

## FETCHING DATA

```
▷ ~
sales = pd.read_csv('retail_sales_10000.csv')
print(sales.head(10))

store = pd.read_csv('stores_50.csv')
print(store.head(10))

[20]
...
      date  transaction_id  store_id  product_category  units_sold  \
0  2023-10-30 22:00:00           1     1025        Clothing       6
1  2023-02-05 20:00:00           2     1031    Electronics       4
2  2023-08-13 14:00:00           3     1049        Clothing      13
3  2023-08-05 07:00:00           4     1041        Clothing       8
4  2023-08-27 22:00:00           5     1017   Furniture      11
5  2023-09-19 01:00:00           6     1047   Groceries      19
6  2023-01-20 10:00:00           7     1030    Electronics       5
7  2023-07-04 10:00:00           8     1039    Electronics       5
8  2023-08-21 10:00:00           9     1031    Electronics       3
9  2023-12-13 18:00:00          10     1019   Furniture       2

      unit_price  cost_per_unit
0        722.09         1307.01
1       1614.59         1264.71
2        980.56         487.95
3       1813.98         269.39
4       1171.52         587.09
5       1860.72         453.96
6       1966.09         1427.65
7       935.09          420.90
8       1152.06         296.37
9       1678.10         1352.23

      store_id  store_name      city  region  store_size_sqft  opening_date
0      1001  Store_1001  Hyderabad  South            4796  2019-10-18
...
6      1007  Store_1007  Bangalore  South            1791  2019-04-20
7      1008  Store_1008  Bangalore  West             4993  2017-08-18
8      1009  Store_1009  Bangalore  West             3264  2022-04-14
9      1010  Store_1010  Kolkata   East             1763  2019-01-25

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

## CHECKING IF ANY NULL VALUES

```
▷ ~
print(sales.isnull().sum())
print(store.isnull().sum())

[28]
...
date          0
transaction_id 0
store_id       0
product_category 0
units_sold     0
unit_price     0
cost_per_unit 0
dtype: int64
store_id       0
store_name     0
city           0
region          0
store_size_sqft 0
opening_date    0
dtype: int64
```

## CHECKING DATATYPE OF COLUMNS

```
▷ ~
print(sales.info())

[30]
...
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype  
---  -- 
0   date        10000 non-null   datetime64[ns] 
1   transaction_id 10000 non-null   int64   
2   store_id     10000 non-null   int64   
3   product_category 10000 non-null   object  
4   units_sold    10000 non-null   int64   
5   unit_price    10000 non-null   float64 
6   cost_per_unit 10000 non-null   float64 
dtypes: datetime64[ns](1), float64(2), int64(3), object(1)
memory usage: 547.0+ KB
None
```

#### TOTAL REVENUE ,TOTAL COST & TOTAL REVENUE BY REGION

```
Total_Revenue = merged['revenue'].sum()
print("Total Revenue: ",Total_Revenue)

Total_cost = merged['cost'].sum()
print("Total cost: ",Total_cost)

revenue_by_region = merged.groupby('region')['revenue'].sum()
print("Revenue by Region: ", revenue_by_region)

[33]
...
Total Revenue: 103306021.35
Total cost: 76697609.69
Revenue by Region: region
East      15838801.13
North     13252670.56
South      42803216.52
West       31411333.14
Name: revenue, dtype: float64
```

#### TOTAL PROFIT & TOTAL PROFIT BY REGION

```
print("Total_profit: ",merged['profit'].sum())

print("TOTAL PROFIT BY REGION: ", merged.groupby('region')['profit'].sum().sort_values(ascending=False))

[38]
...
Total_profit: 26608411.65999996
TOTAL PROFIT BY REGION: region
South    11222966.59
West     7817835.44
East     4067780.17
North    3499829.46
Name: profit, dtype: float64
```

#### TOP STORES

```
top_stores = merged.groupby('store_id')['revenue'].sum()

top_stores = top_stores.sort_values(ascending=False)

print("Top 5 Stores by Revenue: \n", top_stores.head(5))

[47]
...
Top 5 Stores by Revenue:
store_id
1015    2624334.21
1044    2540781.97
1014    2486276.97
1048    2402825.67
1017    2395556.52
Name: revenue, dtype: float64
```

#### CATEGORY WISE TOTAL UNIT SOLD

```
unit_by_category = merged.groupby('product_category')['units_sold'].sum()
print(unit_by_category)

[50]
...
product_category
Clothing      19729
Electronics   19975
Furniture     20921
Groceries    19635
Sports        20380
Name: units_sold, dtype: int64
```

## MONTHLY REVENUE TREND

```
merged['date'] = pd.to_datetime(merged['date'])
merged['month'] = merged['date'].dt.to_period('M')
monthly_revenue = merged.groupby('month')['revenue'].sum().reset_index()
monthly_revenue = monthly_revenue.sort_values('month')
print(monthly_revenue)
print("\nHIGHEST REVENUE MONTH: \n", monthly_revenue.max())
```

[57]

```
...     month      revenue
0  2023-01  7910832.10
1  2023-02  7196483.87
2  2023-03  7209142.11
3  2023-04  7357452.72
4  2023-05  7694151.28
5  2023-06  8210394.80
6  2023-07  7722165.27
7  2023-08  7201510.50
8  2023-09  7338434.02
9  2023-10  7633719.69
10 2023-11  7483466.96
11 2023-12  7234550.15
12 2024-01  7907171.93
13 2024-02  5206545.95
```

```
HIGHEST REVENUE MONTH:
  month      2024-02
  revenue    8210394.8
  dtype: object
```

```
monthly_trend = merged.groupby('month')[['revenue', 'profit']].sum().reset_index()
print(monthly_trend)
```

[ ]

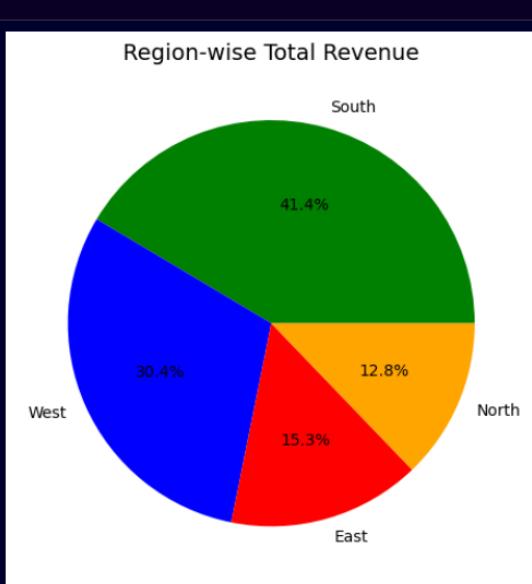
```
...     month      revenue      profit
0  2023-01  7910832.10  2270635.92
1  2023-02  7196483.87  1812660.94
2  2023-03  7209142.11  1580390.51
3  2023-04  7357452.72  1804997.23
4  2023-05  7694151.28  1920576.66
5  2023-06  8210394.80  2334334.62
6  2023-07  7722165.27  1944684.12
7  2023-08  7201510.50  1715384.52
8  2023-09  7338434.02  1965799.68
9  2023-10  7633719.69  1952837.48
10 2023-11  7483466.96  2003211.03
11 2023-12  7234550.15  1901174.13
12 2024-01  7907171.93  1849178.03
13 2024-02  5206545.95  1552546.79
```

```

region_revenue = merged.groupby('region')['revenue'].sum().reset_index()
region_revenue = region_revenue.sort_values('revenue', ascending=False)

plt.figure(figsize=(8,5))
plt.pie(region_revenue['revenue'], labels=region_revenue['region'], autopct='%.1f%%', colors = ['green','blue','red','orange'])
plt.title("Region-wise Total Revenue", fontsize=14)
plt.tight_layout()
plt.show()

```



```

monthly_product = merged.groupby(['month', 'product_category'])['revenue'].sum().reset_index()
top_products = monthly_product.loc[monthly_product.groupby('month')['revenue'].idxmax()]
plt.figure(figsize=(10,6))
plt.plot(top_products['month'].astype(str), top_products['revenue'], marker='o', color='green')
for i, row in top_products.iterrows():
    plt.text(row['month'].strftime('XY-%m'), row['revenue'], row['product_category'],
             ha='center', va='bottom', fontsize=9, rotation=45)
plt.title("Top-Selling Product Each Month", fontsize=16)
plt.xlabel("Month")
plt.ylabel("Revenue of Top Product")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

```

