Pharmaceutical Sales Data Analysis Project Report

1. Project Overview

This project focuses on analyzing pharmaceutical sales data to identify **sales trends, product performance, and time-based patterns**.
The objective was to transform raw sales data into **meaningful insights** that can support business and operational decision-making.

The project follows an end-to-end data analytics workflow using **Python, SQL, and Power BI**, followed by structured reporting.

2. Dataset Description

- **Number of Records:** ~2100 daily observations
- **Data Granularity:** Daily sales
- **Key Columns:**
    - Date
    - Product sales (ATC drug codes)
    - Year
    - Month
    - Weekday

Product Codes and Drug Categories (ATC Classification)

- **M01AB** – Anti-inflammatory & antirheumatic drugs (Acetic acid derivatives)
- **M01AE** – Anti-inflammatory & antirheumatic drugs (Propionic acid derivatives)
- **N02BA** – Analgesics (Salicylic acid derivatives)
- **N02BE** – Analgesics (Pyrazolones)

- **N05B** – Psycholeptics (Anxiolytics)
- **N05C** – Psycholeptics (Hypnotics & sedatives)
- **R03** – Drugs for obstructive airway diseases
- **R06** – Antihistamines for systemic use

These product codes follow the **WHO ATC drug classification system**.

3. Tools and Technologies Used

Python

- Loaded and explored the raw dataset using Pandas
- Checking overview of data set using df.head(10)

	datum	M01AB	M01AE	N02BA	N02BE	N05B	N05C	R03	R06	Year	Month	Hour	Weekday Name
0	1/2/2014	0.00	3.67	3.4	32.40	7.0	0.0	0.0	2.0	2014	1	248	Thursday
1	1/3/2014	8.00	4.00	4.4	50.60	16.0	0.0	20.0	4.0	2014	1	276	Friday
2	1/4/2014	2.00	1.00	6.5	61.85	10.0	0.0	9.0	1.0	2014	1	276	Saturday
3	1/5/2014	4.00	3.00	7.0	41.10	8.0	0.0	3.0	0.0	2014	1	276	Sunday
4	1/6/2014	5.00	1.00	4.5	21.70	16.0	2.0	6.0	2.0	2014	1	276	Monday
5	1/7/2014	0.00	0.00	0.0	0.00	0.0	0.0	0.0	0.0	2014	1	276	Tuesday
6	1/8/2014	5.33	3.00	10.5	26.40	19.0	1.0	10.0	0.0	2014	1	276	Wednesday
7	1/9/2014	7.00	1.68	8.0	25.00	16.0	0.0	3.0	2.0	2014	1	276	Thursday
8	1/10/2014	5.00	2.00	2.0	53.30	15.0	2.0	0.0	2.0	2014	1	276	Friday
9	1/11/2014	5.00	4.34	10.4	52.30	14.0	0.0	1.0	0.2	2014	1	276	Saturday

- Data exploration: Used df.info() to check structure

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2106 entries, 0 to 2105
Data columns (total 13 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   datum         2106 non-null   object
 1   M01AB         2106 non-null   float64
 2   M01AE         2106 non-null   float64
 3   N02BA         2106 non-null   float64
 4   N02BE         2106 non-null   float64
 5   N05B          2106 non-null   float64
 6   N05C          2106 non-null   float64
 7   R03           2106 non-null   float64
 8   R06           2106 non-null   float64
 9   Year          2106 non-null   int64
 10  Month         2106 non-null   int64
 11  Hour          2106 non-null   int64
 12  Weekday Name  2106 non-null   object
dtypes: float64(8), int64(3), object(2)
memory usage: 214.0+ KB
```

- Used df.describe() to check summary

	M01AB	M01AE	N02BA	N02BE	N05B	N05C	R03	R06	Year	Month	Hour
count	2106.000000	2106.000000	2106.000000	2106.000000	2106.000000	2106.000000	2106.000000	2106.000000	2106.000000	2106.000000	2106.000000
mean	5.033683	3.895830	3.880441	29.917095	8.853627	0.593522	5.512262	2.900198	2016.401235	6.344255	275.945869
std	2.737579	2.133337	2.384010	15.590966	5.605605	1.092988	6.428736	2.415816	1.665060	3.386954	1.970547
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	2014.000000	1.000000	190.000000
25%	3.000000	2.340000	2.000000	19.000000	5.000000	0.000000	1.000000	1.000000	2015.000000	3.000000	276.000000
50%	4.990000	3.670000	3.500000	26.900000	8.000000	0.000000	4.000000	2.000000	2016.000000	6.000000	276.000000
75%	6.670000	5.138000	5.200000	38.300000	12.000000	1.000000	8.000000	4.000000	2018.000000	9.000000	276.000000
max	17.340000	14.463000	16.000000	161.000000	54.833333	9.000000	45.000000	15.000000	2019.000000	12.000000	276.000000

- Verified missing values using df.isnull.sum() and data consistency
- Converted date columns to proper format using df[]=pd.to_datetime([])
- Sort the date to ascending order using df=df.sort_values()
- Saved cleaned data for further analysis

SQL (PostgreSQL)

- Imported cleaned data into the database

- Performed aggregation and validation checks
- Analyzed product-wise, weekday-wise, and time-based patterns

- **1. Total records in the dataset -** This analysis verifies the total number of records to ensure the dataset was imported correctly and is complete.

	count bigint
1	2106

- **2. Overall date range-** The earliest and latest dates were identified to understand the time period covered by the sales data.

	earliest_date date	latest_date date
1	2014-01-02	2019-10-08

- **3. Product with highest total sales-** Sales were aggregated by product to identify the top-performing drug category.

	product_code text	total double precision
1	n05c	63005.40270833992

- **4. Product with lowest total sales-** This analysis highlights the product with the lowest sales contribution.

	product_code text	total double precision
1	n07a	1249.9583333330002

- **5. Weekday with highest sales-**Sales were grouped by weekday to determine the peak sales day.

	weekday text	total_sale double precision
1	Saturday	19767.594416663007

- **6. Weekday with lowest sales-** The weekday generating the lowest total sales was identified to understand low-demand periods.

	weekday text	total_sale double precision
1	Thursday	17212.389270837008

- **7. Month-wise sales trend-** Monthly aggregation was performed to analyze sales trends and seasonality.

	month integer	total_sale double precision
1	1	13970.691083319001
2	2	11604.568583321
3	3	11363.605000000007
4	4	10248.718000000008
5	5	9925.967666670002
6	6	8992.860999999995
7	7	8758.503333333003
8	8	9037.611333340003
9	9	11000.856666666996
10	10	12051.0246875
11	11	9534.383333337
12	12	11106.712999996998

- **8. Sales distribution across weekdays-** Weekday-wise sales distribution was analyzed to check whether sales are evenly spread.

	weekday text	total_sale double precision
1	Saturday	19768
2	Sunday	18401
3	Monday	18243
4	Friday	18134
5	Tuesday	18065
6	Wednesd...	17772
7	Thursday	17212

- **9. Highest contributing product (percentage)-** Each product's contribution to total sales was calculated to identify dominant products.

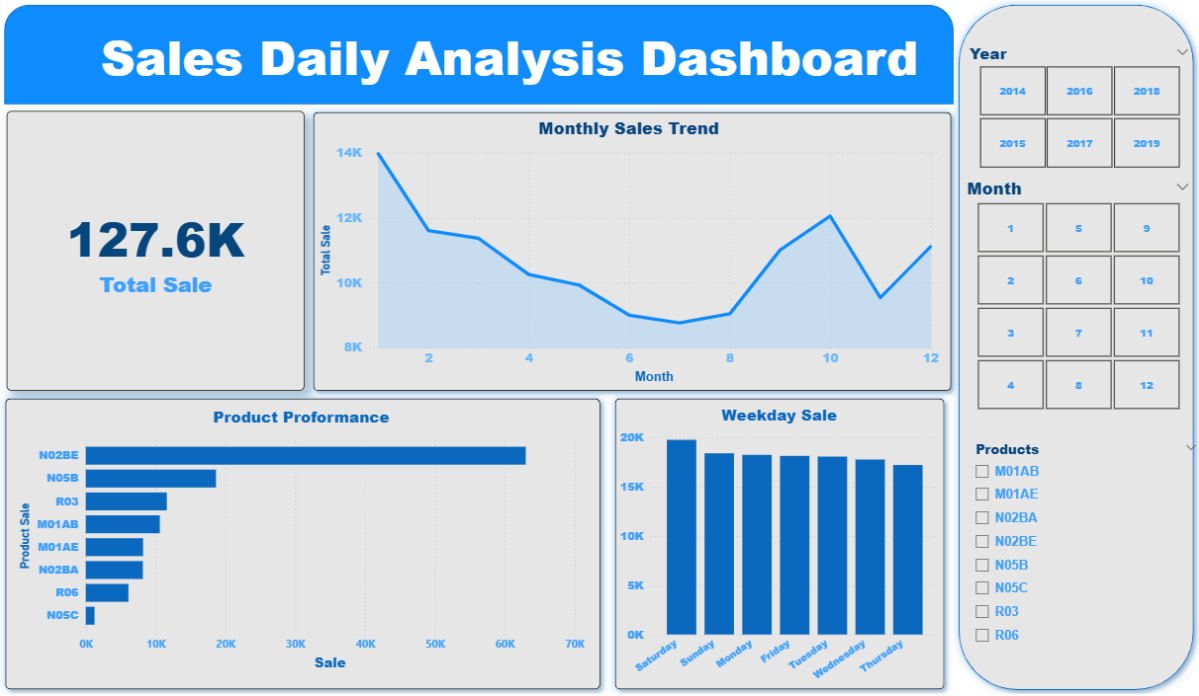
	product_code text	percentage_contribution double precision
1	n05c	49
2	n06a	15
3	n08b	9
4	n02b	8
5	n03c	6
6	n04a	6
7	n09a	5
8	n07a	1

- **10. Average daily sales per product-** Average daily sales were calculated to compare baseline performance across products.

	product_code text	avarage_sales double precision
1	n05c	29.917095303105373
2	n06a	8.85362654319611
3	n08b	5.51226159386135
4	n02b	5.033683325419752
5	n03c	3.895830316160029
6	n04a	3.8804411206082676
7	n09a	2.9001982431149083
8	n07a	0.5935224754667617

Power BI

- Reshaped data using **unpivoting** for flexible analysis
- Created calculated measures for business metrics
- Built an interactive dashboard with slicers



4. Data Preparation and Modeling

Key preparation steps included:

- Correcting data types for dates and numeric fields
- Reshaping wide product columns into a long format with:
    - **Product_Code**
    - **Sales**
- Creating a **Total Sales** measure to represent overall sales volume

This data model supports accurate filtering, aggregation, and visualization.

---

5. Dashboard Design

The Power BI dashboard was designed as a **single-page interactive report** with the following components:

Key Metrics

- **Total Sales** – Overall pharmaceutical sales volume

Visualizations

- **Monthly Sales Trend** – Shows seasonality and trend over time
- **Product Performance** – Compares sales contribution by ATC drug code
- **Weekday Sales Analysis** – Highlights demand variation across weekdays

Interactive Filters (Slicers)

- Year
- Month
- Product Code

---

6. Key Insights

1. **Seasonal Sales Pattern:**
   Sales decline during mid-year and recover toward the end of the year, indicating seasonality.

2. **Product Concentration:**
   A limited number of drug categories contribute a major share of total sales, while some products show relatively low performance.

3. **Weekday Variability:**
   Sales are not evenly distributed across weekdays, suggesting opportunities for operational and sales planning.

4. **Year-wise Comparison:**
   Year-based filtering reveals variations in overall performance, enabling comparative analysis across periods.

7. Business Impact

- Helps identify **high-performing and low-performing drug categories**
- Supports **inventory and supply planning**
- Assists in **sales force and promotion strategy**
- Enables data-driven decision-making through interactive analysis