fzgfal6ep

January 25, 2025

[1]:
```
import pandas as pd #pandas (pd): Data manipulation ke liye use hota hai, jaise␣
 ↪CSV file load karna aur process karna.

import plotly.express as px #plotly.express (px): Data visualization library jo␣
 ↪easy aur quick plots banata hai.
import plotly.graph_objects as go #plotly.graph_objects (go): Advanced and␣
 ↪customizable graphs banane ke liye
import plotly.io as pio   #plotly.io (pio): Graph templates ko customize karne␣
 ↪ke liye
import plotly.colors as colors
pio.templates.default = "plotly_white"  #pio.templates.default = "plotly_white":
 ↪ Default theme white rakha gaya hai graphs ke liye
```

[2]:
```
data = pd.read_csv("Sample - Superstore.csv", encoding='latin-1')␣
 ↪#encoding='latin-1': Special characters ko properly read karne ke liye␣
 ↪encoding use hui hai
```

[3]:
```
data
```

[3]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	\
0	1	CA-2016-152156	11/8/2016	11/11/2016	Second Class	
1	2	CA-2016-152156	11/8/2016	11/11/2016	Second Class	
2	3	CA-2016-138688	6/12/2016	6/16/2016	Second Class	
3	4	US-2015-108966	10/11/2015	10/18/2015	Standard Class	
4	5	US-2015-108966	10/11/2015	10/18/2015	Standard Class	
…	…	…	…	…	…	
9989	9990	CA-2014-110422	1/21/2014	1/23/2014	Second Class	
9990	9991	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	
9991	9992	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	
9992	9993	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	
9993	9994	CA-2017-119914	5/4/2017	5/9/2017	Second Class	

	Customer ID	Customer Name	Segment	Country	City	\
0	CG-12520	Claire Gute	Consumer	United States	Henderson	
1	CG-12520	Claire Gute	Consumer	United States	Henderson	
2	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	
3	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	

```
4       SO-20335    Sean O'Donnell   Consumer  United States  Fort Lauderdale
…          …              …          …            …               …
9989    TB-21400  Tom Boeckenhauer   Consumer  United States           Miami
9990    DB-13060       Dave Brooks   Consumer  United States       Costa Mesa
9991    DB-13060       Dave Brooks   Consumer  United States       Costa Mesa
9992    DB-13060       Dave Brooks   Consumer  United States       Costa Mesa
9993    CC-12220      Chris Cortes   Consumer  United States      Westminster

      … Postal Code  Region       Product ID         Category Sub-Category  \
0     …       42420   South  FUR-BO-10001798        Furniture    Bookcases
1     …       42420   South  FUR-CH-10000454        Furniture       Chairs
2     …       90036    West  OFF-LA-10000240  Office Supplies       Labels
3     …       33311   South  FUR-TA-10000577        Furniture       Tables
4     …       33311   South  OFF-ST-10000760  Office Supplies      Storage
…   …          …     …              …               …          …
9989  …       33180   South  FUR-FU-10001889        Furniture  Furnishings
9990  …       92627    West  FUR-FU-10000747        Furniture  Furnishings
9991  …       92627    West  TEC-PH-10003645       Technology       Phones
9992  …       92627    West  OFF-PA-10004041  Office Supplies        Paper
9993  …       92683    West  OFF-AP-10002684  Office Supplies   Appliances

                                             Product Name     Sales  Quantity  \
0                     Bush Somerset Collection Bookcase  261.9600         2
1     Hon Deluxe Fabric Upholstered Stacking Chairs,…  731.9400         3
2     Self-Adhesive Address Labels for Typewriters b…   14.6200         2
3         Bretford CR4500 Series Slim Rectangular Table  957.5775         5
4                       Eldon Fold 'N Roll Cart System   22.3680         2
…                                                  …       …       …
9989                           Ultra Door Pull Handle   25.2480         3
9990  Tenex B1-RE Series Chair Mats for Low Pile Car…   91.9600         2
9991                            Aastra 57i VoIP phone  258.5760         2
9992  It's Hot Message Books with Stickers, 2 3/4" x 5"   29.6000         4
9993  Acco 7-Outlet Masterpiece Power Center, Wihtou…  243.1600         2

      Discount    Profit
0         0.00   41.9136
1         0.00  219.5820
2         0.00    6.8714
3         0.45 -383.0310
4         0.20    2.5164
…          …       …
9989      0.20    4.1028
9990      0.00   15.6332
9991      0.20   19.3932
9992      0.00   13.3200
9993      0.00   72.9480
```

```
[9994 rows x 21 columns]
```

[4]:
```
data.describe()
```

[4]:
```
            Row ID   Postal Code         Sales     Quantity     Discount  \
count  9994.000000   9994.000000   9994.000000  9994.000000  9994.000000
mean   4997.500000  55190.379428    229.858001     3.789574     0.156203
std    2885.163629  32063.693350    623.245101     2.225110     0.206452
min       1.000000   1040.000000      0.444000     1.000000     0.000000
25%    2499.250000  23223.000000     17.280000     2.000000     0.000000
50%    4997.500000  56430.500000     54.490000     3.000000     0.200000
75%    7495.750000  90008.000000    209.940000     5.000000     0.200000
max    9994.000000  99301.000000  22638.480000    14.000000     0.800000

            Profit
count  9994.000000
mean     28.656896
std     234.260108
min   -6599.978000
25%       1.728750
50%       8.666500
75%      29.364000
max    8399.976000
```

[5]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Row ID         9994 non-null   int64
 1   Order ID       9994 non-null   object
 2   Order Date     9994 non-null   object
 3   Ship Date      9994 non-null   object
 4   Ship Mode      9994 non-null   object
 5   Customer ID    9994 non-null   object
 6   Customer Name  9994 non-null   object
 7   Segment        9994 non-null   object
 8   Country        9994 non-null   object
 9   City           9994 non-null   object
 10  State          9994 non-null   object
 11  Postal Code    9994 non-null   int64
 12  Region         9994 non-null   object
 13  Product ID     9994 non-null   object
 14  Category       9994 non-null   object
 15  Sub-Category   9994 non-null   object
```

```
 16  Product Name   9994 non-null   object
 17  Sales          9994 non-null   float64
 18  Quantity       9994 non-null   int64
 19  Discount       9994 non-null   float64
 20  Profit         9994 non-null   float64
dtypes: float64(3), int64(3), object(15)
memory usage: 1.6+ MB
```

1 Converting Date Columns

[6]:
```
data['Order Date']=pd.to_datetime(data['Order Date'])
data['Ship Date'] = pd.to_datetime(data['Ship Date'])
#Date Conversion: Order Date aur Ship Date columns ko datetime format me␣
 ↪convert kiya gaya hai for date-based analysis.
```

[7]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Row ID         9994 non-null   int64
 1   Order ID       9994 non-null   object
 2   Order Date     9994 non-null   datetime64[ns]
 3   Ship Date      9994 non-null   datetime64[ns]
 4   Ship Mode      9994 non-null   object
 5   Customer ID    9994 non-null   object
 6   Customer Name  9994 non-null   object
 7   Segment        9994 non-null   object
 8   Country        9994 non-null   object
 9   City           9994 non-null   object
 10  State          9994 non-null   object
 11  Postal Code    9994 non-null   int64
 12  Region         9994 non-null   object
 13  Product ID     9994 non-null   object
 14  Category       9994 non-null   object
 15  Sub-Category   9994 non-null   object
 16  Product Name   9994 non-null   object
 17  Sales          9994 non-null   float64
 18  Quantity       9994 non-null   int64
 19  Discount       9994 non-null   float64
 20  Profit         9994 non-null   float64
dtypes: datetime64[ns](2), float64(3), int64(3), object(13)
memory usage: 1.6+ MB
```

2 Adding New Date-Based Columns

[8]:
```
data['Order Month'] = data['Order Date'].dt.month
data['Order Year'] = data['Order Date'].dt.year
data['Order Day of Week'] = data['Order Date'].dt.dayofweek
#Order Month: Order date se month extract karte hain.
#Order Year: Order date se year extract hota hai.
#Order Day of Week: Week ka day (0 for Monday, 6 for Sunday) extract kiya gaya␣
↪hai
```

[9]:
```
data.head()
```

[9]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	\
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	
2	3	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	
3	4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	
4	5	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	

	Customer Name	Segment	Country	City	…	\
0	Claire Gute	Consumer	United States	Henderson	…	
1	Claire Gute	Consumer	United States	Henderson	…	
2	Darrin Van Huff	Corporate	United States	Los Angeles	…	
3	Sean O'Donnell	Consumer	United States	Fort Lauderdale	…	
4	Sean O'Donnell	Consumer	United States	Fort Lauderdale	…	

	Category	Sub-Category	\
0	Furniture	Bookcases	
1	Furniture	Chairs	
2	Office Supplies	Labels	
3	Furniture	Tables	
4	Office Supplies	Storage	

	Product Name	Sales	Quantity	\
0	Bush Somerset Collection Bookcase	261.9600	2	
1	Hon Deluxe Fabric Upholstered Stacking Chairs,…	731.9400	3	
2	Self-Adhesive Address Labels for Typewriters b…	14.6200	2	
3	Bretford CR4500 Series Slim Rectangular Table	957.5775	5	
4	Eldon Fold 'N Roll Cart System	22.3680	2	

	Discount	Profit	Order Month	Order Year	Order Day of Week
0	0.00	41.9136	11	2016	1
1	0.00	219.5820	11	2016	1
2	0.00	6.8714	6	2016	6
3	0.45	-383.0310	10	2015	6
4	0.20	2.5164	10	2015	6

```
[5 rows x 24 columns]
```

3 Monthly Sales Analysis

[10]:
```
sales_by_month = data.groupby('Order Month')['Sales'].sum().reset_index()
```

[11]:
```
sales_by_month
```

[11]:

	Order Month	Sales
0	1	94924.8356
1	2	59751.2514
2	3	205005.4888
3	4	137762.1286
4	5	155028.8117
5	6	152718.6793
6	7	147238.0970
7	8	159044.0630
8	9	307649.9457
9	10	200322.9847
10	11	352461.0710
11	12	325293.5035

[12]:
```
fig = px.line(sales_by_month,
              x='Order Month',
              y='Sales',
              title='Monthly Sales Analysis')
fig.show()
```

[13]:
```
#Data Grouping:
#data.groupby('Order Month')['Sales'].sum() se har month ki total sales nikalte␣
 ↪hain.
#.reset_index() data ko structured format me rakhta hai.
#px.line: Monthly sales trend show karne ke liye line chart banaya gaya hai.
#fig.show(): Graph display karta hai.
```

4 Sales Analysis by Category

[14]:
```
sales_by_category = data.groupby('Category')['Sales'].sum().reset_index()
```

[15]:
```
sales_by_category
```

[15]:

	Category	Sales
0	Furniture	741999.7953
1	Office Supplies	719047.0320
2	Technology	836154.0330

[17]:
```
fig = px.pie(sales_by_category,
             values='Sales',
             names='Category',
             hole=0.5,
             color_discrete_sequence=px.colors.qualitative.Pastel)

fig.update_traces(textposition='inside', textinfo='percent+label')
fig.update_layout(title_text='Sales Analysis by Category',␣
 ↪title_font=dict(size=24))

fig.show()
```

[18]:
```
#groupby('Category'): Category-wise sales nikalte hain.
#Pie Chart:
#px.pie: Sales proportions ko pie chart me show karta hai.
#hole=0.5: Donut-style chart banata hai.
#Pastel Colors: Chart me soft color palette use kiya gaya hai.
```

5 Sales Analysis by Sub-Category

[19]:
```
sales_by_subcategory = data.groupby('Sub-Category')['Sales'].sum().reset_index()
```

[21]:
```
sales_by_subcategory
```

[21]:

	Sub-Category	Sales
0	Accessories	167380.3180
1	Appliances	107532.1610
2	Art	27118.7920
3	Binders	203412.7330
4	Bookcases	114879.9963
5	Chairs	328449.1030
6	Copiers	149528.0300
7	Envelopes	16476.4020
8	Fasteners	3024.2800
9	Furnishings	91705.1640
10	Labels	12486.3120
11	Machines	189238.6310
12	Paper	78479.2060
13	Phones	330007.0540
14	Storage	223843.6080
15	Supplies	46673.5380
16	Tables	206965.5320

[20]:
```
fig = px.bar(sales_by_subcategory,
             x='Sub-Category',
             y='Sales',
```

```
            title='Sales Analysis by Sub-Category')
fig.show()
```

6 Monthly Profit Analysis

[22]:
```
profit_by_month = data.groupby('Order Month')['Profit'].sum().reset_index()
```

[23]:
```
profit_by_month
```

[23]:

	Order Month	Profit
0	1	9134.4461
1	2	10294.6107
2	3	28594.6872
3	4	11587.4363
4	5	22411.3078
5	6	21285.7954
6	7	13832.6648
7	8	21776.9384
8	9	36857.4753
9	10	31784.0413
10	11	35468.4265
11	12	43369.1919

[24]:
```
fig = px.line(profit_by_month,
              x='Order Month',
              y='Profit',
              title='Monthly Profit Analysis')
fig.show()
```

[25]:
```
profit_by_category = data.groupby('Category')['Profit'].sum().reset_index()
```

[26]:
```
profit_by_category
```

[26]:

	Category	Profit
0	Furniture	18451.2728
1	Office Supplies	122490.8008
2	Technology	145454.9481

[27]:
```
fig = px.pie(profit_by_category,
             values='Profit',
             names='Category',
             hole=0.5,
             color_discrete_sequence=px.colors.qualitative.Pastel)

fig.update_traces(textposition='inside', textinfo='percent+label')
```

```
fig.update_layout(title_text='Profit Analysis by Category',␣
 ↪title_font=dict(size=24))

fig.show()
```

7 Profit Analysis by Sub-Category

[29]:
```
profit_by_subcategory = data.groupby('Sub-Category')['Profit'].sum().
 ↪reset_index()
```

[30]:
```
profit_by_subcategory
```

[30]:

	Sub-Category	Profit
0	Accessories	41936.6357
1	Appliances	18138.0054
2	Art	6527.7870
3	Binders	30221.7633
4	Bookcases	-3472.5560
5	Chairs	26590.1663
6	Copiers	55617.8249
7	Envelopes	6964.1767
8	Fasteners	949.5182
9	Furnishings	13059.1436
10	Labels	5546.2540
11	Machines	3384.7569
12	Paper	34053.5693
13	Phones	44515.7306
14	Storage	21278.8264
15	Supplies	-1189.0995
16	Tables	-17725.4811

[31]:
```
fig = px.bar(profit_by_subcategory, x='Sub-Category',
             y='Profit',
             title='Profit Analysis by Sub-Category')
fig.show()
```

[32]:
```
data.head()
```

[32]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	\
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	
2	3	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	
3	4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	
4	5	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	

	Customer Name	Segment	Country	City	…	\

```
0      Claire Gute   Consumer  United States        Henderson  …
1      Claire Gute   Consumer  United States        Henderson  …
2  Darrin Van Huff  Corporate  United States      Los Angeles  …
3   Sean O'Donnell   Consumer  United States  Fort Lauderdale  …
4   Sean O'Donnell   Consumer  United States  Fort Lauderdale  …

          Category  Sub-Category  \
0        Furniture     Bookcases
1        Furniture        Chairs
2  Office Supplies        Labels
3        Furniture        Tables
4  Office Supplies       Storage

                                        Product Name     Sales Quantity  \
0                 Bush Somerset Collection Bookcase  261.9600        2
1  Hon Deluxe Fabric Upholstered Stacking Chairs,…  731.9400        3
2  Self-Adhesive Address Labels for Typewriters b…   14.6200        2
3      Bretford CR4500 Series Slim Rectangular Table  957.5775        5
4                    Eldon Fold 'N Roll Cart System   22.3680        2

  Discount    Profit  Order Month  Order Year  Order Day of Week
0     0.00   41.9136           11        2016                  1
1     0.00  219.5820           11        2016                  1
2     0.00    6.8714            6        2016                  6
3     0.45 -383.0310           10        2015                  6
4     0.20    2.5164           10        2015                  6

[5 rows x 24 columns]
```

8 Sales and Profit Analysis by Customer Segment

[33]:
```
sales_profit_by_segment = data.groupby('Segment').agg({'Sales': 'sum', 'Profit':
 ↪ 'sum'}).reset_index()
color_palette = colors.qualitative.Pastel

fig = go.Figure()
fig.add_trace(go.Bar(x=sales_profit_by_segment['Segment'],
                     y=sales_profit_by_segment['Sales'],
                     name='Sales',
                     marker_color=color_palette[0]))

fig.add_trace(go.Bar(x=sales_profit_by_segment['Segment'],
                     y=sales_profit_by_segment['Profit'],
                     name='Profit',
                     marker_color=color_palette[1]))
```

```
fig.update_layout(title='Sales and Profit Analysis by Customer Segment',
                  xaxis_title='Customer Segment', yaxis_title='Amount')

fig.show()
```

9 analyse sales-to-profit ratio

[34]:
```
sales_profit_by_segment = data.groupby('Segment').agg({'Sales': 'sum', 'Profit':
 ↪ 'sum'}).reset_index()
sales_profit_by_segment['Sales_to_Profit_Ratio'] =␣
 ↪sales_profit_by_segment['Sales'] / sales_profit_by_segment['Profit']
print(sales_profit_by_segment[['Segment', 'Sales_to_Profit_Ratio']])
```

```
       Segment  Sales_to_Profit_Ratio
0     Consumer               8.659471
1    Corporate               7.677245
2  Home Office               7.125416
```

[ ]:
```

```