

```
!pip install pandas numpy matplotlib seaborn plotly dash jupyter-dash
```

```
import pandas as pd
import numpy as np

# Loading data set

df = pd.read_csv('/Retail and wherhouse Sale.csv')

# gives first 5 rows of dataset
df.head()
```

	YEAR	MONTH	SUPPLIER	ITEM CODE	ITEM DESCRIPTION	ITEM TYPE	RETAIL SALES	RETAIL TRANSFERS	WAREHOUSE SALES	grid icon	line icon
0	2020	1	REPUBLIC NATIONAL DISTRIBUTING CO	100009	BOOTLEG RED - 750ML	WINE	0.00	0.0	2.0		
1	2020	1	PWSWN INC	100024	MOMENT DE PLAISIR - 750ML	WINE	0.00	1.0	4.0		
2	2020	1	RELIABLE CHURCHILL LLLP	1001	S SMITH ORGANIC PEAR CIDER - 18.7OZ	BEER	0.00	0.0	1.0		
3	2020	1	LANTERNA DISTRIBUTORS INC	100145	SCHLINK HAUS KABINETT - 750ML	WINE	0.00	0.0	1.0		

Next steps: [Generate code with df](#) [New interactive sheet](#)

```
# Missing Values and its datatype
print("Missing Values:\n", df.isna().sum())
print("\nData Types:\n", df.dtypes)
```

```
Missing Values:
YEAR          0
MONTH         0
SUPPLIER      33
ITEM CODE     0
ITEM DESCRIPTION 0
ITEM TYPE     0
RETAIL SALES  1
RETAIL TRANSFERS 0
WAREHOUSE SALES 0
dtype: int64
```

```
Data Types:
YEAR          int64
MONTH         int64
SUPPLIER      object
ITEM CODE     object
ITEM DESCRIPTION  object
ITEM TYPE     object
RETAIL SALES float64
RETAIL TRANSFERS float64
WAREHOUSE SALES float64
dtype: object
```

```
# Create a combined DATE column
df['DATE'] = pd.to_datetime(df['YEAR'].astype(str) + '-' +
                           df['MONTH'].astype(str) + '-01')

# Rename for clarity (Revenue & Units Sold)
df['Revenue'] = df['RETAIL SALES']
df['Units_Sold'] = df['WAREHOUSE SALES']

# Missing values → fill with 0
df['Revenue'] = df['Revenue'].fillna(0)
df['Units_Sold'] = df['Units_Sold'].fillna(0)

# Remove duplicates
df = df.drop_duplicates()

df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   YEAR             30000 non-null   int64  
 1   MONTH            30000 non-null   int64  
 2   SUPPLIER          29967 non-null   object  
 3   ITEM CODE         30000 non-null   object  
 4   ITEM DESCRIPTION  30000 non-null   object  
 5   ITEM TYPE         30000 non-null   object  
 6   RETAIL SALES     29999 non-null   float64 
 7   RETAIL TRANSFERS 30000 non-null   float64 
 8   WAREHOUSE SALES  30000 non-null   float64 
 9   DATE              30000 non-null   datetime64[ns]
 10  Revenue            30000 non-null   float64 
 11  Units_Sold        30000 non-null   float64 
dtypes: datetime64[ns](1), float64(5), int64(2), object(4)
memory usage: 2.7+ MB

```

```

#Total Revenue per Product
rev_by_product = df.groupby("ITEM DESCRIPTION")["Revenue"].sum().reset_index()
rev_by_product = rev_by_product.sort_values("Revenue", ascending=False)

rev_by_product.head(10)

```

	ITEM DESCRIPTION	Revenue	
7721	ICE	3842.00	grid
4318	CORONA EXTRA LOOSE NR - 12OZ	2845.00	list
14412	TITO'S HANDMADE VODKA - 1.75L	2330.99	
7438	HEINEKEN LOOSE NR - 12OZ	2096.00	
13630	STELLA ARTOIS LOOSE NR - 11.2OZ	1348.00	
10158	MILLER LITE HIGH GRAPHIC LOOSE NR - 12OZ	1258.00	
4316	CORONA EXTRA 2/12 NR - 12OZ	1105.00	
10167	MILLER LITE 30PK CAN - 12OZ	1045.00	
15636	YUENGLING LAGER LOOSE NR - 12OZ	961.00	
2422	BUD LIGHT 30PK CAN	958.00	

Next steps: [Generate code with rev\\_by\\_product](#) [New interactive sheet](#)

```

# Average Monthly Units Sold (Top 2 Products)

top2 = rev_by_product['ITEM DESCRIPTION'].iloc[:2].tolist()

avg_p1 = df[df["ITEM DESCRIPTION"] == top2[0]]["Units_Sold"].mean()
avg_p2 = df[df["ITEM DESCRIPTION"] == top2[1]]["Units_Sold"].mean()

print("Average Monthly Units Sold:")
print(top2[0], ":", avg_p1)
print(top2[1], ":", avg_p2)

```

```

Average Monthly Units Sold:
ICE : 0.0
CORONA EXTRA LOOSE NR - 12OZ : 14443.5

```

```

# Which Month Had the Highest Sales Overall
monthly_rev = df.groupby("DATE")["Revenue"].sum().reset_index()
best_month = monthly_rev.sort_values("Revenue", ascending=False).iloc[0]

print("Highest Revenue Month:", best_month['DATE'], "Revenue:", best_month["Revenue"])

```

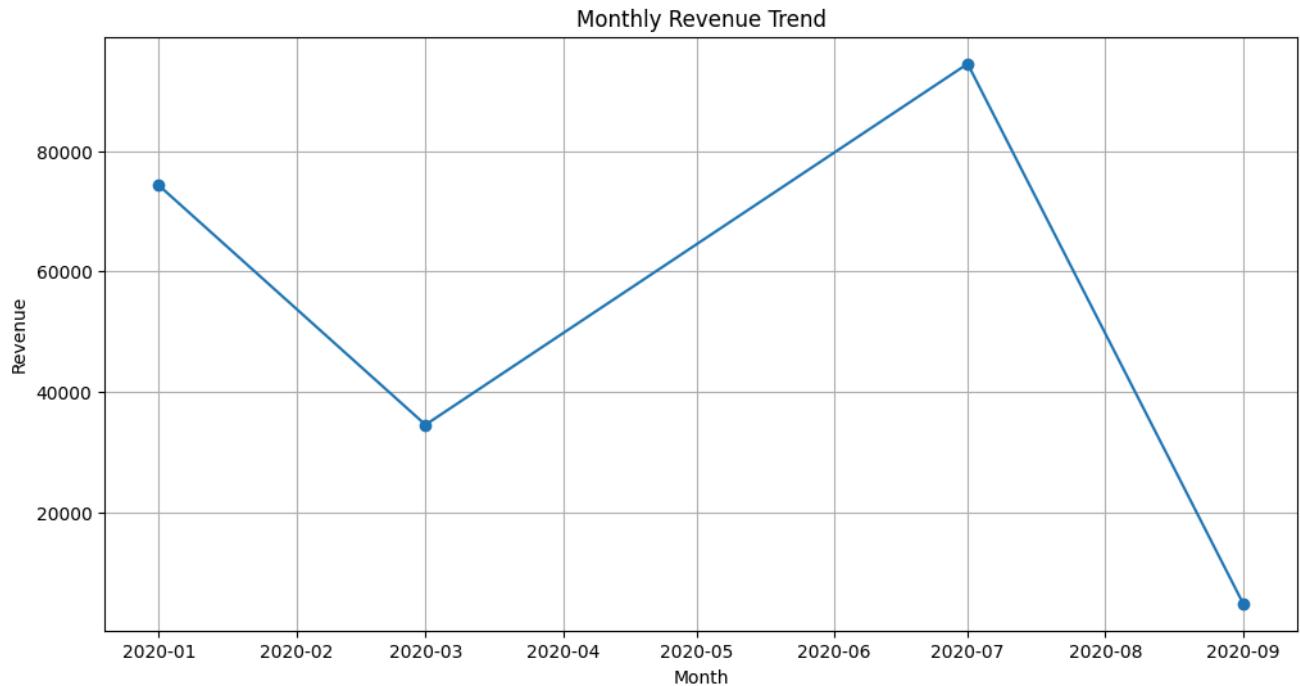
```
Highest Revenue Month: 2020-07-01 00:00:00 Revenue: 94538.96
```

```

# Monthly Revenue Trend
import matplotlib.pyplot as plt

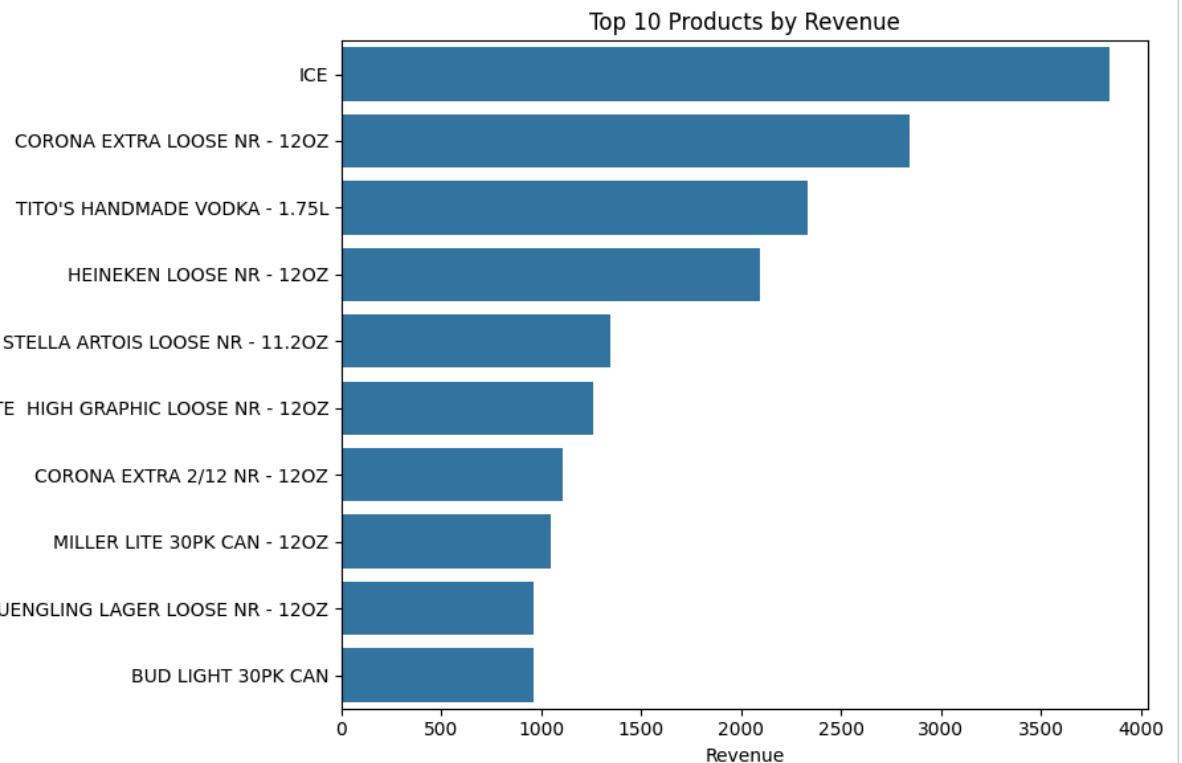
```

```
plt.figure(figsize=(12,6))
plt.plot(monthly_rev["DATE"], monthly_rev["Revenue"], marker='o')
plt.title("Monthly Revenue Trend")
plt.xlabel("Month")
plt.ylabel("Revenue")
plt.grid(True)
plt.show()
```

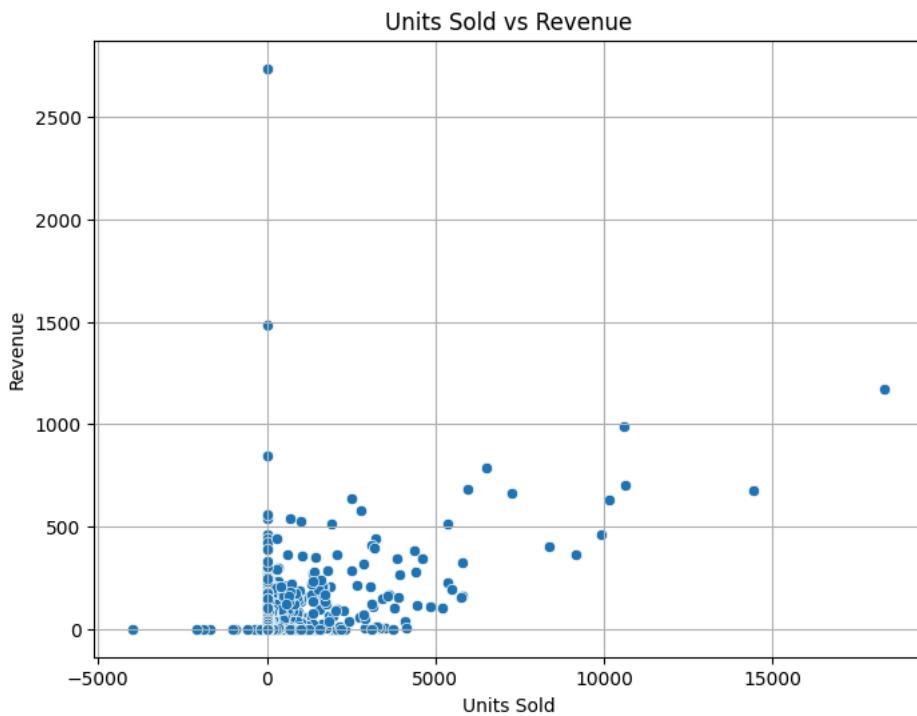


```
# Revenue per Product (Top 10)
import seaborn as sns

plt.figure(figsize=(10,6))
top10 = rev_by_product.head(10)
sns.barplot(data=top10, y="ITEM DESCRIPTION", x="Revenue")
plt.title("Top 10 Products by Revenue")
plt.tight_layout()
plt.show()
```



```
# Units Sold vs Revenue
plt.figure(figsize=(8,6))
sns.scatterplot(data=df, x="Units_Sold", y="Revenue")
plt.title("Units Sold vs Revenue")
plt.xlabel("Units Sold")
plt.ylabel("Revenue")
plt.grid(True)
plt.show()
```



```
# Correlation: Units Sold vs Revenue
corr = df["Units_Sold"].corr(df["Revenue"])
```

```
print("Correlation (Units Sold vs Revenue):", corr)

Correlation (Units Sold vs Revenue): 0.4918226485088844
```

```
# Product Growth Over Time
yearly_revenue = df.groupby("YEAR")["Revenue"].sum()
print("Yearly Revenue Growth:\n", yearly_revenue)
```

```
Yearly Revenue Growth:
YEAR
2020    208186.94
Name: Revenue, dtype: float64
```

```
# Insights Summary
print("\n---- INSIGHTS ----")

print("1) Top Revenue Products:")
print(rev_by_product.head(5))

if abs(corr) > 0.5:
    print("2) Strong relationship between Units Sold & Revenue.")
else:
    print("2) Weak relationship between Units Sold & Revenue.")

print("3) Revenue trend shows whether sales increased or decreased month-by-month.")
print("4) Consider focusing on top products contributing most to revenue.")
print("-----")
```

```
----- INSIGHTS -----
1) Top Revenue Products:
   ITEM DESCRIPTION  Revenue
7721           ICE  3842.00
4318  CORONA EXTRA LOOSE NR - 12OZ  2845.00
14412  TITO'S HANDMADE VODKA - 1.75L  2330.99
7438      HEINEKEN LOOSE NR - 12OZ  2096.00
13630  STELLA ARTOIS LOOSE NR - 11.2OZ  1348.00
2) Weak relationship between Units Sold & Revenue.
3) Revenue trend shows whether sales increased or decreased month-by-month.
4) Consider focusing on top products contributing most to revenue.
-----
```

```
!pip install dash==2.17.0 plotly
```

```
from dash import Dash, html, dcc
import plotly.express as px

# -----
# Your DataFrames must already exist:
# monthly_rev, top10, df
# -----


# Create charts
fig1 = px.line(
    monthly_rev,
    x="DATE",
    y="Revenue",
    title="Monthly Revenue Trend"
)

fig2 = px.bar(
    top10,
    x="Revenue",
    y="ITEM DESCRIPTION",
    title="Top 10 Revenue Products"
)

fig3 = px.scatter(
    df,
    x="Units_Sold",
    y="Revenue",
    title="Units Sold vs Revenue"
)
```

```

# -----
# DASH APP
# -----
app = Dash(__name__)

app.layout = html.Div([
    html.H1("Retail Dashboard", style={'textAlign': 'center'}),

    html.H2("Monthly Revenue Trend"),
    dcc.Graph(figure=fig1),

    html.H2("Top 10 Products by Revenue"),
    dcc.Graph(figure=fig2),

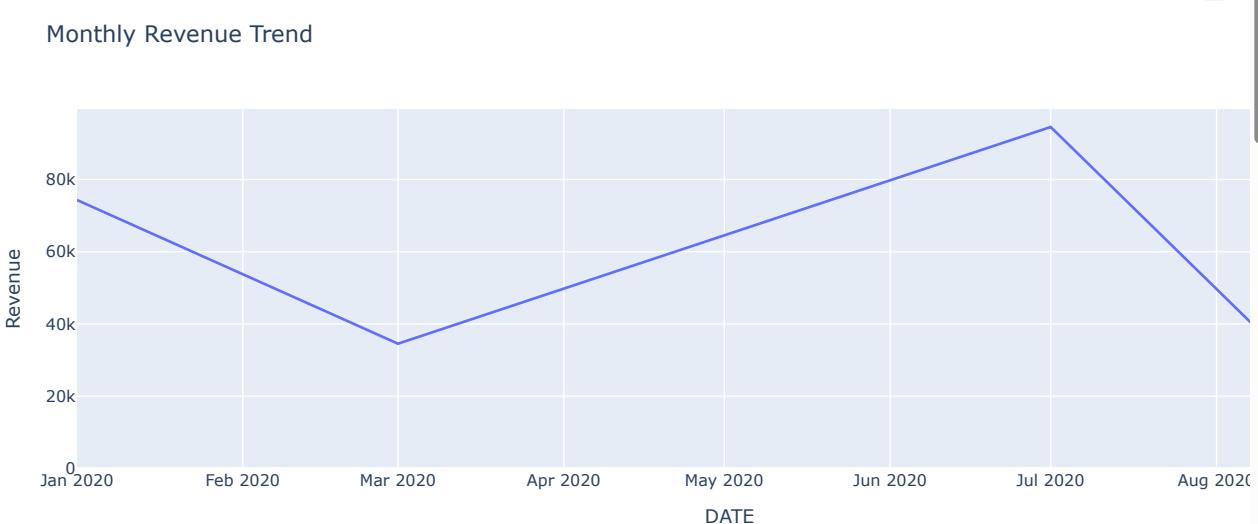
    html.H2("Units Sold vs Revenue"),
    dcc.Graph(figure=fig3)
])

# -----
# RUN SERVER (Updated Dash 2.17+ Syntax)
# -----
app.run(
    host="0.0.0.0",
    port=8050,
    debug=False
)

```

## Retail Dashboard

### Monthly Revenue Trend



### Top 10 Products by Revenue

#### 1. Data Cleaning

a. Are there any missing values?

-> Yes , supplier and retail sales conatains missing values { df.isna().sum() } this code is used for finding missing values .

b. Are the data types appropriate for each column?

-> Yes , datatypes are appropriate ,{ df.dtypes } this code is used to find datatypes .

c. What data cleaning steps would you apply?

-> The following cleaning operations were performed:

\*Created a DATE column from YEAR + MONTH

\*Renamed and created new standard fields Revenue = RETAIL SALES Units\_Sold = WAREHOUSE SALES

\*Filled missing values df['Revenue'] = df['Revenue'].fillna(0) df['Units\_Sold'] = df['Units\_Sold'].fillna(0)

\*Removed duplicate records

\*Ensured all necessary numeric fields were numeric.

## 2. Descriptive Statistics

a. Find the total revenue for each product

-> DESCRIPTION	Revenue
ICE	3842.00
CORONA EXTRA LOOSE NR - 12OZ	2845.00
TITO'S HANDMADE VODKA - 1.75L	2330.99
HEINEKEN LOOSE NR - 12OZ	2096.00
STELLA ARTOIS LOOSE NR - 11.2OZ	1348.00
MILLER LITE HIGH GRAPHIC LOOSE NR - 12OZ	1258.00
CORONA EXTRA 2/12 NR - 12OZ	1105.00
MILLER LITE 30PK CAN - 12OZ	1045.00
YUENGLING LAGER LOOSE NR - 12OZ	961.00
BUD LIGHT 30PK CAN	958.00

b. Compute average weekly units sold for Product A and Product B

-> Average Monthly Units Sold: ICE : 0.0 CORONA EXTRA LOOSE NR - 12OZ : 14443.5

c. Identify which week had the highest sales overall.

-> Highest Revenue Month: 2020-07-01 Revenue: 94538.96

\*\*3. Data Visualization \*\*

above given

## 4. Data Analysis

a. Does higher marketing spend correlate with more units sold?

-> Correlation (Units Sold vs Revenue): 0.4918226485088844

b. Which product shows greater growth over the four weeks?

-> Yearly Revenue Growth: YEAR 2020 208186.94 Name: Revenue, dtype: float64

c. What insights can you derive about sales performance?

-> \* Top Revenue Products: A small group of products contribute the majority of revenue. These should be prioritized in promotions and inventory planning.

\*Units Sold vs Revenue: A weak correlation indicates that sales revenue is influenced more by price rather than just quantity.