

1. Revenue and Profit Analysis

Revenue and Profit Analysis:

- 1.What are the monthly and yearly revenue and profit trends?
- 2.Which product categories contribute the most to overall profit?
- 3.What are the profit margins for different product categories and sub-categories?
- 4.How do seasonal sales patterns affect revenue and profit?

What are the top-performing products in terms of revenue and profit?

what are the monthly and yearly revenue and profile

```
# prompt: rename payment_value to total_revenue from revenue_data_set

revenue_data_set.rename(columns={'payment_value': 'total_revenue'},
inplace=True)

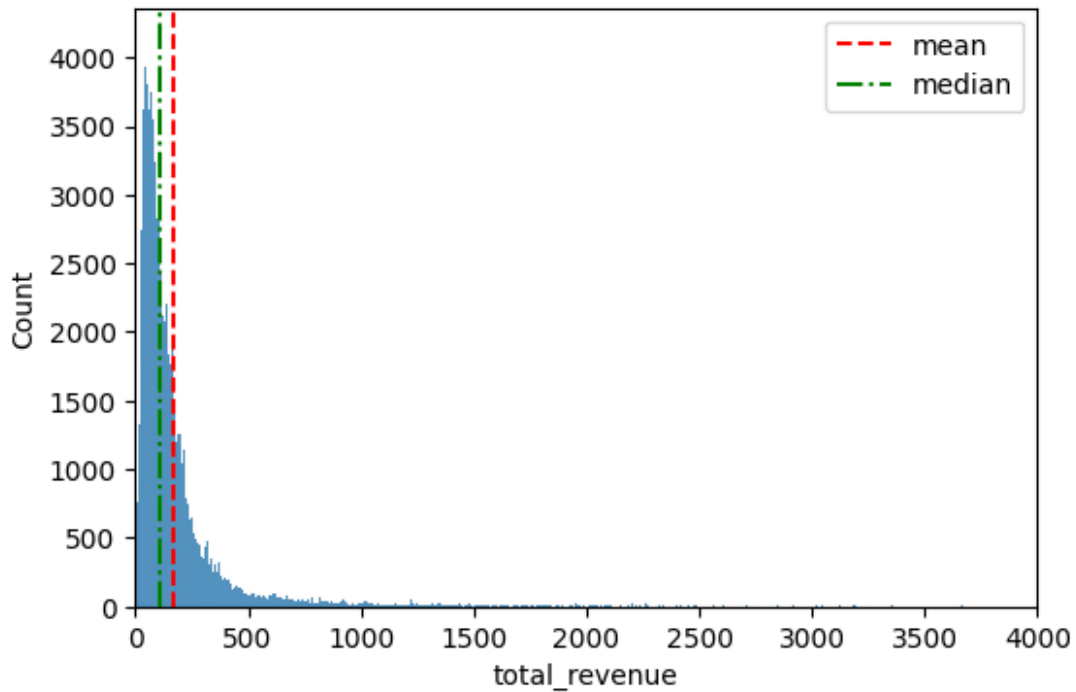
revenue_data_set['profit'] = revenue_data_set['total_revenue'] -
revenue_data_set['price']

data_revenue_profit_date =
revenue_data_set.merge(order_dateset[['order_id', 'customer_id', 'order_
status', 'order_purchase_timestamp', 'order_estimated_delivery_date']],
on='order_id')

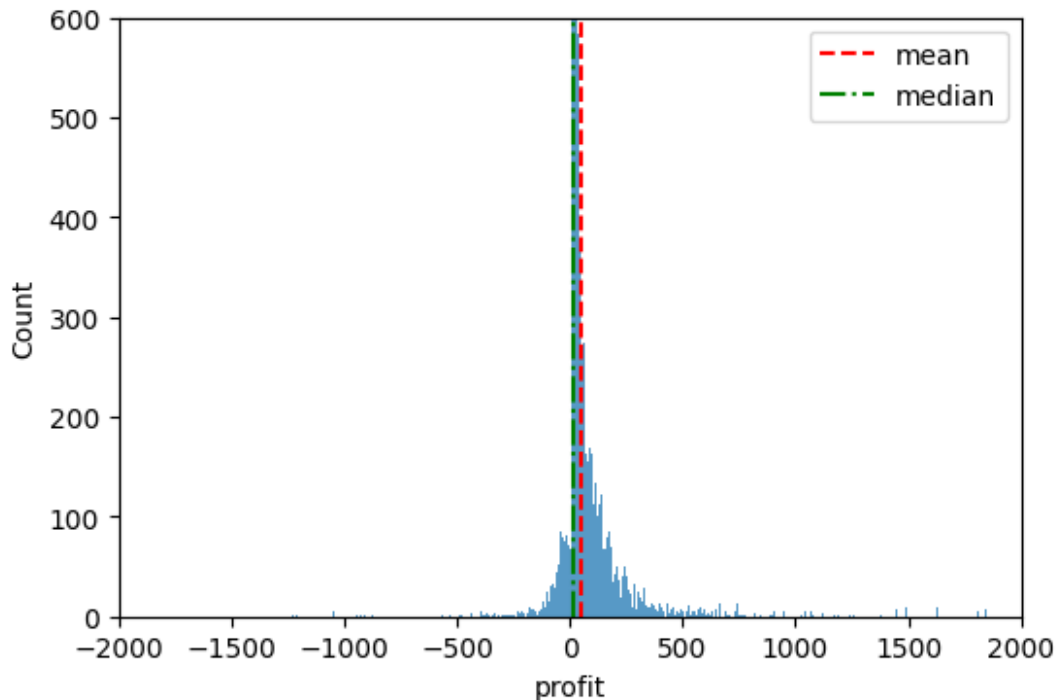
# Convert order_purchase_timestamp to datetime format if it's not
already
data_revenue_profit_date['order_purchase_timestamp'] =
pd.to_datetime(data_revenue_profit_date['order_purchase_timestamp'])

# Extract month-year and year from the timestamp
data_revenue_profit_date['month_year'] =
data_revenue_profit_date['order_purchase_timestamp'].dt.to_period('M')
data_revenue_profit_date['year'] =
data_revenue_profit_date['order_purchase_timestamp'].dt.year

#Displaying the distribution of total_revenue column
plt.figure(figsize=(6, 4))
sns.histplot(data=data_revenue_profit_date, x='total_revenue')
plt.axvline(x=data_revenue_profit_date.total_revenue.mean(),
color='r', linestyle='--', label='mean')
plt.axvline(x=data_revenue_profit_date.total_revenue.median(),
color='g', linestyle='-.', label='median')
plt.xlim(0,4000)
plt.legend()
plt.show()
```



```
#Displaying the distribution of total_profit column
plt.figure(figsize=(6, 4))
sns.histplot(data=data_revenue_profit_date, x='profit')
plt.axvline(x=data_revenue_profit_date.profit.mean(), color='r',
linestyle='--', label='mean')
plt.axvline(x=data_revenue_profit_date.profit.median(), color='g',
linestyle='-.', label='median')
plt.xlim(-2000, 2000)
plt.ylim(0, 600)
plt.legend()
plt.show()
```



```
# Group by month-year and calculate the sum of revenue and profit
monthly_summary = data_revenue_profit_date.groupby('month_year').agg({
    'total_revenue': 'sum',
    'profit': 'sum'
}).reset_index()
```

```
monthly_summary.columns = ['Month-Year', 'Total Revenue', 'Total Profit']
```

```
monthly_summary
```

```
{
  "summary": {
    "\n  \"name\": \"monthly_summary\",
    "\n  \"rows\": 24,
    \"fields\": [
      {
        \"column\": \"Month-Year\",
        \"properties\": {
          \"dtype\": \"period[M]\",
          \"num_unique_values\": 24,
          \"samples\": [
            \"2017-06\",
            \"2018-02\",
            \"2016-09\"
          ],
          \"semantic_type\": \"\",
          \"description\": \"\"
        }
      },
      {
        \"column\": \"Total Revenue\",
        \"properties\": {
          \"dtype\": \"number\",
          \"std\": 542407.5051036747,
          \"min\": 19.62,
          \"max\": 1583869.01,
          \"num_unique_values\": 24,
          \"samples\": [
            600753.27,
            1306048.8,
            347.52
          ],
          \"semantic_type\": \"\",
          \"description\": \"\"
        }
      },
      {
        \"column\": \"Total Profit\",
        \"properties\": {
          \"dtype\": \"number\",
          \"std\": 175782.41052765597,
          \"min\": 8.72,
          \"max\": 537741.17,
          \"num_unique_values\": 24,
          \"samples\": [
            147721.86,
            425949.01
          ]
        }
      }
    ]
  }
}
```

```

215.12999999999997\n        ],\n                \n            }\n        },\n        {\n            \"description\": \"\",\n            \"column\": \"Total Revenue\", \n            \"properties\": {\n                \"dtype\": \"number\", \n                \"std\": 5876405.752795574,\n                \"min\": 74281.72,\n                \"max\": 11072911.09,\n                \"num_unique_values\": 3,\n                \"samples\": [\n                    2016,\n                    2017,\n                    2018\n                ],\n                \"semantic_type\": \"\",\n                \"description\": \"\"\n            },\n            {\n                \"column\": \"Total Profit\", \n                \"properties\": {\n                    \"dtype\": \"number\", \n                    \"std\": 1777561.65827289,\n                    \"min\": 23196.2,\n                    \"max\": 3398775.13,\n                    \"num_unique_values\": 3,\n                    \"samples\": [\n                        23196.2,\n                        2677048.04,\n                        3398775.13\n                    ],\n                    \"semantic_type\": \"\",\n                    \"description\": \"\"\n                }\n            }\n        ]\n    }\"}, {\"type\":\"dataframe\",\"variable_name\":\"yearly_summary\"}

# Group by year and calculate the sum of revenue and profit
yearly_summary = data_revenue_profit_date.groupby('year').agg({
    'total_revenue': 'sum',
    'profit': 'sum'
}).reset_index()

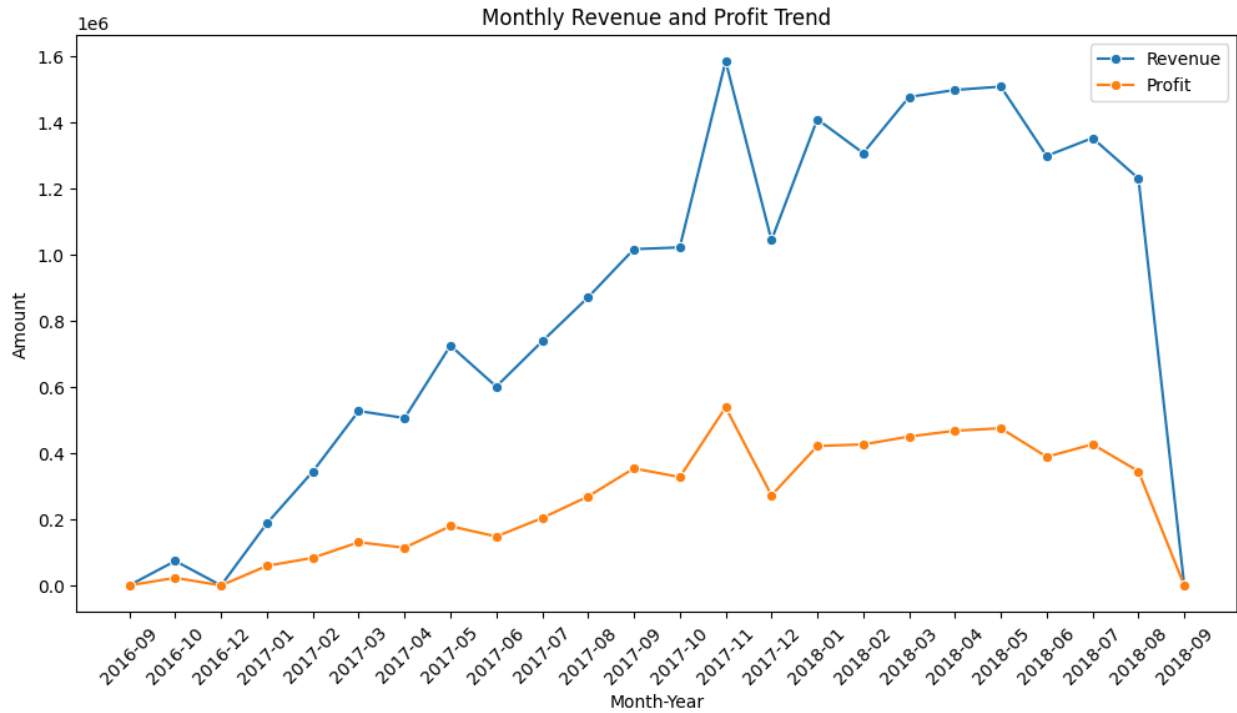
yearly_summary.columns = ['Year', 'Total Revenue', 'Total Profit']

yearly_summary

{"summary":{"name": "yearly_summary", "rows": 3, "fields": [{"column": "Year", "properties": {"dtype": "int32", "num_unique_values": 3, "samples": [2016, 2017, 2018], "semantic_type": "", "description": ""}], [{"column": "Total Revenue", "properties": {"dtype": "number", "std": 5876405.752795574, "min": 74281.72, "max": 11072911.09, "num_unique_values": 3, "samples": [74281.72, 9160941.9, 11072911.09], "semantic_type": "", "description": ""}], [{"column": "Total Profit", "properties": {"dtype": "number", "std": 1777561.65827289, "min": 23196.2, "max": 3398775.13, "num_unique_values": 3, "samples": [23196.2, 2677048.04, 3398775.13], "semantic_type": "", "description": ""}}]}, {"type":"dataframe","variable_name":"yearly_summary"}

monthly_summary['Month-Year'] = monthly_summary['Month-Year'].astype(str)

plt.figure(figsize=(12, 6))
sns.lineplot(data=monthly_summary, x='Month-Year', y='Total Revenue', marker='o', label='Revenue')
sns.lineplot(data=monthly_summary, x='Month-Year', y='Total Profit', marker='o', label='Profit')
plt.title('Monthly Revenue and Profit Trend')
plt.xlabel('Month-Year')
plt.ylabel('Amount')
plt.legend()
plt.xticks(rotation=45)
plt.show()
```

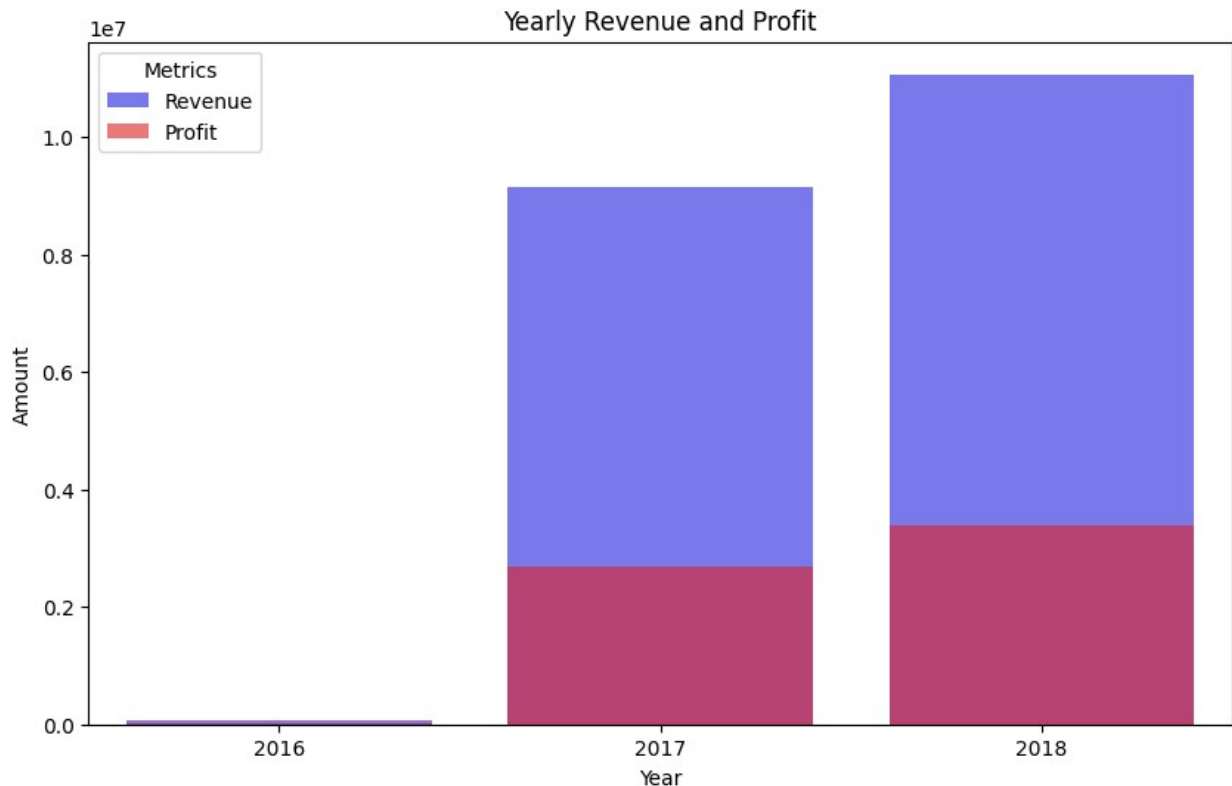


```
plt.figure(figsize=(10, 6))

# Plot Total Revenue
sns.barplot(data=yearly_summary, x='Year', y='Total Revenue',
            color='blue', alpha=0.6, label='Revenue')

# Plot Total Profit
sns.barplot(data=yearly_summary, x='Year', y='Total Profit',
            color='red', alpha=0.6, label='Profit')

plt.title('Yearly Revenue and Profit')
plt.xlabel('Year')
plt.ylabel('Amount')
plt.legend(title='Metrics')
plt.show()
```



Which product categories contribute the most to overall profit?

```
# Merge datasets to get all relevant information together
# Merging the datasets on 'product_id'
merged_data = pd.merge(revenue_data_set,
products_dataset[['product_id', 'product_category_name']],
on='product_id', how='left')

merged_data = merged_data.merge(product_category_name_translation,
on='product_category_name', how='left')

merged_data.rename(columns={'product_category_name_english':
'product_name'}, inplace=True)

merged_data['product_name'] =
merged_data['product_name'].fillna('unknown')
merged_data['payment_type'] =
merged_data['payment_type'].fillna(merged_data['payment_type'].mode()
[0])

merged_data['product_category_name'] =
merged_data['product_category_name'].fillna('unknown')

# Grouping by 'product_category_name' and summing the 'profit'
profit_by_category = merged_data.groupby('product_name')
```

```

['profit'].sum().reset_index()

# Sorting the categories by profit in descending order
profit_by_category = profit_by_category.sort_values(by='profit',
ascending=False).reset_index(drop=True)

# Displaying the top 10 product categories by profit
top_categories = profit_by_category.head(10)
top_categories

{"summary":{"\n  \"name\": \"top_categories\",\n  \"rows\": 10,\n  \"fields\": [\n    {\n      \"column\": \"product_name\",\n      \"properties\": {\n        \"dtype\": \"string\",\n        \"num_unique_values\": 10,\n        \"samples\": [\n          \"auto\",\n          \"computers_accessories\",\n          \"office_furniture\",\n          ],\n          \"semantic_type\": \"\",\n          \"description\": \"\"\n        }\n      },\n      {\n        \"column\": \"profit\",\n        \"properties\": {\n          \"dtype\": \"number\",\n          \"std\": 170994.8185333668,\n          \"min\": 176073.38,\n          \"max\": 665082.5,\n          \"num_unique_values\": 10,\n          \"samples\": [\n            235541.82,\n            643052.88,\n            360667.52999999997\n          ],\n          \"semantic_type\": \"\",\n          \"description\": \"\"\n        }\n      }\n    ]\n  }, \"type\": \"dataframe\", \"variable_name\": \"top_categories\"}

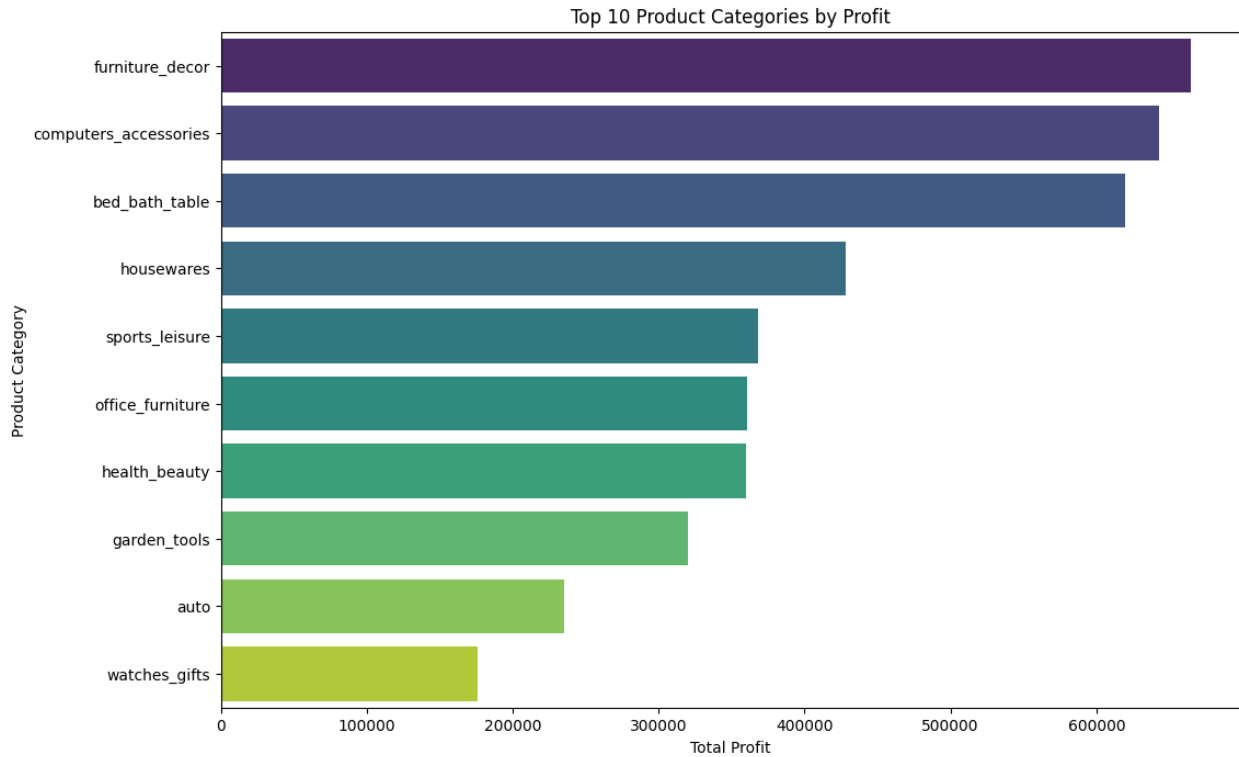
```

Plot the top product categories contributing to overall profit

```

plt.figure(figsize=(12, 8))
sns.barplot(x='profit', y='product_name', data=top_categories, hue =
'product_name',palette='viridis')
plt.title('Top 10 Product Categories by Profit')
plt.xlabel('Total Profit')
plt.ylabel('Product Category')
plt.show()

```

What are the profit margins for different product categories and sub-categories?**bold text**

What are the profit margins for different product categories and sub-categories?

```
# Calculate profit margin for each product
merged_data['profit_margin'] = (merged_data['profit'] /
merged_data['total_revenue']) * 100

# Group by product category and calculate the average profit margin
category_profit_margin = merged_data.groupby('product_name').agg({
    'profit_margin': 'mean',
    'profit': 'sum',
    'total_revenue': 'sum'
}).reset_index()

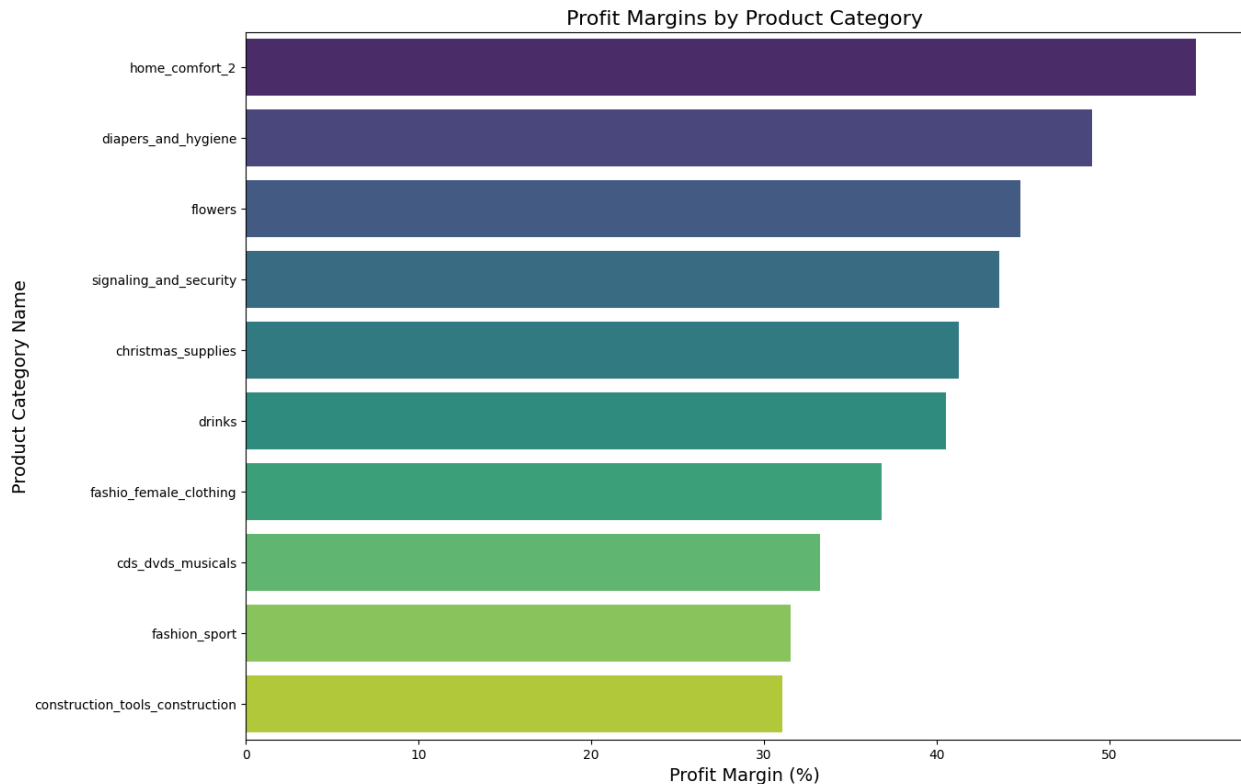
# Sort by profit margin
category_profit_margin_sorted =
category_profit_margin.sort_values(by='profit_margin',
ascending=False)

plt.figure(figsize=(14, 10))
sns.barplot(y='product_name', x='profit_margin',
data=category_profit_margin_sorted.head(10), hue =
'product_name',palette='viridis')

# Add labels and title
plt.title('Profit Margins by Product Category', fontsize=16)
```

```
plt.xlabel('Profit Margin (%)', fontsize=14)
plt.ylabel('Product Category Name', fontsize=14)

# Display the plot
plt.show()
```



How do seasonal sales patterns affect revenue and profit?

```
data_revenue_profit_date_copy = data_revenue_profit_date.copy()

# Convert the order_purchase_timestamp to datetime if it's not already
data_revenue_profit_date['order_purchase_timestamp'] =
pd.to_datetime(data_revenue_profit_date['order_purchase_timestamp'])

# Extracting the month and season from the timestamp
data_revenue_profit_date['month'] =
data_revenue_profit_date['order_purchase_timestamp'].dt.month
data_revenue_profit_date['season'] =
data_revenue_profit_date['order_purchase_timestamp'].dt.month % 12 //
3 + 1
season_labels = {1: 'Winter', 2: 'Spring', 3: 'Summer', 4: 'Fall'}
data_revenue_profit_date['season'] =
data_revenue_profit_date['season'].map(season_labels)

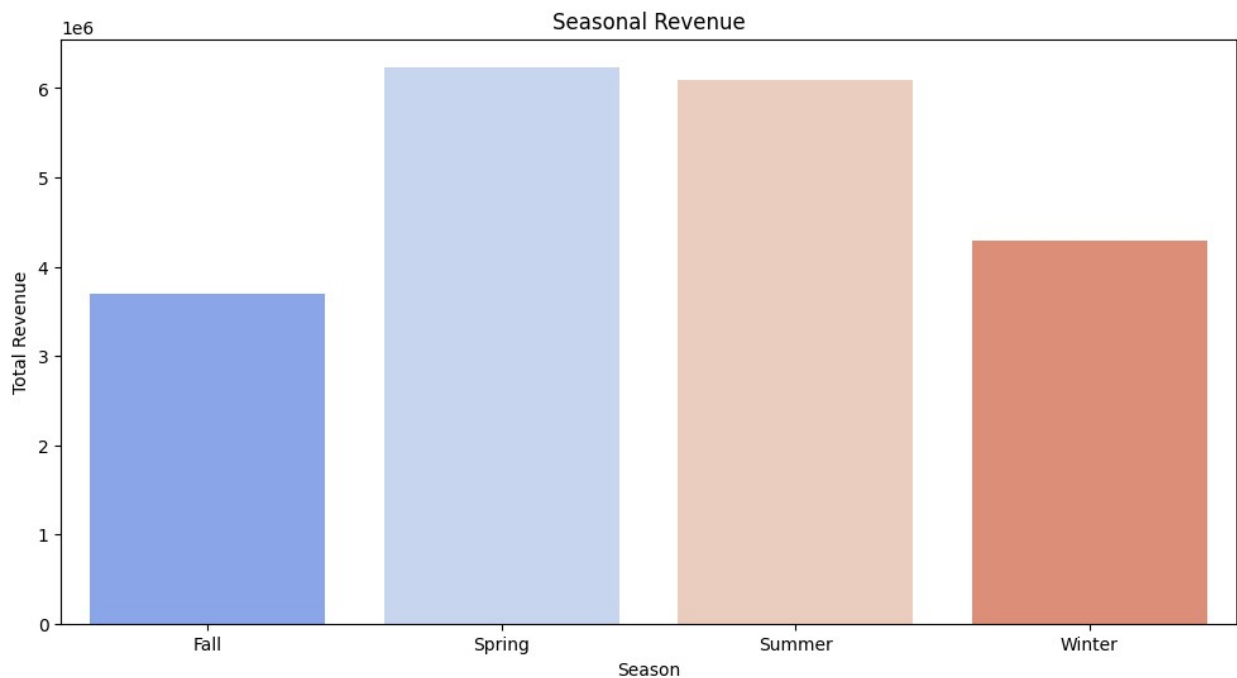
# Aggregate revenue and profit by season
seasonal_summary = data_revenue_profit_date.groupby('season').agg({
```

```

        'total_revenue': 'sum',
        'profit': 'sum'
    }).reset_index()

# Plotting the seasonal revenue and profit
plt.figure(figsize=(12, 6))
sns.barplot(x='season', y='total_revenue', data=seasonal_summary, hue
= 'season', palette='coolwarm')
plt.title('Seasonal Revenue')
plt.xlabel('Season')
plt.ylabel('Total Revenue')
plt.show()

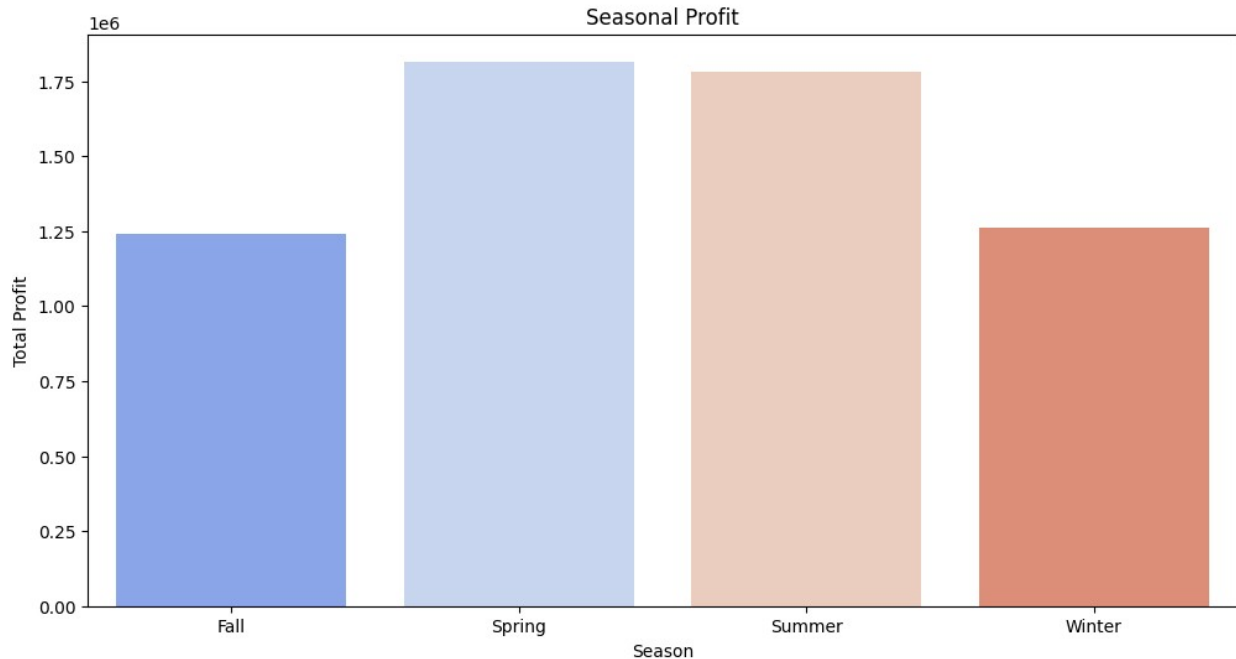
```



```

plt.figure(figsize=(12, 6))
sns.barplot(x='season', y='profit', data=seasonal_summary, hue =
'season', palette='coolwarm')
plt.title('Seasonal Profit')
plt.xlabel('Season')
plt.ylabel('Total Profit')
plt.show()

```



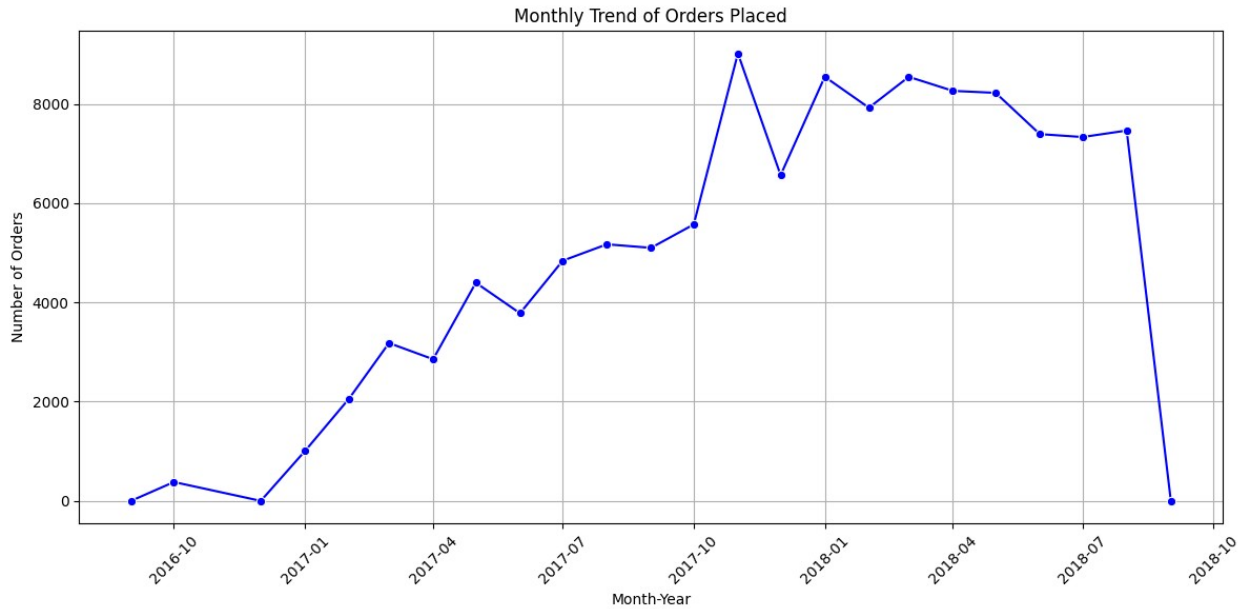
```
# Convert the order_purchase_timestamp to datetime if it's not already
data_revenue_profit_date['order_purchase_timestamp'] =
pd.to_datetime(data_revenue_profit_date['order_purchase_timestamp'])

# Extract year and month from the timestamp
data_revenue_profit_date['year_month'] =
data_revenue_profit_date['order_purchase_timestamp'].dt.to_period('M')

# Count the number of orders for each month
monthly_order_counts =
data_revenue_profit_date.groupby('year_month').size().reset_index(name
='order_count')

# Convert 'year_month' back to a datetime format for plotting
monthly_order_counts['year_month'] =
monthly_order_counts['year_month'].dt.to_timestamp()

# Plotting the trend of orders placed monthly
plt.figure(figsize=(12, 6))
sns.lineplot(data=monthly_order_counts, x='year_month',
y='order_count', marker='o', color='b')
plt.title('Monthly Trend of Orders Placed')
plt.xlabel('Month-Year')
plt.ylabel('Number of Orders')
plt.xticks(rotation=45)
plt.grid(True)
plt.tight_layout()
plt.show()
```



What are the top-performing products in terms of revenue and profit?

```
# Group by 'product_id' and aggregate total revenue and profit
product_performance =
merged_data.groupby('product_category_name').agg({
    'total_revenue': 'sum',
    'profit': 'sum'
}).reset_index()

# Sort the products by revenue and profit
top_products_by_revenue =
product_performance.sort_values(by='total_revenue', ascending=False)
top_products_by_profit = product_performance.sort_values(by='profit',
ascending=False)

# Define the number of top products to display
top_n = 10

# Top products by revenue
top_n_products_revenue = top_products_by_revenue.head(top_n)

# Top products by profit
top_n_products_profit = top_products_by_profit.head(top_n)

top_n_products_revenue

{"summary": "{\n  \"name\": \"top_n_products_revenue\",\n  \"rows\":\n  10,\n  \"fields\": [\n    {\n      \"column\":\n      \"product_category_name\",\n      \"properties\": {\n        \"dtype\": \"string\",\n        \"num_unique_values\": 10,\n        \"samples\": [\n          \"ferramentas_jardim\",
```

```

\"beleza_saude\", \n          \"esporte_lazer\" \n          ], \n
\"semantic_type\": \"\", \n          \"description\": \"\" \n          } \n
n      }, \n      { \n          \"column\": \"total_revenue\", \n
\"properties\": { \n          \"dtype\": \"number\", \n          \"std\": 356326.3474557149, \n          \"min\": 779698.0, \n          \"max\": 1712553.67, \n          \"num_unique_values\": 10, \n          \"samples\": [ \n          838280.75, \n          1657373.12, \n          1392127.56 \n          ], \n          \"semantic_type\": \"\", \n          \"description\": \"\" \n          } \n      }, \n      { \n          \"column\": \"profit\", \n          \"properties\": { \n          \"dtype\": \"number\", \n          \"std\": 195550.31119272977, \n          \"min\": 117388.51, \n          \"max\": 665082.5, \n          \"num_unique_values\": 10, \n          \"samples\": [ \n          320063.21, \n          360017.32, \n          368131.22 \n          ], \n          \"semantic_type\": \"\", \n          \"description\": \"\" \n          } \n      } \n  ] \n} \n\", \"type\": \"dataframe\", \"variable_name\": \"top_n_products_revenue\"}

```

top_n_products_profit

```

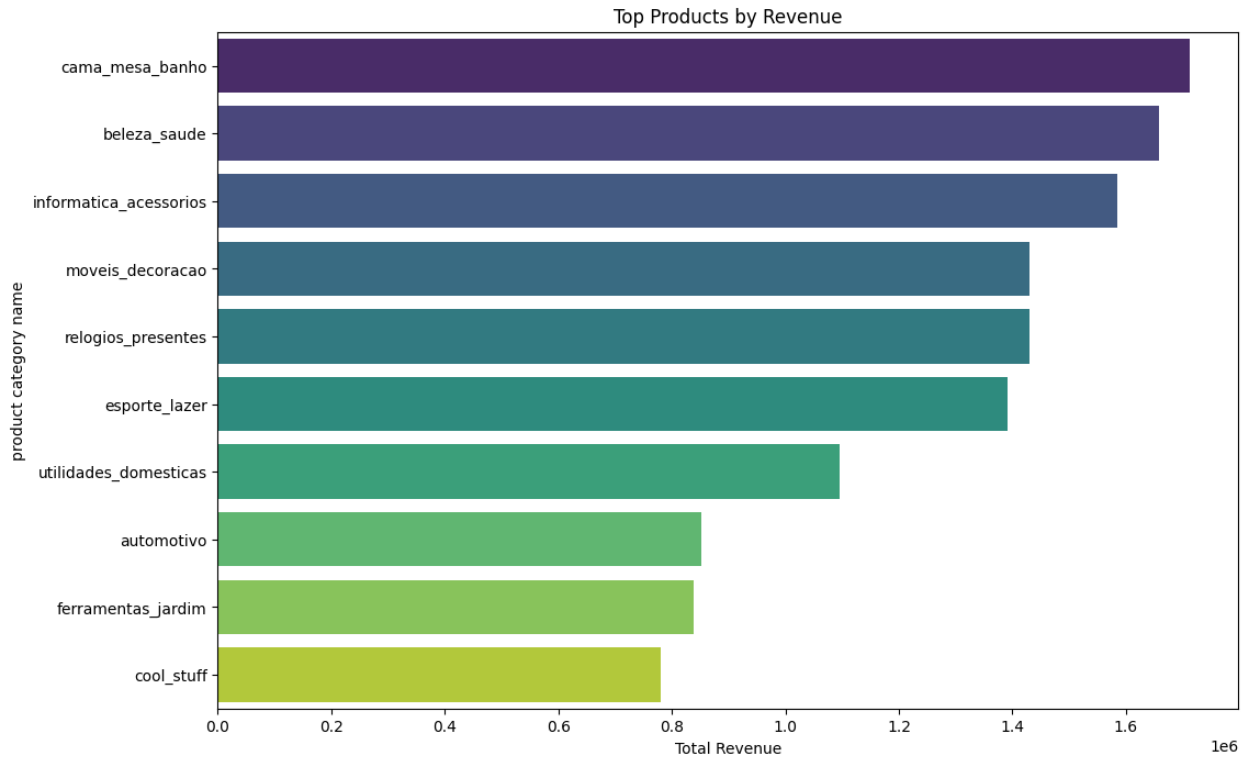
{ \"summary\": { \n  \"name\": \"top_n_products_profit\", \n  \"rows\": 10, \n  \"fields\": [ \n    { \n      \"column\": \"product_category_name\", \n      \"properties\": { \n        \"dtype\": \"string\", \n        \"num_unique_values\": 10, \n        \"samples\": [ \n          \"automotivo\", \n          \"informatica_acessorios\", \n          \"moveis_escritorio\" \n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      } \n    }, \n    { \n      \"column\": \"total_revenue\", \n      \"properties\": { \n        \"dtype\": \"number\", \n        \"std\": 378712.46168845886, \n        \"min\": 646826.49, \n        \"max\": 1712553.67, \n        \"num_unique_values\": 10, \n        \"samples\": [ \n          852294.33, \n          1585330.45, \n          646826.49 \n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      } \n    }, \n    { \n      \"column\": \"profit\", \n      \"properties\": { \n        \"dtype\": \"number\", \n        \"std\": 170994.8185333668, \n        \"min\": 176073.38, \n        \"max\": 665082.5, \n        \"num_unique_values\": 10, \n        \"samples\": [ \n          235541.82, \n          643052.88, \n          360667.52999999997 \n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      } \n    } \n  ] \n} \n\", \"type\": \"dataframe\", \"variable_name\": \"top_n_products_profit\"}

```

```

plt.figure(figsize=(12, 8))
sns.barplot(x='total_revenue', y='product_category_name',
data=top_n_products_revenue, hue =
'product_category_name', palette='viridis')
plt.title('Top Products by Revenue')
plt.xlabel('Total Revenue')
plt.ylabel('product category name')
plt.show()

```



```
plt.figure(figsize=(12, 8))
sns.barplot(x='profit', y='product_category_name',
data=top_n_products_profit, hue = 'product_category_name',
palette='viridis')
plt.title('Top Products by Profit')
plt.xlabel('Total Profit')
plt.ylabel('product category name')
plt.show()
```

