date = categories_time_series['Technology'].idxmax()
latest_technology_peak_value = categories_time_series['Technology'].max()
plt.axvline(x=latest_technology_peak_date, color='red', linestyle='--', linewid
plt.text(latest_technology_peak_date, latest_technology_peak_value, ' Highest H

for year in range(2011, 2015):  # Replace with the range of years in your datas
    plt.axvline(pd.Timestamp(f'{year}-01-01'), color='grey', linestyle='--', li
    plt.axvline(pd.Timestamp(f'{year}-07-01'), color='grey', linestyle='--', li

ax.xaxis.set_major_locator(mdates.YearLocator())
ax.xaxis.set_major_formatter(mdates.DateFormatter('%Y'))

# Add a grid for better readability
plt.grid(True)

# Add legend to the plot
plt.legend()

# Show the plot with all annotations and highlights
plt.show()
```



Is there any pattern with regard to how different product sub-categories sell to different markets?

Here are some observations we can make:

1. Market Clusters:
    - There is two main clusters: 1) Canada, Africa, and EMEA form a cluster with similar sales patterns (low sales), while 2) APAC, EU, US and LATAM form another (more sales). This is important to the following:
2. Category Clusters:
    - We can observe that in Cluster 2, bookcases, copiers, chairs, and phones are more associated with high sales; they share a pattern in their sales, which is higher compared to Cluster 1

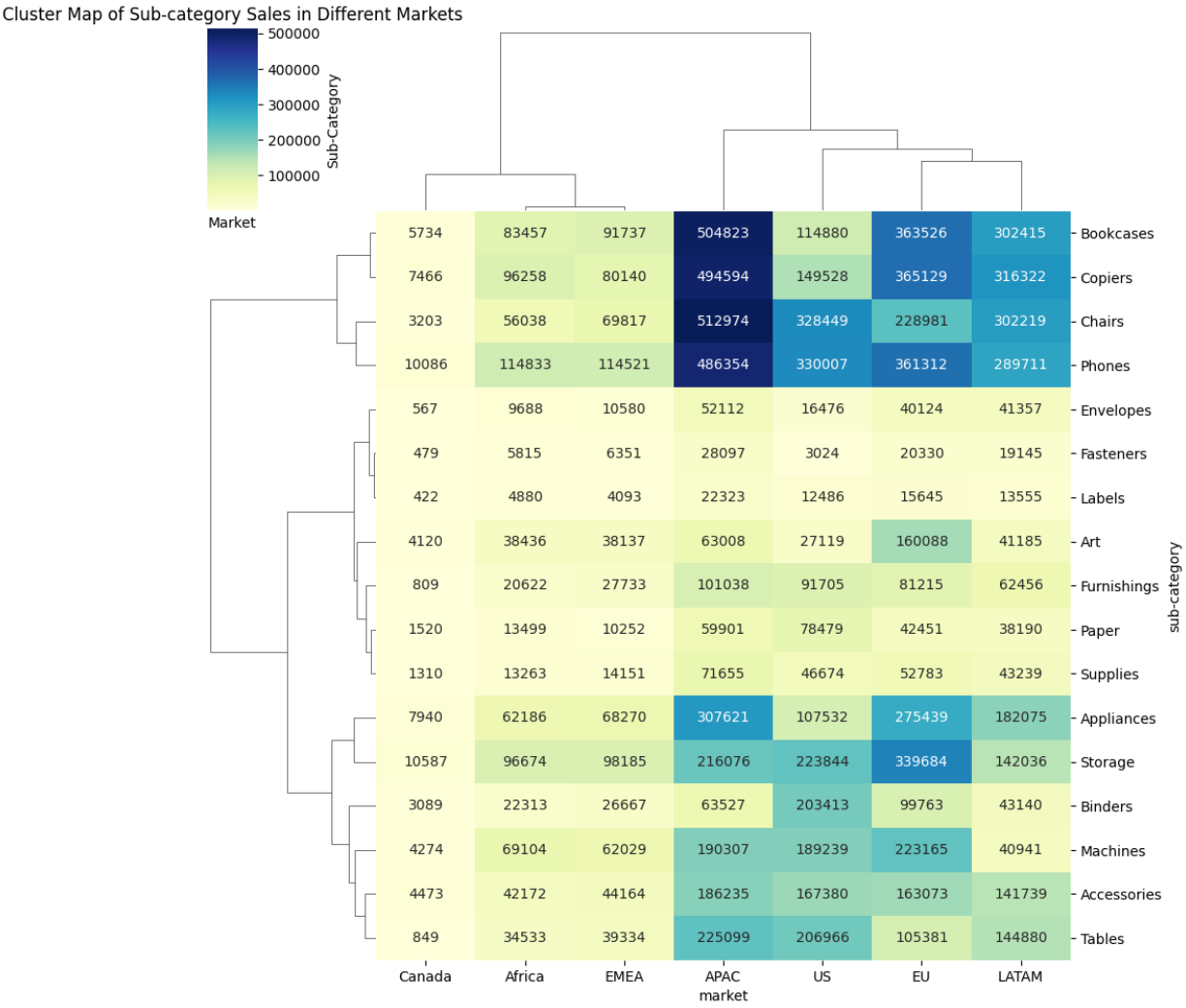
1. Sales Patterns:
    - High Revenue Sub-categories: Phones, Chairs, Copiers,Phones have the highest sales numbers in the US, APAC, LATAM and EU markets (CLUSTER 2).
    - Lower Revenue Sub-categories: Fasteners, Labels, and Art (Art, except for EU) have lower sales figures across all markets, which is visible from the lighter colors.
2. Regional Patterns:
    - The APAC market dominates sales in almost (NOT ALL) every sub-category, shown by the generally darker color intensity in the APAC column.
    - In Cluster 2, APAC, US, and EU exhibit high sales in Storage and Machines, consistent with their clustering, excluding LATAM.

- EMEA, Canada and Africa have overall lower sales in comparison to other markets, as depicted by the lighter shades in their columns.

In [204…

```
import seaborn as sns
import matplotlib.pyplot as plt

market_subcategory_sales = df.pivot_table(values='sales', index='sub-category',

plt.figure(figsize=(14, 10))
sns.clustermap(market_subcategory_sales, annot=True, fmt=".0f", cmap="YlGnBu",
plt.title('Cluster Map of Sub-category Sales in Different Markets')
plt.ylabel('Sub-Category')
plt.xlabel('Market')
plt.show()
```

```
<Figure size 1400x1000 with 0 Axes>
```



In [207…

```
cd /content
```

```
/content
```

In [208…

```
!jupyter nbconvert --to html /content/CITI.ipynb
```

```
[NbConvertApp] Converting notebook /content/CITI.ipynb to html
[NbConvertApp] Writing 1537564 bytes to /content/CITI.html
```