online-sales-analysis

July 1, 2024

[1]:
```
import pandas as pd
```

[2]:
```
df = pd.read_csv('online_sales_data.csv', index_col='Transaction ID')
df
```

[2]:

Transaction ID	Date	Product Category	Product Name	Units Sold	Unit Price	Total Revenue	Region	Payment Method
10001	2024-01-01	Electronics	iPhone 14 Pro	2	999.99	1999.98	North America	Credit Card
10002	2024-01-02	Home Appliances	Dyson V11 Vacuum	1	499.99	499.99	Europe	PayPal
10003	2024-01-03	Clothing	Levi's 501 Jeans	3				
10004	2024-01-04	Books	The Da Vinci Code	4				
10005	2024-01-05	Beauty Products	Neutrogena Skincare Set	1				
…	…	…	…	…				
10236	2024-08-23	Home Appliances	Nespresso Vertuo Next Coffee and Espresso Maker	1				
10237	2024-08-24	Clothing	Nike Air Force 1 Sneakers	3				
10238	2024-08-25	Books	The Handmaid's Tale by Margaret Atwood	3				
10239	2024-08-26	Beauty Products	Sunday Riley Luna Sleeping Night Oil	1				
10240	2024-08-27	Sports	Yeti Rambler 20 oz Tumbler	2				

```
10003                69.99         209.97           Asia     Debit Card
10004                15.99          63.96  North America    Credit Card
10005                89.99          89.99         Europe         PayPal
…                    …              …               …              …
10236               159.99         159.99         Europe         PayPal
10237                90.00         270.00           Asia     Debit Card
10238                10.99          32.97  North America    Credit Card
10239                55.00          55.00         Europe         PayPal
10240                29.99          59.98           Asia    Credit Card

[240 rows x 8 columns]
```

[3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 240 entries, 10001 to 10240
Data columns (total 8 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Date              240 non-null    object
 1   Product Category  240 non-null    object
 2   Product Name      240 non-null    object
 3   Units Sold        240 non-null    int64
 4   Unit Price        240 non-null    float64
 5   Total Revenue     240 non-null    float64
 6   Region            240 non-null    object
 7   Payment Method    240 non-null    object
dtypes: float64(2), int64(1), object(5)
memory usage: 16.9+ KB
```

[4]:
```
df.shape
```

[4]:
```
(240, 8)
```

[5]:
```
df.describe()
```

[5]:
```
       Units Sold   Unit Price  Total Revenue
count  240.000000   240.000000     240.000000
mean     2.158333   236.395583     335.699375
std      1.322454   429.446695     485.804469
min      1.000000     6.500000       6.500000
25%      1.000000    29.500000      62.965000
50%      2.000000    89.990000     179.970000
75%      3.000000   249.990000     399.225000
max     10.000000  3899.990000    3899.990000
```

[6]:
```
df.dtypes
```

[6]:
```
Date                 object
Product Category     object
Product Name         object
Units Sold            int64
Unit Price          float64
Total Revenue       float64
Region               object
Payment Method       object
dtype: object
```

[7]:
```
df['Date'] = pd.to_datetime(df['Date'] )
```

1 Analyze sales trends over time to identify seasonal patterns or growth opportunities.

[8]:
```
periods =  df.groupby(df['Date'].dt.to_period('M'))
```

[9]:
```
monthly_sales = periods['Units Sold'].sum()
```

[10]:
```
revenue_by_month = periods['Total Revenue'].sum()
```

[11]:
```
monthly_sales = monthly_sales.reset_index()
revenue_by_month = revenue_by_month.reset_index()
```

[12]:
```
revenue_by_month['Date'] = revenue_by_month['Date'].dt.to_timestamp()
monthly_sales['Date'] = monthly_sales['Date'].dt.to_timestamp()
```

[13]:
```
revenue_by_month
```

[13]:

	Date	Total Revenue
0	2024-01-01	14548.32
1	2024-02-01	10803.37
2	2024-03-01	12849.24
3	2024-04-01	12451.69
4	2024-05-01	8455.49
5	2024-06-01	7384.55
6	2024-07-01	6797.08
7	2024-08-01	7278.11

[14]:
```
monthly_sales
```

[14]:

	Date	Units Sold
0	2024-01-01	68
1	2024-02-01	77
2	2024-03-01	82
3	2024-04-01	65

```
4 2024-05-01          60
5 2024-06-01          61
6 2024-07-01          53
7 2024-08-01          52
```

2 Explore the popularity of different product categories across regions.

[15]:
```
region_categ_grp = df.groupby(['Region', 'Product Category'])
```

[16]:
```
product_popularity =region_categ_grp['Units Sold'].sum()
```

[17]:
```
product_popularity = product_popularity.to_frame(name='Count')
```

[18]:
```
product_popularity.reset_index(inplace=True)
```

[19]:
```
product_popularity
```

[19]:

	Region	Product Category	Count
0	Asia	Clothing	145
1	Asia	Sports	88
2	Europe	Beauty Products	46
3	Europe	Home Appliances	59
4	North America	Books	114
5	North America	Electronics	66

[20]:
```
region_grp = df.groupby('Region')
category_popularity = region_grp['Product Category'].value_counts()
category_popularity = category_popularity.to_frame('Count')
category_popularity.reset_index(inplace=True)
category_popularity
```

[20]:

	Region	Product Category	Count
0	Asia	Clothing	40
1	Asia	Sports	40
2	Europe	Beauty Products	40
3	Europe	Home Appliances	40
4	North America	Books	40
5	North America	Electronics	40

3 Investigate the impact of payment methods on sales volume or revenue.

[21]:
```
payment_grp = df.groupby('Payment Method')
payment_grp
```

[21]:
```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x00000241AECB9D00>
```

[22]:
```
paym_revenue = payment_grp['Total Revenue'].sum()
```

[23]:
```
paym_revenue = paym_revenue.to_frame(name='Total Revenue')
```

[24]:
```
paym_revenue.reset_index(inplace=True)
```

[25]:
```
paym_revenue
```

[25]:

	Payment Method	Total Revenue
0	Credit Card	51170.86
1	Debit Card	8128.93
2	PayPal	21268.06

[26]:
```
paym_volume = payment_grp['Units Sold'].sum()
paym_volume = paym_volume.to_frame(name='Units Sold')
```

[27]:
```
paym_volume.reset_index(inplace=True)
```

[28]:
```
paym_volume
```

[28]:

	Payment Method	Units Sold
0	Credit Card	268
1	Debit Card	145
2	PayPal	105

[29]:
```
most_used_paym = df['Payment Method'].value_counts()
most_used_paym  = most_used_paym.to_frame(name='Count')
most_used_paym.reset_index(inplace=True)
most_used_paym.columns = ['Payment Method', 'Frequency']
most_used_paym
```

[29]:

	Payment Method	Frequency
0	Credit Card	120
1	PayPal	80
2	Debit Card	40

4 Identify top-selling products within each category to optimize inventory and marketing strategies.

[30]:
```
category_product_grp = df.groupby(['Product Category', 'Product Name'])
```

[31]:
```
category_product_grp_df = category_product_grp['Units Sold'].sum()
```

[32]:
```
category_product_grp_df=category_product_grp_df.to_frame(name='Sum')
```

[33]:
```
category_product_grp_df.reset_index(inplace=True)
```

[34]:
```
category_product_grp_df
```

[34]:

	Product Category	Product Name	Sum
0	Beauty Products	Anastasia Beverly Hills Brow Wiz	2
1	Beauty Products	Biore UV Aqua Rich Watery Essence Sunscreen	1
2	Beauty Products	Caudalie Vinoperfect Radiance Serum	1
3	Beauty Products	CeraVe Hydrating Facial Cleanser	2
4	Beauty Products	Chanel No. 5 Perfume	1
..	…	…	…
229	Sports	Yeti Rambler 20 oz Tumbler	2
230	Sports	Yeti Rambler Bottle	3
231	Sports	Yeti Rambler Tumbler	6
232	Sports	Yeti Roadie 24 Cooler	1
233	Sports	Yeti Tundra Haul Portable Wheeled Cooler	1

[234 rows x 3 columns]

[35]:
```
#pd.set_option('display.max_rows', 234)
```

[36]:
```
filt = category_product_grp_df['Product Category'] == 'Beauty Products'
beauty_df = category_product_grp_df.loc[filt]
beauty_df = beauty_df.sort_values(by='Sum', ascending=False, ignore_index=True)
beauty_df.head(15)
```

[36]:

	Product Category	Product Name	Sum
0	Beauty Products	Anastasia Beverly Hills Brow Wiz	2
1	Beauty Products	First Aid Beauty Ultra Repair Cream	2
2	Beauty Products	L'Occitane Shea Butter Hand Cream	2
3	Beauty Products	Glossier Boy Brow	2
4	Beauty Products	L'Oreal Revitalift Serum	2
5	Beauty Products	CeraVe Hydrating Facial Cleanser	2
6	Beauty Products	Charlotte Tilbury Magic Cream	1
7	Beauty Products	Laneige Water Sleeping Mask	1
8	Beauty Products	NARS Radiant Creamy Concealer	1
9	Beauty Products	Neutrogena Hydro Boost Water Gel	1
10	Beauty Products	Neutrogena Skincare Set	1

```
11  Beauty Products                          Olay Regenerist Face Cream    1
12  Beauty Products  Paula's Choice Skin Perfecting 2% BHA Liquid E…    1
13  Beauty Products                 Shiseido Ultimate Sun Protector    1
14  Beauty Products                          Sunday Riley Good Genes    1
```

[37]:
```
filt = category_product_grp_df['Product Category'] == 'Books'
book_df = category_product_grp_df.loc[filt]
book_df = book_df.sort_values(by='Sum', ascending=False, ignore_index=True)
book_df.head(15)
```

[37]:

	Product Category	Product Name	Sum
0	Books	The Catcher in the Rye by J.D. Salinger	7
1	Books	The Silent Patient by Alex Michaelides	5
2	Books	The Girl with the Dragon Tattoo by Stieg Larsson	5
3	Books	1984 by George Orwell	4
4	Books	To Kill a Mockingbird by Harper Lee	4
5	Books	The Hunger Games by Suzanne Collins	4
6	Books	Atomic Habits by James Clear	4
7	Books	The Girl on the Train by Paula Hawkins	4
8	Books	The Da Vinci Code	4
9	Books	Where the Crawdads Sing by Delia Owens	4
10	Books	Becoming by Michelle Obama	4
11	Books	Dune by Frank Herbert	4
12	Books	The Alchemist by Paulo Coelho	3
13	Books	The Hobbit by J.R.R. Tolkien	3
14	Books	Think and Grow Rich by Napoleon Hill	3

[38]:
```
filt = category_product_grp_df['Product Category'] == 'Clothing'
clothing_df = category_product_grp_df.loc[filt]
clothing_df = clothing_df.sort_values(by='Sum', ascending=False,␣
 ↪ignore_index=True)
clothing_df.head(15)
```

[38]:

	Product Category	Product Name	Sum
0	Clothing	Hanes ComfortSoft T-Shirt	10
1	Clothing	Nike Air Force 1	6
2	Clothing	Gap Essential Crewneck T-Shirt	6
3	Clothing	Forever 21 Graphic Tee	5
4	Clothing	Tommy Hilfiger Polo Shirt	5
5	Clothing	Under Armour HeatGear T-Shirt	5
6	Clothing	Adidas 3-Stripes Shorts	5
7	Clothing	Calvin Klein Boxer Briefs	5
8	Clothing	Uniqlo Airism Mesh Boxer Briefs	4
9	Clothing	Columbia Fleece Jacket	4
10	Clothing	Under Armour Tech 2.0 T-Shirt	4
11	Clothing	Uniqlo Airism Seamless Boxer Briefs	4
12	Clothing	Adidas Originals Trefoil Hoodie	4

```
13         Clothing  Adidas Originals Superstar Sneakers    4
14         Clothing           Puma Suede Classic Sneakers    4
```

[39]:
```
filt = category_product_grp_df['Product Category'] == 'Electronics'
electronics_df = category_product_grp_df.loc[filt]
electronics_df = electronics_df.sort_values(by='Sum', ascending=False,␣
 ↪ignore_index=True)
electronics_df.head(15)
```

[39]:

	Product Category	Product Name	Sum
0	Electronics	Amazon Echo Dot (4th Gen)	4
1	Electronics	Anker PowerCore Portable Charger	4
2	Electronics	Bose SoundLink Revolve+ Speaker	3
3	Electronics	Amazon Fire TV Stick 4K	3
4	Electronics	Sony WH-1000XM4 Headphones	3
5	Electronics	Nintendo Switch	3
6	Electronics	Apple Watch Series 8	3
7	Electronics	GoPro HERO10 Black	3
8	Electronics	Samsung Galaxy Tab S8	2
9	Electronics	Logitech MX Master 3 Mouse	2
10	Electronics	Kindle Paperwhite	2
11	Electronics	Google Nest Hub Max	2
12	Electronics	Garmin Forerunner 945	2
13	Electronics	iPhone 14 Pro	2
14	Electronics	Apple iPad Air	2

[40]:
```
filt = category_product_grp_df['Product Category'] == 'Home Appliances'
home_df = category_product_grp_df.loc[filt]
home_df = home_df.sort_values(by='Sum', ascending=False, ignore_index=True)
home_df.head(15)
```

[40]:

	Product Category	Product Name	Sum
0	Home Appliances	Keurig K-Elite Coffee Maker	3
1	Home Appliances	Instant Pot Duo	3
2	Home Appliances	Eufy RoboVac 11S	3
3	Home Appliances	Philips Airfryer XXL	2
4	Home Appliances	Instant Pot Duo Evo Plus	2
5	Home Appliances	LG OLED TV	2
6	Home Appliances	Ninja Foodi Pressure Cooker	2
7	Home Appliances	Dyson Supersonic Hair Dryer	2
8	Home Appliances	Keurig K-Mini Coffee Maker	2
9	Home Appliances	Anova Precision Cooker	2
10	Home Appliances	Philips Sonicare DiamondClean Toothbrush	2
11	Home Appliances	Cuisinart Coffee Center	2
12	Home Appliances	Crock-Pot 6-Quart Slow Cooker	2
13	Home Appliances	Roomba i7+	2
14	Home Appliances	Breville Smart Grill	2

[41]:
```
filt = category_product_grp_df['Product Category'] == 'Sports'
sports_df = category_product_grp_df.loc[filt]
sports_df = sports_df.sort_values(by='Sum', ascending=False, ignore_index=True)
sports_df.head(15)
```

[41]:

	Product Category	Product Name	Sum
0	Sports	Yeti Rambler Tumbler	6
1	Sports	Spalding NBA Street Basketball	6
2	Sports	Wilson Evolution Basketball	5
3	Sports	Titleist Pro V1 Golf Balls	5
4	Sports	Manduka PRO Yoga Mat	4
5	Sports	Hydro Flask Wide Mouth Water Bottle	4
6	Sports	Adidas FIFA World Cup Football	3
7	Sports	Hydro Flask Standard Mouth Water Bottle	3
8	Sports	Yeti Rambler Bottle	3
9	Sports	Rogue Fitness Kettlebell	3
10	Sports	Babolat Pure Drive Tennis Racket	3
11	Sports	Nike Metcon 6	3
12	Sports	Fitbit Versa 3	3
13	Sports	Fitbit Charge 5	2
14	Sports	Nike Air Zoom Pegasus 37	2

[ ]:
```
```

5 Evaluate the performance of specific products or categories in different regions to tailor marketing campaigns accordingly.

[42]:
```
region_category_grp = df.groupby(['Region', 'Product Category'])
prod_popularity = region_category_grp['Total Revenue'].sum()
prod_popularity = prod_popularity.to_frame(name='Total Revenue')
prod_popularity.reset_index(inplace=True)
```

[43]:
```
prod_popularity
```

[43]:

	Region	Product Category	Total Revenue
0	Asia	Clothing	8128.93
1	Asia	Sports	14326.52
2	Europe	Beauty Products	2621.90
3	Europe	Home Appliances	18646.16
4	North America	Books	1861.93
5	North America	Electronics	34982.41