Superstore_SalesAnalysis

October 15, 2024

SUPER STORE SALES DATA ANALYSIS

0.1 IMPORTING REQUIRED LIBRARIES

[1]:
```
import pandas as pd

import matplotlib.pyplot as plt
%matplotlib inline

import seaborn as sns
```

0.2 IMPORTING THE DATASET

[2]:
```
df =  pd.read_excel("superstore_sales.xlsx")
```

0.3 DATA AUDIT

you can't make your data work for you until you know what data you're talking about. To get a quick idea of what the data looks like, we can call the head function on the data frame. By default, this returns the top five rows, but it can take in a parameter of how many rows to return.

[3]:
```
# first five rows of the dataset
df.head(5)
```

[3]:
```
          order_id order_date  ship_date       ship_mode    customer_name  \
0     AG-2011-2040 2011-01-01 2011-01-06  Standard Class  Toby Braunhardt
1    IN-2011-47883 2011-01-01 2011-01-08  Standard Class      Joseph Holt
2     HU-2011-1220 2011-01-01 2011-01-05    Second Class    Annie Thurman
3  IT-2011-3647632 2011-01-01 2011-01-05    Second Class     Eugene Moren
4    IN-2011-47883 2011-01-01 2011-01-08  Standard Class      Joseph Holt

       segment            state    country  market   region  …  \
0     Consumer      Constantine    Algeria  Africa   Africa  …
1     Consumer  New South Wales  Australia    APAC  Oceania  …
2     Consumer         Budapest    Hungary    EMEA     EMEA  …
3  Home Office        Stockholm     Sweden      EU    North  …
4     Consumer  New South Wales  Australia    APAC  Oceania  …
```

```
          category sub_category                  product_name    sales  \
0  Office Supplies      Storage           Tenex Lockers, Blue  408.300
1  Office Supplies     Supplies      Acme Trimmer, High Speed  120.366
2  Office Supplies      Storage       Tenex Box, Single Width   66.120
3  Office Supplies        Paper  Enermax Note Cards, Premium   44.865
4        Furniture  Furnishings   Eldon Light Bulb, Duo Pack  113.670

   quantity  discount   profit  shipping_cost order_priority  year
0         2       0.0  106.140          35.46         Medium  2011
1         3       0.1   36.036           9.72         Medium  2011
2         4       0.0   29.640           8.17           High  2011
3         3       0.5  -26.055           4.82           High  2011
4         5       0.1   37.770           4.70         Medium  2011

[5 rows x 21 columns]
```

[4]:
```
# Last five rows of the dataset
df.tail()
```

[4]:
```
             order_id order_date  ship_date       ship_mode     customer_name  \
51285  CA-2014-115427 2014-12-31 2015-01-04  Standard Class        Erica Bern
51286    MO-2014-2560 2014-12-31 2015-01-05  Standard Class         Liz Preis
51287  MX-2014-110527 2014-12-31 2015-01-02    Second Class  Charlotte Melton
51288  MX-2014-114783 2014-12-31 2015-01-06  Standard Class     Tamara Dahlen
51289  CA-2014-156720 2014-12-31 2015-01-04  Standard Class      Jill Matthias

         segment             state        country  market   region  …  \
51285  Corporate        California  United States      US     West  …
51286   Consumer  Souss-Massa-Draâ        Morocco  Africa   Africa  …
51287   Consumer           Managua      Nicaragua   LATAM  Central  …
51288   Consumer         Chihuahua         Mexico   LATAM    North  …
51289   Consumer          Colorado  United States      US     West  …

              category sub_category  \
51285  Office Supplies      Binders
51286  Office Supplies      Binders
51287  Office Supplies       Labels
51288  Office Supplies       Labels
51289  Office Supplies    Fasteners

                                   product_name    sales  quantity  \
51285  Cardinal Slant-D Ring Binder, Heavy Gauge Vinyl   13.904         2
51286          Wilson Jones Hole Reinforcements, Clear    3.990         1
51287           Hon Color Coded Labels, 5000 Label Set   26.400         3
51288           Hon Legal Exhibit Labels, Alphabetical    7.120         1
51289                              Bagged Rubber Bands    3.024         3
```

```
       discount   profit  shipping_cost  order_priority  year
51285       0.2   4.5188          0.890          Medium  2014
51286       0.0   0.4200          0.490          Medium  2014
51287       0.0  12.3600          0.350          Medium  2014
51288       0.0   0.5600          0.199          Medium  2014
51289       0.2  -0.6048          0.170          Medium  2014

[5 rows x 21 columns]
```

[5]:
```
# Shape of the dataset
df.shape
```

[5]:
```
(51290, 21)
```

[6]:
```
# Columns present in the dataset
df.columns
```

[6]:
```
Index(['order_id', 'order_date', 'ship_date', 'ship_mode', 'customer_name',
       'segment', 'state', 'country', 'market', 'region', 'product_id',
       'category', 'sub_category', 'product_name', 'sales', 'quantity',
       'discount', 'profit', 'shipping_cost', 'order_priority', 'year'],
      dtype='object')
```

[7]:
```
# A concise summary of the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51290 entries, 0 to 51289
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   order_id        51290 non-null  object
 1   order_date      51290 non-null  datetime64[ns]
 2   ship_date       51290 non-null  datetime64[ns]
 3   ship_mode       51290 non-null  object
 4   customer_name   51290 non-null  object
 5   segment         51290 non-null  object
 6   state           51290 non-null  object
 7   country         51290 non-null  object
 8   market          51290 non-null  object
 9   region          51290 non-null  object
 10  product_id      51290 non-null  object
 11  category        51290 non-null  object
 12  sub_category    51290 non-null  object
 13  product_name    51290 non-null  object
 14  sales           51290 non-null  float64
 15  quantity        51290 non-null  int64
 16  discount        51290 non-null  float64
 17  profit          51290 non-null  float64
 18  shipping_cost   51290 non-null  float64
 19  order_priority  51290 non-null  object
 20  year            51290 non-null  int64
dtypes: datetime64[ns](2), float64(4), int64(2), object(13)
memory usage: 8.2+ MB
```

[8]:
```
# Checking missing values
df.isnull().sum()
```

[8]:
```
order_id          0
order_date        0
ship_date         0
ship_mode         0
customer_name     0
segment           0
state             0
country           0
market            0
region            0
product_id        0
category          0
sub_category      0
product_name      0
sales             0
quantity          0
discount          0
profit            0
shipping_cost     0
order_priority    0
year              0
dtype: int64
```

[9]:
```
# Generating descriptive statistics summary
df.describe()
```

[9]:
```
                          order_date                      ship_date  \
count                          51290                          51290
mean   2013-05-11 21:26:49.155780864  2013-05-15 20:42:42.745174528
min              2011-01-01 00:00:00            2011-01-03 00:00:00
25%              2012-06-19 00:00:00            2012-06-23 00:00:00
50%              2013-07-08 00:00:00            2013-07-12 00:00:00
75%              2014-05-22 00:00:00            2014-05-26 00:00:00
max              2014-12-31 00:00:00            2015-01-07 00:00:00
std                              NaN                            NaN

              sales      quantity      discount        profit  shipping_cost  \
count  51290.000000  51290.000000  51290.000000  51290.000000   51290.000000
```

```
mean      246.490581       3.476545       0.142908      28.641740       26.375818
min         0.444000       1.000000       0.000000   -6599.978000        0.002000
25%        30.758625       2.000000       0.000000       0.000000        2.610000
50%        85.053000       3.000000       0.000000       9.240000        7.790000
75%       251.053200       5.000000       0.200000      36.810000       24.450000
max     22638.480000      14.000000       0.850000    8399.976000      933.570000
std       487.565361       2.278766       0.212280     174.424113       57.296810

               year
count  51290.000000
mean    2012.777208
min     2011.000000
25%     2012.000000
50%     2013.000000
75%     2014.000000
max     2014.000000
std        1.098931
```

---

1 EXPLORATORY DATA ANALYSIS

Q1. what is the total sales trend?

[10]:
```
df['order_date'].min()
```

[10]:
```
Timestamp('2011-01-01 00:00:00')
```

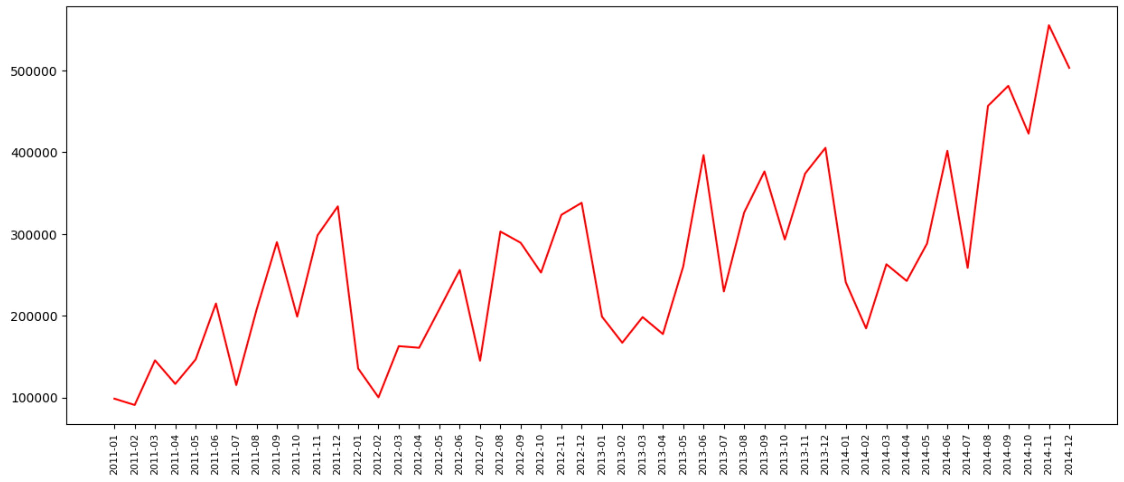
[11]:
```
df['order_date'].max()
```

[11]:
```
Timestamp('2014-12-31 00:00:00')
```

[12]:
```
# Getting month year from order_date
df['month_year'] = df['order_date'].apply(lambda x: x.strftime('%Y-%m'))
```

[13]:
```
# grouping month_year by sales
df_temp = df.groupby('month_year')['sales'].sum().reset_index()
```

[14]:
```
# Setting the figure size
plt.figure(figsize=(15,6))
plt.plot(df_temp['month_year'], df_temp['sales'], color = 'red')
plt.xticks(rotation = 'vertical',size = 8)
plt.show()
```



Q2. which are the top 10 products by sales?

[15]:
```
# Grouping products by sales
product_sales = pd.DataFrame(df.groupby('product_name')['sales'].sum())
```

[16]:
```
# Sorting the dataframe in descending order
product_sales = product_sales.sort_values('sales',ascending = False)
```

[17]:
```
# Top 10 products by sales
product_sales[:10]
```

[17]:
```
                                                       sales
product_name
Apple Smart Phone, Full Size                      86935.7786
Cisco Smart Phone, Full Size                      76441.5306
Motorola Smart Phone, Full Size                   73156.3030
Nokia Smart Phone, Full Size                      71904.5555
Canon imageCLASS 2200 Advanced Copier             61599.8240
Hon Executive Leather Armchair, Adjustable        58193.4841
Office Star Executive Leather Armchair, Adjustable  50661.6840
Harbour Creations Executive Leather Armchair, A…  50121.5160
Samsung Smart Phone, Cordless                     48653.4600
Nokia Smart Phone, with Caller ID                 47877.7857
```

Q3. which are the most selling products?

[18]:
```
# Grouping products by Quantity
most_sales_prod = pd.DataFrame(df.groupby('product_name')['quantity'].sum())
```

[19]:
```
# Sorting the dataframe in descending order
most_sales_prod = most_sales_prod.sort_values('quantity', ascending=False)
```

[20]:
```
# Most selling products
most_sales_prod[:10]
```

[20]:
```
                                        quantity
product_name
Staples                                      876
Cardinal Index Tab, Clear                    337
Eldon File Cart, Single Width                321
Rogers File Cart, Single Width               262
Sanford Pencil Sharpener, Water Color        259
Stockwell Paper Clips, Assorted Sizes        253
Avery Index Tab, Clear                       252
Ibico Index Tab, Clear                       251
Smead File Cart, Single Width                250
Stanley Pencil Sharpener, Water Color        242
```

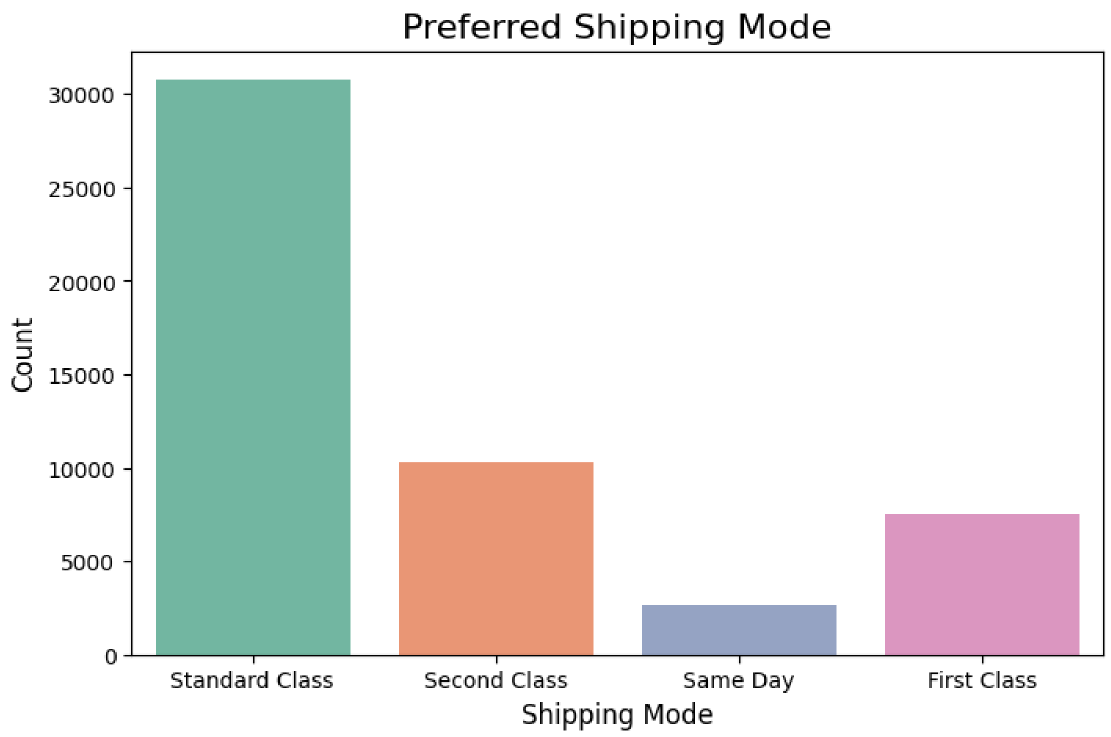
Q4. What is the most preferred ship mode?

[21]:
```
# Setting the figure size
plt.figure(figsize=(8, 5))

# Create a colorful countplot with hue and disable the legend
sns.countplot(x='ship_mode', data=df, hue='ship_mode', palette='Set2',␣
 ↪legend=False)

# Adding title and labels
plt.title('Preferred Shipping Mode', fontsize=16)
plt.xlabel('Shipping Mode', fontsize=12)
plt.ylabel('Count', fontsize=12)

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



1.0.1 Explanation:

- hue='ship_mode': Assigns the color to the categories of the ship_mode column.
- legend=False: Since hue creates a legend by default, we turn it off to keep the plot simple.

Q5. which are the most profitable category and sub-category?

[22]:
```
# Group by 'category' and 'sub_category', and sum the 'profit'
cat_subcat = pd.DataFrame(df.groupby(['category', 'sub_category'])['profit'].
↪sum().reset_index())

# Sort by 'category' and 'profit' in descending order
cat_subcat = cat_subcat.sort_values(['category', 'profit'], ascending=False)
cat_subcat
```

[22]:
```
          category sub_category         profit
14      Technology      Copiers   258567.54818
16      Technology       Phones   216717.00580
13      Technology  Accessories   129626.30620
15      Technology     Machines    58867.87300
4  Office Supplies   Appliances   141680.58940
11 Office Supplies      Storage   108461.48980
6  Office Supplies      Binders    72449.84600
10 Office Supplies        Paper    59207.68270
```

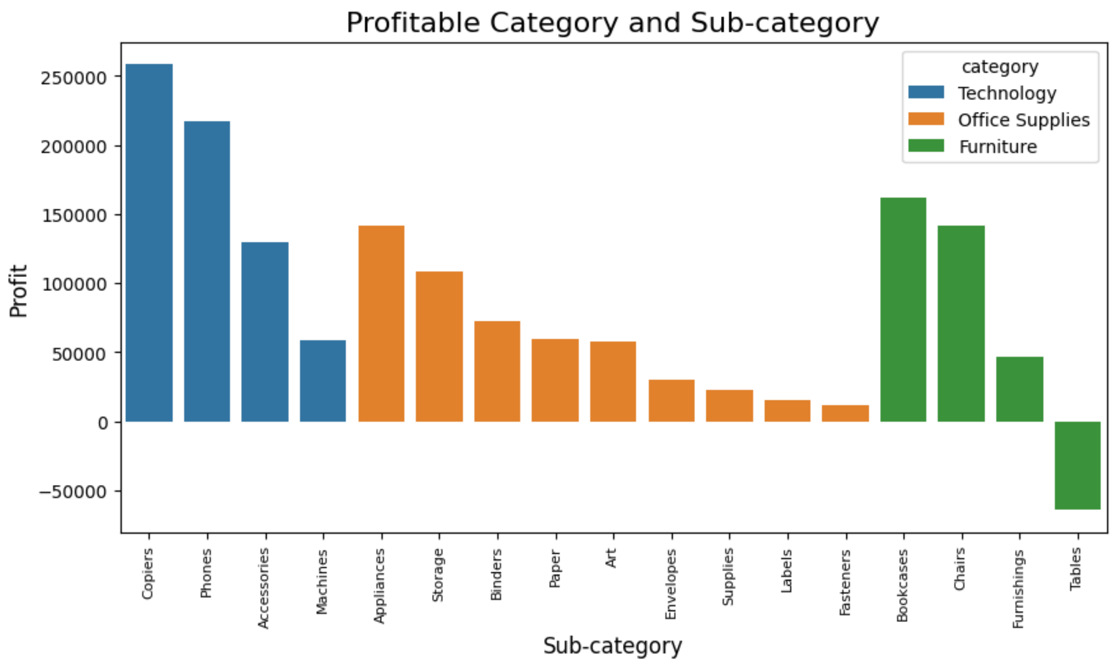
```
5   Office Supplies          Art   57953.91090
7   Office Supplies    Envelopes   29601.11630
12  Office Supplies     Supplies   22583.26310
9   Office Supplies       Labels   15010.51200
8   Office Supplies    Fasteners   11525.42410
0         Furniture    Bookcases  161924.41950
1         Furniture       Chairs  141973.79750
2         Furniture  Furnishings   46967.42550
3         Furniture       Tables  -64083.38870
```

[23]:
```python
plt.figure(figsize=(10, 5))

sns.barplot(x = 'sub_category', y = 'profit', hue = 'category', data=cat_subcat)
plt.xticks(rotation = 'vertical',size = 8)

# Adding title and labels
plt.title('Profitable Category and Sub-category', fontsize=16)
plt.xlabel('Sub-category', fontsize=12)
plt.ylabel('Profit', fontsize=12)

plt.show()
```



Q6. Which are the Top 10 Countries?

```
[24]: # Group by country and sum the sales, then sort in descending order and take␣
       ↪the top 10
      top_countries = df.groupby('country')['sales'].sum().
       ↪sort_values(ascending=False).head(10)

      # Convert the sales to a more readable format
      top_countries_formatted = top_countries.apply(lambda x: f"${x:,.2f}")

      # Display the result
      countries_sales = pd.DataFrame(top_countries_formatted)
      countries_sales
```

```
[24]:                         sales
      country
      United States   $2,297,200.86
      Australia         $925,235.85
      France            $858,931.08
      China             $700,562.03
      Germany           $628,840.03
      Mexico            $622,590.62
      India             $589,650.10
      United Kingdom    $528,576.30
      Indonesia         $404,887.50
      Brazil            $361,106.42
```

Q7. Which are the Top 10 Costumers?

```
[25]: # Group by customer name and sum the sales, then sort in descending order and␣
       ↪take the top 10 customers
      top_customers = df.groupby('customer_name')['sales'].sum().
       ↪sort_values(ascending=False).head(10)

      # Convert the sales to a more readable format
      top_customers_formatted = pd.DataFrame(top_customers.apply(lambda x: f"${x:,.
       ↪2f}"))

      top_customers_formatted
```

```
[25]:                          sales
      customer_name
      Tom Ashbrook         $40,488.07
      Tamara Chand         $37,457.33
      Greg Tran            $35,550.95
      Christopher Conant  $35,187.08
      Sean Miller          $35,170.93
      Bart Watters         $32,310.45
      Natalie Fritzler     $31,781.26
```

```
Fred Hopkins        $30,400.67
Jane Waco           $30,288.45
Hunter Lopez        $30,243.57
```

~~~THE END~~~

```
[ ]:
```
~~~