



SQL + Python Data Analytics Project



Title : ZARA Sales Analysis Using SQL and Python

Overview :

The Product Sales Dataset is a comprehensive collection of data designed to provide insights into the performance of various products across different categories. This dataset includes essential attributes such as product categories, names, prices, and sales volumes, making it a valuable resource for businesses and analysts alike.

The ZARA logo is displayed in a large, bold, black serif font.

Dataset :

<https://www.kaggle.com/datasets/xontoloyo/data-penjualan-zara?select=zara.csv>



File Edit View Run Kernel Settings Help

Trusted

Code ▾

JupyterLab ⌂ Python 3 (ipykernel) ○ ⏺ ⏹

```
[1]: import pandas as pd  
import matplotlib as plt  
import seaborn as sn
```

⟳ ⏺ ⏹ ⏵ ⏴ ⏷ ⏸ ⏹

```
[2]: pip install pymysql
```

```
Requirement already satisfied: pymysql in c:\users\amar\anaconda3\lib\site-packages (1.1.1)  
Note: you may need to restart the kernel to use updated packages.
```

```
[3]: import pymysql
```

```
[4]: pip install mysql-connector-python
```

```
Requirement already satisfied: mysql-connector-python in c:\users\amar\anaconda3\lib\site-packages (9.1.0)  
Note: you may need to restart the kernel to use updated packages.
```

```
[5]: import mysql.connector
```

```
[7]: conn = pymysql.connect(  
    host = "localhost",  
    user = "root",  
    password = "amar",  
    database = "zara"  
)
```

```
[74]: query = 'select * from clothes'
```

```
[86]: data = pd.read_sql(query,conn)  
data
```

```
[7]: conn = pymysql.connect(  
    host = "localhost",  
    user = "root",  
    password = "amar",  
    database = "zara"  
)
```

```
[74]: query = 'select * from clothes'
```

```
[86]: data = pd.read_sql(query,conn)  
data
```



C:\Users\AMAR\AppData\Local\Temp\ipykernel_22884\4223859399.py:1: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider using SQLAlchemy.
data = pd.read_sql(query,conn)

	Product ID	Product Position	Promotion	Product Category	Seasonal	Sales Volume	brand	url	sku	name	description	price	currency
0	188771	Aisle	No	Clothing	No	654	Zara	https://www.zara.com/us/en/tuxedo-jacket-p0889...	324052738-800-46	TUXEDO JACKET	Straight fit blazer. Pointed lapel collar and ...	169.0	USD
1	180176	End-cap	Yes	Clothing	Yes	2220	Zara	https://www.zara.com/us/en/slim-fit-suit-jacket-p0889...	335342680-800-44	SLIM FIT SUIT JACKET	Slim fit jacket. Notched lapel collar.	129.0	USD

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

	Product ID	Sales Volume	price
count	252.000000	252.000000	252.000000
mean	153370.503968	1823.702381	86.252540
std	26160.444549	697.703748	52.083205
min	110075.000000	529.000000	7.990000
25%	131053.750000	1243.000000	49.900000
50%	151681.500000	1839.500000	79.900000
75%	175669.750000	2398.750000	109.000000
max	199631.000000	2989.000000	439.000000

```
[25]: data.columns
```

```
[25]: Index(['Product ID', 'Product Position', 'Promotion', 'Product Category',
       'Seasonal', 'Sales Volume', 'brand', 'url', 'sku', 'name',
       'description', 'price', 'currency', 'scraped_at', 'terms', 'section'],
       dtype='object')
```

```
+ X □ ▶ ■ C ▶ Code [17]: JupyterLab ▾ Python 3 (ipykernel) □ ▾
```

[17]: `data.isnull().sum()`

[17]:

```
Product ID      0
Product Position 0
Promotion        0
Product Category 0
Seasonal          0
Sales Volume     0
brand            0
url              0
sku              0
name             0
description      0
price            0
currency          0
scraped_at       0
terms            0
section          0
dtype: int64
```

[21]: `data.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 252 entries, 0 to 251
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype  
 ---  -- 
 0   Product ID      252 non-null    int64  
 1   Product Position 252 non-null    object  
 2   Promotion        252 non-null    object  
 3   Product Category 252 non-null    object
```

MYSQL Query

```
5 -- 1 What is the average sales volume for products with promotions compared to those without?  
6  
7 • select promotion, avg(`Sales Volume`) as avg_sales_volume from clothes group by promotion;  
8
```



The screenshot shows the MySQL Workbench interface with the 'Result Grid' tab selected. The query results are displayed in a table with two columns: 'promotion' and 'avg_sales_volume'. There are two rows: one for 'No' with a value of 1820.5455, and one for 'Yes' with a value of 1827.1750.

promotion	avg_sales_volume
No	1820.5455
Yes	1827.1750

Python

What is the average sales volume for products with promotions compared to those without?

```
: query = """ SELECT Promotion, AVG(`Sales Volume`) AS avg_sales_volume FROM clothes GROUP BY Promotion;"""  
cur.execute(query)  
data = cur.fetchall()  
  
data  
]

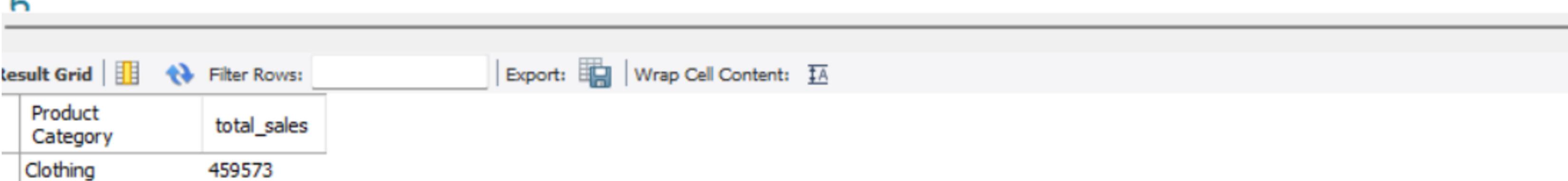
```
: (('No', Decimal('1820.5455')), ('Yes', Decimal('1827.1750')))
```


```

MYSQL Query

```
9 -- 2.Which product categories have the highest and lowest sales volumes?  
.0  
.1 • select `Product Category`, sum(`Sales Volume`) as total_sales  
.2 from clothes  
.3 group by `Product Category`  
.4 order by total_sales desc;  
.5  
.6
```

Python



Product Category	total_sales
Clothing	459573

▼ Which product categories have the highest and lowest sales volumes?

```
[99]: query = """ select `Product Category`, sum(`Sales Volume`) as total_sales from clothes group by `Product Category` order by total_sales desc;"""  
cur.execute(query)  
data = cur.fetchall()  
data  
  
[99]: (('Clothing', Decimal('459573'))),
```

MYSQL Query

```
17 -- 3.Identify the top 5 most expensive products for each category.
```

```
18 WITH RankedProducts AS (SELECT `product category`,name,price,ROW_NUMBER() OVER  
19 (PARTITION BY `product category` ORDER BY price DESC) AS ran  
20 FROM clothes  
21 )  
22 SELECT `product category`,name,price  
23 FROM RankedProducts  
24 WHERE ran <= 5;
```

```
25
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

product category	name	price
Clothing	CROPPED LEATHER JACKET	439
Clothing	SUEDE JACKET	349
Clothing	LEATHER JACKET	299
Clothing	VINTAGE EFFECT LEATHER BOMBER JACKET	299
Clothing	WOOL BLEND TEXTURED JACKET	189

Python

Identify the top 5 most expensive products for each category.

```
[41]: query = """ WITH RankedProducts AS (SELECT `Product Category` AS category,`name`, `price` ,ROW_NUMBER()  
OVER (PARTITION BY `Product Category` ORDER BY `price` DESC) AS gu  
FROM `clothes`  
)  
SELECT category,`name`, `price` FROM RankedProducts WHERE gu <= 5;"""  
cur.execute(query)  
data = cur.fetchall()  
data
```

```
[41]: (('Clothing', 'CROPPED LEATHER JACKET', 439.0),  
('Clothing', 'SUEDE JACKET', 349.0),  
('Clothing', 'LEATHER JACKET', 299.0),  
('Clothing', 'VINTAGE EFFECT LEATHER BOMBER JACKET', 299.0),  
('Clothing', 'WOOL BLEND TEXTURED JACKET', 189.0))
```

MYSQL Query

```
26 -- 4.How many products are seasonal vs. non-seasonal, and what is their average price?  
27  
28 • select seasonal, COUNT(*) AS Product_Count, avg(price) from clothes group by seasonal;  
^~
```

Result Grid			
	seasonal	Product_Count	avg(price)
▶	No	124	86.42290322580627
	Yes	128	86.0874999999998

Python

How many products are seasonal vs. non-seasonal, and what is their average price?

```
53]: query = """select seasonal, COUNT(*) AS Product_Count, avg(price) from clothes group by seasonal;"""  
cur.execute(query)  
data = cur.fetchall()  
data
```



```
53]: (('No', 124, 86.42290322580627), ('Yes', 128, 86.0874999999998))
```

MYSQL Query

```
-- 5 Retrieve all products in a specific section (e.g., "MAN") with a price above a certain threshold.  
• select * from clothes where section = "man" and price >100;  
•  
•
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

Product ID	Product Position	Promotion	Product Category	Seasonal	Sales Volume	brand	url	sku	name	description
188771	Aisle	No	Clothing	No	654	Zara	https://www.zara.com/us/en/tuxedo-jacket-p08896675.html	324052738-800-46	TUXEDO JACKET	Straight fit blazer. Pointed lapel collar and double-breasted closure.
180176	End-cap	Yes	Clothing	Yes	2220	Zara	https://www.zara.com/us/en/slim-fit-suit-jacket-p08896675.html	335342680-800-44	SLIM FIT SUIT JACKET	Slim fit jacket. Notched lapel collar. Long sleeves.
112917	Aisle	Yes	Clothing	Yes	1568	Zara	https://www.zara.com/us/en/stretch-suit-jacket-p08896675.html	328303236-420-44	STRETCH SUIT JACKET	Slim fit jacket made of viscose blend fabric.
192936	End-cap	No	Clothing	Yes	2942	Zara	https://www.zara.com/us/en/double-faced-jacket-p08896675.html	312368260-800-2	DOUBLE FACED JACKET	Jacket made of faux leather faux shearling.
182157	Aisle	No	Clothing	No	2421	Zara	https://www.zara.com/us/en/suit-jacket-in-100-linen-p08896675.html	322972485-431-46	SUIT JACKET IN 100% LINEN	Straight fit blazer made of linen. Notch lapel.

Result Grid | Form Editor

Python

Retrieve all products in a specific section (e.g., "MAN") with a price above a certain threshold.

```
[83]: query = """select * from clothes where section = "man" and price >100 limit 10;"""
cur.execute(query)
data = cur.fetchall()
data
```

```
[83]: ((188771,
      'Aisle',
      'No',
      'Clothing',
      'No',
      654,
      'Zara',
      'https://www.zara.com/us/en/tuxedo-jacket-p08896675.html',
```

MYSQL Query

```
35 -- 6 Find products where the description contains specific keywords like "jacket."  
36  
37 • select * from clothes where description like "%jacket%";  
38
```

Result Grid											
	Product ID	Product Position	Promotion	Product Category	Seasonal	Sales Volume	brand	url	sku	name	description
▶	185102	Aisle	No	Clothing	No	2823	Zara	https://www.zara.com/us/en/basic-puffer-jack...	272145190-250-2	BASIC PUFFER JACKET	Puffer jacket made of tear-resistant rip
	180176	End-cap	Yes	Clothing	Yes	2220	Zara	https://www.zara.com/us/en/slim-fit-suit-jacket...	335342680-800-44	SLIM FIT SUIT JACKET	Slim fit jacket. Notched lapel collar. Lor
	112917	Aisle	Yes	Clothing	Yes	1568	Zara	https://www.zara.com/us/en/stretch-suit-jacke...	328303236-420-44	STRETCH SUIT JACKET	Slim fit jacket made of viscose blend fa
	192936	End-cap	No	Clothing	Yes	2942	Zara	https://www.zara.com/us/en/double-faced-jack...	312368260-800-2	DOUBLE FACED JACKET	Jacket made of faux leather faux shea
	117590	End-cap	No	Clothing	No	2968	Zara	https://www.zara.com/us/en/contrasting-collar...	320298385-807-2	CONTRASTING COLLAR JACKET	Relaxed fit jacket. Contrasting lapel co

Python

Find products where the description contains specific keywords like "jacket."

```
: query = """select * from clothes where description like "%jacket%" limit 10;"""
cur.execute(query)
data = cur.fetchall()
data

: ((185102,
  'Aisle',
  'No',
  'Clothing',
  'No',
  2823,
  'Zara',
  'https://www.zara.com/us/en/basic-puffer-jacket-p06985450.html',
  272145190-250-2,
  'BASIC PUFFER JACKET',
  'Puffer jacket made of tear-resistant rip')
```

MYSQL Query

```
-- 7 Calculate total sales volume by product category and position (e.g., Aisle vs. End-cap).  
1 • select `Product Category` AS category,  
2   `Product Position` AS position,  
3   sum(`Sales Volume`) AS total_sales_volume  
4   from clothes group by `Product Category`, `Product Position`;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

category	position	total_sales_volume
Clothing	Aisle	177396
Clothing	End-cap	152930
Clothing	Front of Store	129247

Python

- ▼ Calculate total sales volume by product category and position (e.g., Aisle vs. End-cap). ↴

```
[97]: query = """select `Product Category` AS category, `Product Position` AS position, sum(`Sales Volume`) AS total_sales_volume from clothes group by `Product Category`, `Product Position`"""  
[97]: cur.execute(query)  
[97]: data = cur.fetchall()  
[97]: data  
[97]: [ ('Clothing', 'Aisle', Decimal('177396')),  
[97]:      ('Clothing', 'End-cap', Decimal('152930')),  
[97]:      ('Clothing', 'Front of Store', Decimal('129247'))]
```

MYSQL Query

```
46 -- 8 Summarize the total revenue generated per category.  
47  
48 • select `product category`,sum(price * `sales volume`) as total_revenue  
49 from clothes group by `product category`;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	product category	total_revenue		
▶	Clothing	38988476.48		

Python

Summarize the total revenue generated per category.

```
[02]: query = """select `product category`, sum(price * `sales volume`) from clothes group by `product category`;"""\n        cur.execute(query)\n        data = cur.fetchall()\n        data\n\n[02]: (('Clothing', 38988476.48),)
```

Visualization in python

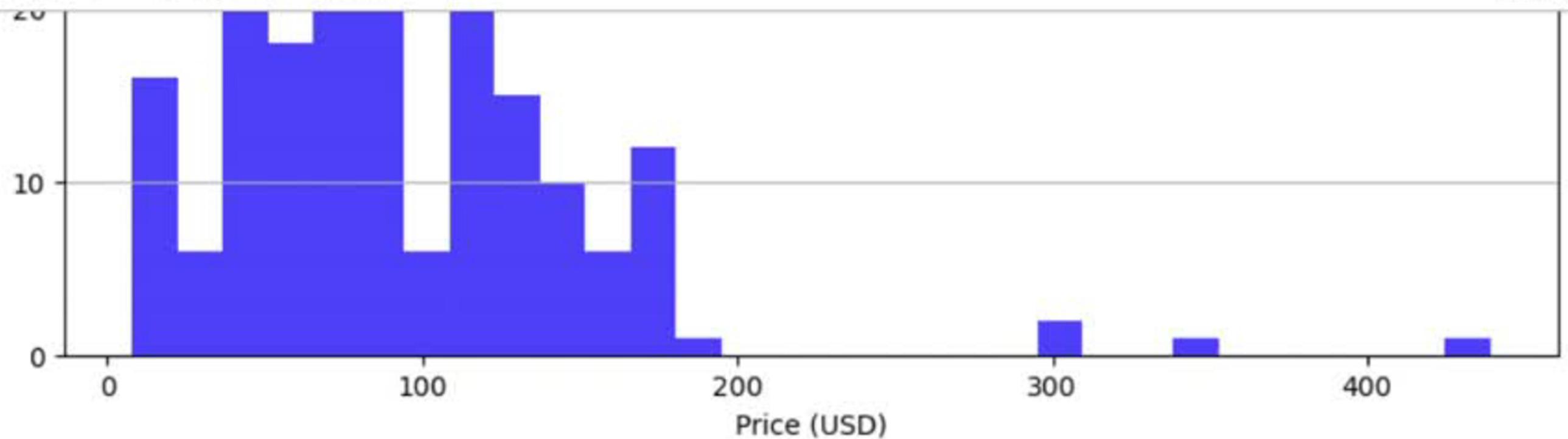
jupyter sql +python project Last Checkpoint: yesterday



Trusted

File Edit View Run Kernel Settings Help

JupyterLab Python 3 (ipykernel)



```
[244]: # Count products by category
category_counts = data['Product Category'].value_counts()

# Plot bar chart
plt.figure(figsize=(10, 6))
category_counts.plot(kind='bar', color='orange')
plt.title('Count of Products by Category')
plt.xlabel('Product Category')
plt.ylabel('Count')
plt.xticks(rotation=10)
plt.show()
```

Count of Products by Category

